

**Research & Vehicle Technology**

**“Infotainment Systems Product Development”**

**Feature – Wireless Interface Router Server**

**Infotainment Subsystem Part Specific Specification (SPSS)**

Version 1.8

**UNCONTROLLED COPY IF PRINTED**

**Version Date: January 18, 2022**

**FORD CONFIDENTIALF**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Ver** | **Notes** | |
| **February 9, 2018** | **1.0** | **Initial Release** |  |
|  |  |  |  |
| **May 22, 2018** | **1.1** |  | |
|  | WIR-CLD-REQ-276161/B-Wireless Interface Router Server | | MBORREL4: Removed access token handling |
|  | WIR-CLD-REQ-276162/C-Wireless Interface Router Client1 | | MBORREL4: Removed access token handling |
|  | WIR-CLD-REQ-289670/C-Wireless Interface Router Client2 | | MBORREL4: Removed access token handling |
|  | WIR-IIR-REQ-289532/B-WIRServer\_Rx | | MBORREL4: Added MD's |
|  | MD-REQ-304019/A-AutosarNM | | MBORREL4: New MD for autosar wake up signal |
|  | MD-REQ-304038/A-VehicleMode | | MBORREL4: New MD for keyoff load signal |
|  | WIR-REQ-295922/B-Responding To Queries From Applications Requesting Connectivity | | MBORREL4: Updated content per team review |
|  | STR-503858/B-Requirements | | MBORREL4: Removed REQ-295931 |
|  | WIR-REQ-295928/B-Types Of Intents | | MBORREL4: Updated content per team review |
|  | WIR-REQ-295929/B-Intent Priority | | MBORREL4: Updated content per team review |
|  | WIR-REQ-295932/B-Intent Structure | | MBORREL4: Updated content per team review |
|  | WIR-REQ-295934/B-Intents And Possible Interfaces | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296061/B-Bandwidth Shaping | | MBORREL4: Updated content per team review |
|  | WIR-REQ-295943/B-Intent Policy Update Acknowledgement | | MBORREL4: Updated content per team review |
|  | WIR-UC-REQ-296114/B-Policy not updated in WIRServer / WIRClients | | MBORREL4: Updated content per team review |
|  | WIR-REQ-295960/B-Purpose | | MBORREL4: Updated content per team review |
|  | WIR-REQ-295963/B-Diagnostics Information Storage | | MBORREL4: Updated content per team review |
|  | WIR-REQ-295964/B-Diagnostics Information Sending To Cloud | | MBORREL4: Updated content per team review |
|  | WIR-REQ-295965/B-Sending The App Requests And Interfaces Provided To App To WIRServer Central Controller | | MBORREL4: Updated content per team review |
|  | WIR-UC-REQ-296112/B-WIR provides diagnostics information to cloud based on request from Cloud | | MBORREL4: Updated content per team review |
|  | WIR-REQ-295970/B-Data usage and edge interface | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296012/B-Receive requests from local controller | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296013/B-Process requests from Local controller: | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296018/B-Removing application requests from scheduling queue | | MBORREL4: Updated content per team review |
|  | STR-503991/B-Requirements | | MBORREL4: Added REQ-310878-880 |
|  | WIR-REQ-296025/B-Off-peak determination of off-peak wake up qualification | | MBORREL4: Updated content per team review |
|  | WIR-REQ-310878/A-Requests during offpeak | | MBORREL4: New req. per team review |
|  | WIR-REQ-296029/B-Off-peak wake up | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296030/B-Off-peak wake up battery restrictions | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296031/B-OFF-Peak activities | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296032/B-Off-peak termination | | MBORREL4: Updated content per team review |
|  | WIR-REQ-310879/A-Off-peak timer | | MBORREL4: New req. per team review |
|  | WIR-REQ-296033/B-Off-peak termination notification to applications | | MBORREL4: Updated req name per team review |
|  | WIR-REQ-310880/A-Off-peak completion notification to cloud | | MBORREL4: New req. per team review |
|  | WIR-REQ-296034/B-Off-peak and Ignition | | MBORREL4: Updated content per team review |
|  | STR-503988/B-Requirements | | MBORREL4: Added REQ-311551-561 |
|  | WIR-REQ-296038/B-WIRClient1 WIFI status | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296039/B-WIRClient2 WIFI status | | MBORREL4: Updated content per team review |
|  | WIR-REQ-311551/A-WI-FI ON/OFF settings | | MBORREL4: New req. per team review |
|  | WIR-REQ-311552/A-WI-FI Station mode ON/OFF settings | | MBORREL4: New req. per team review |
|  | WIR-REQ-311553/A-User initiated scan | | MBORREL4: New req. per team review |
|  | WIR-REQ-311554/A-User initiated connect to an access point | | MBORREL4: New req. per team review |
|  | WIR-REQ-311555/A-User initiated disconnect from an access point | | MBORREL4: New req. per team review |
|  | WIR-REQ-311556/A-Selecting An Access Point | | MBORREL4: New req. per team review |
|  | WIR-REQ-296040/B-Arbitration between WIRClient1WIFI and WIRClient2 WIFI | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296041/B-WIFI and special policy | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296042/B-WIFI disconnect | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296043/B-Connecting both WIRClient1WIFI and WIRClient2 WIFI to same access point | | MBORREL4: Updated content per team review |
|  | WIR-REQ-296044/B-WIFI restrictions | | MBORREL4: Updated content per team review |
|  | WIR-REQ-311557/A-Handling failures | | MBORREL4: New req. per team review |
|  | WIR-REQ-311558/A-WIFI hotspot SSID / Password in HMI display | | MBORREL4: New req. per team review |
|  | WIR-REQ-311559/A-WIFI Network availability notiification | | MBORREL4: New req. per team review |
|  | WIR-REQ-311560/A-WIFI network availability notification HMI | | MBORREL4: New req. per team review |
|  | WIR-REQ-311561/A-WIFI network availability notification default setting | | MBORREL4: New req. per team review |
|  | WIR-FUN-REQ-296074/B-Performance and General Requirements | | MBORREL4: Updated Function name |
|  | STR-504083/B-Requirements | | MBORREL4: Added REQ-310882-883 |
|  | WIR-REQ-296078/B-Providing interfaces | | MBORREL4: Updated content per team review |
|  | WIR-REQ-310882/A-Response to privacy mode settings | | MBORREL4: New req. per team review |
|  | WIR-REQ-310883/A-eCall / Emergency assistance / ERA-GLONASS related requirements | | MBORREL4: New req. per team review |
|  | WIR-REQ-311562/A-Checking CCS policies | | MBORREL4: New req. per team review |
|  | WIR-REQ-296095/B-Configuration Parameters | | MBORREL4: Updated content per team review |
|  |  |  |  |
| **August 21, 2018** | **1.2** |  | |
|  | WIR-IIR-REQ-289532/C-WIRServer\_Rx | | MBORREL4: Removed REQ-304019 (already done per existing functionality, see Embedded Modem Common Functions v2 TCU SPSS) |
|  | WIR-REQ-295943/C-Intent Policy Update Acknowledgement | | MBORREL4: Updated table per team review / FTCP updates |
|  | WIR-REQ-295947/B-Tunnel Support | | MBORREL4: Changed WIRClient1 to WIRServer |
|  | WIR-REQ-295973/B-Data usage period reset | | MBORREL4: MBORREL4: Updated content per team review / FTCP updates |
|  | STR-503991/C-Requirements | | MBORREL4: Added REQ-321008 per team review / FTCP updates |
|  | WIR-REQ-296027/B-Cloud response to off-peak request from vehicle | | MBORREL4: Updated content per team review / FTCP updates |
|  | WIR-REQ-296028/B-Repeated off-peak wake up | | MBORREL4: Updated content per team review / FTCP updates |
|  | WIR-REQ-296029/C-Off-peak wake up | | MBORREL4: Updated content per team review / FTCP updates |
|  | WIR-REQ-296030/C-Off-peak wake up restrictions | | MBORREL4: Updated title and content per team review / FTCP updates |
|  | WIR-REQ-296031/C-OFF-Peak activities | | MBORREL4: Updated content per team review / FTCP updates |
|  | WIR-REQ-296032/C-Off-peak termination | | MBORREL4: Updated content per team review / FTCP updates |
|  | WIR-REQ-310879/B-Off-peak timer | | MBORREL4: Updated content per team review / FTCP updates |
|  | WIR-REQ-321008/A-Off-peak error codes | | MBORREL4: New req. per team review / FTCP updates |
|  | WIR-UC-REQ-296098/B-Off peak wake up | | MBORREL4: Updated scenario desc. per team review / FTCP updates |
|  | WIR-REQ-311558/B-WIFI hotspot SSID / Password in HMI display | | MBORREL4: Updated to clarify WIRClient2's Hotspot |
|  |  |  |  |
| **April 24, 2019** | **1.3** |  | |
|  | STR-366262/B-Terminology and Abbreviations | | MBORREL4: Added VDS |
|  | STR-487077/B-Architectural Design | | MBORREL4: Added REQ-350862 |
|  | WIR-CLD-REQ-350862/A-Wireless Interface Router Client3 | | MBORREL4: New req. |
|  | STR-366290/C-Physical Mapping of Classes | | MBORREL4: Added Client3 |
|  | WIR-REQ-295914/B-Local Controller Purpose | | MBORREL4: Added WIRClient3 |
|  | WIR-UC-REQ-296107/B-Local controller provides connection interface to requesting application | | MBORREL4: Added WIRClient3 |
|  | WIR-UC-REQ-296108/B-Central controller not available | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-295928/C-Types Of Intents | | MBORREL4: Added WIRClient3 content |
|  | WIR-REQ-295934/C-Intents And Possible Interfaces | | MBORREL4: Added WIRClient3 to possible interfaces |
|  | WIR-REQ-296061/C-Bandwidth Shaping | | MBORREL4: Added WIRClient3 |
|  | STR-503861/B-Requirements | | MBORREL4: Added REQ-350866 |
|  | WIR-REQ-295936/B-Purpose | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-350866/A-WIRClient3 Policies | | MBORREL4: New req. |
|  | WIR-REQ-295946/B-Purpose | | MBORREL4: Added WIRClient3 |
|  | WIR-UC-REQ-296112/C-WIR provides diagnostics information to cloud based on request from Cloud | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-295967/B-Data usage calculation | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-295968/B-Data usage calculation scope | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-295969/B-Data usage aggregation per application | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-295970/C-Data usage and edge interface | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-295984/B-Trigger | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-296012/C-Receive requests from local controller | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-296015/B-Ability to control already provided interface | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-296016/B-Scheduling application requests | | MBORREL4: Added WIRClient3 |
|  | WIR-UC-REQ-296096/B-Central controller provides connection interface to local controller | | MBORREL4: Added WIRClient3 |
|  | WIR-UC-REQ-296097/B-Interface not available | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-296075/B-Boot up | | MBORREL4: Added WIRClient3 |
|  | WIR-REQ-296077/B-WIRServer, WIRClient1, WIRClient2, and WIRClient3 interfaces | | MBORREL4: Updated req name and added WIRClient3 |
|  |  |  |  |
| **November 22, 2019** | **1.4** |  | |
|  | STR-356400/C-Overview | | MBORREL4: Updated text |
|  | STR-366262/C-Terminology and Abbreviations | | MBORREL4: Updated table |
|  | STR-487077/C-Architectural Design | | MBORREL4: Added REQ-370364, REQ-370370 |
|  | WIR-CLD-REQ-276161/C-Wireless Interface Router Server | | MBORREL4: Updated req. |
|  | WIR-CLD-REQ-276162/D-Wireless Interface Router Client1 | | MBORREL4: Updated req. |
|  | WIR-CLD-REQ-289670/D-Wireless Interface Router Client2 | | MBORREL4: Updated req. |
|  | WIR-CLD-REQ-350862/B-Wireless Interface Router Client3 | | MBORREL4: New req. |
|  | WIR-CLD-REQ-370364/A-Wireless Interface Router Client4 | | MBORREL4: New req. |
|  | WIR-CLD-REQ-370370/A-Wireless Interface Router Client5 | | MBORREL4: New req. |
|  | STR-366290/D-Physical Mapping of Classes | | MBORREL4: Updated table |
|  | STR-503850/B-Functional Definition | | MBORREL4: Added FUN-REQ-370034 |
|  | WIR-REQ-295914/C-Local Controller Purpose | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-296107/C-Local controller provides connection interface to requesting application | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-296108/C-Central controller not available | | MBORREL4: Updated req. |
|  | STR-503858/C-Requirements | | MBORREL4: Added REQ-369964, REQ-370231 |
|  | WIR-REQ-295928/D-Types Of Intents | | MBORREL4: Updated req. |
|  | WIR-REQ-295932/C-Intent Structure | | MBORREL4: Updated req. |
|  | WIR-REQ-369964/A-Intent processing | | MBORREL4: New req. |
|  | WIR-REQ-295934/D-Intents And Possible Interfaces | | MBORREL4: Updated req. |
|  | WIR-REQ-296061/D-Bandwidth Shaping | | MBORREL4: Updated req. |
|  | WIR-REQ-370231/A-APN3 and APN4 | | MBORREL4: New req. |
|  | STR-503859/B-Use Cases | | MBORREL4: Added REQ-369967-369971 |
|  | WIR-UC-REQ-369967/A-Amazon application request special intent for WiFi | | MBORREL4: New req. |
|  | WIR-UC-REQ-369968/A-WiFi HotSpot application request special intent | | MBORREL4: New req. |
|  | WIR-UC-REQ-369969/A-FCI application request special intent | | MBORREL4: New req. |
|  | WIR-UC-REQ-369970/A-Wrong APN ID for special intent | | MBORREL4: New req. |
|  | WIR-UC-REQ-369971/A-Other application request special intent | | MBORREL4: New req. |
|  | STR-503861/C-Requirements | | MBORREL4: Added REQ-369995-998, REQ-370380-381 |
|  | WIR-REQ-295936/C-Purpose | | MBORREL4: Updated req. |
|  | WIR-REQ-295937/B-Global Application ID | | MBORREL4: Updated title and req. |
|  | WIR-REQ-295939/B-Intent Policy Table Privilege Breakdown | | MBORREL4: Updated req. |
|  | WIR-REQ-295940/B-Intent Policy Table Structure | | MBORREL4: Updated req. |
|  | WIR-REQ-370380/A-WIRClient4 Policies | | MBORREL4: New req. |
|  | WIR-REQ-370381/A-Z2 Dual Modem Policy Limitation | | MBORREL4: New req. |
|  | WIR-REQ-295942/B-Intent Policy Update | | MBORREL4: Updated req. |
|  | WIR-REQ-369995/A-Sends Error for Corrupt policy file | | MBORREL4: New req. |
|  | WIR-REQ-295943/D-Intent Policy Update Acknowledgement | | MBORREL4: Updated req. |
|  | WIR-REQ-369996/A-Major version and Minor version | | MBORREL4: New req. |
|  | WIR-REQ-369997/A-Default intent Policy file | | MBORREL4: New req. |
|  | WIR-REQ-369998/A-Intent policy update/rejection based on version | | MBORREL4: New req. |
|  | WIR-UC-REQ-296113/B-Cloud sends updated policy table to WIR | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-296114/C-Policy not updated in WIRServer / WIRClients | | MBORREL4: Updated req. |
|  | STR-503864/B-Requirements | | MBORREL4: Added REQ-370382-383 |
|  | WIR-REQ-295946/C-Purpose | | MBORREL4: Updated req. |
|  | WIR-REQ-295948/B-WIRClients IP Address | | MBORREL4: Updated title and req. |
|  | WIR-REQ-295949/B-Ignition Cycle When Tunnel IP Address Is Already In Use | | MBORREL4: Updated req. |
|  | WIR-REQ-295950/B-IP Address Assignation | | MBORREL4: Updated req. |
|  | WIR-REQ-295951/B-IP Aliasing | | MBORREL4: Updated req. |
|  | WIR-REQ-370382/A-Allow WIRClient2 Edge Interface Access To Ethernet Connected ECU’s | | MBORREL4: New req. |
|  | WIR-REQ-370383/A-Allow WIRClient5 Edge Interface Access To Ethernet Connected ECU’s | | MBORREL4: New req. |
|  | WIR-REQ-295953/B-Allow WIRClient Applications To Access Edge Interface Of Other Ethernet Connected ECU's | | MBORREL4: Updated title and req. |
|  | STR-503867/B-Requirements | | MBORREL4: Added REQ-370232 |
|  | WIR-REQ-295963/C-Diagnostics Information Storage | | MBORREL4: Updated req. |
|  | WIR-REQ-370232/A-On demand diagnostic request | | MBORREL4: New req. |
|  | WIR-UC-REQ-296112/D-WIR provides diagnostics information to cloud based on request from Cloud | | MBORREL4: Updated req. |
|  | STR-503870/B-Requirements | | MBORREL4: Added REQ-369999 |
|  | WIR-REQ-295967/C-Data usage calculation | | MBORREL4: Updated req. |
|  | WIR-REQ-295968/C-Data usage calculation scope | | MBORREL4: Updated req. |
|  | WIR-REQ-295969/C-Data usage aggregation per application | | MBORREL4: Updated req. |
|  | WIR-REQ-295970/D-Data usage and edge interface | | MBORREL4: Updated req. |
|  | WIR-REQ-295981/B-SiriusXM data usage report | | MBORREL4: Changed title and content |
|  | WIR-REQ-369999/A-Sending Data Usage Information to cloud on Ignition OFF using an FTCP alert | | MBORREL4: New req. |
|  | WIR-REQ-295984/C-Trigger | | MBORREL4: Updated req. |
|  | WIR-REQ-295986/B-WIFI Connect Reminder Popup User Action | | MBORREL4: Updated req. |
|  | WIR-REQ-295987/B-User Selects OK | | MBORREL4: Updated req. |
|  | WIR-REQ-295988/B-User Selects Remind Me Later | | MBORREL4: Updated req. |
|  | WIR-REQ-295990/B-User Selects “No” Check Box | | MBORREL4: Updated title and req. |
|  | WIR-REQ-295991/B-Master Reset | | MBORREL4: Updated req. |
|  | STR-503985/B-Requirements | | MBORREL4: Added REQ-370384 |
|  | WIR-REQ-296012/D-Receive requests from local controller | | MBORREL4: Updated req. |
|  | WIR-REQ-296015/C-Ability to control already provided interface | | MBORREL4: Updated req. |
|  | WIR-REQ-296016/C-Scheduling application requests | | MBORREL4: Updated req. |
|  | WIR-REQ-370384/A-Z2 Modem Limitation | | MBORREL4: New req. |
|  | WIR-UC-REQ-296096/C-Central controller provides connection interface to local controller | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-296097/C-Interface not available | | MBORREL4: Updated req. |
|  | WIR-REQ-310878/B-Requests during offpeak | | MBORREL4: Updated req. |
|  | STR-503988/C-Requirements | | MBORREL4: Added REQ-370000-002, REQ-370266-267, REQ-370385 |
|  | WIR-REQ-370385/A-WIRClient5 WIFI status | | MBORREL4: New req. |
|  | WIR-REQ-311553/B-User initiated scan | | MBORREL4: Updated req. |
|  | WIR-REQ-311554/B-User initiated connect to an access point | | MBORREL4: Updated req. |
|  | WIR-REQ-311555/B-User initiated disconnect from an access point | | MBORREL4: Updated req. |
|  | WIR-REQ-311556/B-Selecting An Access Point | | MBORREL4: Updated req. |
|  | WIR-REQ-296040/C-Arbitration between WIRClients WIFI | | MBORREL4: Updated title and content |
|  | WIR-REQ-296041/C-WIFI and special policy | | MBORREL4: Updated req. |
|  | WIR-REQ-296042/C-WIFI disconnect | | MBORREL4: Updated req. |
|  | WIR-REQ-296043/C-Connecting multiple WIRClients WIFI to same access point | | MBORREL4: Updated title and content |
|  | WIR-REQ-370000/A-WIFI scan when gear moved park and engine is not running | | MBORREL4: New req. |
|  | WIR-REQ-370001/A-WIFI scan when driver restriction is disabled and engine is running | | MBORREL4: New req. |
|  | WIR-REQ-370002/A-Excluding WIRClient1 projection mode AP from scan result | | MBORREL4: New req. |
|  | WIR-REQ-370266/A-Scan and connecting to an AP while WIRServer or WIRClient2 is not provisioned | | MBORREL4: New req. |
|  | WIR-REQ-370267/A-Factory WiFi network support | | MBORREL4: New req. |
|  | STR-503989/B-Use Cases | | MBORREL4: Added REQ-370030-033, REQ-370386-389 |
|  | WIR-UC-REQ-296102/B-Connect to WIFI Access point | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-296103/B-WIRClient1 WI-FI is not available | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-296104/B-WIRClient2 WI-FI is not available | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-370386/A-WIRClient5 WI-FI is not available | | MBORREL4: New req. |
|  | WIR-UC-REQ-296105/B-WIRClient1 and WIRClient2 WI-FI are not available | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-370387/A-WIRClient1 and WIRClient5 WI-FI are not available | | MBORREL4: New req. |
|  | WIR-UC-REQ-370388/A-WIRClient2 and WIRClient5 WI-FI are not available | | MBORREL4: New req. |
|  | WIR-UC-REQ-370389/A-WIRClient1, WIRClient2 and WIRClient5 WI-FI are not available | | MBORREL4: New req. |
|  | WIR-UC-REQ-296106/B-Known access point not available | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-370033/A-Driver restriction is disabled and engine is off | | MBORREL4: New req. |
|  | WIR-UC-REQ-370030/A-WiFi is connected but no application is requesting WiFi connection | | MBORREL4: New req. |
|  | WIR-UC-REQ-370031/A-WIRServer disconnected from an AP and an application requests WiFi connection | | MBORREL4: New req. |
|  | WIR-UC-REQ-370032/A-Last application releases WIR WiFi connection | | MBORREL4: New req. |
|  | WIR-REQ-296075/C-Boot up | | MBORREL4: Updated req. |
|  | WIR-REQ-296077/C-WIRServer, WIRClient1, WIRClient2, WIRClient3, WIRClient4 and WIRClient5 interfaces | | MBORREL4: Updated req. |
|  | WIR-FUN-REQ-370034/A-Captive Portal Check | | MBORREL4: New function |
|  | STR-705251/A-Requirements | | MBORREL4: New STR |
|  | WIR-REQ-370003/A-Requesting WLAN connection | | MBORREL4: New req. |
|  | WIR-REQ-370004/A-Caching WiFi Hotspot password | | MBORREL4: New req. |
|  | WIR-REQ-370005/A-Retrieving saved WiFi Hotspot password | | MBORREL4: New req. |
|  | WIR-REQ-370006/A-Common WiFi Hotspot SSID/password database | | MBORREL4: New req. |
|  | WIR-REQ-370007/A-Captive portal check request | | MBORREL4: New req. |
|  | WIR-REQ-370008/A-CPC procedure | | MBORREL4: New req. |
|  | WIR-REQ-370009/A-Central controller | | MBORREL4: New req. |
|  | WIR-REQ-370010/A-Request WEB engine to display landing page | | MBORREL4: New req. |
|  | WIR-REQ-370011/A-WLAN internet connection lost | | MBORREL4: New req. |
|  | WIR-REQ-370012/A-CPC check procedure | | MBORREL4: New req. |
|  | STR-705252/A-Use Cases | | MBORREL4: New STR |
|  | WIR-UC-REQ-369972/A-WIRClient1 Wi-Fi connected to a Wi-Fi Hotspot first time and redirected to landing page | | MBORREL4: New req. |
|  | WIR-UC-REQ-369973/A-WIRClient1 Wi-Fi connected to an Wi-Fi Hotspot first time and Wi-Fi Hotspot doesn’t have landing page | | MBORREL4: New req. |
|  | WIR-UC-REQ-369974/A-WIRClient1/WIRClient2 Wi-Fi connected to a Wi-Fi Hotspot connected previously | | MBORREL4: New req. |
|  | WIR-UC-REQ-369975/A-WIRClient1/WIRClient2 Wi-Fi connected to an open Wi-Fi Hotspot connected previously and redirected to landing page | | MBORREL4: New req. |
|  | WIR-UC-REQ-369976/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time | | MBORREL4: New req. |
|  | WIR-UC-REQ-369977/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters verification code recei | | MBORREL4: New req. |
|  | WIR-UC-REQ-369978/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time and lading page displayed | | MBORREL4: New req. |
|  | WIR-UC-REQ-369979/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters password with on-screen | | MBORREL4: New req. |
|  | WIR-UC-REQ-369980/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again | | MBORREL4: New req. |
|  | WIR-UC-REQ-369981/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again and landing page displayed | | MBORREL4: New req. |
|  | WIR-UC-REQ-369982/A-WIRClient1/WIRClient2 Wi-Fi connected to password protected Wi-Fi Hotspot again but authentication fails | | MBORREL4: New req. |
|  | WIR-UC-REQ-369983/A-Wi-Fi connection lost after Wi-Fi Hotspot timeout | | MBORREL4: New req. |
|  | WIR-UC-REQ-369984/A-Wi-Fi connection lost due to Wi-Fi signal becomes too weak | | MBORREL4: New req. |
|  | WIR-UC-REQ-369985/A-Wi-Fi connection lost during connection setup due to Wi-Fi signal becomes weak | | MBORREL4: New req. |
|  | WIR-UC-REQ-369986/A-Customer selects other screen while landing page displayed | | MBORREL4: New req. |
|  | WIR-UC-REQ-369987/A-Other screen overrides while landing page displayed | | MBORREL4: New req. |
|  | WIR-UC-REQ-369988/A-WEB engine supports different WIRClient1 screen size | | MBORREL4: New req. |
|  | WIR-UC-REQ-369989/A-CPC get internet connection success after multiple try | | MBORREL4: New req. |
|  | WIR-UC-REQ-369990/A-CPC get internet connection failure after max try | | MBORREL4: New req. |
|  | WIR-UC-REQ-369991/A-Password enter screen interrupted by other screen | | MBORREL4: New req. |
|  | WIR-UC-REQ-369992/A-WEB engine fails to display landing page | | MBORREL4: New req. |
|  | WIR-UC-REQ-369993/A-WEB engine opens 2nd WEB page while landing page is displayed---TBD | | MBORREL4: New req. |
|  | STR-705253/A-White Box View | | MBORREL4: New STR |
|  | STR-504086/B-Requirements | | MBORREL4: Added REQ-370035 |
|  | WIR-REQ-296095/C-Configuration Parameters | | MBORREL4: Updated req. |
|  | WIR-REQ-370035/A-Master Reset and Configuration Parameters | | MBORREL4: New req. |
|  |  |  |  |
| **May 22, 2020** | **1.5** |  | |
|  | WIR-CLD-REQ-276161/D-Wireless Interface Router Server+ | | MBORREL4: Updated to include AV roles |
|  | WIR-CLD-REQ-276161/E-Wireless Interface Router Server | | MBORREL4: Removed token line item |
|  | WIR-CLD-REQ-276162/E-Wireless Interface Router Client1 | | MBORREL4: Updated to include AV roles |
|  | WIR-CLD-REQ-289670/E-Wireless Interface Router Client2+ | | MBORREL4: Updated to include AV roles |
|  | WIR-CLD-REQ-289670/F-Wireless Interface Router Client2 | | MBORREL4: Removed APN3/4 |
|  | WIR-CLD-REQ-350862/C-Wireless Interface Router Client3 | | MBORREL4: Updated to include AV roles |
|  | WIR-CLD-REQ-370364/B-Wireless Interface Router Client4 | | MBORREL4: Updated to include AV roles |
|  | WIR-CLD-REQ-370370/B-Wireless Interface Router Client5 | | MBORREL4: Updated to include AV roles |
|  | WIR-IIR-REQ-289531/B-WIRServer\_Tx | | MBORREL4: Added REQ-386195. Added all SoA API's |
|  | WIR-IIR-REQ-289532/D-WIRServer\_Rx | | MBORREL4: Added all SoA API's |
|  | WIR-REQ-295914/D-Local Controller Purpose | | GDOUGHE1: Updated req to make common MBORREL4: Updated req. |
|  | WIR-REQ-295918/B-Return Unique Id | | MBORREL4: Updated req. |
|  | WIR-REQ-295919/B-Flow | | MBORREL4: Updated req. |
|  | WIR-REQ-295920/B-Maintain AllocationID’s | | MBORREL4: Updated req. and title |
|  | WIR-REQ-295921/B-Remove AllocationID’s From Queue Once Application Notifies It Is No Longer Needed | | MBORREL4: Updated req. and title |
|  | WIR-REQ-295922/C-Responding To Queries From Applications Requesting Connectivity | | MBORREL4: Updated req. |
|  | WIR-UC-REQ-296107/D-Local controller provides connection interface to requesting application | | GDOUGHE1: Updated req to make common |
|  | WIR-UC-REQ-296108/D-Central controller not available | | GDOUGHE1: Updated req to make common |
|  | STR-503858/D-Requirements | | GDOUGHE1: Updated reqs to make common MBORREL4: Removed REQ-370231, REQ-295933. Added REQ-385871 |
|  | WIR-REQ-295927/B-Intent Purpose | | MBORREL4: Updated req. |
|  | WIR-REQ-295928/E-Types Of Intents | | GDOUGHE1: Removed off peak for client3/4/5 MBORREL4: Removed special policy |
|  | WIR-REQ-295932/D-Intent Structure | | GDOUGHE1: Removed APN3/4 MBORREL4: Updated table |
|  | WIR-REQ-369964/B-Intent processing | | GDOUGHE1: Updated foreground processing MBORREL4: Updated req. |
|  | WIR-REQ-295934/E-Intents And Possible Interfaces | | GDOUGHE1: Removed APN3/4 MBORREL4: Removed APN3/4 and special intent |
|  | WIR-REQ-296061/E-Bandwidth Shaping | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-385871/A-China Basic Service and Entertainment Service | | MBORREL4: New req. |
|  | STR-503859/C-Use Cases | | MBORREL4: Removed REQ-369967-971 |
|  | STR-503861/D-Requirements | | GDOUGHE1: Updated reqs to make common MBORREL4: Added REQ-385872 |
|  | WIR-REQ-295936/D-Purpose | | GDOUGHE1: Update to make common MBORREL4: Updated req. |
|  | WIR-REQ-295937/C-Global Application ID | | MBORREL4: Updated req. |
|  | WIR-REQ-295939/C-Intent Policy Table Privilege Breakdown | | GDOUGHE1: Removed APN3/4 MBORREL4: Updated table |
|  | WIR-REQ-295941/B-Intent Policy Storage | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295942/C-Intent Policy Update | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295943/E-Intent Policy Update Acknowledgement | | GDOUGHE1: Updated req to make common MBORREL4: Updated req. |
|  | WIR-REQ-295944/B-Intent Policy Update Notification To Applications | | MBORREL4: Updated req. |
|  | WIR-REQ-385872/A-Intent policy update pull request | | GDOUGHE1: Updated req to remove DID and replace with hardwired value MBORREL4: New req. |
|  | WIR-UC-REQ-296113/C-Cloud sends updated policy table to WIR | | GDOUGHE1: Updated req to make common |
|  | WIR-UC-REQ-296114/D-Policy not updated in WIRServer / WIRClients | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295946/D-Purpose | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295948/C-WIRClients IP Address | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295949/C-Ignition Cycle When Tunnel IP Address Is Already In Use | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295950/C-IP Address Assignation | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295951/C-IP Aliasing | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295953/C-Allow WIRClient Applications To Access Edge Interface Of Other Ethernet Connected ECU's | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295954/B-WIRClient Interfacing With WIRServer Example | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295956/B-Forwarding The Request To WIRClient WIFI Example Proposal | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295957/B-Receiving Response On WIRClient Module Example Proposal | | GDOUGHE1: Updated req to make common |
|  | STR-503867/C-Requirements | | GDOUGHE1: Updated reqs to make common MBORREL4: Removed REQ-295962 |
|  | WIR-REQ-295961/B-WIFI Diagnostics | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-388321/A-Reporting WiFi Connection Event to SDN | | GDOUGHE1: New req |
|  | WIR-REQ-295965/C-Sending The App Requests And Interfaces Provided To App To WIRServer Central Controller | | MBORREL4: Updated req. |
|  | STR-503868/B-Use Cases | | GDOUGHE1: Updated req to make common |
|  | WIR-UC-REQ-296112/E-WIR provides diagnostics information to cloud based on request from Cloud | | GDOUGHE1: Updated req to make common |
|  | STR-503870/C-Requirements | | GDOUGHE1: Updated reqs to make common |
|  | WIR-REQ-295967/D-Data usage calculation | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295968/D-Data usage calculation scope | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295969/D-Data usage aggregation per application | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-295970/E-Data usage and edge interface | | MBORREL4: Removed APN3/4, updated table |
|  | WIR-REQ-295972/B-Data usage storage | | GDOUGHE1: Updated req |
|  | STR-503985/C-Requirements | | GDOUGHE1: Updated reqs to make common |
|  | WIR-REQ-296012/E-Receive requests from local controller | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-296013/C-Process requests from Local controller: | | GDOUGHE1: Removed special intent |
|  | WIR-REQ-296015/D-Ability to control already provided interface | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-296016/D-Scheduling application requests | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-388353/A-Disabling cellular or WiFi for non-connected vehicle | | GDOUGHE1: New req |
|  | WIR-UC-REQ-296096/D-Central controller provides connection interface to local controller | | GDOUGHE1: Updated req to make common |
|  | WIR-UC-REQ-296097/D-Interface not available | | GDOUGHE1: Updated req to make common |
|  | STR-503991/D-Requirements | | MBORREL4: Added REQ-385873 |
|  | WIR-REQ-385873/A-VPSM subscription and callback handling during off-peak service | | MBORREL4: New req. |
|  | STR-503988/D-Requirements | | GDOUGHE1: Updated reqs to make common |
|  | WIR-REQ-370385/B-WIRClient5 WIFI status | | GDOUGHE1: Updated req |
|  | WIR-REQ-311553/C-User initiated scan | | GDOUGHE1: Updated req |
|  | WIR-REQ-311554/C-User initiated connect to an access point | | GDOUGHE1: Updated req |
|  | WIR-REQ-311555/C-User initiated disconnect from an access point | | GDOUGHE1: Updated req |
|  | WIR-REQ-296040/D-Arbitration between WIRClients WIFI | | GDOUGHE1: Updated req |
|  | WIR-REQ-370000/B-WIFI scan when gear moved park and engine is not running | | MBORREL4: Updated req. |
|  | WIR-REQ-370001/B-WIFI scan when driver restriction is disabled and engine is running | | GDOUGHE1: Updated req |
|  | WIR-REQ-296075/D-Boot up | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-296077/D-WIRServer and WIRClient interfaces | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-310883/B-eCall / Emergency assistance / ERA-GLONASS / Emergency Connect related requirements | | GDOUGHE1: Updated req to make common |
|  | WIR-REQ-296095/D-Configuration Parameters | | GDOUGHE1: Added WiFi Connect Interval DID MBORREL4: Updated table |
|  |  |  |  |
| **July 6, 2020** | **1.6** |  | |
|  | MD-REQ-386195/B-PolicyTablePullRequestQuery | | MBORREL4: Changed to Query |
|  | MD-REQ-380265/C-NetworkInterfaceAllocation | | MBORREL4: Removed LanPreferred Flag |
|  | MD-REQ-380265/C-NetworkInterfaceAllocation | | MBORREL4: Removed LanPreferred Flag |
|  | STR-503850/C-Functional Definition | | MBORREL4: Added REQ-391547 |
|  | WIR-REQ-295939/D-Intent Policy Table Privilege Breakdown | | MBORREL4: Updated table and req |
|  | WIR-REQ-385872/B-Intent policy update pull request+ | | MBORREL4: Updated req. |
|  | WIR-REQ-385872/C-Intent policy update pull request | | MBORREL4: Updated req. |
|  | STR-503870/D-Requirements | | MBORREL4: Removed REQ-295981 |
|  | WIR-REQ-295970/F-Data usage and edge interface | | MBORREL4: Updated req. |
|  | STR-503988/E-Requirements | | MBORREL4: Added REQ-392769, REQ-392787 |
|  | WIR-REQ-370385/C-WIRClient5 WIFI status | | MBORREL4: Updated req. |
|  | WIR-REQ-311553/D-User initiated scan | | MBORREL4: Updated req. |
|  | WIR-REQ-296040/E-Arbitration between WIRClients WIFI | | MBORREL4: Updated req. |
|  | WIR-REQ-311558/C-WIFI hotspot SSID in HMI display | | MBORREL4: Updated req. |
|  | WIR-REQ-370002/B-Excluding WIRClient1 projection mode AP from scan result | | MBORREL4: Updated req. |
|  | WIR-REQ-370267/B-Factory WiFi network support | | MBORREL4: Updated req. |
|  | WIR-REQ-392769/A-Network connection password failure | | MBORREL4: New req. |
|  | WIR-REQ-392787/A-Hidden network display after disconnect | | MBORREL4: New req. |
|  | STR-504083/C-Requirements | | MBORREL4: Removed REQ-311562 |
|  | WIR-REQ-310882/B-Response to privacy mode settings | | MBORREL4: Updated req. |
|  | WIR-REQ-310883/C-eCall / Emergency assistance / ERA-GLONASS / Emergency Connect related requirements | | MBORREL4: Updated req. |
|  | WIR-REQ-370003/B-Requesting WLAN connection | | MBORREL4: Generalized the content |
|  | WIR-REQ-370004/B-Caching WiFi Hotspot password | | MBORREL4: Generalized the content |
|  | WIR-REQ-370005/B-Retrieving saved WiFi Hotspot password | | MBORREL4: Generalized the content |
|  | WIR-REQ-370006/B-Common WiFi Hotspot SSID/password database | | MBORREL4: Generalized the content |
|  | WIR-REQ-370007/B-Captive portal check request | | MBORREL4: Generalized the content |
|  | WIR-UC-REQ-369974/B-WIRClient Wi-Fi connected to a Wi-Fi Hotspot connected previously | | MBORREL4: Updated title, added Client5 |
|  | WIR-UC-REQ-369975/B-WIRClient Wi-Fi connected to an open Wi-Fi Hotspot connected previously and redirected to landing page | | MBORREL4: Updated title, added Client5 |
|  | WIR-UC-REQ-369976/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot first time | | MBORREL4: Updated title, added Client5 |
|  | WIR-UC-REQ-369977/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters verification code recei | | MBORREL4: Updated title, added Client5 |
|  | WIR-UC-REQ-369978/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot first time and lading page displayed | | MBORREL4: Updated title, added Client5 |
|  | WIR-UC-REQ-369979/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters password with on-screen | | MBORREL4: Updated title, added Client5 |
|  | WIR-UC-REQ-369980/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot again | | MBORREL4: Updated title, added Client5 |
|  | WIR-UC-REQ-369981/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot again and landing page displayed | | MBORREL4: Updated title, added Client5 |
|  | WIR-UC-REQ-369982/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot again but authentication fails | | MBORREL4: Updated title, added Client5 |
|  | WIR-UC-REQ-369983/B-Wi-Fi connection lost after Wi-Fi Hotspot timeout | | MBORREL4: Added Client5 |
|  | WIR-UC-REQ-369984/B-Wi-Fi connection lost due to Wi-Fi signal becomes too weak | | MBORREL4: Added Client5 |
|  | WIR-UC-REQ-369985/B-Wi-Fi connection lost during connection setup due to Wi-Fi signal becomes weak | | MBORREL4: Added Client5 |
|  | WIR-FUN-REQ-391547/A-DNS Server and RPZ | | MBORREL4: New req. |
|  | STR-770363/A-Requirements | | MBORREL4: New STR. |
|  | WIR-REQ-391548/A-Configure DNS Server | | MBORREL4: New req. |
|  | WIR-REQ-391549/A-RPZ Zone File | | MBORREL4: New req. |
|  | WIR-REQ-391550/A-Maintain Whitelist Domain | | MBORREL4: New req. |
|  | WIR-REQ-391551/A-Storing Base Domain Whitelist Locally | | MBORREL4: New req. |
|  | WIR-REQ-391552/A-Configure Caching and Forwarding DNS Server | | MBORREL4: New req. |
|  | WIR-REQ-391553/A-Forwarding Requests to DNS Entities | | MBORREL4: New req. |
|  | WIR-REQ-391554/A-Logging and Auditing | | MBORREL4: New req. |
|  | WIR-REQ-391555/A-Updating Whitelist | | MBORREL4: New req. |
|  | WIR-REQ-391556/A-Restarting DNS Resolver | | MBORREL4: New req. |
|  | WIR-REQ-391565/A-Bypassing DNS RPZ for Exception Application | | MBORREL4: New req. |
|  | WIR-REQ-391927/A-Maintaining whitelist and approved application list when disabling feature from CCS reset, factory reset and master reset | | MBORREL4: New req. |
|  | STR-770362/A-Use Cases | | MBORREL4: New STR. |
|  | WIR-UC-REQ-391557/A-Configuring DNS Server | | MBORREL4: New req. |
|  | WIR-UC-REQ-391558/A-Application requesting for DNS Resolution | | MBORREL4: New req. |
|  | WIR-UC-REQ-391559/A-Updating URL’s | | MBORREL4: New req. |
|  | WIR-UC-REQ-391592/A-Bypassing DNS RPZ for Exception Application | | MBORREL4: New req. |
|  | STR-770364/A-White Box View | | MBORREL4: New STR. |
|  | STR-770365/A-Activity Diagrams | | MBORREL4: New STR. |
|  | WIR-ACT-REQ-391563/A-Interaction Among Application- DNS Resolver & DNS Server | | MBORREL4: New req. |
|  | WIR-ACT-REQ-391564/A-Updating URL's | | MBORREL4: New req. |
|  | STR-770366/A-Sequence Diagrams | | MBORREL4: New STR. |
|  | WIR-SD-REQ-391560/A-Application Requests for DNS Resolution - Happy Path: | | MBORREL4: New req. |
|  | WIR-SD-REQ-391561/A-Application Request for DNS Resolution - URL is not present /updated in Whitelist | | MBORREL4: New req. |
|  | WIR-SD-REQ-391562/A-Application Request for DNS Resolution - Exception Application Case | | MBORREL4: New req. |
|  | WIR-REQ-296095/E-Configuration Parameters | | MBORREL4: Updated table |
|  |  |  |  |
| **November 19, 2020** | **1.7** |  | |
|  | STR-366290/E-Physical Mapping of Classes | | MBORREL4: Added DuerOS |
|  | WIR-IIR-REQ-289531/C-WIRServer\_Tx | | MBORREL4: Added REQ-402837-839 |
|  | MD-REQ-402837/A-ReadPolicyTable | | MBORREL4: New req. |
|  | MD-REQ-402838/A-EnableSyncWifiStaMode | | MBORREL4: New req. |
|  | MD-REQ-402839/A-DisableSyncWifiStaMode | | MBORREL4: New req. |
|  | WIR-IIR-REQ-289532/E-WIRServer\_Rx | | MBORREL4: Added REQ-402837-839 |
|  | MD-REQ-402837/A-ReadPolicyTable | | MBORREL4: New req. |
|  | MD-REQ-402838/A-EnableSyncWifiStaMode | | MBORREL4: New req. |
|  | MD-REQ-402839/A-DisableSyncWifiStaMode | | MBORREL4: New req. |
|  | WIR-REQ-295932/E-Intent Structure | | MBORREL4: Updated req. |
|  | WIR-REQ-295939/E-Intent Policy Table Privilege Breakdown | | MBORREL4: Updated req. |
|  | WIR-REQ-295942/D-Intent Policy Update | | MBORREL4: Updated req. |
|  | WIR-REQ-385872/D-Intent policy update pull request | | MBORREL4: Updated req. |
|  | STR-503864/C-Requirements | | MBORREL4: Removed REQ-295949 |
|  | WIR-REQ-295948/D-WIRClients IP Address | | MBORREL4: Updated req. |
|  | STR-503867/D-Requirements | | MBORREL4: Added REQ-402383 |
|  | WIR-REQ-388321/B-Reporting WiFi Connection Event to SDN | | MBORREL4: Updated req. |
|  | WIR-REQ-402383/A-Controlling diagnostic data upload by CCS setting | | MBORREL4: New req. |
|  | STR-503870/E-Requirements | | MBORREL4: Added REQ-402384 |
|  | WIR-REQ-402384/A-Controlling data usage data upload by CCS setting | | MBORREL4: New req. |
|  | STR-503985/D-Requirements | | MBORREL4; Added REQ-402361 |
|  | WIR-REQ-402361/A-Support for China (DuerOS) | | MBORREL4: New req. |
|  | STR-503988/F-Requirements | | MBORREL4: Added REQ-402796-797 |
|  | WIR-REQ-370000/C-WIFI scan when gear moved park and engine is not running | | MBORREL4: Updated req. |
|  | WIR-REQ-370267/C-Factory WiFi network support | | MBORREL4: Updated req. |
|  | WIR-REQ-402796/A-RocketSetup Support - API to Save WiFi AP list | | MBORREL4: New req. |
|  | WIR-REQ-402797/A-API to control WIRClient1 WLAN STA mode | | MBORREL4: New req. |
|  | WIR-REQ-370009/B-Central controller | | MBORREL4: Updated req. |
|  | WIR-REQ-296095/F-Configuration Parameters | | MBORREL4: Corrected typo in first sentence. Updated table |
|  |  |  |  |
| **January 18, 2022** | **1.8** |  | |
|  | STR-366262/D-Terminology and Abbreviations | | MBORREL4: Added PDC |
|  | WIR-CLD-REQ-289670/G-Wireless Interface Router Client2 | | GDOUGHE1: Removed data usage calculation |
|  | WIR-CLD-REQ-350862/D-Wireless Interface Router Client3 | | GDOUGHE1: Remove LAN management, data usage calculation |
|  | WIR-CLD-REQ-370364/C-Wireless Interface Router Client4 | | GDOUGHE1: Remove data usage calculation |
|  | WIR-CLD-REQ-370370/C-Wireless Interface Router Client5 | | GDOUGHE1: Removed data usage calculation |
|  | STR-366290/F-Physical Mapping of Classes | | MBORREL4: Added PDC |
|  | MD-REQ-380258/C-VlanAdd | | GDOUGHE1: Updated ifaceType literals |
|  | MD-REQ-380262/C-VnmReset | | GDOUGHE1: Updated EcuType literals |
|  | MD-REQ-380265/D-NetworkInterfaceAllocation | | GDOUGHE1: Updated InterfaceType literals. MBORREL4: Added WPA3 |
|  | MD-REQ-380267/C-NetworkInterfaceAllocationStatusInd | | GDOUGHE1: Updated NetworkInterfaceType literals |
|  | MD-REQ-380268/C-NetworkInterfaceDownInd | | GDOUGHE1: Updated NetworkInterfaceType literals |
|  | MD-REQ-380269/C-NetworkInterfaceUpInd | | GDOUGHE1: Updated NetworkInterfaceType literals |
|  | MD-REQ-380274/C-Policy | | GDOUGHE1: Updated Policy literals |
|  | MD-REQ-380275/C-PolicyInd | | GDOUGHE1: Updated Policy literals |
|  | MD-REQ-380276/C-PolicyTableInd | | GDOUGHE1: Updated Policy literals |
|  | MD-REQ-380278/C-InterfaceTableInd | | GDOUGHE1: Updated ifaceTable literals |
|  | MD-REQ-380281/B-ScanApimAPs | | MBORREL4: Added WPA3 |
|  | MD-REQ-380283/B-ConnectToApimAP | | MBORREL4: Added WPA3 |
|  | MD-REQ-380288/B-GetNetworkDetails | | MBORREL4: Added WPA3 |
|  | MD-REQ-380294/B-WifiConnectedInd | | MBORREL4: Added WPA3 |
|  | MD-REQ-380298/C-WlanNQM | | GDOUGHE1: Updated EcuType and NetworkInterfaceType literals |
|  | MD-REQ-380299/C-CellNQM | | GDOUGHE1: Updated EcuType literals |
|  | MD-REQ-380304/B-ScanWiFiNetworks | | MBORREL4: Added WPA3 |
|  | MD-REQ-380305/B-ConnectWiFiAP | | MBORREL4: Added WPA3 |
|  | MD-REQ-380307/B-ProfileUpdate | | MBORREL4: Added WPA3 |
|  | MD-REQ-380309/C-StateUpdateInd | | MBORREL4: Added WPA3 |
|  | MD-REQ-380312/B-ProfileUpdateInd | | MBORREL4: Added WPA3 |
|  | MD-REQ-380258/C-VlanAdd | | GDOUGHE1: Updated ifaceType literals |
|  | MD-REQ-380262/C-VnmReset | | GDOUGHE1: Updated EcuType literals |
|  | MD-REQ-380265/D-NetworkInterfaceAllocation | | GDOUGHE1: Updated InterfaceType literals. MBORREL4: Added WPA3 |
|  | MD-REQ-380274/C-Policy | | GDOUGHE1: Updated Policy literals |
|  | MD-REQ-380281/B-ScanApimAPs | | MBORREL4: Added WPA3 |
|  | MD-REQ-380283/B-ConnectToApimAP | | MBORREL4: Added WPA3 |
|  | MD-REQ-380288/B-GetNetworkDetails | | MBORREL4: Added WPA3 |
|  | MD-REQ-380298/C-WlanNQM | | GDOUGHE1: Updated EcuType and NetworkInterfaceType literals |
|  | MD-REQ-380299/C-CellNQM | | GDOUGHE1: Updated EcuType literals |
|  | MD-REQ-380300/C-WlanNQMInd | | GDOUGHE1: Updated EcuType and NetworkInterfaceType literals |
|  | MD-REQ-380304/B-ScanWiFiNetworks | | MBORREL4: Added WPA3 |
|  | MD-REQ-380305/B-ConnectWiFiAP | | MBORREL4: Added WPA3 |
|  | MD-REQ-380307/B-ProfileUpdate | | MBORREL4: Added WPA3 |
|  | MD-REQ-380312/B-ProfileUpdateInd | | MBORREL4: Added WPA3 |
|  | STR-503850/D-Functional Definition | | MBORREL4: Added REQ-470990. Removed REQ-391547 |
|  | WIR-REQ-295917/B-Process Intent Requests | | GDOUGHE1: Corrected processes to process |
|  | STR-503858/E-Requirements | | MBORREL4: Added REQ-470984, REQ-470985 |
|  | WIRv2-REQ-470984/A-Intent Structure | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-369964/C-Intent processing | | MBORREL4: Updated req. |
|  | WIRv2-REQ-470985/A-Intents And Possible Interfaces | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-295938/B-Intent Policy Table Privileges | | GDOUGHE1: Corrected byte to bytes |
|  | WIR-REQ-295939/F-Intent Policy Table Privilege Breakdown | | MBORREL4: Updated req. |
|  | WIR-REQ-385872/E-Intent policy update pull request | | GDOUGHE1: Updated req for clarification |
|  | WIR-REQ-295963/D-Diagnostics Information Storage | | MBORREL4: Corrected typo |
|  | WIR-REQ-388321/C-Reporting WiFi Connection Event to SDN | | GDOUGHE1: Updated FTCP alert name and clarified requirement |
|  | WIR-REQ-295970/G-Data usage and edge interface+ | | GDOUGHE1: Updated req for pending data usage report behavior |
|  | WIR-REQ-295970/H-Data usage and edge interface | | MBORREL4: Updated req. |
|  | WIR-REQ-402384/B-Controlling data usage data upload by CCS setting | | MBORREL4: Updated req. |
|  | STR-503988/G-Requirements | | MBORREL4: Added REQ-470986, REQ-470987, REQ-470988, REQ-470989 |
|  | WIR-REQ-311557/B-Handling failures | | GDOUGHE1: Updated requirement |
|  | WIR-REQ-370000/D-WIFI scan when gear moved park and engine is not running | | GDOUGHE1: Updated req to clarify scan stop behavior |
|  | WIR-REQ-470986/A-Prevent customer to connect to projection mode AP manually | | MBORREL4: New req. |
|  | WIR-REQ-370267/D-Factory WiFi network support | | GDOUGHE1: Updated requirement for ECG1 |
|  | WIRv2-REQ-470987/A-Factory WiFi network support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470988/A-Personal & Portable Profile Support - API to Save WIFI Setup | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470989/A-Supporting same SSID and different security type AP | | MBORREL4: New req. for FNV3 |
|  | WIR-FUN-REQ-470990/A-Phoenix Support | | MBORREL4: New function for FNV3 |
|  | STR-1012614/A-Requirements | | MBORREL4: New section |
|  | WIR-REQ-470991/A-Support Android OS | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470992/A-FNV2 CM Client | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470993/A-WIR Implementation | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470994/A-Link Bandwidth Support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470995/A-WIRClient2 WLAN support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470996/A-Data Usage Per Application | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470997/A-BSSID support on WiFi configuration | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470998/A-Android Settings Items for ConnectivityManager | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-470999/A-Wifi Configuration Items | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471000/A-WiFi Auto Connect Support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471001/A-MAC Randomization Support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471002/A-Wi-Fi Preferred Network Offload (PNO) Scan Support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471003/A-Dual WLAN Support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471004/A-WLAN Command and Response Routing | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471005/A-WiFi Scoring Algorithm Support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471006/A-WiFi Health Monitoring | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471007/A-WiFi Throughput Estimation Support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471008/A-WPA3 Support | | MBORREL4: New req. for FNV3 |
|  | WIR-REQ-471009/A-Networking and Device Discovery | | MBORREL4: New req. for FNV3 |
|  | STR-705251/B-Requirements | | MBORREL4: Added REQ-471010 |
|  | WIR-REQ-370008/B-CPC procedure | | GDOUGHE1: Grammatical correction |
|  | WIR-REQ-471010/A-Not supporting HTTP captive portal | | MBORREL4: New req. |
|  | WIR-UC-REQ-369986/B-Customer selects another screen while landing page displayed | | MBORREL4: Corrected typo, updated title |
|  | STR-504086/C-Requirements | | MBORREL4: Added REQ-470 |
|  | WIR-REQ-296095/G-Configuration Parameters | | MBORREL4: Updated table |
|  | WIRv2-REQ-471011/A-Configuration Parameters | | MBORREL4: New req. for FNV3 |

**Table of Contents**

[Revision History 2](#_Toc93426703)

[1 Overview 23](#_Toc93426704)

[1.1 Terminology and Abbreviations 23](#_Toc93426705)

[2 Architectural Design 25](#_Toc93426706)

[2.1 WIR-CLD-REQ-276161/E-Wireless Interface Router Server 25](#_Toc93426707)

[2.2 WIR-CLD-REQ-276162/E-Wireless Interface Router Client1 25](#_Toc93426708)

[2.3 WIR-CLD-REQ-289670/G-Wireless Interface Router Client2 25](#_Toc93426709)

[2.4 WIR-CLD-REQ-350862/D-Wireless Interface Router Client3 25](#_Toc93426710)

[2.5 WIR-CLD-REQ-370364/C-Wireless Interface Router Client4 26](#_Toc93426711)

[2.6 WIR-CLD-REQ-370370/C-Wireless Interface Router Client5 26](#_Toc93426712)

[2.7 Physical Mapping of Classes 26](#_Toc93426713)

[2.8 WIRServer Interface 26](#_Toc93426714)

[2.8.1 WIR-IIR-REQ-289531/C-WIRServer\_Tx 26](#_Toc93426715)

[2.8.2 WIR-IIR-REQ-289532/E-WIRServer\_Rx 95](#_Toc93426716)

[3 Functional Definition 163](#_Toc93426717)

[3.1 WIR-FUN-REQ-295913/A-Local Controller Client 163](#_Toc93426718)

[3.1.1 Requirements 163](#_Toc93426719)

[3.1.2 Use Cases 166](#_Toc93426720)

[3.1.3 White Box View 166](#_Toc93426721)

[3.2 WIR-FUN-REQ-295926/A-Intents 167](#_Toc93426722)

[3.2.1 Requirements 167](#_Toc93426723)

[3.2.2 Use Cases 170](#_Toc93426724)

[3.2.3 White Box View 170](#_Toc93426725)

[3.3 WIR-FUN-REQ-295935/A-Policy Manager 171](#_Toc93426726)

[3.3.1 Requirements 171](#_Toc93426727)

[3.3.2 Use Cases 174](#_Toc93426728)

[3.3.3 White Box View 175](#_Toc93426729)

[3.4 WIR-FUN-REQ-295945/A-Tunnel Manager 176](#_Toc93426730)

[3.4.1 Requirements 176](#_Toc93426731)

[3.4.2 Use Cases 180](#_Toc93426732)

[3.4.3 White Box View 180](#_Toc93426733)

[3.5 WIR-FUN-REQ-295959/A-Diagnostics 181](#_Toc93426734)

[3.5.1 Requirements 181](#_Toc93426735)

[3.5.2 Use Cases 183](#_Toc93426736)

[3.5.3 White Box View 184](#_Toc93426737)

[3.6 WIR-FUN-REQ-295966/A-Data Usage 185](#_Toc93426738)

[3.6.1 Requirements 185](#_Toc93426739)

[3.6.2 Use Cases 186](#_Toc93426740)

[3.6.3 White Box View 186](#_Toc93426741)

[3.7 WIR-FUN-REQ-295982/A-WIFI Connect Reminders - HMI 187](#_Toc93426742)

[3.7.1 Requirements 187](#_Toc93426743)

[3.7.2 Use Cases 188](#_Toc93426744)

[3.7.3 White Box View 188](#_Toc93426745)

[3.8 WIR-FUN-REQ-295996/A-Central Controller 189](#_Toc93426746)

[3.8.1 Requirements 189](#_Toc93426747)

[3.8.2 Use Cases 190](#_Toc93426748)

[3.8.3 White Box View 191](#_Toc93426749)

[3.9 WIR-FUN-REQ-295999/A-Off-Peak Handler 192](#_Toc93426750)

[3.9.1 Requirements 192](#_Toc93426751)

[3.9.2 Use Cases 194](#_Toc93426752)

[3.9.3 White Box View 195](#_Toc93426753)

[3.10 WIR-FUN-REQ-295997/A-Wifi Management 196](#_Toc93426754)

[3.10.1 Requirements 196](#_Toc93426755)

[3.10.2 Use Cases 202](#_Toc93426756)

[3.10.3 White Box View 206](#_Toc93426757)

[3.11 WIR-FUN-REQ-296074/B-Performance and General Requirements 207](#_Toc93426758)

[3.11.1 Requirements 207](#_Toc93426759)

[3.11.2 Use Cases 207](#_Toc93426760)

[3.11.3 White Box View 207](#_Toc93426761)

[3.12 WIR-FUN-REQ-470990/A-Phoenix Support 208](#_Toc93426762)

[3.12.1 Requirements 208](#_Toc93426763)

[3.13 WIR-FUN-REQ-370034/A-Captive Portal Check 211](#_Toc93426764)

[3.13.1 Requirements 211](#_Toc93426765)

[3.13.2 Use Cases 213](#_Toc93426766)

[3.13.3 White Box View 222](#_Toc93426767)

[3.14 WIR-FUN-REQ-296080/A-WIRServer Configuration 223](#_Toc93426768)

[3.14.1 Requirements 223](#_Toc93426769)

[3.14.2 Use Cases 226](#_Toc93426770)

[3.14.3 White Box View 226](#_Toc93426771)

[4 Appendix: Reference Documents 227](#_Toc93426772)

# Overview

The Wireless Interface Router feature has an opportunity to reduce greatly Ford’s ongoing connectivity costs by leveraging non-metered client mode Wi-Fi and driving cellular usage at cheaper, off peak times. Further, the WIR gives vehicle applications an opportunity to utilize any vehicle edge interface directly, without any intermediate protocol, like CAN.

WIR enables vehicle software applications to reach the cloud via any edge network interface on the vehicle. Further, WIR can schedule notifications to applications when more cost favorable networks are available. The architecture of WIR is expandable to new edge network interfaces and is able to support new ECUs as they integrate with ECG via Ethernet

Routing data through SYNC, TCU, and TCU-B Wi-Fi connections will drive down the overall costs of cellular data consumption. Once implemented this service can be used for any connected service within SYNC, Cluster, ADAS, SDS, ADSIM, etc. that requires off board communication to a 1st, 2nd, or 3rd party server.

With the introduction of Ethernet as an internal vehicle network the ability to allow various subsystems to establish an IP connection to an off board server (IP Based Pass through) is a key enabler. These subsystems include any Ethernet connected node in the vehicle: ECG, SYNC, TCU, TCU-B, ADAS, SDS, and ADSIM etc. The off board destinations may be in Ford environment or third party systems not necessarily in Ford enterprise IT Systems. Enabling various services offered by these off board destinations will result in customer satisfaction and delight.

IP Based Pass through works with Wireless Interface Router (aka Connectivity Manager) in the ECG, TCU, TCU-B, and SYNC, an In-Vehicle Policy Management application and with an authorization/policy management application in the NG SDN. These will be the key components to provide session authorization, management, and security authentication for this service. Once implemented, this service can be used by any Ethernet connected service within SYNC, Cluster, ADAS, SDS, and ADSIM etc. that requires an off board connection to a 1st, 2nd, or third party destination address.

Some WiFi network service provider has captive portal page, aka landing page. Once WLAN client is connected to WiFi Hotspot, all IP traffic is ignored except HTTP. Also, all HTTP request is redirected to captive portal page where the customer should accept terms and conditions before getting internet connection through WiFi Hotspot. WIR sends HTTP request to predetermined URL to see response is normal (2XX OK) or redirection (3XX redirection). WIR sends URL and local IP address to WEB engine to display landing page on SYNC display, so the customer can interact with the page. WIR sends internet connection ready notification to WIR application when it receives normal response.

## Terminology and Abbreviations

The following table lists terminologies that are used in this document along with a brief description.

| **Term** | **Description** |
| --- | --- |
| ADSIM | Automated Driving System Interface Module |
| CAN | Controller Area Network |
| CCS | Customer Connectivity Settings |
| CHR | Connected HMI Radio |
| CPC Ca | Captive Portal Check |
| ECG | Enhanced Central Gateway |
| ECU | Electronic Control Unit |
| FNV | Fully Networked Vehicle |
| FTCP | Ford Telematics Control Protocol |
| GRE | Generic Routing and Encapsulation |
| HMI | Human Machine Interface |
| IPC | Inter Process Communication |
| PDC | Phoenix Domain Controller |
| SDN | Software Delivery Network |
| SDS | Self Driving System |
| SoA | Service Oriented Architecture |
| TCU | Telematics Control Unit |
| TCU-B | Telematics Control Unit B |
| WIR | Wireless Interface Router |
| WLAN | Wireless Local Area Network |

# Architectural Design

## WIR-CLD-REQ-276161/E-Wireless Interface Router Server

The Wireless Interface Router Server (WIRServer) is responsible for the tasks listed below:

* Interfaces with Wireless interface router Clients
* Responsible for bandwidth shaping
* Responsible for diagnostics
* Responsible for calculating data usage
* Responsible for policy validation
* Responsible for cloud interface
* Responsible for captive portal check
* Responsible for asking landing page display to WEB engine

Please review the implementation guide/ block diagram to locate the WIRServer class.

## WIR-CLD-REQ-276162/E-Wireless Interface Router Client1

The Wireless Interface Router Client1 (WIRClient1) is responsible for the tasks listed below:

* Handles connectivity request from applications
* Responsible for calculating data usage
* Responsible for policy validation
* Responsible for setting up Tunnels
  + Routing table instead of Tunnels for AV
* Responsible for handling WIFI connection
* Interfaces with central connection controller
* Providing connections to applications when central controller is not active
* Responsible for landing page display by WEB engine and responds according to customer input

Please review the implementation guide/ block diagram to locate the WIRClient1 class.

## WIR-CLD-REQ-289670/G-Wireless Interface Router Client2

The Wireless Interface Router Client2 (WIRClient2) is responsible for the tasks listed below:

* Handles connectivity request from applications
* Responsible for policy validation
* Responsible for setting up Tunnels
  + Routing table instead of Tunnels for AV
* Responsible for setting up Cellular APN1 and APN2.
* Responsible for handling WIFI connection
* Interfaces with central connection controller
* Providing connections to applications when central controller is not active

Please review the implementation guide/ block diagram to locate the WIRClient2 class.

## WIR-CLD-REQ-350862/D-Wireless Interface Router Client3

The Wireless Interface Router Client3 (WIRClient3) is responsible for the tasks listed below:

* Handles connectivity request from applications
* Responsible for policy validation
* Responsible for setting up Tunnels
  + Routing table instead of Tunnels for AV
* Responsible for handling WIFI connection
* Interfaces with central connection controller
* Providing connections to applications when central controller is not active

Please review the implementation guide/ block diagram to locate the WIRClient3 class.

## WIR-CLD-REQ-370364/C-Wireless Interface Router Client4

The Wireless Interface Router Client4 (WIRClient4) is responsible for the tasks listed below:

* Handles connectivity request from applications
* Responsible for policy validation
* Responsible for setting up Tunnels
  + Routing table instead of Tunnels for AV
* Responsible for handling WIFI connection
* Interfaces with central connection controller
* Providing connections to applications when central controller is not active

Please review the implementation guide/ block diagram to locate the WIRClient4 class.

## WIR-CLD-REQ-370370/C-Wireless Interface Router Client5

The Wireless Interface Router Client5 (WIRClient5) is responsible for the tasks listed below:

* Handles connectivity request from applications
* Responsible for policy validation
* Responsible for setting up Tunnels
  + Routing table instead of Tunnels for AV
* Responsible for setting up Cellular APN1 and APN2
* Responsible for handling WIFI connection
* Interfaces with central connection controller
* Providing connections to applications when central controller is not active

Please review the implementation guide/ block diagram to locate the WIRClient5 class.

## Physical Mapping of Classes

The table below shows an example of how the logical classes that make up the WIR feature may be mapped into physical modules. This mapping is an example only (FNV2, Z2, FNV3) and does not necessarily carryover to other carlines or vehicle architectures.

|  |  |
| --- | --- |
| **Logical Class** | **Physical Module (ECU)** |
| WIRServer | ECG |
| WIRClient1 | APIM, DuerOS, PDC |
| WIRClient2 | TCU |
| WIRClient3 | SDS |
| WIRClient4 | ADSIM |
| WIRClient5 | TCU-B |

## WIRServer Interface

### WIR-IIR-REQ-289531/C-WIRServer\_Tx

#### MD-REQ-386195/B-PolicyTablePullRequestQuery

Message Type: FTCP

This query is used by the WIRServer to request a pull of the policy table.

**Note:** Refer to the latest “Ford Telematics Communication Protocol Specification” and Protofile for the most up to date FTCP messages/definitions.

|  |  |
| --- | --- |
| **Alert** | **Description** |
| PolicyTablePullRequestQuery | Alert sent to the cloud to pull the policy table |

#### MD-REQ-380247/A-CellActivate

This API is used internally between WIRClient and WIRServer to request activation of a cellular interface. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | CellApnType | | Enum | - | - | Available APNs for cellular data |
|  |  | |  | Internet | 0x0 | Metered User Internet APN |
|  |  | |  | Ford | 0x1 | Non-metered Ford APN |
|  |  | |  | Tethering | 0x2 | Metered Mobile Hot Spot APN |
|  |  | |  | FOTA | 0x3 | FOTA APN |
|  |  | |  | HTTP | 0x4 | HTTP APN |
| **Response** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | CellRet | | Enum | - | - | Return code |
|  |  | |  | CELL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CELL\_SUCCESS | 0x1 | Success |
| R | cause | | String | - | Char Value:0-255  No String length limit | Activation cause code |
| R | IfaceId | | Int32 | - | 0-4294967295 | Interface Id used internally by CM |
| R | IfaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | ipV4Address | | String | - | Char Value:0-255  No String length limit | Interface IP address V4 |
| R | ipV4DnsPrimary | | String | - | Char Value:0-255  No String length limit | Primary DNS IP address V4 |
| R | ipV4DnsSecondary | | String | - | Char Value:0-255  No String length limit | Secondary DNS IP address V4 |
| R | ipV4Gateway | | String | - | Char Value:0-255  No String length limit | Gateway IP address V4 |
| R | ipV6Address | | String | - | Char Value:0-255  No String length limit | Interface IP address V6 |
| R | ipV6DnsPrimary | | String | - | Char Value:0-255  No String length limit | Primary DNS IP address V6 |
| R | ipV6DnsSecondary | | String | - | Char Value:0-255  No String length limit | Secondary DNS IP address V6 |
| R | ipV6Gateway | | String | - | Char Value:0-255  No String length limit | Gateway IP address V6 |
| R | mtuIpV4 | | Int32 | - | 0-4294967295 | Interface MTU V4 |
| R | mtuIpV6 | | Int32 | - | 0-4294967295 | Interface MTU V6 |

#### MD-REQ-380249/A-CellDeactivate

This API is used internally between WIRClient and WIRServer to request deactivation of a cellular interface. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | CellApnType | | Enum | - | - | Available APNs for cellular data |
|  |  | |  | Internet | 0x0 | Metered User Internet APN |
|  |  | |  | Ford | 0x1 | Non-metered Ford APN |
|  |  | |  | Tethering | 0x2 | Metered Mobile Hot Spot APN |
|  |  | |  | FOTA | 0x3 | FOTA APN |
|  |  | |  | HTTP | 0x4 | HTTP APN |
| **Response** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | CellRet | | Enum | - | - | Return code |
|  |  | |  | CELL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CELL\_SUCCESS | 0x1 | Success |
| R | cause | | String | - | Char Value:0-255  No String length limit | Activation cause code |

#### MD-REQ-380252/A-CellStatistics

This API is used internally by WIRClient and WIRServer to request cellular traffic statistics of a cellular interface. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | StatsCmdType | | Enum | - | - | Command types for cellular data traffic statistics |
|  |  | |  | Disable | 0x0 | Disable reporting of statistics |
|  |  | |  | Enable | 0x1 | Enable reporting of statistics |
|  |  | |  | Query | 0x2 | Query Statistics |
| R | reportingWindow | | Int32 | - | 0-4294967295 | Reporting Window in seconds |
| **Response** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | CellRet | | Enum | - | - | Return code |
|  |  | |  | CELL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CELL\_SUCCESS | 0x1 | Success |
| R | PKTS\_TX\_V4 | | Int32 | - | 0-4294967295 | Transmitted IPv4 packets |
| R | PKTS\_RX\_V4 | | Int32 | - | 0-4294967295 | Received IPv4 packets |
| R | BYTES\_TX\_V4 | | Int64 | - | 0-18446744073709551615 | Transmitted IPv4 bytes |
| R | BYTES\_RX\_V4 | | Int64 | - | 0-18446744073709551615 | Received IPv4 bytes |
| R | PKTS\_DROPPED\_TX | | Int32 | - | 0-4294967295 | Dropped transmit IPv4 packets |
| R | PKTS\_DROPPED\_RX | | Int32 | - | 0-4294967295 | Dropped receive IPv4 packets |

#### MD-REQ-380258/C-VlanAdd

This API is used internally by WIRClient and WIRServer to request a new VLAN configuration. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | ifaceEcu | | String | - | Char Value:0-255  No String length limit | Interface ECU enum in string format |
| R | mtu | | String | - | Char Value:0-255  No String length limit | Interface MTU in string format |
| R | svcLevel | | String | - | Char Value:0-255  No String length limit | Service Level of the VLAN in string format |
| R | ifaceType | | Enum | - | - | Network Interface Type over which the VLAN would be setup |
|  |  | |  | None | 0x0 |  |
|  |  | |  | TcuCell | 0x1 |  |
|  |  | |  | TcuWifi | 0x2 |  |
|  |  | |  | SyncWifi | 0x3 |  |
|  |  | |  | SyncApplink | 0x4 | Out of scope |
|  |  | |  | TcubCell | 0x5 | Applicable to AV vehicles Only |
|  |  | |  | TcubWifi | 0x6 | Applicable to AV vehicles Only |
|  |  | |  | EspmLan | 0x7 | ESPM+ LAN  Applicable to AV vehicles Only |
|  |  | |  | DdsmLan | 0x8 | DDSM LAN  Applicable to AV vehicles Only |
|  |  | |  | Error | 0x9 |  |
| R | vlanEndpointType | | Enum | - | - | VLAN Endpoint type |
|  |  | |  | Host | 0x0 | Host type endpoint |
|  |  | |  | Gateway | 0x1 | Gateway type endpoint |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | ipV4Address | | String | - | Char Value:0-255  No String length limit | VLAN IP Address |

#### MD-REQ-380259/A-VlanRemove

This API is used internally by WIRClient and WIRServer to request removal of old VLAN configuration. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |

#### MD-REQ-380260/A-VlanLink

This API is used internally by WIRClient and WIRServer to request link VLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Network Interface as enumerated in the OS |
| R | serviceLevel | | String | - | Char Value:0-255  No String length limit | Service Level of the VLAN in string format |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Network Interface as enumerated in the OS |
| R | serviceLevel | | String | - | Char Value:0-255  No String length limit | Service Level of the VLAN in string format |
| R | linkId | | String |  | Char Value:0-255  No String length limit | Unique link ID generated by Networking |

#### MD-REQ-380261/A-VlanUnlink

This API is used internally by WIRClient and WIRServer to request unlink VLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | linkId | | String |  | Char Value:0-255  No String length limit | Unique link ID generated by Networking |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | linkId | | String |  | Char Value:0-255  No String length limit | Unique link ID generated by Networking |

#### MD-REQ-380262/C-VnmReset

This API is used internally by WIRClient and WIRServer to request a VLAN configuration reset. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | EcuType | | Enum | - | - | Path Id used internally by CM |
|  |  | |  | Unknown | 0x0 | Error |
|  |  | |  | ECG | 0x1 |  |
|  |  | |  | TCU | 0x2 |  |
|  |  | |  | SYNC | 0x3 |  |
|  |  | |  | TCU\_B | 0x4 | Applicable to AV vehicles Only |
|  |  | |  | SDS | 0x5 | Applicable to AV vehicles Only |
|  |  | |  | ADSIM | 0x6 | Applicable to AV vehicles Only |
|  |  | |  | DDSM | 0x7 | Applicable to AV vehicles Only |
|  |  | |  | ADM | 0x8 | Applicable to AV vehicles Only |
|  |  | |  | ESPM | 0x9 | Applicable to AV vehicles |
|  |  | |  | SDM | 0x10 | Applicable to AV vehicles |
| **Response** | | | | | | |
| R | EcuType | | Enum | - | - | Path Id used internally by CM |
|  |  | |  | Unknown | 0x0 | Error |
|  |  | |  | ECG | 0x1 |  |
|  |  | |  | TCU | 0x2 |  |
|  |  | |  | SYNC | 0x3 |  |
|  |  | |  | TCU\_B | 0x4 |  |
|  |  | |  | SDS | 0x5 |  |
|  |  | |  | ADSIM | 0x6 |  |
|  |  | |  | DDSM | 0x7 |  |
|  |  | |  | ADM | 0x8 |  |
|  |  | |  | ESPM | 0x9 | Applicable to AV vehicles |
|  |  | |  | SDM | 0x10 | Applicable to AV vehicles |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |

#### MD-REQ-380263/A-VnmLinkNetwork

This API is used internally by WIRClient and WIRServer to request to link a network to VLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | ifaceEcu | | String | - | Char Value:0-255  No String length limit | Interface ECU enum in string format |
| R | interfaceZero | | String | - | Char Value:0-255  No String length limit | User’s provided interface name to be linked |
| R | interfaceOne | | String | - | Char Value:0-255  No String length limit | Network interface to be linked |
| R | ipAddr | | String | - | Char Value:0-255  No String length limit | Gateway ip address associated with network interface |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | linkId | | String | - | Char Value:0-255  No String length limit | Unique link ID generated by Networking |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |

#### MD-REQ-380264/A-VnmUnlinkNetwork

This API is used internally by WIRClient and WIRServer to request to unlink a network to VLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | ifaceEcu | | String | - | Char Value:0-255  No String length limit | Interface ECU enum in string format |
| R | linkId | | String | - | Char Value:0-255  No String length limit | Unique link ID generated by Networking |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |

#### MD-REQ-380265/D-NetworkInterfaceAllocation

This API is used internally by WIRClient and WIRServer to request a new network interface. WIR also uses this API for its response. The response will return success or failure, actual interface is returned through NetworkInterfaceAllocationStatusInd.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | | One-Shot (A-Synch) | | | |
| **QoS Level** | | | Default | | | |
| **Retained** | | | No | | | |
|  | | | | | | |
| **R/O** | **Name** | **Type** | | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | String | | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | Int32 | | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| R | IntentType | Enum | | - | - | Intent Type |
|  |  |  | | Foreground | 0x0 |  |
|  |  |  | | Background\_  BestEffort | 0x1 |  |
|  |  |  | | Background\_  Guaranteed | 0x2 |  |
|  |  |  | | Special | 0x3 | N/A |
|  |  |  | | OffPeak | 0x4 | N/A for AV vehicles |
|  |  |  | | LAN\_Only | 0x5 | Applicable to AV vehicles Only |
| R | InterfaceType | Enum | | - | - | Interface Type |
|  |  |  | | None | 0x0 |  |
|  |  |  | | TcuCell | 0x1 |  |
|  |  |  | | TcuWifi | 0x2 |  |
|  |  |  | | SyncWifi | 0x3 |  |
|  |  |  | | SyncApplink | 0x4 | Out of scope |
|  |  |  | | TcubCell | 0x5 | Applicable to AV vehicles Only |
|  |  |  | | TcubWifi | 0x6 | Applicable to AV vehicles Only |
|  |  |  | | EspmLan | 0x7 | Applicable to AV vehicles Only |
|  |  |  | | DdsmLan | 0x8 | Applicable to AV vehicles Only |
|  |  |  | | Error | 0x9 |  |
| R | CellApnType | Enum | | - | - | Cellular APN Type |
|  |  |  | | Internet | 0x0 |  |
|  |  |  | | Ford | 0x1 |  |
|  |  |  | | Tethering | 0x2 |  |
|  |  |  | | FOTA | 0x3 |  |
|  |  |  | | HTTP | 0x4 |  |
| R | PriorityLevel | Enum | | - | - | Priority Level |
|  |  |  | | Priority 0 | 0x0 |  |
|  |  |  | | Priority 1 | 0x1 |  |
|  |  |  | | Priority 2 | 0x2 |  |
|  |  |  | | Priority 3 | 0x3 |  |
|  |  |  | | Priority 4 | 0x4 |  |
| R | expiry | Int32 | | - | 0-4294967295 | Expiration timer |
| R | OffPeakFlag | Enum | | - | - | Offpeak flag. N/A for AV vehicles |
|  |  |  | | No | 0x0 |  |
|  |  |  | | Yes | 0x1 |  |
| R | CellularOnlyFlag | Enum | | - | - | Cellular Only Flag |
|  |  |  | | No | 0x0 |  |
|  |  |  | | Yes | 0x1 |  |
| R | WifiPreferredFlag | Enum | | - | - | Wifi preferred flag |
|  |  |  | | No | 0x0 |  |
|  |  |  | | Yes | 0x1 |  |
| R | ssid | String | | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | String | |  | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | Enum | | - | - | Optional - Channel of AP |
|  |  |  | | Reserved | 0x00 |  |
|  |  |  | | WLAN\_CH\_1 | 0x01 |  |
|  |  |  | | WLAN\_CH\_2 | 0x02 |  |
|  |  |  | | WLAN\_CH\_3 | 0x03 |  |
|  |  |  | | … | … |  |
|  |  |  | | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | Enum | | - | - | Security settings to use |
|  |  |  | | Reserved | 0x00 |  |
|  |  |  | | Open | 0x01 | Open or no security |
|  |  |  | | WEP | 0x02 | WEP |
|  |  |  | | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  |  | | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  |  | | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  |  | | MAX | 0x06 |  |
| R | WlanWepSecurity |  | |  |  | WEP Settings |
|  |  | String | | Key | Char Value:0-255  No String length limit |  |
|  |  | Int32 | | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity |  | |  |  | WPS Settings |
|  |  | WlanWpsType | |  |  | Type |
|  |  | String | |  | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | String | |  | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | Enum | | - | - | Pair cipher |
|  |  |  | | PAIR\_MIN | 0x0 |  |
|  |  |  | | PAIR\_NONE | 0x1 | None |
|  |  |  | | PAIR\_TKIP | 0x2 | TKIP |
|  |  |  | | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  |  | | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  |  | | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | Enum | | - | - | Group cipher |
|  |  |  | | GROUP\_MIN | 0x0 |  |
|  |  |  | | GROUP\_NONE | 0x1 | None |
|  |  |  | | GROUP\_TKIP | 0x2 | TKIP |
|  |  |  | | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  |  | | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  |  | | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | TBD | | - | - | TBD |
| R | WlanIpv4AddrType | Enum | | - | - |  |
|  |  |  | | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  |  | | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  |  | | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  |  | | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  |  | | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  |  | | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | - | | - | - |  |
|  |  | String | | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | String | | Netmask | Char Value:0-255  No String length limit | Netmask of current connection |
|  |  | String | | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | String | | dnsPref | Char Value:0-255  No String length limit | Preferred DNS server |
|  |  | String | | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | Enum | | - | - |  |
|  |  |  | | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  |  | | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  |  | | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  |  | | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | TBD | | - | - | TBD |
| R | ExclusiveUse | Boolean | | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | Enum | | - | - | Origin of the profile |
|  |  |  | | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  |  | | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  |  | | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  |  | | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  |  | | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | Int64 | | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | Boolean | | - | 0/1 | Hidden AP |
| R | IfaceNametoLink | String | | - | Char Value:0-255  No String length limit | AP SSID |
| R | data\_size | Int32 | | - | 0-4294967295 | Data size, if applicable. Applicable to AV vehicles Only |
| R | Bw\_downlink | Int32 | | - | 0-4294967295 | Application required downlink bandwidth. Applicable to AV vehicles Only |
| R | Bw\_uplink | Int32 | | - | 0-4294967295 | Application required uplink bandwidth. Applicable to AV vehicles Only |
| **Response** | | | | | | |
| R | appId | String | | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | Int32 | | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| R | WirRet | Enum | | - | - | Return Command |
|  |  |  | | WIR\_ERROR | 0x0 | Error/Failure |
|  |  |  | | WIR\_SUCCESS | 0x1 | Success |

#### MD-REQ-380266/A-NetworkInterfaceRelease

This API is used internally by WIRClient and WIRServer to request a network interface release. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | | Int32 | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | | Int32 | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| R | WirRet | | Enum | - | - | Return Command |
|  |  | |  | WIR\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WIR\_SUCCESS | 0x1 | Success |

#### MD-REQ-380267/C-NetworkInterfaceAllocationStatusInd

This API is used internally by WIRClient and WIRServer to broadcast network interface allocation status.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | | OnChange | | | |
| **QoS Level** | | | Default | | | |
| **Retained** | | | No | | | |
|  | | | | | | |
| **R/O** | **Name** | **Type** | | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | - | | - | - | N/A |
| **Response** | | | | | | |
| R | appId | String | | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | Int32 | | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| R | NetworkInterface  AllocationStatus | Enum | | - | - | Network Allocation Status |
|  |  |  | | NET\_IFACE\_ALLOC\_FAILURE | 0x0 | Error/Failure |
|  |  |  | | NET\_IFACE\_ALLOC\_SUCCESS | 0x1 | Success |
|  |  |  | | NET\_IFACE\_ALLOC\_INQUEUE | 0x2 | In queue/In progress |
|  |  |  | | NET\_IFACE\_ALLOC\_ILLEGAL | 0x3 | Not supported by policy |
| R | NetworkInterfaceType | Enum | | - | - | Interface Type |
|  |  |  | | IFACE\_NONE | 0x0 | None selected/specified |
|  |  |  | | IFACE\_TCUCELL | 0x1 | Cellular interface on TCU |
|  |  |  | | IFACE\_TCUWIFI | 0x2 | WLAN interface on TCU |
|  |  |  | | IFACE\_SYNCWIFI | 0x3 | WLAN interface on SYNC |
|  |  |  | | IFACE\_SYNCAPPL | 0x4 | AppLink interface on SYNC (currently out of scope for CM) |
|  |  |  | | IFACE\_TCUBCELL | 0x5 | Cellular interface on TCU-B. Applicable to AV vehicles Only |
|  |  |  | | IFACE\_TCUBWIFI | 0x6 | WLAN interface on TCU-B. Applicable to AV vehicles Only |
|  |  |  | | IFACE\_ESPMLAN | 0x7 | ESPM+ LAN. Applicable to AV vehicles Only |
|  |  |  | | IFACE\_DDSMLAN | 0x8 | DDSM LAN.  Applicable to AV vehicles Only |
| R | ipAddr | String | | - | Char Value:0-255  No String length limit | IP Address assigned to new network interface |

#### MD-REQ-380268/C-NetworkInterfaceDownInd

This API is used internally by WIRClient and WIRServer to broadcast network interface down status.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | | OnChange | | | |
| **QoS Level** | | | Default | | | |
| **Retained** | | | No | | | |
|  | | | | | | |
| **R/O** | **Name** | **Type** | | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | - | | - | - | N/A |
| **Response** | | | | | | |
| R | appId | String | | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | Int32 | | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| R | NetworkInterface  AllocationStatus | Enum | | - | - | Network Allocation Status |
|  |  |  | | NET\_IFACE\_ALLOC\_FAILURE | 0x0 | Error/Failure |
|  |  |  | | NET\_IFACE\_ALLOC\_SUCCESS | 0x1 | Success |
|  |  |  | | NET\_IFACE\_ALLOC\_INQUEUE | 0x2 | In queue/In progress |
|  |  |  | | NET\_IFACE\_ALLOC\_ILLEGAL | 0x3 | Not supported by policy |
| R | NetworkInterfaceType | Enum | | - | - | Interface Type |
|  |  |  | | IFACE\_NONE | 0x0 | None selected/specified |
|  |  |  | | IFACE\_TCUCELL | 0x1 | Cellular interface on TCU |
|  |  |  | | IFACE\_TCUWIFI | 0x2 | WLAN interface on TCU |
|  |  |  | | IFACE\_SYNCWIFI | 0x3 | WLAN interface on SYNC |
|  |  |  | | IFACE\_SYNCAPPL | 0x4 | AppLink interface on SYNC (currently out of scope for CM) |
|  |  |  | | IFACE\_TCUBCELL | 0x5 | Cellular interface on TCU-B. Applicable to AV vehicles Only |
|  |  |  | | IFACE\_TCUBWIFI | 0x6 | WLAN interface on TCU-B. Applicable to AV vehicles Only |
|  |  |  | | IFACE\_ESPMLAN | 0x7 | ESPM+ LAN. Applicable to AV vehicles Only |
|  |  |  | | IFACE\_DDSMLAN | 0x8 | DDSM LAN.  Applicable to AV vehicles Only |

#### MD-REQ-380269/C-NetworkInterfaceUpInd

This API is used internally by WIRClient and WIRServer to broadcast network interface up status.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | | OnChange | | | |
| **QoS Level** | | | Default | | | |
| **Retained** | | | No | | | |
|  | | | | | | |
| **R/O** | **Name** | **Type** | | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | - | | - | - | N/A |
| **Response** | | | | | | |
| R | appId | String | | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | Int32 | | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| R | NetworkInterface  AllocationStatus | Enum | | - | - | Network Allocation Status |
|  |  |  | | NET\_IFACE\_ALLOC\_FAILURE | 0x0 | Error/Failure |
|  |  |  | | NET\_IFACE\_ALLOC\_SUCCESS | 0x1 | Success |
|  |  |  | | NET\_IFACE\_ALLOC\_INQUEUE | 0x2 | In queue/In progress |
|  |  |  | | NET\_IFACE\_ALLOC\_ILLEGAL | 0x3 | Not supported by policy |
| R | NetworkInterfaceType | Enum | | - | - | Interface Type |
|  |  |  | | IFACE\_NONE | 0x0 | None selected/specified |
|  |  |  | | IFACE\_TCUCELL | 0x1 | Cellular interface on TCU |
|  |  |  | | IFACE\_TCUWIFI | 0x2 | WLAN interface on TCU |
|  |  |  | | IFACE\_SYNCWIFI | 0x3 | WLAN interface on SYNC |
|  |  |  | | IFACE\_SYNCAPPL | 0x4 | AppLink interface on SYNC (currently out of scope for CM) |
|  |  |  | | IFACE\_TCUBCELL | 0x5 | Cellular interface on TCU-B. Applicable to AV vehicles Only |
|  |  |  | | IFACE\_TCUBWIFI | 0x6 | WLAN interface on TCU-B. Applicable to AV vehicles Only |
|  |  |  | | IFACE\_ESPMLAN | 0x7 | ESPM+ LAN. Applicable to AV vehicles Only |
|  |  |  | | IFACE\_DDSMLAN | 0x8 | DDSM LAN.  Applicable to AV vehicles Only |
| R | ipAddr | String | | - | Char Value:0-255  No String length limit | IP Address assigned to new network interface |

#### MD-REQ-380272/A-WebViewDisplay

This API is used internally by WIRClient and WIRServer to request WEB View display. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | ipAddr | | String | - | Char Value:0-255  No String length limit | IP address to be used for connection |
| R | uri | | String | - | Char Value:0-255  No String length limit | WEB page URI |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | WirRet | | Enum | - | - | Return Command |
|  |  | |  | WIR\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WIR\_SUCCESS | 0x1 | Success |

#### MD-REQ-380274/C-Policy

This API is used internally by WIRClient and WIRServer to request a network policy for an application. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| **Response** | | | | | | |
| R | CtrlRet | | Enum | - | - |  |
|  |  | |  | CTRL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CTRL\_SUCCESS | 0x1 | Success |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | Policy | | Int32 | - | 0-4294967295 | Network policy, bit flag |
|  |  | |  | TCU\_CELLULAR\_APN1 | 0x0001 |  |
|  |  | |  | WIFI | 0x0002 |  |
|  |  | |  | TCU\_CELLULAR\_APN2 | 0x0004 |  |
|  |  | |  | DNS\_RPZ\_EXCEPTION | 0x0008 |  |
|  |  | |  | ESPM\_LAN | 0x0010 | Applicable to AV vehicles Only |
|  |  | |  | DDSM\_LAN | 0x0020 | Applicable to AV vehicles Only |
|  |  | |  | OFF\_PEAK | 0x0040 | N/A |
|  |  | |  | RESERVED | 0x0080 |  |
|  |  | |  | RESERVED | 0x0100 |  |
|  |  | |  | FACTORY\_WIFI | 0x0200 |  |
|  |  | |  | TCU\_B\_CELLULAR\_APN1 | 0x0400 | Applicable to AV vehicles Only |
|  |  | |  | TCU\_B\_CELLULAR\_APN2 | 0x0800 | Applicable to AV vehicles Only |
|  |  | |  | WIFI\_CONNECTION\_CHANGE\_REPORT | 0x1000 |  |
|  |  | |  | SYSTEM\_APP | 0x2000 |  |
|  |  | |  | RESERVED | 0x4000 |  |
|  |  | |  | RESERVED | 0x8000 |  |

#### MD-REQ-380275/C-PolicyInd

This API is used internally by WIRClient and WIRServer to broadcast a notification of a network policy change for an application.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | Policy | | Int32 | - | 0-4294967295 | Network policy, bit flag |
|  |  | |  | TCU\_CELLULAR\_APN1 | 0x0001 |  |
|  |  | |  | WIFI | 0x0002 |  |
|  |  | |  | TCU\_CELLULAR\_APN2 | 0x0004 |  |
|  |  | |  | DNS\_RPZ\_EXCEPTION | 0x0008 |  |
|  |  | |  | ESPM\_LAN | 0x0010 | Applicable to AV vehicles Only |
|  |  | |  | DDSM\_LAN | 0x0020 | Applicable to AV vehicles Only |
|  |  | |  | OFF\_PEAK | 0x0040 | N/A |
|  |  | |  | RESERVED | 0x0080 |  |
|  |  | |  | RESERVED | 0x0100 |  |
|  |  | |  | FACTORY\_WIFI | 0x0200 |  |
|  |  | |  | TCU\_B\_CELLULAR\_APN1 | 0x0400 | Applicable to AV vehicles Only |
|  |  | |  | TCU\_B\_CELLULAR\_APN2 | 0x0800 | Applicable to AV vehicles Only |
|  |  | |  | WIFI\_CONNECTION\_CHANGE\_REPORT | 0x1000 |  |
|  |  | |  | SYSTEM\_APP | 0x2000 |  |
|  |  | |  | RESERVED | 0x4000 |  |
|  |  | |  | RESERVED | 0x8000 |  |

#### MD-REQ-380276/C-PolicyTableInd

This API is used internally by WIRClient and WIRServer to broadcast a notification of a network policy table update.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | Num\_entry | | Int32 |  | 0-4294967295 | Number of entries for following two fields |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | Policy | | Int32 | - | 0-4294967295 | Network policy, bit flag |
|  |  | |  | TCU\_CELLULAR\_APN1 | 0x0001 |  |
|  |  | |  | WIFI | 0x0002 |  |
|  |  | |  | TCU\_CELLULAR\_APN2 | 0x0004 |  |
|  |  | |  | DNS\_RPZ\_EXCEPTION | 0x0008 |  |
|  |  | |  | ESPM\_LAN | 0x0010 | Applicable to AV vehicles Only |
|  |  | |  | DDSM\_LAN | 0x0020 | Applicable to AV vehicles Only |
|  |  | |  | OFF\_PEAK | 0x0040 | N/A |
|  |  | |  | RESERVED | 0x0080 |  |
|  |  | |  | RESERVED | 0x0100 |  |
|  |  | |  | FACTORY\_WIFI | 0x0200 |  |
|  |  | |  | TCU\_B\_CELLULAR\_APN1 | 0x0400 | Applicable to AV vehicles Only |
|  |  | |  | TCU\_B\_CELLULAR\_APN2 | 0x0800 | Applicable to AV vehicles Only |
|  |  | |  | WIFI\_CONNECTION\_CHANGE\_REPORT | 0x1000 |  |
|  |  | |  | SYSTEM\_APP | 0x2000 |  |
|  |  | |  | RESERVED | 0x4000 |  |
|  |  | |  | RESERVED | 0x8000 |  |
| R | policyTableVersionMajor | | Int32 | - | 0-4294967295 | Policy Table Version Major |
| R | policyTableVersionMinor | | Int32 | - | 0-4294967295 | Policy Table Version Minor |

#### MD-REQ-380277/A-PolicyTableStatusInd

This API is used internally by WIRClient and WIRServer to broadcast a response to policy table indication.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | CtrlRet | | Enum | - | - |  |
|  |  | |  | CTRL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CTRL\_SUCCESS | 0x1 | Success |

#### MD-REQ-380278/C-InterfaceTableInd

This API is used internally by WIRClient and WIRServer to broadcast available network interfaces table update indication.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | Num\_entry | | Int32 | - | 0-4294967295 | Number of entries for following two fields |
| R | ifaceTable | | Int32 | - | 0-4294967295 | Bit flag of available interfaces |
|  |  | |  | TCU\_CELLULAR\_APN1 | 0x0001 |  |
|  |  | |  | WIFI | 0x0002 |  |
|  |  | |  | TCU\_CELLULAR\_APN2 | 0x0004 |  |
|  |  | |  | ESPM\_LAN | 0x0010 | Applicable to AV vehicles Only |
|  |  | |  | DDSM\_LAN | 0x0020 | Applicable to AV vehicles Only |
|  |  | |  | OFF\_PEAK | 0x0040 | N/A |
|  |  | |  | FACTORY\_WIFI | 0x0200 | Applicable to AV vehicles Only |
|  |  | |  | TCU\_B\_CELLULAR\_APN1 | 0x0400 | Applicable to AV vehicles Only |
|  |  | |  | TCU\_B\_CELLULAR\_APN2 | 0x0800 | Applicable to AV vehicles Only |
| R | policyTableVersionMajor | | Int32 | - | 0-4294967295 | Policy Table Version Major |
| R | policyTableVersionMinor | | Int32 | - | 0-4294967295 | Policy Table Version Minor |

#### MD-REQ-380279/A-Enable

This API is used internally by WIRClient and WIRServer to request enabling WLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380280/A-Disable

This API is used internally by WIRClient and WIRServer to request disabling WLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380281/B-ScanApimAPs

This API is used internally by WIRClient and WIRServer to request a scan of available APs. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | scanId | | Int64 | - | 0-18446744073709551615 | ID value used to track scans |
| R | passive | | Boolean | - | 0/1 | true = passive scan (no pkts tx'd), false = active scan |
| R | WlanChannel | | Enum | - | - | Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | ssid | | String | - | Char Value:0-255  No String length limit | list of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required |
| R | Count | | Int32 | - | 0-4294967295 | Number of APs found |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID – scan result |
| R | bssid | | String | - | Char Value:0-255  No String length limit | MAC address of AP – scan result |
| R | WlanChannel | | Enum | - | - | Channel of AP – scan result |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | Wlan Bandwidth | | Enum | - | - |  |
|  |  | |  | WLAN\_BW\_MIN | 0x0 |  |
|  |  | |  | WLAN\_BW\_MHZ20 | 0x1 | 20MHz channel |
|  |  | |  | WLAN\_BW\_MHZ40 | 0x2 | 40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz |
|  |  | |  | WLAN\_BW\_MHZ80 | 0x3 | 80MHz channel; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ8080 | 0x4 | 80-80MHz, 2 non-contiguous 80MHz; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ160 | 0x5 | 160MHz channel; must 11ac |
|  |  | |  | WLAN\_BW\_MAX | 0x6 |  |
| R | rssi | | Int32 | - | 0-4294967295 | RSSI of beacon/probe response |
| R | WlanSecurity | | Enum | - | - | Security settings of scan results |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | isWpsSupported | | Boolean | - | 0/1 | does AP support WPS |
| R | isEssSupported | | Boolean | - | 0/1 | is AP part of an Extended Service Set |
| R | ipV4Addr | | String | - | Char Value:0-255  No String length limit | IPv4 address |
| R | netmask | | String | - | Char Value:0-255  No String length limit | Net mask |
| R | connected | | Boolean | - | 0/1 | connected to AP |
| R | prevConnected | | Boolean | - | 0/1 | Previously connected AP |
| R | isHidden | | Boolean | - | 0/1 | Hidden AP |
| R | WifiConnectionType | | Enum | - | - |  |
|  |  | |  | CONNECTION\_TYPE\_PIN | 0x0 | PIN |
|  |  | |  | CONNECTION\_TYPE\_WPS\_PIN | 0x1 | WPS (Wi-Fi Protected Setup) PIN |
|  |  | |  | CONNECTION\_TYPE\_WPS\_PBC | 0x2 | WPS Push Button Control |
|  |  | |  | CONNECTION\_TYPE\_OPEN | 0x3 | Open WiFi |
|  |  | |  | CONNECTION\_TYPE\_HIDDEN\_SECURE | 0x4 | Hidden + security, SSID is not broadcasted |
|  |  | |  | CONNECTION\_TYPE\_HIDDEN\_OPEN | 0x5 | Hidden + open, SSID is not broadcasted |
| R | EWifiConnectionStatus | | Enum | - | - |  |
|  |  | |  | WIFI\_STATUS\_SUCCESS | 0 | Success |
|  |  | |  | WIFI\_AP\_SCAN\_SUCCESS | 1 | Scan Successful |
|  |  | |  | WIFI\_NO\_APS\_AVAILABLE | 2 | No Access Points available |
|  |  | |  | WIFI\_CONNECTED | 3 | Connected to Access Point |
|  |  | |  | WIFI\_NOT\_CONNECTED | 4 | Wifi is not connected to Access Point |
|  |  | |  | WIFI\_OTH\_FAIL | 5 | Authentication Failure |
|  |  | |  | WIFI\_CONNECTING | 6 | Connecting to Access Point |
|  |  | |  | WIFI\_SECURITY\_MISSMATCH | 7 | There is security type mismatch between what requested and what is in actual |
|  |  | |  | WIFI\_AUTHENTICATING | 8 | In authenticating |
|  |  | |  | WIFI\_ABORTED | 9 | Connection aborted |
|  |  | |  | WIFI\_WRONG\_PASSWORD | 10 | Failed with wrong password |
|  |  | |  | WIFI\_WRONG\_WPS\_PIN | 11 | Failed with wrong WPS PIN |
|  |  | |  | WIFI\_CONNECTION\_PROGRESS | 12 | In connecting |
|  |  | |  | WIFI\_RESTRICTED\_NETWORK | 13 | Connection failure due to trying to connect to a restricted network eg : TCU hotspot |
|  |  | |  | WIFI\_NOT\_IMPLEMENTED | 101 | Not Implemented |
|  |  | |  | WIFI\_STATUS\_FAIL | 102 | Failure |
|  |  | |  | WIFI\_STATUS\_FAIL\_OTHER | 203 | Other Failure or Connecting to AP Failed |
|  |  | |  | WIFI\_CANCEL\_CONNECTION | 104 | Cancelled |
| R | complete | | Boolean | - | 0/1 | TRUE - complete  FALSE- more to come |

#### MD-REQ-380282/A-CancelScanApimAPs

This API is used internally by WIRClient and WIRServer to request to cancel scanning for available APs. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380283/B-ConnectToApimAP

This API is used internally by WIRClient and WIRServer to request to connect to an available AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType |  |  | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - | - | TBD |
| R | WlanIpv4AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |
| **Response** | | | | | | |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID – scan result |
| R | bssid | | String | - | Char Value:0-255  No String length limit | MAC address of AP – scan result |
| R | WlanChannel | | Enum | - | - | Channel of AP – scan result |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | Wlan Bandwidth | | Enum | - | - |  |
|  |  | |  | WLAN\_BW\_MIN | 0x0 |  |
|  |  | |  | WLAN\_BW\_MHZ20 | 0x1 | 20MHz channel |
|  |  | |  | WLAN\_BW\_MHZ40 | 0x2 | 40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz |
|  |  | |  | WLAN\_BW\_MHZ80 | 0x3 | 80MHz channel; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ8080 | 0x4 | 80-80MHz, 2 non-contiguous 80MHz; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ160 | 0x5 | 160MHz channel; must 11ac |
|  |  | |  | WLAN\_BW\_MAX | 0x6 |  |
| R | rssi | | Int32 | - | 0-4294967295 | RSSI of beacon/probe response |
| R | WlanSecurity | | Enum | - | - | Security settings of scan results |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | isWpsSupported | | Boolean | - | 0/1 | does AP support WPS |
| R | isEssSupported | | Boolean | - | 0/1 | is AP part of an Extended Service Set |
| R | ipV4Addr | | String | - | Char Value:0-255  No String length limit | IPv4 address |
| R | netmask | | String | - | Char Value:0-255  No String length limit | Net mask |
| R | connected | | Boolean | - | 0/1 | connected to AP |
| R | prevConnected | | Boolean | - | 0/1 | Previously connected AP |
| R | isHidden | | Boolean | - | 0/1 | Hidden AP |
| R | WifiConnectionType | | Enum | - | - |  |
|  |  | |  | CONNECTION\_TYPE\_PIN | 0x0 | PIN |
|  |  | |  | CONNECTION\_TYPE\_WPS\_PIN | 0x1 | WPS (Wi-Fi Protected Setup) PIN |
|  |  | |  | CONNECTION\_TYPE\_WPS\_PBC | 0x2 | WPS Push Button Control |
|  |  | |  | CONNECTION\_TYPE\_OPEN | 0x3 | Open WiFi |
|  |  | |  | CONNECTION\_TYPE\_HIDDEN\_SECURE | 0x4 | Hidden + security, SSID is not broadcasted |
|  |  | |  | CONNECTION\_TYPE\_HIDDEN\_OPEN | 0x5 | Hidden + open, SSID is not broadcasted |
| R | EWifiConnectionStatus | | Enum | - | - |  |
|  |  | |  | WIFI\_STATUS\_SUCCESS | 0 | Success |
|  |  | |  | WIFI\_AP\_SCAN\_SUCCESS | 1 | Scan Successful |
|  |  | |  | WIFI\_NO\_APS\_AVAILABLE | 2 | No Access Points available |
|  |  | |  | WIFI\_CONNECTED | 3 | Connected to Access Point |
|  |  | |  | WIFI\_NOT\_CONNECTED | 4 | Wifi is not connected to Access Point |
|  |  | |  | WIFI\_OTH\_FAIL | 5 | Authentication Failure |
|  |  | |  | WIFI\_CONNECTING | 6 | Connecting to Access Point |
|  |  | |  | WIFI\_SECURITY\_MISSMATCH | 7 | There is security type mismatch between what requested and what is in actual |
|  |  | |  | WIFI\_AUTHENTICATING | 8 | In authenticating |
|  |  | |  | WIFI\_ABORTED | 9 | Connection aborted |
|  |  | |  | WIFI\_WRONG\_PASSWORD | 10 | Failed with wrong password |
|  |  | |  | WIFI\_WRONG\_WPS\_PIN | 11 | Failed with wrong WPS PIN |
|  |  | |  | WIFI\_CONNECTION\_PROGRESS | 12 | In connecting |
|  |  | |  | WIFI\_RESTRICTED\_NETWORK | 13 | Connection failure due to trying to connect to a restricted network eg : TCU hotspot |
|  |  | |  | WIFI\_NOT\_IMPLEMENTED | 101 | Not Implemented |
|  |  | |  | WIFI\_STATUS\_FAIL | 102 | Failure |
|  |  | |  | WIFI\_STATUS\_FAIL\_OTHER | 203 | Other Failure or Connecting to AP Failed |
|  |  | |  | WIFI\_CANCEL\_CONNECTION | 104 | Cancelled |

#### MD-REQ-380284/A-CancelConnectToApimAP

This API is used internally by WIRClient and WIRServer to request to cancel connecting to an available AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380285/A-DisconnectFromApimAP

This API is used internally by WIRClient and WIRServer to request to disconnect from an available AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380286/A-ForgetApimAP

This API is used internally by WIRClient and WIRServer to request to forget an available AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | ssid | | String | - |  | SSID of AP |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380287/A-GetWpsPin

This API is used internally by WIRClient and WIRServer to request to get a WPS PIN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | WlanWpsSecurity | | Enum | - | - |  |
|  |  | |  | WLAN\_WPS\_MIN | 0x0 |  |
|  |  | |  | WLAN\_WPS\_KEYPAD | 0x1 | AP supplies the PIN |
|  |  | |  | WLAN\_WPS\_PIN | 0x2 | STA supplies the PIN |
|  |  | |  | WLAN\_WPS\_PBC | 0x3 | Pushbutton |
|  |  | |  | WLAN\_WPS\_MAX | 0x4 |  |
|  | pin | | String | - | Char Value:0-255  No String length limit | The Pin when using keypad type |

#### MD-REQ-380288/B-GetNetworkDetails

This API is used internally by WIRClient and WIRServer to request to get network details. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType | - |  | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - | - | TBD |
| R | WlanIpv4AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |

#### MD-REQ-380289/A-GetEnableStatus

This API is used internally by WIRClient and WIRServer to request to get enable status. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | status | | Boolean | - | 0/1 | True - enabled, false-disabled |

#### MD-REQ-380290/A-SetNotification

This API is used internally by WIRClient and WIRServer to request to set notification on/off. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | on | | Boolean | - | 0/1 | On or off |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380291/A-GetNotification

This API is used internally by WIRClient and WIRServer to request to get notification on/off status. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | on | | Boolean | - | 0/1 | On or off |

#### MD-REQ-380292/A-WifiAvailableInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that a WiFi AP is available.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |

#### MD-REQ-380293/A-WifiUnavailableInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that a WiFi AP is unavailable.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |

#### MD-REQ-380294/B-WifiConnectedInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that a WiFi client is connected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType | - |  | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA/WPA32-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - |  | TBD |
| R | WlanIpv4AddrType | | Enum | - |  |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |

#### MD-REQ-380295/A-WifiDisconnectedInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that a WiFi client is disconnected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |

#### MD-REQ-380296/A-WifiSignalStrengthInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify that AP’s RSSI was received from a WLAN client.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | rssi | | Int32 | - | 0-4294967295 | Signal Strength |

#### MD-REQ-380297/A-WifiMacAddressInd

This API is used internally by WIRClient and WIRServer to broadcast a notification to notify a client MAC address.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | macAddr | | String | - | Char Value:0-255  No String length limit | MAC Address |

#### MD-REQ-380298/C-WlanNQM

This API is used internally by WIRClient and WIRServer to request a WLAN network quality measurement. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | EcuType | | Enum | - | - | ECU type |
|  |  | |  | ECU\_UNK | 0x0 | Error |
|  |  | |  | ECU\_ECG | 0x1 | ECG |
|  |  | |  | ECU\_TCU | 0x2 | TCU |
|  |  | |  | ECU\_SYNC | 0x3 | SYNC |
|  |  | |  | ECU\_TCU\_B | 0x4 | TCU-B. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDS | 0x5 | SDS. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADSIM | 0x6 | ADSIM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_DDSM | 0x7 | DDSM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADM | 0x8 | ADM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ESPM | 0x9 | ESPM+. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDM | 0x10 | SDM. Applicable to AV vehicles Only |
| R | NetworkInterface  Type | | Enum | - | - |  |
|  |  | |  | IFACE\_NONE | 0x0 | None selected/specified |
|  |  | |  | IFACE\_TCUCELL | 0x1 | Cellular interface on TCU |
|  |  | |  | IFACE\_TCUWIFI | 0x2 | WLAN interface on TCU |
|  |  | |  | IFACE\_SYNCWIFI | 0x3 | WLAN interface on SYNC |
|  |  | |  | IFACE\_SYNCAPPL | 0x4 | AppLink interface on SYNC (currently out of scope for CM) |
|  |  | |  | IFACE\_TCUBCELL | 0x5 | Cellular interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_TCUBWIFI | 0x6 | WLAN interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ESPMLAN | 0x7 | ESPM+ LAN. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_DDSMLAN | 0x8 | DDSM LAN. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ERR | 0x9 | Error condition |
| **Response** | | | | | | |
| R | NqmRet | | Enum | - | - |  |
|  |  | |  | NQM\_ERROR | 0x0 | Error/Failure |
|  |  | |  | NQM\_SUCCESS | 0x1 | Success |
| R | EcuType | | Enum | - | - | ECU type |
|  |  | |  | ECU\_UNK | 0x0 | Error |
|  |  | |  | ECU\_ECG | 0x1 | ECG |
|  |  | |  | ECU\_TCU | 0x2 | TCU |
|  |  | |  | ECU\_SYNC | 0x3 | SYNC |
|  |  | |  | ECU\_TCU\_B | 0x4 | TCU-B |
|  |  | |  | ECU\_SDS | 0x5 | SDS |
|  |  | |  | ECU\_ADSIM | 0x6 | ADSIM |
|  |  | |  | ECU\_DDSM | 0x7 | DDSM |
|  |  | |  | ECU\_ADM | 0x8 | ADM |
|  |  | |  | ECU\_ESPM | 0x9 | ESPM+. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDM | 0x10 | SDM. Applicable to AV vehicles Only |
| R | NetworkInterface  Type | | Enum | - | - |  |
|  |  | |  | IFACE\_NONE | 0x0 | None selected/specified |
|  |  | |  | IFACE\_TCUCELL | 0x1 | Cellular interface on TCU |
|  |  | |  | IFACE\_TCUWIFI | 0x2 | WLAN interface on TCU |
|  |  | |  | IFACE\_SYNCWIFI | 0x3 | WLAN interface on SYNC |
|  |  | |  | IFACE\_SYNCAPPL | 0x4 | AppLink interface on SYNC (currently out of scope for CM) |
|  |  | |  | IFACE\_TCUBCELL | 0x5 | Cellular interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_TCUBWIFI | 0x6 | WLAN interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ESPMLAN | 0x7 | ESPM+ LAN. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_DDSMLAN | 0x8 | DDSM LAN. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ERR | 0x9 | Error condition |
| R | WlanStaSmState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_SM\_STATE\_OFF | 0x0 | WLAN off |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ENABLING | 0x1 | Enabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ON | 0x2 | On |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTING | 0x3 | Connecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISABLING | 0x6 | Disabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ERROR | 0x7 | Error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_AUTHERROR | 0x8 | Authentication error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_NWNOTFOUND | 0x9 | Network not found |
| R | macAddr | | String | - | Char Value:0-255  No String length limit | MAC Address |
| R | WlanStaConnState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTED | 0x0 | Disconnected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_WPS\_ACTIVE | 0x1 | WPS pending |
|  |  | |  | WLAN\_STA\_CON\_STATE\_ASSOCIATING | 0x2 | Associating |
|  |  | |  | WLAN\_STA\_CON\_STATE\_IP\_ADDRESSING | 0x3 | Getting IP address |
|  |  | |  | WLAN\_STA\_CON\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_CON\_STATE\_AUTHERROR | 0x6 | Authentication error |
|  |  | |  | WLAN\_STA\_CON\_STATE\_NWNOTFOUND | 0x7 | Network not found |
| R | connSsid | | String | - | Char Value:0-255 | Connected AP's SSID |
| R | connBssid | | String | - | No String length limit | Connected AP's BSSID |
| R | conn80211Tech | | String | - | Char Value:0-255 | Connected AP's 802.11 technology, 802.11a, … |
| R | connChannel | | Int32 | - | 0-4294967295 | Connected AP's channel |
| R | bandwidth | | Int32 | - | 0-4294967295 | Connected AP's bandwidth |
| R | dataRate | | Int32 | - | 0-4294967295 | Connected AP's data rate (bps) |
| R | signalStrength | | Int32 | - | 0-4294967295 | RSSI (dBm) |
| R | signalNoise | | Int32 | - | 0-4294967295 | Noise level (dBm) |
| R | signalSNR | | Int32 | - | 0-4294967295 | SNR (dB) |
| R | ipAddr | | String | - | Char Value:0-255  No String length limit | IP address |
| R | ipSubnet | | String | - | Char Value:0-255  No String length limit | Subnet mask |
| R | ipGateway | | String | - | Char Value:0-255  No String length limit | GW IP address |
| R | ipDnsPref | | String | - | Char Value:0-255  No String length limit | IP address of primary DNS |
| R | ipDnsAlt | | String | - | Char Value:0-255  No String length limit | IP address of secondary DNS |

#### MD-REQ-380299/C-CellNQM

This API is used internally by WIRClient and WIRServer to request a cellular network quality measurement. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | EcuType | | Enum | - | - | ECU type |
|  |  | |  | ECU\_UNK | 0x0 | Error |
|  |  | |  | ECU\_ECG | 0x1 | ECG |
|  |  | |  | ECU\_TCU | 0x2 | TCU |
|  |  | |  | ECU\_SYNC | 0x3 | SYNC |
|  |  | |  | ECU\_TCU\_B | 0x4 | TCU-B. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDS | 0x5 | SDS. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADSIM | 0x6 | ADSIM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_DDSM | 0x7 | DDSM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADM | 0x8 | ADM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ESPM | 0x9 | ESPM+ |
|  |  | |  | ECU\_SDM | 0x10 | SDM |
| **Response** | | | | | | |
| R | NqmRet | | Enum | - | - |  |
|  |  | |  | NQM\_ERROR | 0x0 | Error/Failure |
|  |  | |  | NQM\_SUCCESS | 0x1 | Success |
| R | EcuType | | Enum | - | - | ECU type |
|  |  | |  | ECU\_UNK | 0x0 | Error |
|  |  | |  | ECU\_ECG | 0x1 | ECG |
|  |  | |  | ECU\_TCU | 0x2 | TCU |
|  |  | |  | ECU\_SYNC | 0x3 | SYNC |
|  |  | |  | ECU\_TCU\_B | 0x4 | TCU-B. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDS | 0x5 | SDS. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADSIM | 0x6 | ADSIM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_DDSM | 0x7 | DDSM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADM | 0x8 | ADM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ESPM | 0x9 | ESPM+ |
|  |  | |  | ECU\_SDM | 0x10 | SDM |
| R | mcc | | Int32 | - | 0-4294967295 | Mobile country code |
| R | mnc | | Int32 | - | 0-4294967295 | Mobile network code |
| R | three\_digit\_mnc | | Boolean | - | 0/1 | 3 digits MNC or 2 digits MNC |
| R | CellularNwType | | Enum | - | - | Cellular Network Type |
|  |  | |  | NO\_NW\_TYPE | 0x0 | No network service |
|  |  | |  | GSM\_NW | 0x1 | GSM |
|  |  | |  | UMTS\_NW | 0x2 | UMTS |
|  |  | |  | LTE\_NW | 0x4 | LTE |
| R | CellularRatType | | Enum | - | - | Cellular RAT Type |
|  |  | |  | NO\_NW | 0x0 | No network available |
|  |  | |  | GSM | 0x1 | GSM |
|  |  | |  | GPRS | 0x2 | GPRS |
|  |  | |  | EDGE | 0x3 | EDGE |
|  |  | |  | UMTS | 0x4 | UMTS |
|  |  | |  | HSPA\_P | 0x5 | HSPA and HSPA+ |
|  |  | |  | LTE | 0x6 | LTE |
| R | signalStrength | | Int32 | - | 0-4294967295 | Signal Strength |
| R | CellularRegStatus | | Enum | - | - |  |
|  |  | |  | REG\_NO\_SRV | 0x0 | Registered but no service |
|  |  | |  | LIMITED\_SRV | 0x1 | Lmited service |
|  |  | |  | SRV | 0x2 | Full service |
| R | CellularSrvStatus | | Enum | - | - |  |
|  |  | |  | NO\_SRV | 0x0 | No service |
|  |  | |  | CS\_ONLY | 0x1 | CS only |
|  |  | |  | PS\_ONLY | 0x2 | PS only |
|  |  | |  | CS\_PS | 0x3 | CS and PS |
| R | CellularPwrStatus | | Enum | - | - |  |
|  |  | |  | MODEM\_PWR\_OFF | 0x0 | Modem Power Off |
|  |  | |  | MODEM\_PWR\_CELL\_OFF | 0x1 | Low Power Mode - Cellular Off |
|  |  | |  | MODEM\_PWR\_CELL\_ON | 0x2 | Cellular On |
|  |  | |  | MODEM\_PWR\_RESETTING | 0x3 | Power Reset |
|  |  | |  | MODEM\_PWR\_TESTING | 0x4 | Factory Test Mode |
|  |  | |  | MODEM\_PWR\_LPM\_IMMINENT | 0x5 | Just prior to Low Power Mode |
|  |  | |  | MODEM\_PWR\_RESET\_IMMINENT | 0x6 | Just prior to Resetting |
|  |  | |  | MODEM\_PWR\_RESET\_COMPLETE | 0x7 | modem back in service |

#### MD-REQ-380304/B-ScanWiFiNetworks

This API is used internally by WIRClient and WIRServer to request a WiFi network scan. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | passive | | Boolean | - | 0/1 | true = passive scan (no pkts tx'd), false = active scan |
| R | WlanChannel | | Enum | - | - | List of channels of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | ssid | | String | - | Char Value:0-255  No String length limit | List of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required |
| R | scanId | | Int64 | - | 0-18446744073709551615 | ID value used to track scans |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | scanId | | Int64 |  | 0-18446744073709551615 | ID value used to track scans |
| R | WlanChannel | | Enum | - | - | List of channels of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | ssid | | String | - | Char Value:0-255  No String length limit | List of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required |
| R | count | | Int32 | - | 0-4294967295 | number of APs found, repeats ap field |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID – scan result |
| R | bssid | | String | - | Char Value:0-255  No String length limit | MAC address of AP – scan result |
| R | WlanChannel | | Enum | - | - | Channel of AP – scan result |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | Wlan Bandwidth | | Enum | - | - |  |
|  |  | |  | WLAN\_BW\_MIN | 0x0 |  |
|  |  | |  | WLAN\_BW\_MHZ20 | 0x1 | 20MHz channel |
|  |  | |  | WLAN\_BW\_MHZ40 | 0x2 | 40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz |
|  |  | |  | WLAN\_BW\_MHZ80 | 0x3 | 80MHz channel; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ8080 | 0x4 | 80-80MHz, 2 non-contiguous 80MHz; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ160 | 0x5 | 160MHz channel; must 11ac |
|  |  | |  | WLAN\_BW\_MAX | 0x6 |  |
| R | rssi | | Int32 | - | 0-4294967295 | RSSI of beacon/probe response |
| R | WlanSecurity | | Enum | - | - | Security settings of scan results |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | isWpsSupported | | Boolean | - | 0/1 | does AP support WPS |
| R | isEssSupported | | Boolean | - | 0/1 | is AP part of an Extended Service Set |
| R | scanResultsComplete | | Boolean | - | 0/1 | All the scan results have been reported or not |

#### MD-REQ-380305/B-ConnectWiFiAP

This API is used internally by WIRClient and WIRServer to request to connect to an AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType | - |  | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - |  | TBD |
| R | WlanIpv4AddrType | | Enum | - |  |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface Name |

#### MD-REQ-380306/A-DisconnectWiFiAP

This API is used internally by WIRClient and WIRServer to request to disconnect from an AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |

#### MD-REQ-380307/B-ProfileUpdate

This API is used internally by WIRClient and WIRServer to request to update a profile. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | txId | | String | - | Char Value:0-255  No String length limit | Transaction ID |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | txId | | String | - | Char Value:0-255  No String length limit | Transaction ID |
| R | updateComplete | | Boolean | - | 0/1 | Is it last segment? |
| R | profile\_cnt | | String | - | Char Value:0-255  No String length limit | Number of entries of following 2 fields |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType | - | - | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - |  | TBD |
| R | WlanIpv4AddrType | | Enum | - |  |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |

#### MD-REQ-380308/A-ConnectionStatus

This API is used internally by WIRClient and WIRServer to request connection status. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID |
| R | bssid | | String | - | Char Value:0-255  No String length limit | BSSID, MAC address |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID |
| R | bssid | | String | - | Char Value:0-255  No String length limit | BSSID, MAC address |
| R | WlanStaConnState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTED | 0x0 | Disconnected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_WPS\_ACTIVE | 0x1 | WPS pending |
|  |  | |  | WLAN\_STA\_CON\_STATE\_ASSOCIATING | 0x2 | Associating |
|  |  | |  | WLAN\_STA\_CON\_STATE\_IP\_ADDRESSING | 0x3 | Getting IP address |
|  |  | |  | WLAN\_STA\_CON\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_CON\_STATE\_AUTHERROR | 0x6 | Authentication error |
|  |  | |  | WLAN\_STA\_CON\_STATE\_NWNOTFOUND | 0x7 | Network not found |
| R | WlanStaSmState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_SM\_STATE\_OFF | 0x0 | WLAN off |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ENABLING | 0x1 | Enabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ON | 0x2 | On |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTING | 0x3 | Connecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISABLING | 0x6 | Disabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ERROR | 0x7 | Error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_AUTHERROR | 0x8 | Authentication error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_NWNOTFOUND | 0x9 | Network not found |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |

#### MD-REQ-380309/C-StateUpdateInd

This API is used internally by WIRClient and WIRServer to broadcast an update of ECU’s station mode connection state.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | WlanEcu | | Enum | - | - |  |
|  |  | |  | TCU\_WLAN | 0 | TCU WLAN |
|  |  | |  | SYNC\_WLAN | 1 | SYNC WLAN |
|  |  | |  | TCU\_B\_WLAN | 2 | TCU-B WLAN. Applicable to AV vehicles Only |
| R | WlanStaConnState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTED | 0x0 | Disconnected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_WPS\_ACTIVE | 0x1 | WPS pending |
|  |  | |  | WLAN\_STA\_CON\_STATE\_ASSOCIATING | 0x2 | Associating |
|  |  | |  | WLAN\_STA\_CON\_STATE\_IP\_ADDRESSING | 0x3 | Getting IP address |
|  |  | |  | WLAN\_STA\_CON\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_CON\_STATE\_AUTHERROR | 0x6 | Authentication error |
|  |  | |  | WLAN\_STA\_CON\_STATE\_NWNOTFOUND | 0x7 | Network not found |
| R | WlanStaSmState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_SM\_STATE\_OFF | 0x0 | WLAN off |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ENABLING | 0x1 | Enabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ON | 0x2 | On |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTING | 0x3 | Connecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISABLING | 0x6 | Disabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ERROR | 0x7 | Error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_AUTHERROR | 0x8 | Authentication error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_NWNOTFOUND | 0x9 | Network not found |
| R | macAddr | | String | - | Char Value:0-255  No String length limit | MAC Address |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID |
| R | bssid | | String | - | Char Value:0-255  No String length limit | BSSID |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | bandwidth | | Int32 | - | 0-4294967295 |  |
| R | WlanTech | | Enum | - | - |  |
|  |  | |  | Unk | 0x0 | Unknown |
|  |  | |  | B | 0x1 | 11b; 2.4Ghz, Max data rate 11Mbps |
|  |  | |  | A | 0x2 | 11a, 5.0Ghz, Max data rate 54Mbps |
|  |  | |  | G | 0x3 | 11g, 2.4Ghz, Max data rate 54Mbps |
|  |  | |  | N | 0x4 | 11n, 2.4/5.0Ghz, Max data rate - 1ss (20/40): 72.2/150Mbps |
|  |  | |  | AC | 0x5 | 11ac, 2.4/5.0Ghz, Max data rate - 1ss (20/40/80/160) : 78/200/433/866Mbps |
| R | dataRate | | Int32 | - | 0-4294967295 | bps |
| R | signalStrength | | Int32 | - | 0-4294967295 | Signal strength, dBm |
| R | signalNoise | | Int32 | - | 0-4294967295 | Noise level, dBm |
| R | signalSNR | | Int32 | - | 0-4294967295 | SNR, dB |
| R | ipAddr | | String | - | Char Value:0-255  No String length limit | IP address |
| R | ipSubnet | | String | - | Char Value:0-255  No String length limit | Subnet mask |
| R | ipGateway | | String | - | Char Value:0-255  No String length limit | GW IP address |
| R | ipDnsPref | | String | - | Char Value:0-255  No String length limit | DNS primary |
| R | ipDnsAlt | | String | - | Char Value:0-255  No String length limit | DNS secondary |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface name |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |

#### MD-REQ-380310/A-StationModeStats

This API is used internally by WIRClient and WIRServer to request WLAN station mode statistics. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface Name |
| R | StatsCmdType | | Enum | - | - |  |
|  |  | |  | DISABLE | 0x0 | Disable |
|  |  | |  | ENABLE | 0x1 | Enable |
|  |  | |  | QUERY | 0x2 | Query current state |
| R | reportingWindow | | Int32 | - | 0-4294967295 | Seconds |
| **Response** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface Name |
| R | StatsCmdType | | Enum | - | - |  |
|  |  | |  | DISABLE | 0x0 | Disable |
|  |  | |  | ENABLE | 0x1 | Enable |
|  |  | |  | QUERY | 0x2 | Query current state |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | pktsTxV4 | | Int64 | - | 0-18446744073709551615 | Number of packets transmitted |
| R | pktsRxV4 | | Int64 | - | 0-18446744073709551615 | Number of packets received |
| R | bytesTxV4 | | Int64 | - | 0-18446744073709551615 | Number of bytes transmitted |
| R | bytesRxV4 | | Int64 | - | 0-18446744073709551615 | Number of bytes received |
| R | PktsDroppedTx | | Int64 | - | 0-18446744073709551615 | Number of dropped packets transmitted |
| R | PktsDroppedRx | | Int64 | - | 0-18446744073709551615 | Number of dropped packets received |

#### MD-REQ-380312/B-ProfileUpdateInd

This API is used internally by WIRClient and WIRServer to broadcast a profile update indication.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | WlanProfileAction | | Enum | - | - |  |
|  |  | |  | ACT\_UPDATE | 0x0 | Add/update |
|  |  | |  | ACT\_REMOVE | 0x1 | Remove |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType | - |  | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - |  | TBD |
| R | WlanIpv4AddrType | | Enum | - |  |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |

#### MD-REQ-380314/A-WhsInfo

This API is used internally by WIRClient and WIRServer to request WIFI HotSpot Information. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| **Response** | | | | | | |
| R | Intf | | String | - | Char Value:0-255  No String length limit | The name of the interface |
| R | macAddr | | String | - | Char Value:0-255  No String length limit | MAC address of the interface |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of our AP |
| R | password | | String | - | Char Value:0-255  No String length limit | Password |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |

#### MD-REQ-402837/A-ReadPolicyTable

This API is used by WIRServer (internally) and WIRClients to request current policy table. WIRServer also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | policyTableLength | | Int32 | - | 0-4294967295 | Length of policyTable |
| R | policyTable | | Binary | - | GPB data | GPB compressed policy table |

#### MD-REQ-402838/A-EnableSyncWifiStaMode

This API is used by WIRServer (internally) and WIRClients to enable WIRClient1 WiFi STA mode. WIRServer also uses this API for its response.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | |
| **Method Type** | | One-Shot (Synch) | | | | | |
| **QoS Level** | | Default | | | | | |
| **Retained** | | No | | | | | |
|  | | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | | |
| R | WirRet | | Enum | - | - | Return Command |
|  |  | |  | WIR\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WIR\_SUCCESS | 0x1 | Success |

#### MD-REQ-402839/A-DisableSyncWifiStaMode

This API is used by WIRServer (internally) and WIRClients to disable WIRClient1 WiFi STA mode. WIRServer also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | WirRet | | Enum | - | - | Return Command |
|  |  | |  | WIR\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WIR\_SUCCESS | 0x1 | Success |

### WIR-IIR-REQ-289532/E-WIRServer\_Rx

#### MD-REQ-027149/A-IgnitionStatus\_St (TcSE ROIN-225464-1)

Message Type: Status

Signal used to indicate ignition state.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Literals** | **Value** | **Description** |
| Type | - | - | Indicates ignition state |
|  | Unknown | 0x0 |  |
|  | Off | 0x1 |  |
|  | Accessory | 0x2 |  |
|  | Run | 0x4 |  |
|  | Start | 0x8 |  |
|  | Invalid | 0xF |  |

#### MD-REQ-199634/A-BSBattSOC

Message Type: Status

Status used to indicate the vehicle battery’s state of charge.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Literals** | **Value** | **Description** |
| Type | - | - | Battery's state of charge in percent |
|  | 0-127 | 0x0 - 0x7F |  |

#### MD-REQ-201601/A-Delay\_Accy

Message Type: Status

This signal is used indicate whether Delayed Accessory is active or not.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Literals** | **Value** | **Description** |
| Type | - | - | Status of delayed accessory |
|  | Off | 0x00 |  |
|  | On | 0x01 |  |

#### MD-REQ-028253/A-GearLeverPosition\_St (TcSE ROIN-282103-1)

Message Type: Status

Status used to indicate the current gear selected.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Literals** | **Value** | **Description** |
| Type | - | - | Used to indicate current gear selected. |
|  | Park | 0x0 |  |
|  | Reverse | 0x1 |  |
|  | Neutral | 0x2 |  |
|  | Drive | 0x3 |  |
|  | Sport\_DriveSport | 0x4 |  |
|  | Low | 0x5 |  |
|  | First | 0x6 |  |
|  | Second | 0x7 |  |
|  | Third | 0x8 |  |
|  | Fourth | 0x9 |  |
|  | Fifth | 0xA |  |
|  | Sixth | 0xB |  |
|  | Undefined\_Treat\_as\_Fault | 0xC |  |
|  | Undefined\_Treat\_as\_Fault1 | 0xD |  |
|  | Unknown\_Position | 0xE |  |
|  | Fault | 0xF |  |

#### MD-REQ-014025/A-VehicleSpeed\_St (TcSE ROIN-223023-1)

Message Type: Status

Status used to indicate vehicle speed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Literals** | **Value** | **Description** |
| Type | - | - | Indicates vehicle speed.  Unit: kph  Resolution:0.01  Offset:0 |
|  | kph | 0x0 to 0xFFFF |  |

#### MD-REQ-304038/A-VehicleMode

Message Type: Status

Signal to manage the key off load of the vehicle when the Ignition is OFF.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Literals** | **Value** | **Description** |
| Type | - | - | - |
|  | NormalPower | 0x0 |  |
|  | FactoryPower | 0x1 |  |
|  | TransportPower | 0x2 |  |
|  | Hibernate | 0x3 |  |
|  | CriticalBattery | 0x4 |  |
|  | NotUsed | 0x5 - 0xF |  |

#### MD-REQ-380247/A-CellActivate

This API is used internally between WIRClient and WIRServer to request activation of a cellular interface. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | CellApnType | | Enum | - | - | Available APNs for cellular data |
|  |  | |  | Internet | 0x0 | Metered User Internet APN |
|  |  | |  | Ford | 0x1 | Non-metered Ford APN |
|  |  | |  | Tethering | 0x2 | Metered Mobile Hot Spot APN |
|  |  | |  | FOTA | 0x3 | FOTA APN |
|  |  | |  | HTTP | 0x4 | HTTP APN |
| **Response** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | CellRet | | Enum | - | - | Return code |
|  |  | |  | CELL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CELL\_SUCCESS | 0x1 | Success |
| R | cause | | String | - | Char Value:0-255  No String length limit | Activation cause code |
| R | IfaceId | | Int32 | - | 0-4294967295 | Interface Id used internally by CM |
| R | IfaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | ipV4Address | | String | - | Char Value:0-255  No String length limit | Interface IP address V4 |
| R | ipV4DnsPrimary | | String | - | Char Value:0-255  No String length limit | Primary DNS IP address V4 |
| R | ipV4DnsSecondary | | String | - | Char Value:0-255  No String length limit | Secondary DNS IP address V4 |
| R | ipV4Gateway | | String | - | Char Value:0-255  No String length limit | Gateway IP address V4 |
| R | ipV6Address | | String | - | Char Value:0-255  No String length limit | Interface IP address V6 |
| R | ipV6DnsPrimary | | String | - | Char Value:0-255  No String length limit | Primary DNS IP address V6 |
| R | ipV6DnsSecondary | | String | - | Char Value:0-255  No String length limit | Secondary DNS IP address V6 |
| R | ipV6Gateway | | String | - | Char Value:0-255  No String length limit | Gateway IP address V6 |
| R | mtuIpV4 | | Int32 | - | 0-4294967295 | Interface MTU V4 |
| R | mtuIpV6 | | Int32 | - | 0-4294967295 | Interface MTU V6 |

#### MD-REQ-380248/A-CellActivateInd

This API is used internally by WIRClient and WIRServer to broadcast an activation state change of a cellular interface.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | IfaceId | | Int32 | - | 0-4294967295 | Interface Id used internally by CM |
| R | IfaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | ipV4Address | | String | - | Char Value:0-255  No String length limit | Interface IP address V4 |
| R | ipV4DnsPrimary | | String | - | Char Value:0-255  No String length limit | Primary DNS IP address V4 |
| R | ipV4DnsSecondary | | String | - | Char Value:0-255  No String length limit | Secondary DNS IP address V4 |
| R | ipV4Gateway | | String | - | Char Value:0-255  No String length limit | Gateway IP address V4 |
| R | ipV6Address | | String | - | Char Value:0-255  No String length limit | Interface IP address V6 |
| R | ipV6DnsPrimary | | String | - | Char Value:0-255  No String length limit | Primary DNS IP address V6 |
| R | ipV6DnsSecondary | | String | - | Char Value:0-255  No String length limit | Secondary DNS IP address V6 |
| R | ipV6Gateway | | String | - | Char Value:0-255  No String length limit | Gateway IP address V6 |
| R | mtuIpV4 | | Int32 | - | 0-4294967295 | Interface MTU V4 |
| R | mtuIpV6 | | Int32 | - | 0-4294967295 | Interface MTU V6 |

#### MD-REQ-380249/A-CellDeactivate

This API is used internally between WIRClient and WIRServer to request deactivation of a cellular interface. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | CellApnType | | Enum | - | - | Available APNs for cellular data |
|  |  | |  | Internet | 0x0 | Metered User Internet APN |
|  |  | |  | Ford | 0x1 | Non-metered Ford APN |
|  |  | |  | Tethering | 0x2 | Metered Mobile Hot Spot APN |
|  |  | |  | FOTA | 0x3 | FOTA APN |
|  |  | |  | HTTP | 0x4 | HTTP APN |
| **Response** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | CellRet | | Enum | - | - | Return code |
|  |  | |  | CELL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CELL\_SUCCESS | 0x1 | Success |
| R | cause | | String | - | Char Value:0-255  No String length limit | Activation cause code |

#### MD-REQ-380250/A-CellDeactivateInd

This API is used internally by WIRClient and WIRServer to broadcast a deactivation state change of a cellular interface.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | CellRet | | Enum | - | - | Return code |
|  |  | |  | CELL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CELL\_SUCCESS | 0x1 | Success |
| R | cause | | String | - | Char Value:0-255  No String length limit | Activation cause code |

#### MD-REQ-380251/A-CellLinkPropertyInd

This API is used internally by WIRClient and WIRServer to broadcast cellular interface link properties updates.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | IfaceId | | Int32 | - | 0-4294967295 | Interface Id used internally by CM |
| R | IfaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | ipV4Address | | String | - | Char Value:0-255  No String length limit | Interface IP address V4 |
| R | ipV4DnsPrimary | | String | - | Char Value:0-255  No String length limit | Primary DNS IP address V4 |
| R | ipV4DnsSecondary | | String | - | Char Value:0-255  No String length limit | Secondary DNS IP address V4 |
| R | ipV4Gateway | | String | - | Char Value:0-255  No String length limit | Gateway IP address V4 |
| R | ipV6Address | | String | - | Char Value:0-255  No String length limit | Interface IP address V6 |
| R | ipV6DnsPrimary | | String | - | Char Value:0-255  No String length limit | Primary DNS IP address V6 |
| R | ipV6DnsSecondary | | String | - | Char Value:0-255  No String length limit | Secondary DNS IP address V6 |
| R | ipV6Gateway | | String | - | Char Value:0-255  No String length limit | Gateway IP address V6 |
| R | mtuIpV4 | | Int32 | - | 0-4294967295 | Interface MTU V4 |
| R | mtuIpV6 | | Int32 | - | 0-4294967295 | Interface MTU V6 |
| R | IfaceId | | Int32 | - | 0-4294967295 | Interface Id used internally by CM |

#### MD-REQ-380252/A-CellStatistics

This API is used internally by WIRClient and WIRServer to request cellular traffic statistics of a cellular interface. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | StatsCmdType | | Enum | - | - | Command types for cellular data traffic statistics |
|  |  | |  | Disable | 0x0 | Disable reporting of statistics |
|  |  | |  | Enable | 0x1 | Enable reporting of statistics |
|  |  | |  | Query | 0x2 | Query Statistics |
| R | reportingWindow | | Int32 | - | 0-4294967295 | Reporting Window in seconds |
| **Response** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | CellRet | | Enum | - | - | Return code |
|  |  | |  | CELL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CELL\_SUCCESS | 0x1 | Success |
| R | PKTS\_TX\_V4 | | Int32 | - | 0-4294967295 | Transmitted IPv4 packets |
| R | PKTS\_RX\_V4 | | Int32 | - | 0-4294967295 | Received IPv4 packets |
| R | BYTES\_TX\_V4 | | Int64 | - | 0-18446744073709551615 | Transmitted IPv4 bytes |
| R | BYTES\_RX\_V4 | | Int64 | - | 0-18446744073709551615 | Received IPv4 bytes |
| R | PKTS\_DROPPED\_TX | | Int32 | - | 0-4294967295 | Dropped transmit IPv4 packets |
| R | PKTS\_DROPPED\_RX | | Int32 | - | 0-4294967295 | Dropped receive IPv4 packets |

#### MD-REQ-380253/A-CellStatisticsInd

This API is used internally by WIRClient and WIRServer to broadcast cellular interface traffic statistics.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | CellRet | | Enum | - | - | Return code |
|  |  | |  | CELL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CELL\_SUCCESS | 0x1 | Success |
| R | PKTS\_TX\_V4 | | Int32 | - | 0-4294967295 | Transmitted IPv4 packets |
| R | PKTS\_RX\_V4 | | Int32 | - | 0-4294967295 | Received IPv4 packets |
| R | BYTES\_TX\_V4 | | Int64 | - | 0-18446744073709551615 | Transmitted IPv4 bytes |
| R | BYTES\_RX\_V4 | | Int64 | - | 0-18446744073709551615 | Received IPv4 bytes |
| R | PKTS\_DROPPED\_TX | | Int32 | - | 0-4294967295 | Dropped transmit IPv4 packets |
| R | PKTS\_DROPPED\_RX | | Int32 | - | 0-4294967295 | Dropped receive IPv4 packets |
| R | connId | | Int32 | - | 0-4294967295 | Interface ID used internally by CM |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface name as enumerated in the OS |
| R | CellRet | | Enum | - | - | Return code |

#### MD-REQ-380254/A-CellSessionInd

This API is used internally by WIRClient and WIRServer to broadcast cellular interface new session information.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | uuID | | String | - | Char Value:0-255  No String length limit | Unique Id used internally by CM |

#### MD-REQ-380255/A-DataStatusInd

This API is used internally by WIRClient and WIRServer to broadcast cellular data capability of a cellular interface.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | IndicationType | | Enum | - |  |  |
|  |  | |  | Networking | 0x0 | Indicates netmgrd/networking |
|  |  | |  | CellularData | 0x1 | Indicates cellular data |
|  |  | |  | CellularFlowCtrl | 0x2 | Indicates cellular flow control |
| R | StatusType | | Enum | - | - | DCM/Netmgrd data capability |
|  |  | |  | Unavailable | 0x0 | Data path not available |
|  |  | |  | Available | 0x1 | Data path available |

#### MD-REQ-380258/C-VlanAdd

This API is used internally by WIRClient and WIRServer to request a new VLAN configuration. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | ifaceEcu | | String | - | Char Value:0-255  No String length limit | Interface ECU enum in string format |
| R | mtu | | String | - | Char Value:0-255  No String length limit | Interface MTU in string format |
| R | svcLevel | | String | - | Char Value:0-255  No String length limit | Service Level of the VLAN in string format |
| R | ifaceType | | Enum | - | - | Network Interface Type over which the VLAN would be setup |
|  |  | |  | None | 0x0 |  |
|  |  | |  | TcuCell | 0x1 |  |
|  |  | |  | TcuWifi | 0x2 |  |
|  |  | |  | SyncWifi | 0x3 |  |
|  |  | |  | SyncApplink | 0x4 | Out of scope |
|  |  | |  | TcubCell | 0x5 | Applicable to AV vehicles Only |
|  |  | |  | TcubWifi | 0x6 | Applicable to AV vehicles Only |
|  |  | |  | EspmLan | 0x7 | ESPM+ LAN  Applicable to AV vehicles Only |
|  |  | |  | DdsmLan | 0x8 | DDSM LAN  Applicable to AV vehicles Only |
|  |  | |  | Error | 0x9 |  |
| R | vlanEndpointType | | Enum | - | - | VLAN Endpoint type |
|  |  | |  | Host | 0x0 | Host type endpoint |
|  |  | |  | Gateway | 0x1 | Gateway type endpoint |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | ipV4Address | | String | - | Char Value:0-255  No String length limit | VLAN IP Address |

#### MD-REQ-380259/A-VlanRemove

This API is used internally by WIRClient and WIRServer to request removal of old VLAN configuration. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |

#### MD-REQ-380260/A-VlanLink

This API is used internally by WIRClient and WIRServer to request link VLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Network Interface as enumerated in the OS |
| R | serviceLevel | | String | - | Char Value:0-255  No String length limit | Service Level of the VLAN in string format |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Network Interface as enumerated in the OS |
| R | serviceLevel | | String | - | Char Value:0-255  No String length limit | Service Level of the VLAN in string format |
| R | linkId | | String |  | Char Value:0-255  No String length limit | Unique link ID generated by Networking |

#### MD-REQ-380261/A-VlanUnlink

This API is used internally by WIRClient and WIRServer to request unlink VLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | linkId | | String |  | Char Value:0-255  No String length limit | Unique link ID generated by Networking |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | vlanEndpointId | | Int32 | - | 0-4294967295 | Endpoint Id used internally by CM |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |
| R | vlanName | | String | - | Char Value:0-255  No String length limit | Unique Vlan name |
| R | vlanId | | String | - | Char Value:0-255  No String length limit | Unique Vlan Id |
| R | linkId | | String |  | Char Value:0-255  No String length limit | Unique link ID generated by Networking |

#### MD-REQ-380262/C-VnmReset

This API is used internally by WIRClient and WIRServer to request a VLAN configuration reset. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | EcuType | | Enum | - | - | Path Id used internally by CM |
|  |  | |  | Unknown | 0x0 | Error |
|  |  | |  | ECG | 0x1 |  |
|  |  | |  | TCU | 0x2 |  |
|  |  | |  | SYNC | 0x3 |  |
|  |  | |  | TCU\_B | 0x4 | Applicable to AV vehicles Only |
|  |  | |  | SDS | 0x5 | Applicable to AV vehicles Only |
|  |  | |  | ADSIM | 0x6 | Applicable to AV vehicles Only |
|  |  | |  | DDSM | 0x7 | Applicable to AV vehicles Only |
|  |  | |  | ADM | 0x8 | Applicable to AV vehicles Only |
|  |  | |  | ESPM | 0x9 | Applicable to AV vehicles |
|  |  | |  | SDM | 0x10 | Applicable to AV vehicles |
| **Response** | | | | | | |
| R | EcuType | | Enum | - | - | Path Id used internally by CM |
|  |  | |  | Unknown | 0x0 | Error |
|  |  | |  | ECG | 0x1 |  |
|  |  | |  | TCU | 0x2 |  |
|  |  | |  | SYNC | 0x3 |  |
|  |  | |  | TCU\_B | 0x4 |  |
|  |  | |  | SDS | 0x5 |  |
|  |  | |  | ADSIM | 0x6 |  |
|  |  | |  | DDSM | 0x7 |  |
|  |  | |  | ADM | 0x8 |  |
|  |  | |  | ESPM | 0x9 | Applicable to AV vehicles |
|  |  | |  | SDM | 0x10 | Applicable to AV vehicles |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |

#### MD-REQ-380263/A-VnmLinkNetwork

This API is used internally by WIRClient and WIRServer to request to link a network to VLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | ifaceEcu | | String | - | Char Value:0-255  No String length limit | Interface ECU enum in string format |
| R | interfaceZero | | String | - | Char Value:0-255  No String length limit | User’s provided interface name to be linked |
| R | interfaceOne | | String | - | Char Value:0-255  No String length limit | Network interface to be linked |
| R | ipAddr | | String | - | Char Value:0-255  No String length limit | Gateway ip address associated with network interface |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | linkId | | String | - | Char Value:0-255  No String length limit | Unique link ID generated by Networking |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |

#### MD-REQ-380264/A-VnmUnlinkNetwork

This API is used internally by WIRClient and WIRServer to request to unlink a network to VLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | ifaceEcu | | String | - | Char Value:0-255  No String length limit | Interface ECU enum in string format |
| R | linkId | | String | - | Char Value:0-255  No String length limit | Unique link ID generated by Networking |
| **Response** | | | | | | |
| R | pathId | | Int32 | - | 0-4294967295 | Path Id used internally by CM |
| R | origEcu | | String | - | Char Value:0-255  No String length limit | Origination ECU enum in string format |
| R | VlanRet | | Enum | - | - | Return code for command |
|  |  | |  | VLAN\_ERROR | 0x0 |  |
|  |  | |  | VLAN\_SUCCESS | 0x1 |  |

#### MD-REQ-380265/D-NetworkInterfaceAllocation

This API is used internally by WIRClient and WIRServer to request a new network interface. WIR also uses this API for its response. The response will return success or failure, actual interface is returned through NetworkInterfaceAllocationStatusInd.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | | One-Shot (A-Synch) | | | |
| **QoS Level** | | | Default | | | |
| **Retained** | | | No | | | |
|  | | | | | | |
| **R/O** | **Name** | **Type** | | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | String | | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | Int32 | | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| R | IntentType | Enum | | - | - | Intent Type |
|  |  |  | | Foreground | 0x0 |  |
|  |  |  | | Background\_  BestEffort | 0x1 |  |
|  |  |  | | Background\_  Guaranteed | 0x2 |  |
|  |  |  | | Special | 0x3 | N/A |
|  |  |  | | OffPeak | 0x4 | N/A for AV vehicles |
|  |  |  | | LAN\_Only | 0x5 | Applicable to AV vehicles Only |
| R | InterfaceType | Enum | | - | - | Interface Type |
|  |  |  | | None | 0x0 |  |
|  |  |  | | TcuCell | 0x1 |  |
|  |  |  | | TcuWifi | 0x2 |  |
|  |  |  | | SyncWifi | 0x3 |  |
|  |  |  | | SyncApplink | 0x4 | Out of scope |
|  |  |  | | TcubCell | 0x5 | Applicable to AV vehicles Only |
|  |  |  | | TcubWifi | 0x6 | Applicable to AV vehicles Only |
|  |  |  | | EspmLan | 0x7 | Applicable to AV vehicles Only |
|  |  |  | | DdsmLan | 0x8 | Applicable to AV vehicles Only |
|  |  |  | | Error | 0x9 |  |
| R | CellApnType | Enum | | - | - | Cellular APN Type |
|  |  |  | | Internet | 0x0 |  |
|  |  |  | | Ford | 0x1 |  |
|  |  |  | | Tethering | 0x2 |  |
|  |  |  | | FOTA | 0x3 |  |
|  |  |  | | HTTP | 0x4 |  |
| R | PriorityLevel | Enum | | - | - | Priority Level |
|  |  |  | | Priority 0 | 0x0 |  |
|  |  |  | | Priority 1 | 0x1 |  |
|  |  |  | | Priority 2 | 0x2 |  |
|  |  |  | | Priority 3 | 0x3 |  |
|  |  |  | | Priority 4 | 0x4 |  |
| R | expiry | Int32 | | - | 0-4294967295 | Expiration timer |
| R | OffPeakFlag | Enum | | - | - | Offpeak flag. N/A for AV vehicles |
|  |  |  | | No | 0x0 |  |
|  |  |  | | Yes | 0x1 |  |
| R | CellularOnlyFlag | Enum | | - | - | Cellular Only Flag |
|  |  |  | | No | 0x0 |  |
|  |  |  | | Yes | 0x1 |  |
| R | WifiPreferredFlag | Enum | | - | - | Wifi preferred flag |
|  |  |  | | No | 0x0 |  |
|  |  |  | | Yes | 0x1 |  |
| R | ssid | String | | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | String | |  | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | Enum | | - | - | Optional - Channel of AP |
|  |  |  | | Reserved | 0x00 |  |
|  |  |  | | WLAN\_CH\_1 | 0x01 |  |
|  |  |  | | WLAN\_CH\_2 | 0x02 |  |
|  |  |  | | WLAN\_CH\_3 | 0x03 |  |
|  |  |  | | … | … |  |
|  |  |  | | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | Enum | | - | - | Security settings to use |
|  |  |  | | Reserved | 0x00 |  |
|  |  |  | | Open | 0x01 | Open or no security |
|  |  |  | | WEP | 0x02 | WEP |
|  |  |  | | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  |  | | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  |  | | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  |  | | MAX | 0x06 |  |
| R | WlanWepSecurity |  | |  |  | WEP Settings |
|  |  | String | | Key | Char Value:0-255  No String length limit |  |
|  |  | Int32 | | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity |  | |  |  | WPS Settings |
|  |  | WlanWpsType | |  |  | Type |
|  |  | String | |  | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | String | |  | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | Enum | | - | - | Pair cipher |
|  |  |  | | PAIR\_MIN | 0x0 |  |
|  |  |  | | PAIR\_NONE | 0x1 | None |
|  |  |  | | PAIR\_TKIP | 0x2 | TKIP |
|  |  |  | | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  |  | | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  |  | | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | Enum | | - | - | Group cipher |
|  |  |  | | GROUP\_MIN | 0x0 |  |
|  |  |  | | GROUP\_NONE | 0x1 | None |
|  |  |  | | GROUP\_TKIP | 0x2 | TKIP |
|  |  |  | | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  |  | | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  |  | | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | TBD | | - | - | TBD |
| R | WlanIpv4AddrType | Enum | | - | - |  |
|  |  |  | | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  |  | | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  |  | | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  |  | | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  |  | | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  |  | | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | - | | - | - |  |
|  |  | String | | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | String | | Netmask | Char Value:0-255  No String length limit | Netmask of current connection |
|  |  | String | | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | String | | dnsPref | Char Value:0-255  No String length limit | Preferred DNS server |
|  |  | String | | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | Enum | | - | - |  |
|  |  |  | | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  |  | | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  |  | | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  |  | | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | TBD | | - | - | TBD |
| R | ExclusiveUse | Boolean | | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | Enum | | - | - | Origin of the profile |
|  |  |  | | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  |  | | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  |  | | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  |  | | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  |  | | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | Int64 | | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | Boolean | | - | 0/1 | Hidden AP |
| R | IfaceNametoLink | String | | - | Char Value:0-255  No String length limit | AP SSID |
| R | data\_size | Int32 | | - | 0-4294967295 | Data size, if applicable. Applicable to AV vehicles Only |
| R | Bw\_downlink | Int32 | | - | 0-4294967295 | Application required downlink bandwidth. Applicable to AV vehicles Only |
| R | Bw\_uplink | Int32 | | - | 0-4294967295 | Application required uplink bandwidth. Applicable to AV vehicles Only |
| **Response** | | | | | | |
| R | appId | String | | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | Int32 | | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| R | WirRet | Enum | | - | - | Return Command |
|  |  |  | | WIR\_ERROR | 0x0 | Error/Failure |
|  |  |  | | WIR\_SUCCESS | 0x1 | Success |

#### MD-REQ-380266/A-NetworkInterfaceRelease

This API is used internally by WIRClient and WIRServer to request a network interface release. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | | Int32 | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | allocId | | Int32 | - | 0-4294967295 | Allocation ID will be assigned by WIR |
| R | WirRet | | Enum | - | - | Return Command |
|  |  | |  | WIR\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WIR\_SUCCESS | 0x1 | Success |

#### MD-REQ-380270/A-WIRClientRegisterInd

This API is used internally by WIRClient and WIRServer to register a WIRClient.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |

#### MD-REQ-380271/A-WIRClientDeregisterInd

This API is used internally by WIRClient and WIRServer to deregister a WIRClient.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |

#### MD-REQ-380272/A-WebViewDisplay

This API is used internally by WIRClient and WIRServer to request WEB View display. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | ipAddr | | String | - | Char Value:0-255  No String length limit | IP address to be used for connection |
| R | uri | | String | - | Char Value:0-255  No String length limit | WEB page URI |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | WirRet | | Enum | - | - | Return Command |
|  |  | |  | WIR\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WIR\_SUCCESS | 0x1 | Success |

#### MD-REQ-380273/B-HeartBeat

This API is used internally by WIRClient and WIRServer to report a heartbeat message. Initial type heartbeat is a one-shot message and regular type heartbeat occurs every 20 seconds while interface is provided.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | N/A |
| **Response** | | | | | | |
| R | hbMsg | | String | - |  | Payload with a message string |
| R | HeartBeatType | | Enum | - | - | HeartBeat event type |
|  |  | |  | HB\_INITIAL | 0x0 | Initial HeartBeat Type |
|  |  | |  | HB\_REGULAR | 0x1 | Regular Heartbeat Type |
| R | hbIndex | | Int32 | - | 0-4294967295 | HeartBeat event index |
| R | hbTimer | | Int32 | - | 0-4294967295 | HeartBeat event timer duration provided by ECG |

#### MD-REQ-380274/C-Policy

This API is used internally by WIRClient and WIRServer to request a network policy for an application. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| **Response** | | | | | | |
| R | CtrlRet | | Enum | - | - |  |
|  |  | |  | CTRL\_ERROR | 0x0 | Error/Failure |
|  |  | |  | CTRL\_SUCCESS | 0x1 | Success |
| R | appId | | String | - | Char Value:0-255  No String length limit | WIR Client AppId |
| R | Policy | | Int32 | - | 0-4294967295 | Network policy, bit flag |
|  |  | |  | TCU\_CELLULAR\_APN1 | 0x0001 |  |
|  |  | |  | WIFI | 0x0002 |  |
|  |  | |  | TCU\_CELLULAR\_APN2 | 0x0004 |  |
|  |  | |  | DNS\_RPZ\_EXCEPTION | 0x0008 |  |
|  |  | |  | ESPM\_LAN | 0x0010 | Applicable to AV vehicles Only |
|  |  | |  | DDSM\_LAN | 0x0020 | Applicable to AV vehicles Only |
|  |  | |  | OFF\_PEAK | 0x0040 | N/A |
|  |  | |  | RESERVED | 0x0080 |  |
|  |  | |  | RESERVED | 0x0100 |  |
|  |  | |  | FACTORY\_WIFI | 0x0200 |  |
|  |  | |  | TCU\_B\_CELLULAR\_APN1 | 0x0400 | Applicable to AV vehicles Only |
|  |  | |  | TCU\_B\_CELLULAR\_APN2 | 0x0800 | Applicable to AV vehicles Only |
|  |  | |  | WIFI\_CONNECTION\_CHANGE\_REPORT | 0x1000 |  |
|  |  | |  | SYSTEM\_APP | 0x2000 |  |
|  |  | |  | RESERVED | 0x4000 |  |
|  |  | |  | RESERVED | 0x8000 |  |

#### MD-REQ-380279/A-Enable

This API is used internally by WIRClient and WIRServer to request enabling WLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380280/A-Disable

This API is used internally by WIRClient and WIRServer to request disabling WLAN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380281/B-ScanApimAPs

This API is used internally by WIRClient and WIRServer to request a scan of available APs. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | scanId | | Int64 | - | 0-18446744073709551615 | ID value used to track scans |
| R | passive | | Boolean | - | 0/1 | true = passive scan (no pkts tx'd), false = active scan |
| R | WlanChannel | | Enum | - | - | Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | ssid | | String | - | Char Value:0-255  No String length limit | list of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required |
| R | Count | | Int32 | - | 0-4294967295 | Number of APs found |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID – scan result |
| R | bssid | | String | - | Char Value:0-255  No String length limit | MAC address of AP – scan result |
| R | WlanChannel | | Enum | - | - | Channel of AP – scan result |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | Wlan Bandwidth | | Enum | - | - |  |
|  |  | |  | WLAN\_BW\_MIN | 0x0 |  |
|  |  | |  | WLAN\_BW\_MHZ20 | 0x1 | 20MHz channel |
|  |  | |  | WLAN\_BW\_MHZ40 | 0x2 | 40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz |
|  |  | |  | WLAN\_BW\_MHZ80 | 0x3 | 80MHz channel; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ8080 | 0x4 | 80-80MHz, 2 non-contiguous 80MHz; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ160 | 0x5 | 160MHz channel; must 11ac |
|  |  | |  | WLAN\_BW\_MAX | 0x6 |  |
| R | rssi | | Int32 | - | 0-4294967295 | RSSI of beacon/probe response |
| R | WlanSecurity | | Enum | - | - | Security settings of scan results |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | isWpsSupported | | Boolean | - | 0/1 | does AP support WPS |
| R | isEssSupported | | Boolean | - | 0/1 | is AP part of an Extended Service Set |
| R | ipV4Addr | | String | - | Char Value:0-255  No String length limit | IPv4 address |
| R | netmask | | String | - | Char Value:0-255  No String length limit | Net mask |
| R | connected | | Boolean | - | 0/1 | connected to AP |
| R | prevConnected | | Boolean | - | 0/1 | Previously connected AP |
| R | isHidden | | Boolean | - | 0/1 | Hidden AP |
| R | WifiConnectionType | | Enum | - | - |  |
|  |  | |  | CONNECTION\_TYPE\_PIN | 0x0 | PIN |
|  |  | |  | CONNECTION\_TYPE\_WPS\_PIN | 0x1 | WPS (Wi-Fi Protected Setup) PIN |
|  |  | |  | CONNECTION\_TYPE\_WPS\_PBC | 0x2 | WPS Push Button Control |
|  |  | |  | CONNECTION\_TYPE\_OPEN | 0x3 | Open WiFi |
|  |  | |  | CONNECTION\_TYPE\_HIDDEN\_SECURE | 0x4 | Hidden + security, SSID is not broadcasted |
|  |  | |  | CONNECTION\_TYPE\_HIDDEN\_OPEN | 0x5 | Hidden + open, SSID is not broadcasted |
| R | EWifiConnectionStatus | | Enum | - | - |  |
|  |  | |  | WIFI\_STATUS\_SUCCESS | 0 | Success |
|  |  | |  | WIFI\_AP\_SCAN\_SUCCESS | 1 | Scan Successful |
|  |  | |  | WIFI\_NO\_APS\_AVAILABLE | 2 | No Access Points available |
|  |  | |  | WIFI\_CONNECTED | 3 | Connected to Access Point |
|  |  | |  | WIFI\_NOT\_CONNECTED | 4 | Wifi is not connected to Access Point |
|  |  | |  | WIFI\_OTH\_FAIL | 5 | Authentication Failure |
|  |  | |  | WIFI\_CONNECTING | 6 | Connecting to Access Point |
|  |  | |  | WIFI\_SECURITY\_MISSMATCH | 7 | There is security type mismatch between what requested and what is in actual |
|  |  | |  | WIFI\_AUTHENTICATING | 8 | In authenticating |
|  |  | |  | WIFI\_ABORTED | 9 | Connection aborted |
|  |  | |  | WIFI\_WRONG\_PASSWORD | 10 | Failed with wrong password |
|  |  | |  | WIFI\_WRONG\_WPS\_PIN | 11 | Failed with wrong WPS PIN |
|  |  | |  | WIFI\_CONNECTION\_PROGRESS | 12 | In connecting |
|  |  | |  | WIFI\_RESTRICTED\_NETWORK | 13 | Connection failure due to trying to connect to a restricted network eg : TCU hotspot |
|  |  | |  | WIFI\_NOT\_IMPLEMENTED | 101 | Not Implemented |
|  |  | |  | WIFI\_STATUS\_FAIL | 102 | Failure |
|  |  | |  | WIFI\_STATUS\_FAIL\_OTHER | 203 | Other Failure or Connecting to AP Failed |
|  |  | |  | WIFI\_CANCEL\_CONNECTION | 104 | Cancelled |
| R | complete | | Boolean | - | 0/1 | TRUE - complete  FALSE- more to come |

#### MD-REQ-380282/A-CancelScanApimAPs

This API is used internally by WIRClient and WIRServer to request to cancel scanning for available APs. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380283/B-ConnectToApimAP

This API is used internally by WIRClient and WIRServer to request to connect to an available AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType |  |  | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - | - | TBD |
| R | WlanIpv4AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |
| **Response** | | | | | | |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID – scan result |
| R | bssid | | String | - | Char Value:0-255  No String length limit | MAC address of AP – scan result |
| R | WlanChannel | | Enum | - | - | Channel of AP – scan result |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | Wlan Bandwidth | | Enum | - | - |  |
|  |  | |  | WLAN\_BW\_MIN | 0x0 |  |
|  |  | |  | WLAN\_BW\_MHZ20 | 0x1 | 20MHz channel |
|  |  | |  | WLAN\_BW\_MHZ40 | 0x2 | 40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz |
|  |  | |  | WLAN\_BW\_MHZ80 | 0x3 | 80MHz channel; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ8080 | 0x4 | 80-80MHz, 2 non-contiguous 80MHz; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ160 | 0x5 | 160MHz channel; must 11ac |
|  |  | |  | WLAN\_BW\_MAX | 0x6 |  |
| R | rssi | | Int32 | - | 0-4294967295 | RSSI of beacon/probe response |
| R | WlanSecurity | | Enum | - | - | Security settings of scan results |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | isWpsSupported | | Boolean | - | 0/1 | does AP support WPS |
| R | isEssSupported | | Boolean | - | 0/1 | is AP part of an Extended Service Set |
| R | ipV4Addr | | String | - | Char Value:0-255  No String length limit | IPv4 address |
| R | netmask | | String | - | Char Value:0-255  No String length limit | Net mask |
| R | connected | | Boolean | - | 0/1 | connected to AP |
| R | prevConnected | | Boolean | - | 0/1 | Previously connected AP |
| R | isHidden | | Boolean | - | 0/1 | Hidden AP |
| R | WifiConnectionType | | Enum | - | - |  |
|  |  | |  | CONNECTION\_TYPE\_PIN | 0x0 | PIN |
|  |  | |  | CONNECTION\_TYPE\_WPS\_PIN | 0x1 | WPS (Wi-Fi Protected Setup) PIN |
|  |  | |  | CONNECTION\_TYPE\_WPS\_PBC | 0x2 | WPS Push Button Control |
|  |  | |  | CONNECTION\_TYPE\_OPEN | 0x3 | Open WiFi |
|  |  | |  | CONNECTION\_TYPE\_HIDDEN\_SECURE | 0x4 | Hidden + security, SSID is not broadcasted |
|  |  | |  | CONNECTION\_TYPE\_HIDDEN\_OPEN | 0x5 | Hidden + open, SSID is not broadcasted |
| R | EWifiConnectionStatus | | Enum | - | - |  |
|  |  | |  | WIFI\_STATUS\_SUCCESS | 0 | Success |
|  |  | |  | WIFI\_AP\_SCAN\_SUCCESS | 1 | Scan Successful |
|  |  | |  | WIFI\_NO\_APS\_AVAILABLE | 2 | No Access Points available |
|  |  | |  | WIFI\_CONNECTED | 3 | Connected to Access Point |
|  |  | |  | WIFI\_NOT\_CONNECTED | 4 | Wifi is not connected to Access Point |
|  |  | |  | WIFI\_OTH\_FAIL | 5 | Authentication Failure |
|  |  | |  | WIFI\_CONNECTING | 6 | Connecting to Access Point |
|  |  | |  | WIFI\_SECURITY\_MISSMATCH | 7 | There is security type mismatch between what requested and what is in actual |
|  |  | |  | WIFI\_AUTHENTICATING | 8 | In authenticating |
|  |  | |  | WIFI\_ABORTED | 9 | Connection aborted |
|  |  | |  | WIFI\_WRONG\_PASSWORD | 10 | Failed with wrong password |
|  |  | |  | WIFI\_WRONG\_WPS\_PIN | 11 | Failed with wrong WPS PIN |
|  |  | |  | WIFI\_CONNECTION\_PROGRESS | 12 | In connecting |
|  |  | |  | WIFI\_RESTRICTED\_NETWORK | 13 | Connection failure due to trying to connect to a restricted network eg : TCU hotspot |
|  |  | |  | WIFI\_NOT\_IMPLEMENTED | 101 | Not Implemented |
|  |  | |  | WIFI\_STATUS\_FAIL | 102 | Failure |
|  |  | |  | WIFI\_STATUS\_FAIL\_OTHER | 203 | Other Failure or Connecting to AP Failed |
|  |  | |  | WIFI\_CANCEL\_CONNECTION | 104 | Cancelled |

#### MD-REQ-380284/A-CancelConnectToApimAP

This API is used internally by WIRClient and WIRServer to request to cancel connecting to an available AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380285/A-DisconnectFromApimAP

This API is used internally by WIRClient and WIRServer to request to disconnect from an available AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380286/A-ForgetApimAP

This API is used internally by WIRClient and WIRServer to request to forget an available AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | ssid | | String | - |  | SSID of AP |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380287/A-GetWpsPin

This API is used internally by WIRClient and WIRServer to request to get a WPS PIN. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | WlanWpsSecurity | | Enum | - | - |  |
|  |  | |  | WLAN\_WPS\_MIN | 0x0 |  |
|  |  | |  | WLAN\_WPS\_KEYPAD | 0x1 | AP supplies the PIN |
|  |  | |  | WLAN\_WPS\_PIN | 0x2 | STA supplies the PIN |
|  |  | |  | WLAN\_WPS\_PBC | 0x3 | Pushbutton |
|  |  | |  | WLAN\_WPS\_MAX | 0x4 |  |
|  | pin | | String | - | Char Value:0-255  No String length limit | The Pin when using keypad type |

#### MD-REQ-380288/B-GetNetworkDetails

This API is used internally by WIRClient and WIRServer to request to get network details. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType | - |  | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - | - | TBD |
| R | WlanIpv4AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |

#### MD-REQ-380289/A-GetEnableStatus

This API is used internally by WIRClient and WIRServer to request to get enable status. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | status | | Boolean | - | 0/1 | True - enabled, false-disabled |

#### MD-REQ-380290/A-SetNotification

This API is used internally by WIRClient and WIRServer to request to set notification on/off. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | on | | Boolean | - | 0/1 | On or off |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |

#### MD-REQ-380291/A-GetNotification

This API is used internally by WIRClient and WIRServer to request to get notification on/off status. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| **Response** | | | | | | |
| R | appId | | String | - | Char Value:0-255  No String length limit | Requesting app ID |
| R | HmiRet | | Enum | - | - |  |
|  |  | |  | HMI\_ERROR | 0x0 | Error/Failure |
|  |  | |  | HMI\_SUCCESS | 0x1 | Success |
| R | on | | Boolean | - | 0/1 | On or off |

#### MD-REQ-380298/C-WlanNQM

This API is used internally by WIRClient and WIRServer to request a WLAN network quality measurement. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | EcuType | | Enum | - | - | ECU type |
|  |  | |  | ECU\_UNK | 0x0 | Error |
|  |  | |  | ECU\_ECG | 0x1 | ECG |
|  |  | |  | ECU\_TCU | 0x2 | TCU |
|  |  | |  | ECU\_SYNC | 0x3 | SYNC |
|  |  | |  | ECU\_TCU\_B | 0x4 | TCU-B. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDS | 0x5 | SDS. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADSIM | 0x6 | ADSIM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_DDSM | 0x7 | DDSM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADM | 0x8 | ADM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ESPM | 0x9 | ESPM+. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDM | 0x10 | SDM. Applicable to AV vehicles Only |
| R | NetworkInterface  Type | | Enum | - | - |  |
|  |  | |  | IFACE\_NONE | 0x0 | None selected/specified |
|  |  | |  | IFACE\_TCUCELL | 0x1 | Cellular interface on TCU |
|  |  | |  | IFACE\_TCUWIFI | 0x2 | WLAN interface on TCU |
|  |  | |  | IFACE\_SYNCWIFI | 0x3 | WLAN interface on SYNC |
|  |  | |  | IFACE\_SYNCAPPL | 0x4 | AppLink interface on SYNC (currently out of scope for CM) |
|  |  | |  | IFACE\_TCUBCELL | 0x5 | Cellular interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_TCUBWIFI | 0x6 | WLAN interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ESPMLAN | 0x7 | ESPM+ LAN. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_DDSMLAN | 0x8 | DDSM LAN. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ERR | 0x9 | Error condition |
| **Response** | | | | | | |
| R | NqmRet | | Enum | - | - |  |
|  |  | |  | NQM\_ERROR | 0x0 | Error/Failure |
|  |  | |  | NQM\_SUCCESS | 0x1 | Success |
| R | EcuType | | Enum | - | - | ECU type |
|  |  | |  | ECU\_UNK | 0x0 | Error |
|  |  | |  | ECU\_ECG | 0x1 | ECG |
|  |  | |  | ECU\_TCU | 0x2 | TCU |
|  |  | |  | ECU\_SYNC | 0x3 | SYNC |
|  |  | |  | ECU\_TCU\_B | 0x4 | TCU-B |
|  |  | |  | ECU\_SDS | 0x5 | SDS |
|  |  | |  | ECU\_ADSIM | 0x6 | ADSIM |
|  |  | |  | ECU\_DDSM | 0x7 | DDSM |
|  |  | |  | ECU\_ADM | 0x8 | ADM |
|  |  | |  | ECU\_ESPM | 0x9 | ESPM+. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDM | 0x10 | SDM. Applicable to AV vehicles Only |
| R | NetworkInterface  Type | | Enum | - | - |  |
|  |  | |  | IFACE\_NONE | 0x0 | None selected/specified |
|  |  | |  | IFACE\_TCUCELL | 0x1 | Cellular interface on TCU |
|  |  | |  | IFACE\_TCUWIFI | 0x2 | WLAN interface on TCU |
|  |  | |  | IFACE\_SYNCWIFI | 0x3 | WLAN interface on SYNC |
|  |  | |  | IFACE\_SYNCAPPL | 0x4 | AppLink interface on SYNC (currently out of scope for CM) |
|  |  | |  | IFACE\_TCUBCELL | 0x5 | Cellular interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_TCUBWIFI | 0x6 | WLAN interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ESPMLAN | 0x7 | ESPM+ LAN. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_DDSMLAN | 0x8 | DDSM LAN. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ERR | 0x9 | Error condition |
| R | WlanStaSmState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_SM\_STATE\_OFF | 0x0 | WLAN off |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ENABLING | 0x1 | Enabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ON | 0x2 | On |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTING | 0x3 | Connecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISABLING | 0x6 | Disabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ERROR | 0x7 | Error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_AUTHERROR | 0x8 | Authentication error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_NWNOTFOUND | 0x9 | Network not found |
| R | macAddr | | String | - | Char Value:0-255  No String length limit | MAC Address |
| R | WlanStaConnState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTED | 0x0 | Disconnected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_WPS\_ACTIVE | 0x1 | WPS pending |
|  |  | |  | WLAN\_STA\_CON\_STATE\_ASSOCIATING | 0x2 | Associating |
|  |  | |  | WLAN\_STA\_CON\_STATE\_IP\_ADDRESSING | 0x3 | Getting IP address |
|  |  | |  | WLAN\_STA\_CON\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_CON\_STATE\_AUTHERROR | 0x6 | Authentication error |
|  |  | |  | WLAN\_STA\_CON\_STATE\_NWNOTFOUND | 0x7 | Network not found |
| R | connSsid | | String | - | Char Value:0-255 | Connected AP's SSID |
| R | connBssid | | String | - | No String length limit | Connected AP's BSSID |
| R | conn80211Tech | | String | - | Char Value:0-255 | Connected AP's 802.11 technology, 802.11a, … |
| R | connChannel | | Int32 | - | 0-4294967295 | Connected AP's channel |
| R | bandwidth | | Int32 | - | 0-4294967295 | Connected AP's bandwidth |
| R | dataRate | | Int32 | - | 0-4294967295 | Connected AP's data rate (bps) |
| R | signalStrength | | Int32 | - | 0-4294967295 | RSSI (dBm) |
| R | signalNoise | | Int32 | - | 0-4294967295 | Noise level (dBm) |
| R | signalSNR | | Int32 | - | 0-4294967295 | SNR (dB) |
| R | ipAddr | | String | - | Char Value:0-255  No String length limit | IP address |
| R | ipSubnet | | String | - | Char Value:0-255  No String length limit | Subnet mask |
| R | ipGateway | | String | - | Char Value:0-255  No String length limit | GW IP address |
| R | ipDnsPref | | String | - | Char Value:0-255  No String length limit | IP address of primary DNS |
| R | ipDnsAlt | | String | - | Char Value:0-255  No String length limit | IP address of secondary DNS |

#### MD-REQ-380299/C-CellNQM

This API is used internally by WIRClient and WIRServer to request a cellular network quality measurement. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | EcuType | | Enum | - | - | ECU type |
|  |  | |  | ECU\_UNK | 0x0 | Error |
|  |  | |  | ECU\_ECG | 0x1 | ECG |
|  |  | |  | ECU\_TCU | 0x2 | TCU |
|  |  | |  | ECU\_SYNC | 0x3 | SYNC |
|  |  | |  | ECU\_TCU\_B | 0x4 | TCU-B. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDS | 0x5 | SDS. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADSIM | 0x6 | ADSIM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_DDSM | 0x7 | DDSM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADM | 0x8 | ADM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ESPM | 0x9 | ESPM+ |
|  |  | |  | ECU\_SDM | 0x10 | SDM |
| **Response** | | | | | | |
| R | NqmRet | | Enum | - | - |  |
|  |  | |  | NQM\_ERROR | 0x0 | Error/Failure |
|  |  | |  | NQM\_SUCCESS | 0x1 | Success |
| R | EcuType | | Enum | - | - | ECU type |
|  |  | |  | ECU\_UNK | 0x0 | Error |
|  |  | |  | ECU\_ECG | 0x1 | ECG |
|  |  | |  | ECU\_TCU | 0x2 | TCU |
|  |  | |  | ECU\_SYNC | 0x3 | SYNC |
|  |  | |  | ECU\_TCU\_B | 0x4 | TCU-B. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDS | 0x5 | SDS. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADSIM | 0x6 | ADSIM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_DDSM | 0x7 | DDSM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADM | 0x8 | ADM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ESPM | 0x9 | ESPM+ |
|  |  | |  | ECU\_SDM | 0x10 | SDM |
| R | mcc | | Int32 | - | 0-4294967295 | Mobile country code |
| R | mnc | | Int32 | - | 0-4294967295 | Mobile network code |
| R | three\_digit\_mnc | | Boolean | - | 0/1 | 3 digits MNC or 2 digits MNC |
| R | CellularNwType | | Enum | - | - | Cellular Network Type |
|  |  | |  | NO\_NW\_TYPE | 0x0 | No network service |
|  |  | |  | GSM\_NW | 0x1 | GSM |
|  |  | |  | UMTS\_NW | 0x2 | UMTS |
|  |  | |  | LTE\_NW | 0x4 | LTE |
| R | CellularRatType | | Enum | - | - | Cellular RAT Type |
|  |  | |  | NO\_NW | 0x0 | No network available |
|  |  | |  | GSM | 0x1 | GSM |
|  |  | |  | GPRS | 0x2 | GPRS |
|  |  | |  | EDGE | 0x3 | EDGE |
|  |  | |  | UMTS | 0x4 | UMTS |
|  |  | |  | HSPA\_P | 0x5 | HSPA and HSPA+ |
|  |  | |  | LTE | 0x6 | LTE |
| R | signalStrength | | Int32 | - | 0-4294967295 | Signal Strength |
| R | CellularRegStatus | | Enum | - | - |  |
|  |  | |  | REG\_NO\_SRV | 0x0 | Registered but no service |
|  |  | |  | LIMITED\_SRV | 0x1 | Lmited service |
|  |  | |  | SRV | 0x2 | Full service |
| R | CellularSrvStatus | | Enum | - | - |  |
|  |  | |  | NO\_SRV | 0x0 | No service |
|  |  | |  | CS\_ONLY | 0x1 | CS only |
|  |  | |  | PS\_ONLY | 0x2 | PS only |
|  |  | |  | CS\_PS | 0x3 | CS and PS |
| R | CellularPwrStatus | | Enum | - | - |  |
|  |  | |  | MODEM\_PWR\_OFF | 0x0 | Modem Power Off |
|  |  | |  | MODEM\_PWR\_CELL\_OFF | 0x1 | Low Power Mode - Cellular Off |
|  |  | |  | MODEM\_PWR\_CELL\_ON | 0x2 | Cellular On |
|  |  | |  | MODEM\_PWR\_RESETTING | 0x3 | Power Reset |
|  |  | |  | MODEM\_PWR\_TESTING | 0x4 | Factory Test Mode |
|  |  | |  | MODEM\_PWR\_LPM\_IMMINENT | 0x5 | Just prior to Low Power Mode |
|  |  | |  | MODEM\_PWR\_RESET\_IMMINENT | 0x6 | Just prior to Resetting |
|  |  | |  | MODEM\_PWR\_RESET\_COMPLETE | 0x7 | modem back in service |

#### MD-REQ-380300/C-WlanNQMInd

This API is used internally by WIRClient and WIRServer to broadcast an update of WLAN network quality measurement.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | |
| **Method Type** | | OnChange | | | | | |
| **QoS Level** | | Default | | | | | |
| **Retained** | | No | | | | | |
|  | | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | | |
| R | EcuType | | Enum | - | - | ECU type |
|  |  | |  | ECU\_UNK | 0x0 | Error |
|  |  | |  | ECU\_ECG | 0x1 | ECG |
|  |  | |  | ECU\_TCU | 0x2 | TCU |
|  |  | |  | ECU\_SYNC | 0x3 | SYNC |
|  |  | |  | ECU\_TCU\_B | 0x4 | TCU-B. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDS | 0x5 | SDS. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADSIM | 0x6 | ADSIM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_DDSM | 0x7 | DDSM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ADM | 0x8 | ADM. Applicable to AV vehicles Only |
|  |  | |  | ECU\_ESPM | 0x9 | ESPM+. Applicable to AV vehicles Only |
|  |  | |  | ECU\_SDM | 0x10 | SDM. Applicable to AV vehicles Only |
| R | NetworkInterfaceType | | Enum | - | - |  |
|  |  | |  | IFACE\_NONE | 0x0 | None selected/specified |
|  |  | |  | IFACE\_TCUCELL | 0x1 | Cellular interface on TCU |
|  |  | |  | IFACE\_TCUWIFI | 0x2 | WLAN interface on TCU |
|  |  | |  | IFACE\_SYNCWIFI | 0x3 | WLAN interface on SYNC |
|  |  | |  | IFACE\_SYNCAPPL | 0x4 | AppLink interface on SYNC (currently out of scope for CM) |
|  |  | |  | IFACE\_TCUBCELL | 0x5 | Cellular interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_TCUBWIFI | 0x6 | WLAN interface on TCU-B. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ESPMLAN | 0x7 | ESPM+ LAN. Applicable to AV vehicles Only |
|  |  | |  | IFACE\_DDSMLAN | 0x8 | DDSM LAN.  Applicable to AV vehicles Only |
|  |  | |  | IFACE\_ERR | 0x9 | Error condition |
| R | WlanStaSmState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_SM\_STATE\_OFF | 0x0 | WLAN off |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ENABLING | 0x1 | Enabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ON | 0x2 | On |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTING | 0x3 | Connecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISABLING | 0x6 | Disabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ERROR | 0x7 | Error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_AUTHERROR | 0x8 | Authentication error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_NWNOTFOUND | 0x9 | Network not found |
| R | macAddr | | String | - | Char Value:0-255  No String length limit | MAC Address |
| R | WlanStaConnState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTED | 0x0 | Disconnected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_WPS\_ACTIVE | 0x1 | WPS pending |
|  |  | |  | WLAN\_STA\_CON\_STATE\_ASSOCIATING | 0x2 | Associating |
|  |  | |  | WLAN\_STA\_CON\_STATE\_IP\_ADDRESSING | 0x3 | Getting IP address |
|  |  | |  | WLAN\_STA\_CON\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_CON\_STATE\_AUTHERROR | 0x6 | Authentication error |
|  |  | |  | WLAN\_STA\_CON\_STATE\_NWNOTFOUND | 0x7 | Network not found |
| R | connSsid | | String | - | Char Value:0-255  No String length limit | Connected AP's SSID |
| R | connBssid | | String | - | Char Value:0-255  No String length limit | Connected AP's BSSID |
| R | conn80211Tech | | String | - | Char Value:0-255  No String length limit | Connected AP's 802.11 technology, 802.11a, … |
| R | connChannel | | Int32 | - | 0-4294967295 | Connected AP's channel |
| R | bandwidth | | Int32 | - | 0-4294967295 | Connected AP's bandwidth |
| R | dataRate | | Int32 | - | 0-4294967295 | Connected AP's data rate (bps) |
| R | signalStrength | | Int32 | - | 0-4294967295 | RSSI (dBm) |
| R | signalNoise | | Int32 | - | 0-4294967295 | Noise level (dBm) |
| R | signalSNR | | Int32 | - | 0-4294967295 | SNR (dB) |
| R | ipAddr | | String | - | Char Value:0-255  No String length limit | IP address |
| R | ipSubnet | | String | - | Char Value:0-255  No String length limit | Subnet mask |
| R | ipGateway | | String | - | Char Value:0-255  No String length limit | GW IP address |
| R | ipDnsPref | | String | - | Char Value:0-255  No String length limit | IP address of primary DNS |
| R | ipDnsAlt | | String | - | Char Value:0-255  No String length limit | IP address of secondary DNS |

#### MD-REQ-380301/A-CellNQMInd

This API is used internally by WIRClient and WIRServer to broadcast an update of cellular network quality measurement.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | mcc | | Int32 | - | 0-4294967295 | Mobile country code |
| R | mnc | | Int32 | - | 0-4294967295 | Mobile network code |
| R | three\_digit\_mnc | | Boolean | - | 0/1 | 3 digits MNC or 2 digits MNC |
| R | CellularNwType | | Enum | - | - | Cellular Network Type |
|  |  | |  | NO\_NW\_TYPE | 0x0 | No network service |
|  |  | |  | GSM\_NW | 0x1 | GSM |
|  |  | |  | UMTS\_NW | 0x2 | UMTS |
|  |  | |  | LTE\_NW | 0x4 | LTE |
| R | CellularRatType | | Enum | - | - | Cellular RAT Type |
|  |  | |  | NO\_NW | 0x0 | No network available |
|  |  | |  | GSM | 0x1 | GSM |
|  |  | |  | GPRS | 0x2 | GPRS |
|  |  | |  | EDGE | 0x3 | EDGE |
|  |  | |  | UMTS | 0x4 | UMTS |
|  |  | |  | HSPA\_P | 0x5 | HSPA and HSPA+ |
|  |  | |  | LTE | 0x6 | LTE |
| R | signalStrength | | Int32 | - | 0-4294967295 | Signal Strength |
| R | CellularRegStatus | | Enum | - | - |  |
|  |  | |  | REG\_NO\_SRV | 0x0 | Registered but no service |
|  |  | |  | LIMITED\_SRV | 0x1 | Lmited service |
|  |  | |  | SRV | 0x2 | Full service |
| R | CellularSrvStatus | | Enum | - | - |  |
|  |  | |  | NO\_SRV | 0x0 | No service |
|  |  | |  | CS\_ONLY | 0x1 | CS only |
|  |  | |  | PS\_ONLY | 0x2 | PS only |
|  |  | |  | CS\_PS | 0x3 | CS and PS |
| R | CellularPwrStatus | | Enum | - | - |  |
|  |  | |  | MODEM\_PWR\_OFF | 0x0 | Modem Power Off |
|  |  | |  | MODEM\_PWR\_CELL\_OFF | 0x1 | Low Power Mode - Cellular Off |
|  |  | |  | MODEM\_PWR\_CELL\_ON | 0x2 | Cellular On |
|  |  | |  | MODEM\_PWR\_RESETTING | 0x3 | Power Reset |
|  |  | |  | MODEM\_PWR\_TESTING | 0x4 | Factory Test Mode |
|  |  | |  | MODEM\_PWR\_LPM\_IMMINENT | 0x5 | Just prior to Low Power Mode |
|  |  | |  | MODEM\_PWR\_RESET\_IMMINENT | 0x6 | Just prior to Resetting |
|  |  | |  | MODEM\_PWR\_RESET\_COMPLETE | 0x7 | modem back in service |

#### MD-REQ-380304/B-ScanWiFiNetworks

This API is used internally by WIRClient and WIRServer to request a WiFi network scan. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | passive | | Boolean | - | 0/1 | true = passive scan (no pkts tx'd), false = active scan |
| R | WlanChannel | | Enum | - | - | List of channels of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | ssid | | String | - | Char Value:0-255  No String length limit | List of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required |
| R | scanId | | Int64 | - | 0-18446744073709551615 | ID value used to track scans |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | scanId | | Int64 |  | 0-18446744073709551615 | ID value used to track scans |
| R | WlanChannel | | Enum | - | - | List of channels of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | ssid | | String | - | Char Value:0-255  No String length limit | List of SSIDs to specifically scan for. Used to scan for hidden networks. "passive" will be ignored as an active scan is required |
| R | count | | Int32 | - | 0-4294967295 | number of APs found, repeats ap field |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID – scan result |
| R | bssid | | String | - | Char Value:0-255  No String length limit | MAC address of AP – scan result |
| R | WlanChannel | | Enum | - | - | Channel of AP – scan result |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | Wlan Bandwidth | | Enum | - | - |  |
|  |  | |  | WLAN\_BW\_MIN | 0x0 |  |
|  |  | |  | WLAN\_BW\_MHZ20 | 0x1 | 20MHz channel |
|  |  | |  | WLAN\_BW\_MHZ40 | 0x2 | 40MHz channel; must use 11n/11ac. Few devices support on 2.4Ghz |
|  |  | |  | WLAN\_BW\_MHZ80 | 0x3 | 80MHz channel; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ8080 | 0x4 | 80-80MHz, 2 non-contiguous 80MHz; must use 11ac |
|  |  | |  | WLAN\_BW\_MHZ160 | 0x5 | 160MHz channel; must 11ac |
|  |  | |  | WLAN\_BW\_MAX | 0x6 |  |
| R | rssi | | Int32 | - | 0-4294967295 | RSSI of beacon/probe response |
| R | WlanSecurity | | Enum | - | - | Security settings of scan results |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | isWpsSupported | | Boolean | - | 0/1 | does AP support WPS |
| R | isEssSupported | | Boolean | - | 0/1 | is AP part of an Extended Service Set |
| R | scanResultsComplete | | Boolean | - | 0/1 | All the scan results have been reported or not |

#### MD-REQ-380305/B-ConnectWiFiAP

This API is used internally by WIRClient and WIRServer to request to connect to an AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType | - |  | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - |  | TBD |
| R | WlanIpv4AddrType | | Enum | - |  |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface Name |

#### MD-REQ-380306/A-DisconnectWiFiAP

This API is used internally by WIRClient and WIRServer to request to disconnect from an AP. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |

#### MD-REQ-380307/B-ProfileUpdate

This API is used internally by WIRClient and WIRServer to request to update a profile. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | txId | | String | - | Char Value:0-255  No String length limit | Transaction ID |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | txId | | String | - | Char Value:0-255  No String length limit | Transaction ID |
| R | updateComplete | | Boolean | - | 0/1 | Is it last segment? |
| R | profile\_cnt | | String | - | Char Value:0-255  No String length limit | Number of entries of following 2 fields |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType | - | - | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - |  | TBD |
| R | WlanIpv4AddrType | | Enum | - |  |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |

#### MD-REQ-380308/A-ConnectionStatus

This API is used internally by WIRClient and WIRServer to request connection status. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID |
| R | bssid | | String | - | Char Value:0-255  No String length limit | BSSID, MAC address |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| **Response** | | | | | | |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID |
| R | bssid | | String | - | Char Value:0-255  No String length limit | BSSID, MAC address |
| R | WlanStaConnState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTED | 0x0 | Disconnected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_WPS\_ACTIVE | 0x1 | WPS pending |
|  |  | |  | WLAN\_STA\_CON\_STATE\_ASSOCIATING | 0x2 | Associating |
|  |  | |  | WLAN\_STA\_CON\_STATE\_IP\_ADDRESSING | 0x3 | Getting IP address |
|  |  | |  | WLAN\_STA\_CON\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_CON\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_CON\_STATE\_AUTHERROR | 0x6 | Authentication error |
|  |  | |  | WLAN\_STA\_CON\_STATE\_NWNOTFOUND | 0x7 | Network not found |
| R | WlanStaSmState | | Enum | - | - |  |
|  |  | |  | WLAN\_STA\_SM\_STATE\_OFF | 0x0 | WLAN off |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ENABLING | 0x1 | Enabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ON | 0x2 | On |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTING | 0x3 | Connecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_CONNECTED | 0x4 | Connected |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISCONNECTING | 0x5 | Disconnecting |
|  |  | |  | WLAN\_STA\_SM\_STATE\_DISABLING | 0x6 | Disabling |
|  |  | |  | WLAN\_STA\_SM\_STATE\_ERROR | 0x7 | Error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_AUTHERROR | 0x8 | Authentication error |
|  |  | |  | WLAN\_STA\_SM\_STATE\_NWNOTFOUND | 0x9 | Network not found |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |

#### MD-REQ-380310/A-StationModeStats

This API is used internally by WIRClient and WIRServer to request WLAN station mode statistics. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface Name |
| R | StatsCmdType | | Enum | - | - |  |
|  |  | |  | DISABLE | 0x0 | Disable |
|  |  | |  | ENABLE | 0x1 | Enable |
|  |  | |  | QUERY | 0x2 | Query current state |
| R | reportingWindow | | Int32 | - | 0-4294967295 | Seconds |
| **Response** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface Name |
| R | StatsCmdType | | Enum | - | - |  |
|  |  | |  | DISABLE | 0x0 | Disable |
|  |  | |  | ENABLE | 0x1 | Enable |
|  |  | |  | QUERY | 0x2 | Query current state |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | pktsTxV4 | | Int64 | - | 0-18446744073709551615 | Number of packets transmitted |
| R | pktsRxV4 | | Int64 | - | 0-18446744073709551615 | Number of packets received |
| R | bytesTxV4 | | Int64 | - | 0-18446744073709551615 | Number of bytes transmitted |
| R | bytesRxV4 | | Int64 | - | 0-18446744073709551615 | Number of bytes received |
| R | PktsDroppedTx | | Int64 | - | 0-18446744073709551615 | Number of dropped packets transmitted |
| R | PktsDroppedRx | | Int64 | - | 0-18446744073709551615 | Number of dropped packets received |

#### MD-REQ-380311/A-StationModeStatsInd

This API is used internally by WIRClient and WIRServer to broadcast an update of WLAN station mode statistics.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | ifaceName | | String | - | Char Value:0-255  No String length limit | Interface Name |
| R | StatsCmdType | | Enum | - | - |  |
|  |  | |  | DISABLE | 0x0 | Disable |
|  |  | |  | ENABLE | 0x1 | Enable |
|  |  | |  | QUERY | 0x2 | Query current state |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | pktsTxV4 | | Int64 | - | 0-18446744073709551615 | Number of packets transmitted |
| R | pktsRxV4 | | Int64 | - | 0-18446744073709551615 | Number of packets received |
| R | bytesTxV4 | | Int64 | - | 0-18446744073709551615 | Number of bytes transmitted |
| R | bytesRxV4 | | Int64 | - | 0-18446744073709551615 | Number of bytes received |
| R | PktsDroppedTx | | Int64 | - | 0-18446744073709551615 | Number of dropped packets transmitted |
| R | PktsDroppedRx | | Int64 | - | 0-18446744073709551615 | Number of dropped packets received |

#### MD-REQ-380312/B-ProfileUpdateInd

This API is used internally by WIRClient and WIRServer to broadcast a profile update indication.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | WlanProfileAction | | Enum | - | - |  |
|  |  | |  | ACT\_UPDATE | 0x0 | Add/update |
|  |  | |  | ACT\_REMOVE | 0x1 | Remove |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of network |
| R | bssid | | String | - | Char Value:0-255  No String length limit | Optional - BSSID of AP |
| R | WlanChannel | | Enum | - | - | Optional - Channel of AP |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | WLAN\_CH\_1 | 0x01 |  |
|  |  | |  | WLAN\_CH\_2 | 0x02 |  |
|  |  | |  | WLAN\_CH\_3 | 0x03 |  |
|  |  | |  | … | … |  |
|  |  | |  | WLAN\_CH\_165 | 0xA5 |  |
| R | WlanSecurity | | Enum | - | - | Security settings to use |
|  |  | |  | Reserved | 0x00 |  |
|  |  | |  | Open | 0x01 | Open or no security |
|  |  | |  | WEP | 0x02 | WEP |
|  |  | |  | WPS | 0x03 | WPS (WiFi Protected Setup) |
|  |  | |  | WPA\_Personal | 0x04 | WPA/WPA2/WPA3 Personal (passkey) |
|  |  | |  | WPA\_Enterprise | 0x05 | WPA/WPA2/WPA3 Enterprise (EAP-PEAP/EAP-TLS/etc) (not supported) |
|  |  | |  | MAX | 0x06 |  |
| R | WlanWepSecurity | | - | - | - | WEP Settings |
|  |  | | String | Key | Char Value:0-255  No String length limit |  |
|  |  | | Int32 | defaultKeyIndex | 0-4294967295 |  |
| R | WlanWpsSecurity | | - | - | - | WPS Settings |
|  |  | | WlanWpsType | - |  | Type |
|  |  | | String | - | Char Value:0-255  No String length limit | Pin |
| R | WlanWpaPersonalSecurity | | String | - | Char Value:0-255  No String length limit | WPA/WPA2/WPA3-Personal Settings - password |
| R | WlanPairwiseCipher | | Enum | - | - | Pair cipher |
|  |  | |  | PAIR\_MIN | 0x0 |  |
|  |  | |  | PAIR\_NONE | 0x1 | None |
|  |  | |  | PAIR\_TKIP | 0x2 | TKIP |
|  |  | |  | PAIR\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | PAIR\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | PAIR\_MAX | 0x5 |  |
| R | WlanGroupCipher | | Enum | - | - | Group cipher |
|  |  | |  | GROUP\_MIN | 0x0 |  |
|  |  | |  | GROUP\_NONE | 0x1 | None |
|  |  | |  | GROUP\_TKIP | 0x2 | TKIP |
|  |  | |  | GROUP\_CCMP | 0x3 | CCMP/AES |
|  |  | |  | GROUP\_TKIPCCMP | 0x4 | Mixmode – tkip or ccmp |
|  |  | |  | GROUP\_MAX | 0x5 |  |
| R | WlanWpaEnterprise | | TBD | - |  | TBD |
| R | WlanIpv4AddrType | | Enum | - |  |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV4\_ADDR\_NONE | 0x1 | No IPv4 Addressing is used |
|  |  | |  | WLAN\_IPV4\_ADDR\_STATIC | 0x2 | Static IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_CLIENT | 0x3 | DHCP Client IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_DHCP\_SERVER | 0x4 | DHCP Server IPv4 Address |
|  |  | |  | WLAN\_IPV4\_ADDR\_MAX | 0x5 |  |
| R | WlanIpv4Addr | | - | - | - |  |
|  |  | | String | Ip | Char Value:0-255  No String length limit | IP address of current connection |
|  |  | | String | Netmask | Char Value:0-255  No String length limit | Netmask of currenct connection |
|  |  | | String | Gateway | Char Value:0-255  No String length limit | default gateway of current connection |
|  |  | | String | dnsPref | Char Value:0-255  No String length limit | Prefered DNS server |
|  |  | | String | dnsAlt | Char Value:0-255  No String length limit | Secondary DNS server |
| R | WlanIpv6AddrType | | Enum | - | - |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_MIN | 0x0 |  |
|  |  | |  | WLAN\_IPV6\_ADDR\_NONE | 0x1 | No IPv6 Addressing is used |
|  |  | |  | WLAN\_IPV6\_ADDR\_STATIC | 0x2 | Static IPv6 Address |
|  |  | |  | WLAN\_IPV6\_ADDR\_MAX | 0x3 |  |
| R | WlanIpv6Addr | | TBD | - | - |  |
| R | ExclusiveUse | | Boolean | - | 0/1 | Exclusive to WIR Client |
| R | WlanProfileSource | | Enum | - | - | Origin of the profile |
|  |  | |  | PROFSRC\_DEFAULT | 0x0 | Default internal |
|  |  | |  | PROFSRC\_HMI | 0x1 | HMI/User configured |
|  |  | |  | PROFSRC\_CLOUD | 0x2 | Cloud pushed |
|  |  | |  | PROFSRC\_SPECIAL | 0x3 | Special Intent |
|  |  | |  | PROFSRC\_PROV | 0x4 | EOL provisioned |
| R | timestamp | | Int64 | - | 0-18446744073709551615 | Last known time stamp |
| R | hidden | | Boolean | - | 0/1 | Hidden AP |

#### MD-REQ-380313/A-WlanStateInd

This API is used internally by WIRClient and WIRServer to broadcast a WLAN station mode state update.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | wlanState | | Boolean | - | 0/1 | WLAN On/Off |
| R | wlanNotifState | | Int32 | - | - | Notification On/Off |

#### MD-REQ-380314/A-WhsInfo

This API is used internally by WIRClient and WIRServer to request WIFI HotSpot Information. WIR also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (A-Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| **Response** | | | | | | |
| R | Intf | | String | - | Char Value:0-255  No String length limit | The name of the interface |
| R | macAddr | | String | - | Char Value:0-255  No String length limit | MAC address of the interface |
| R | ssid | | String | - | Char Value:0-255  No String length limit | SSID of our AP |
| R | password | | String | - | Char Value:0-255  No String length limit | Password |
| R | WlanRet | | Enum | - | - |  |
|  |  | |  | WLAN\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WLAN\_SUCCESS | 0x1 | Success |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |

#### MD-REQ-380315/A-SignalUpdateInd

This API is used internally by WIRClient and WIRServer to broadcast a WLAN signal update.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | OnChange | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | ifId | | Int32 | - | 0-4294967295 | Interface ID |
| R | rssi | | Int32 | - | 0-4294967295 | RSSI of the currently connected AP |

#### MD-REQ-402837/A-ReadPolicyTable

This API is used by WIRServer (internally) and WIRClients to request current policy table. WIRServer also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | policyTableLength | | Int32 | - | 0-4294967295 | Length of policyTable |
| R | policyTable | | Binary | - | GPB data | GPB compressed policy table |

#### MD-REQ-402838/A-EnableSyncWifiStaMode

This API is used by WIRServer (internally) and WIRClients to enable WIRClient1 WiFi STA mode. WIRServer also uses this API for its response.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | |
| **Method Type** | | One-Shot (Synch) | | | | | |
| **QoS Level** | | Default | | | | | |
| **Retained** | | No | | | | | |
|  | | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | | |
| R | WirRet | | Enum | - | - | Return Command |
|  |  | |  | WIR\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WIR\_SUCCESS | 0x1 | Success |

#### MD-REQ-402839/A-DisableSyncWifiStaMode

This API is used by WIRServer (internally) and WIRClients to disable WIRClient1 WiFi STA mode. WIRServer also uses this API for its response.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Method Type** | | One-Shot (Synch) | | | | |
| **QoS Level** | | Default | | | | |
| **Retained** | | No | | | | |
|  | | | | | | |
| **R/O** | **Name** | | **Type** | **Literals** | **Value** | **Description** |
| **Request** | | | | | | |
| - | - | | - | - | - | - |
| **Response** | | | | | | |
| R | WirRet | | Enum | - | - | Return Command |
|  |  | |  | WIR\_ERROR | 0x0 | Error/Failure |
|  |  | |  | WIR\_SUCCESS | 0x1 | Success |

# Functional Definition

## WIR-FUN-REQ-295913/A-Local Controller Client

### Requirements

#### WIR-REQ-295914/D-Local Controller Purpose

The Local controller resides in all WIRClients and WIRServer and can be extended seamlessly to other potential Ethernet interfacing ECU’s in the future (Example instrument cluster, ADAS etc.). The central controller resides in WIRServer. Typically, the Local controller interfaces with applications, queues / schedules the requests from applications and interfaces with central controller to provide the appropriate network interface to the applications. The Local controller also provides network interface on its own when Central controller is not available.

#### WIR-REQ-295915/A-Local Controller Interface With Central Controller



Whenever there is a request from application for an interface to the local controller the local controller requests for an interface from the central controller. If the central controller is available and provides an interface, the local controller provides the same interface to requesting application. If the central controller is not available, the local controller provides interface on its own to the requesting application.

The strategy outlined above shall be followed for every request.

#### WIR-REQ-295916/A-Receive Intent Requests

The Local controller shall interface with connectivity requesting apps and receives the intent requests from these applications.

#### WIR-REQ-295917/B-Process Intent Requests

The Local controller shall process the intent request by communicating with central controller from WIRServer. If central controller is not available, Local controller shall act on its own and process the intent requests.

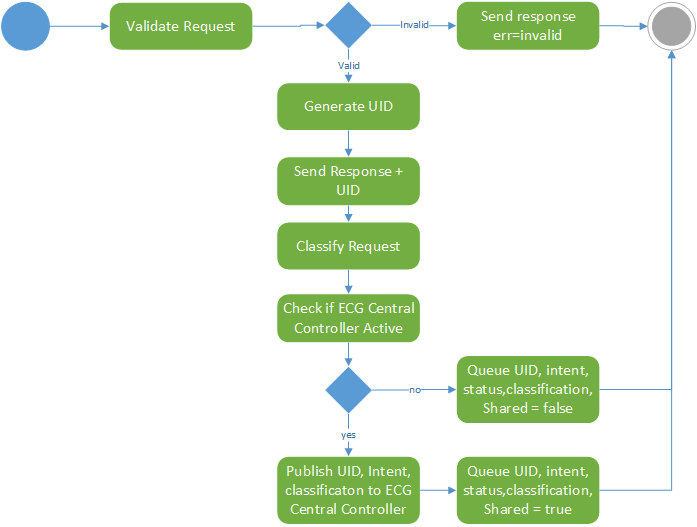
#### WIR-REQ-295918/B-Return Unique Id

The Local controller shall process the intent request and provide a unique identifier (AllocationID) as a response to the connectivity requesting application.

The Local controller and requesting application shall use the unique identifier (AllocationID) as a reference for all future interactions.

#### WIR-REQ-295919/B-Flow

The below UML diagram describes the flow of connection controller module (UID equivalent to AllocationID).



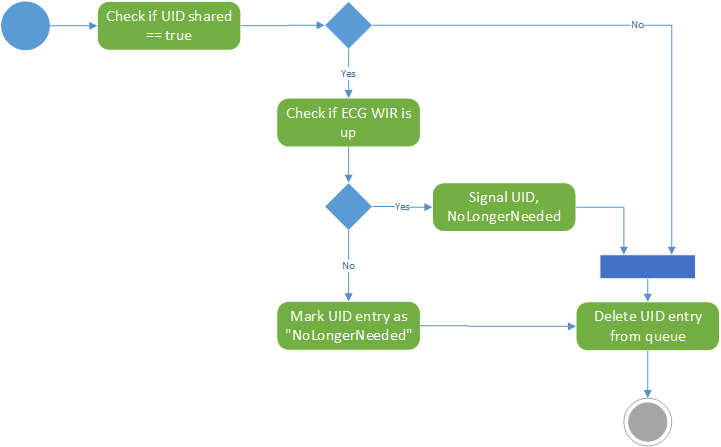
#### WIR-REQ-295920/B-Maintain AllocationID’s

The Local connection controller shall maintain the active AllocationID’s and their status whether

1. Interface provided
2. Interface is used
3. Network interface available
4. Network interface not available

#### WIR-REQ-295921/B-Remove AllocationID’s From Queue Once Application Notifies It Is No Longer Needed

The following UML activity depicts the behavior of the local connection controller when an application signals an AllocationID is complete and no longer needed (UID is equivalent to AllocationID).



#### WIR-REQ-295922/C-Responding To Queries From Applications Requesting Connectivity

The Local controller will also be responsible for responding to queries such as interface availability, interface health, etc. from connectivity demanding applications. The Local controller will use the unique identifier (AllocationID) to identify the specific query referenced.

The Local controller shall use the unique ID (AllocationID) as a reference to respond to queries.

#### WIR-REQ-295923/A-Query Response Details

The Local controller shall provide one of the following statuses back to the requesting application

1. Success – If interface has been provided successfully
2. Not supported – if the requested interface for the application is not supported
3. In progress – If the requested interface is not yet available to be provided
4. Interface – The IP address of the interface provided
5. Type of Interface – Whether the interface is WIFI (SYNC / TCU), Cellular etc.

#### WIR-REQ-295924/A-Scheduling

The local controller is also responsible for scheduling the different connectivity interface requests received from applications if central controller is not available.

#### WIR-REQ-295925/A-Removal From The Scheduling Queue

Once connection has been provided to application the request shall be removed from scheduling queue by local controller.

### Use Cases

#### WIR-UC-REQ-296107/D-Local controller provides connection interface to requesting application

|  |  |
| --- | --- |
| **Actors** | WIRServer, Any WIRClient |
| **Pre-conditions** | 1. Application requests WIR local controller for connection interface |
| **Scenario Description** | Local controller works with central controller to provide interfaces to requesting application |
| **Post-conditions** | Local controller successfully sends the request to the central controller  Central controller processes the request and provides interface back to the local controller |
| **Interfaces** |  |
| **Notes** | Exception use cases  Central controller not available |

#### WIR-UC-REQ-296108/D-Central controller not available

|  |  |
| --- | --- |
| **Actors** | WIRServer, Any WIRClient |
| **Pre-conditions** | 1. Application requests WIR local controller for connection interface |
| **Scenario Description** | Local controller requests central controller to provide interfaces to requesting application. But central controller is not available |
| **Post-conditions** | Local controller provides interface to the requesting application on its own. |
| **Interfaces** |  |
| **Notes** |  |

### White Box View

## WIR-FUN-REQ-295926/A-Intents

### Requirements

#### WIR-REQ-295927/B-Intent Purpose

When applications request network interfaces via the controller API, intent is included as a parameter of the request. The intent parameter provides the WIR context information about when, and with what kind of interface, to service the request.

#### WIR-REQ-295928/E-Types Of Intents

The WIRClient1 applications can leverage the following types of intents:

1. Fore ground
2. Back ground
3. Back ground guaranteed

The WIRClient2 applications can leverage the following types of intents:

1. Fore ground
2. Back ground
3. Back ground guaranteed
4. Offpeak

The WIRClient3 applications can leverage the following types of intents:

1. Fore ground
2. Back ground
3. Back ground guaranteed

The WIRClient4 applications can leverage the following types of intents

1. Fore ground
2. Back ground
3. Back ground guaranteed

The WIRClient5 applications can leverage the following types of intents

1. Fore ground
2. Back ground
3. Back ground guaranteed

The WIRServer applications can leverage the following types of intents:

1. Fore ground
2. Back ground
3. Back ground guaranteed
4. Offpeak

#### WIR-REQ-295929/B-Intent Priority

The WIRServer applications shall specify the priority of the request along with the intent.

If priority is selected then that application will be treated with higher priority compared to an application which does not select the priority flag.

#### WIR-REQ-295930/A-Off-Peak

Due to vehicle battery power constraints, WIRClient1 applications are not allowed to use off-peak flags.

#### WIR-REQ-295932/E-Intent Structure

The Logical intent structure is as defined below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Intent** | **Priority**  **Level** | **Expiry**  **timer** | **Cellular Only** | **WiFi Preferred** | **Timer expiration** | **Supported**  **Interface** | **Allowed app** |
| Foreground | High  Medium  Low (default) | NA | Y: Supports only cellular only  N: Default value. Check WiFi Preferred | Y : WiFi and fallback to cellular  N : Cellular and fallback to WiFi | NA | Cellular APN1, Cellular APN2,  WiFi | All |
| Background | NA | NA | NA | NA | Return failure | WiFi | All except FCI app |
| Background guaranteed | NA | 1~N | NA | NA | Off-peak=Y : Convert to off-peak  Off-peak=N : Convert to foreground cellular  Once foreground intent is provided, WIR shall not save the request anymore. | WiFi and fallback to cellular.  After fallback to cellular and WiFi becomes available, WIR shall provide WiFi again. | All except FCI app |
| Off-peak | High  Medium | NA | NA | NA | NA | Cellular APN1 | All |

For DuerOS target, Foreground and Background intent shall have a NetId field which will be passed to VNM.

#### WIRv2-REQ-470984/A-Intent Structure

The Logical intent structure is as defined below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Intent** | **Priority**  **Level** | **Expiry**  **timer** | **Cellular Only** | **WiFi Preferred** | **Timer expiration** | **Supported**  **Interface** | **Allowed app** |
| Foreground | High  Medium  Low (default) | NA | Y: Supports only cellular only  N: Default value. Check WiFi Preferred | Y : WiFi and fallback to cellular  N : Cellular and fallback to WiFi | NA | Cellular APN1, Cellular APN2,  WiFi | All |
| Background | NA | NA | NA | NA | Return failure | WiFi | All except FCI app |
| Background guaranteed | NA | 1~N | NA | NA | Fallback to cellular connection | WiFi and fallback to cellular.  After fallback to cellular and WiFi becomes available, WIR shall provide WiFi again. | All except FCI app |

#### WIR-REQ-369964/C-Intent processing

1. Foreground intent

Foreground intent is for cellular or WiFi interface. If “WiFi Preferred” enabled, WIR shall try WiFi and fallback to cellular interface. If “WiFi Preferred” not enabled, WIR shall try cellular and fallback to WiFi interface. If “Cellular only” is enabled, WIR shall try cellular and returns failure if cellular service is not available. If “Cellular only” is not enabled, WIR shall check WiFi Preferred flag. If “cellular only” is not provided, WIR shall set it as not enabled.

1. Background intent

WIR shall try to allocate WiFi interface. {TBD- After ignition off and on, WIR shall cancel any pending background intent request.}

1. Background guaranteed intent

WIR shall try to allocate WiFi interface until time out. Timer value 0 is invalid and WIR shall return invalid parameter error.

If off-peak is not set, upon timer expiration, WIR shall convert it to foreground intent and fallback to cellular allocation. Once cellular interface is provided, WIR shall not persist the request.

If off-peak is set, upon timer expiration, WIR shall convert it to off-peak intent.

After WIRServer off, WIR shall cancel any pending background guaranteed intent request without off-peak flag.

If off-peak flag is set, WIR shall check off-peak is allowed DID. If off-peak is allowed, WIR shall convert it to off-peak intent regardless timer expiration state. If off-peak is not allowed, WIR shall return “off-peak not allowed” failure.

#### WIR-REQ-295934/E-Intents And Possible Interfaces

The below table describes at a high level the different intents and their possible interfaces:

|  |  |  |
| --- | --- | --- |
| **Intent state** | **Possible interfaces** | **Traffic classifications** |
| Foreground | WIRClient2 APN1,  WIRClient5 APN1,  WIRClient1 WLAN services,  WIRClient2 WLAN services,  WIRClient5 WLAN services | Traffic classification – 1  Traffic classification – 2  Traffic classification – 3 |
| Background, Background guaranteed but timer not expired | WIRClient1 WLAN services,  WIRClient2 WLAN services,  WIRClient5 WLAN services | Traffic classification – 4  Traffic classification – 5 |
| Background guaranteed timer expired and off-peak selected | WIRClient2 APN1 off-peak, | Traffic classification – 4 |
| Background guaranteed timer expired and off-peak not selected | WIRClient2 APN1,  WIRClient5 APN1 | Traffic classification – 4 |
| Off- peak | WIRClient2 APN 1 off-peak, | Traffic classification – 4  Traffic classification – 5 |

#### WIRv2-REQ-470985/A-Intents And Possible Interfaces

The below table describes at a high level the different intents and their possible interfaces:

|  |  |  |
| --- | --- | --- |
| **Intent state** | **Possible interfaces** | **Traffic classifications** |
| Foreground | WIRClient2 APN1,  WIRClient5 APN1,  WIRClient1 WLAN services,  WIRClient2 WLAN services,  WIRClient5 WLAN services | Traffic classification – 1  Traffic classification – 2  Traffic classification – 3 |
| Background, Background guaranteed but timer not expired | WIRClient1 WLAN services,  WIRClient2 WLAN services,  WIRClient5 WLAN services | Traffic classification – 4  Traffic classification – 5 |
| Background guaranteed timer expired | WIRClient2 APN1 WIRClient5 APN1,  WIRClient2 APN2,  WIRClient5 APN2 | Traffic classification – 4 |

#### WIR-REQ-296061/E-Bandwidth Shaping

WIRClients / Server shall support Bandwidth shaping to achieve Traffic classification priority. For example, an application with priority high shall have a better connection than an application with priority Low.

#### WIR-REQ-385871/A-China Basic Service and Entertainment Service

Basic service shall use APN1 through foreground intent.

Entertainment service shall use APN2 through foreground intent.

Both service’s APN selection is controlled through policy table and policy table shall have correct value to support APN1 and APN2.

Cellular Only flag shall be set to enabled to avoid switching to WiFi.

### Use Cases

### White Box View

## WIR-FUN-REQ-295935/A-Policy Manager

### Requirements

#### WIR-REQ-295936/D-Purpose

Applications shall request connectivity interface from WIR by sending an intent request. WIR shall validate the application intent request against the intent privileges set for every application. The connectivity privileges of every application shall be stored in all applicable WIRClients and WIRServer. These privileges shall be updatable over the air from backend.

#### WIR-REQ-295937/C-Global Application ID

Every application shall have a global application ID which shall be maintained in the Ford cloud. This global application ID shall be used by WIR to identify applications, validate applications and map intent and URL privileges for the application. The individual applications when requesting network interfaces shall supply the Application ID along with the intent.

#### WIR-REQ-295938/B-Intent Policy Table Privileges

Each application intent privilege can be specified using a 2 bytes value.

#### WIR-REQ-295939/F-Intent Policy Table Privilege Breakdown

Each application intent privilege can be specified using a 2 bytes value. The encoding of the 2 byte value is defined below (ex. bit 0 is intended for WIRClient2 cellular APN1, bit 1 is intended for WIFI etc.):

|  |  |  |
| --- | --- | --- |
| **Interfaces** | **Bit position**  **1- enabled**  **0 - disabled** | **Value** |
| WIRClient2 Cellular APN1 | 0 (LSB) | 0x0001 |
| WIFI | 1 | 0x0002 |
| WIRClient2 Cellular APN2 | 2 | 0x0004 |
| Reserved | 3 | 0x0008 |
| Reserved | 4 | 0x0010 |
| Reserved | 5 | 0x0020 |
| Reserved (Offpeak) | 6 | 0x0040 |
| Reserved | 7 | 0x0080 |
| Reserved | 8 | 0x0100 |
| Factory WIFI | 9 | 0x0200 |
| WIRClient5 Cellular APN1 | 10 | 0x0400 |
| WIRClient5 Cellular APN2 | 11 | 0x0800 |
| WiFi connection change report | 12 | 0x1000 |
| System App | 13 | 0x2000 |
| Reserved Modem Preference\*\* | 14 | 0x4000 |
| Reserved | 15 (MSB) | 0x8000 |

\*\*Reserved Modem Preference is used for future dual modem load balancing assignment flexibility and bit position will be set to determine the preference as follows: 0 – disabled = TCU, 1 – enabled = TCU-B.

An interface is allowed when corresponding bit is set. Bit 0 is LSB (0x01) and bit 15 is MSB (0x80).

“WIRClient2 Cellular APN1“ and “WIRClient2 Cellular APN2“ are mutually exclusive. Each application shall enable only one of the interfaces and shall not enable both. If both APN1 and APN2 are enabled on policy table, WIRServer shall allocate using APN1 and provide the interface to the application.

WIRServer shall only allow a policy table read request (via ReadPolicyTable) to applications with the “System App” bit set.

#### WIR-REQ-295940/B-Intent Policy Table Structure

The structure of the intent policy table is as defined below:

|  |  |
| --- | --- |
| **Application ID** | **Connectivity intent permissions** |
|  |  |

#### WIR-REQ-295941/B-Intent Policy Storage

The applicable WIRClient local controller shall store intent policy privilege.

#### WIR-REQ-350866/A-WIRClient3 Policies

The WIRServer shall maintain the policies for WIRClient3. WIRClient3 shall have an updatable policy table in the future.

#### WIR-REQ-370380/A-WIRClient4 Policies

The WIRServer shall maintain the policies for WIRClient4. WIRClient4 shall have an updatable policy table in the future.

#### WIR-REQ-370381/A-Z2 Dual Modem Policy Limitation

Applications shall be privileged to use one of either WIRClient2 or WIRClient5, but not both client’s interfaces. The policy will be updatable to switch between clients, but not to allow applications to use both client interfaces.

#### WIR-REQ-295942/D-Intent Policy Update

Whenever there is an intent policy table change then Cloud / Backend pushes this information to the WIR policy manager residing in WIRServer. The policy table pushed by cloud shall be complete table, not delta. The WIRServer then validates the change, updates its own policy table and transmits the policy table update to each applicable WIRClient. Each applicable WIRClient shall in turn update their own policy tables.

WIRServer shall send a response to backend after it updates the policy table. After sending the response, WIRServer shall continue to update other client’s policy tables.

Backend shall wait 60 seconds to send another policy table update request to same vehicle.

The intent policy table update shall be as depicted in the diagram below:



#### WIR-REQ-369995/A-Sends Error for Corrupt policy file

WIR server shall not update the policy table if a corrupt policy file is received, instead it shall send an error.

#### WIR-REQ-295943/E-Intent Policy Update Acknowledgement

Once policy table is updated successfully each WIRClient shall provide an acknowledgement to WIRServer.

If a WIRClient fails to update its local copy and reports error, the WIRServer shall retry 5 times unless the error is a permanent error.

The WIRServer shall notify the cloud whether the policy updates are successful or failed. In case of failure, WIRServer shall notify the cloud with the appropriate error code

|  |  |
| --- | --- |
| **Policy Table Update Error codes** | **Error message/ Description** |
| OTHER\_ERRORS | All other Errors |
| PAYLOAD\_ERROR | There is error in the Payload |
| ECU\_POLICY\_UPDATE\_FAILED | Policy update in WIRClient failed |

PAYLOAD\_ERROR shall set an error description field as “RETRY: failed to parse command”. The cloud shall check “RETRY:” tag in the error description to see if it should resend policy table. WIRServer can set “RETRY:xxx” tag to any valid error codes if resend is required and the cloud shall resend the request whenever it sees “RETRY:xxx” tag regardless error code.

WIRServer shall set “RETRY:” field only when WIRServer update failed. All other ECU update failure shall not trigger retry.

WIRServer shall send an update result with following 2 fields after succeeded or failed to Ford backend. Failure shall be determined after predetermined number of retry by WIRServer.

Update result: success or failure

Description: Optional text field (ECU that failed to update policy, Ex., SYNC update failed)

#### WIR-REQ-295944/B-Intent Policy Update Notification To Applications

The WIRClients in addition to updating the policy table shall also publish the message via callback so that the individual application shall know there is a change in policy privilege. Each application shall also be able to request the current policy table via WIR API.

#### WIR-REQ-369996/A-Major version and Minor version

WIR server shall send the received Major and Minor Versions to cloud as part WIRPolicyUpdateCommand Response

#### WIR-REQ-369997/A-Default intent Policy file

WIRServer shall maintain the Default policy file with Application ID and Intent Policy Permissions. This policy file shall be updated upon receiving WIRPolicyUpdateCommand from cloud.

Default policy file shall be part of system image. If policy table is not available due to corruption, deleted or any other reason, WIRServer shall generate new policy table from default policy table.

#### WIR-REQ-369998/A-Intent policy update/rejection based on version

WIR Server shall update polices if it receives the upgraded version number and shall reject the policy update if it receives the downgraded policy version number than it currently holds.

#### WIR-REQ-385872/E-Intent policy update pull request

Intent policy update pull request follows query, query response and correlated alert FTCP command sequence.

WIRServer sends query and Ford backend response with query response. Correlated alert is sent by WIRServer.

WIRServer shall send a policy update pull query at once per every DID POLICY\_PULL\_IGN\_CYCLE ignition cycle if policy table has non-default version. When WIRServer has default policy table, it shall send a policy update pull query at every ignition cycle once FTCP connection has been established and all ECUs are provisioned or not present. For any pull request failure, WIRServer shall retry 3 times with 5 min delay.

When SW OTA updates is ready and WIR policy table is updated, WIR policy table push shall be arranged in before.

WIRServer shall set Major and Minor version of local policy table to the query.

Ford backend shall compare Major and Minor version of the query and from policy table version on the server. If Major version or Minor version doesn’t match, the server shall send a policy update request. Otherwise, the server shall not send a policy update request.

Ford backend shall send query response which has result code and ErrorDetail.

Result code shall be one of following.

* Success: Ford backend has newer policy table
* No newer version available: Vehicle has already latest policy table
* No policy table on server: Server doesn’t have available policy table
* Other error: Other errors happened

ErrorDetail – Text description for above error code. In case success, ErrorDetail shall be empty.

If result code of query response is set to Success, Ford backend shall set policy table to the query response. Otherwise, policy table shall not be part of query response.

WIRServer shall send correlated alert with following result code and ErrorDetail if Ford backend sent policy table on the query response.

Result code shall be one of following.

* Success: Correct policy table received
* Incorrect policy table: Received policy table format is incorrect
* Other error: Other errors happened

ErrorDetail – Text description for above error code. In case success, ErrorDetail shall be empty.

WIRServer shall apply new policy table after it sent correlated alert with Success result code.

### Use Cases

#### WIR-UC-REQ-296113/C-Cloud sends updated policy table to WIR

|  |  |
| --- | --- |
| **Actors** | WIRServer, Applicable WIRClients, Ford Cloud |
| **Pre-conditions** | 1. Ford cloud sends updated policy table to vehicle |
| **Scenario Description** | WIRServer receives updated policy table from Ford Cloud and sends the updated policy to applicable WIRClients |
| **Post-conditions** | WIRServer receives the updated policy table from Ford Cloud  WIRServer updates the policy and sends policies to applicable WIRClients  Each applicable WIRClient updates the policy table and sends confirmation to WIRServer  WIRServer sends confirmation to Ford cloud |
| **Interfaces** |  |
| **Notes** | Exception use case   1. Policy not updated in WIRServer / WIRClients |

#### WIR-UC-REQ-296114/D-Policy not updated in WIRServer / WIRClients

|  |  |
| --- | --- |
| **Actors** | WIRServer, Applicable WIRClients, Ford Cloud |
| **Pre-conditions** | 1. Ford cloud sends updated policy table to vehicle |
| **Scenario Description** | WIRServer receives updated policy table from Ford Cloud and but not able to update policy table in vehicle |
| **Post-conditions** | WIRServer receives the updated policy table from Ford Cloud  WIRServer updates the policy and sends policies to applicable WIRClients  A WIRClient could not update the policy  WIRServer sends failure to Ford cloud |
| **Interfaces** |  |
| **Notes** |  |

### White Box View

## WIR-FUN-REQ-295945/A-Tunnel Manager

### Requirements

#### WIR-REQ-295946/D-Purpose

The main purpose of WIR tunnel concept is to enable all networking channels, which includes exposing and sharing network interfaces between the WIRClients and WIRServer.

* WIRClient1’s Wi-Fi is exposed to WIRClients and WIRServer
* WIRClient2’s Cellular and Wi-Fi interfaces are accessible from WIRClients and WIRServer
* WIRClient5’s Cellular and Wi-Fi interfaces are accessible from WIRClients and WIRServer

#### WIR-REQ-295947/B-Tunnel Support

The WIRServer module shall support tunneling protocol. Each tunnel is used for virtualizing a network interface.

#### WIR-REQ-295948/D-WIRClients IP Address

WIRClients shall support receiving IP address from WIRServer module. The WIRServer module and WIRClient modules shall make sure that all tunnel IP address are randomly assigned when WIR central controller starts.

#### WIR-REQ-295950/C-IP Address Assignation

WIRClients shall support assigning the IP address received from WIRServer to the tunnel endpoint.

#### WIR-REQ-295951/C-IP Aliasing

WIRClients shall support IP aliasing in WIR, IP aliasing is used to add IP addresses to existing Ethernet device (eth0) to represent tunnel source addresses (ex. 10.11.0.1 and 10.11.0.5 on WIRClient1 side). It is also used to assign IP address (ex. 10.1.0.5) to created tunnel device (tun1).

#### WIR-REQ-295952/A-Allow WIRClient1 Edge Interface Access To Ethernet Connected ECU’s

WIRClient1 shall allow usage of edge interface namely WIFI from another Ethernet connected ECU via tunnel concept.

#### WIR-REQ-370382/A-Allow WIRClient2 Edge Interface Access To Ethernet Connected ECU’s

WIRClient2 shall allow usage of edge interfaces namely WIFI and cellular network from another Ethernet connected ECU via tunnel concept.

#### WIR-REQ-370383/A-Allow WIRClient5 Edge Interface Access To Ethernet Connected ECU’s

WIRClient5 shall allow usage of edge interfaces namely WIFI and cellular network from another Ethernet connected ECU via tunnel concept.

#### WIR-REQ-295953/C-Allow WIRClient Applications To Access Edge Interface Of Other Ethernet Connected ECU's

WIRClients outside of WIRClient2 shall allow applications within those WIRClients to access WIRClient2 cellular network and WIRClient2 WIFI network via Tunneling concept.

WIRClients outside of WIRClient1 shall allow applications within those WIRClients to access WIRClient1 WIFI network via Tunneling concept.

WIRClients outside of WIRClient5 shall allow applications within those WIRClients to access WIRClient5 cellular network and WIRClient5 WIFI network via Tunneling concept.

#### WIR-REQ-295954/B-WIRClient Interfacing With WIRServer Example

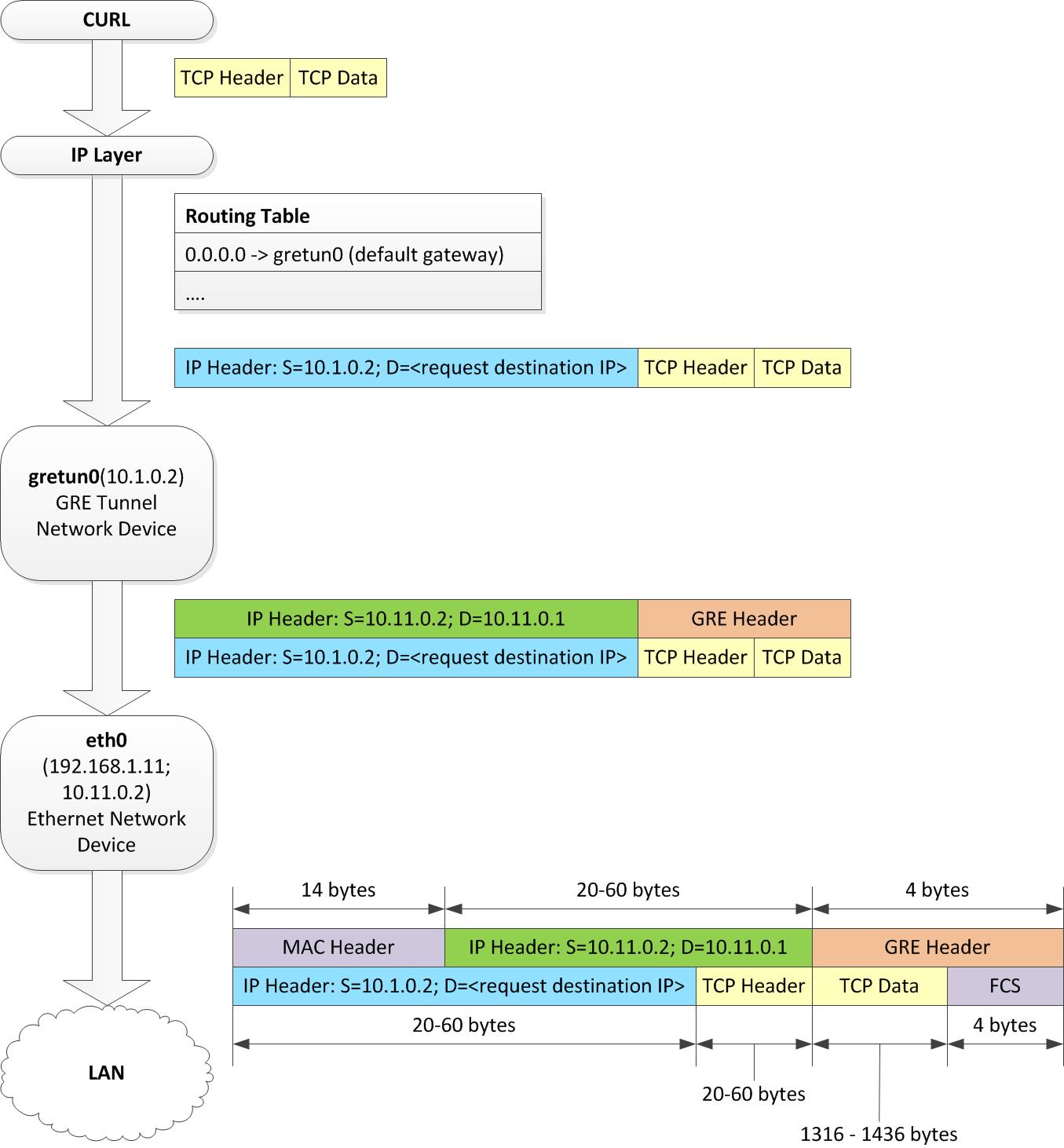
The below diagram outlines (provides an example) of how WIRClient1 will interface with WIRServer and expose its edge interfaces:



GRE tunnel encapsulation flow

#### WIR-REQ-295955/A-Sending A Request From WIRServer

Figures below provide more details on how data exchange works in case of using tunnels. In particular, they show how packages are created and sent from WIRServer to WIRClient1 and Internet and back. These sections assume a client application on the WIRServer would like to use a CURL library to make a cloud request. This request is routed over WIRClient1’s Wi-Fi interface.



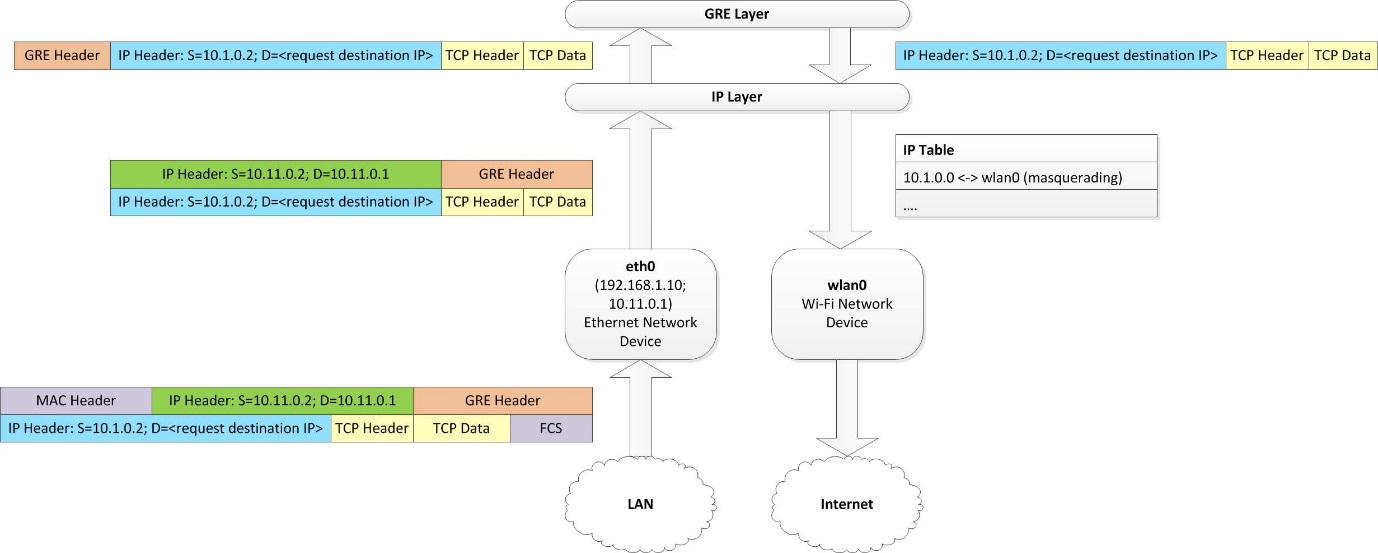
Sending request on the WIRServer

CURL is configured to use gretun0 network device to send its packets. That is why it uses gretun0 IP address as source address. The destination IP address is the final cloud destination. According to the routing table rules that we configure in advance, all packages, coming to Internet, should be sent via tunneling device (gretun1). Tunneling device wraps original IP packet, adding GRE header and new IP header to it. As there is a tunnel established from address 10.11.0.2 to address 10.11.0.1, gretun0 sets in newly added IP header. Notice that the tunneling device IP (10.1.0.2) is not presented in external IP header.

After that packet is wrapped by Ethernet device eth0 with a MAC header and sent to LAN.

This is an example proposal.

#### WIR-REQ-295956/B-Forwarding The Request To WIRClient WIFI Example Proposal

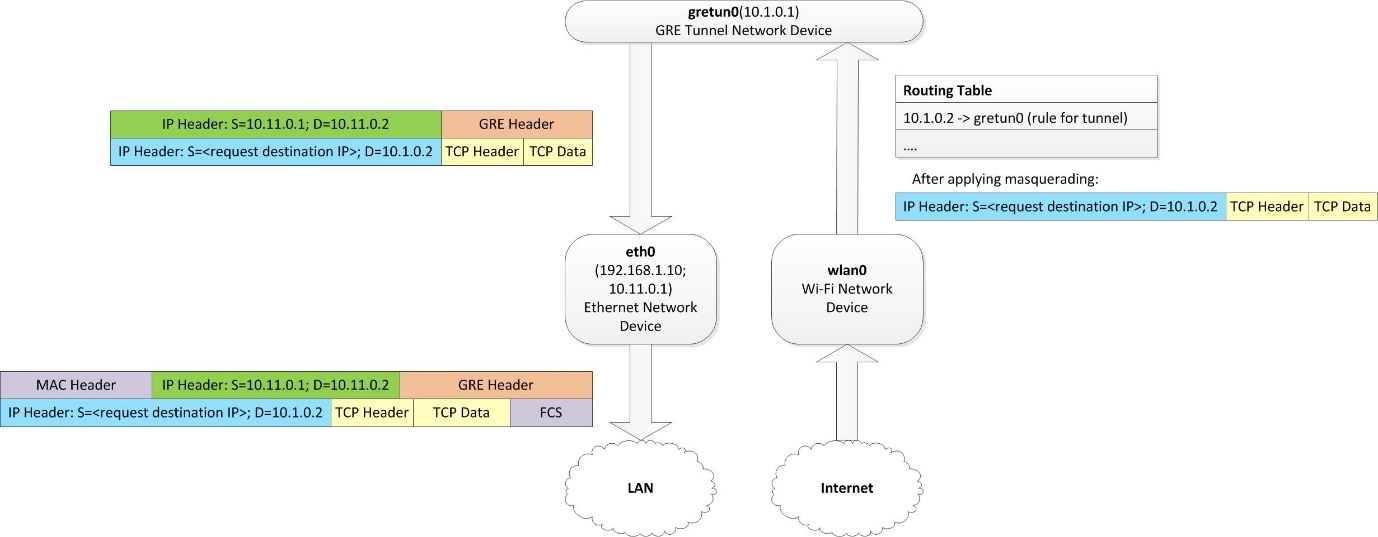


Sending request on WIRClient1

When packet arrives at the WIRClient1 Ethernet interface, it consists of the original IP packet, wrapped in a GRE Header, wrapped in a local IP packet, and finally wrapped in a MAC header. The Ethernet device removes MAC header and FCS and passes the local IP packet to the IP layer. The IP layer makes sure that current system has the IP address (10.11.0.1), which is the destination address in the IP packet header, and removes the IP header and passes the packet to the GRE layer. The GRE interface removes the GRE header and the original IP packet back to the IP routing layer. The IP routing layer sees the packet source address is 10.1.0.2 and matches the packet to an IP routing rule which routes all packets from the 10.1.0.2 IP address to the network device wlan0 (in this particular example we assume that we are sending packets via Wi-Fi). The packet is then transmitted via the WIRClient1 Wi-Fi interface.

#### WIR-REQ-295957/B-Receiving Response On WIRClient Module Example Proposal

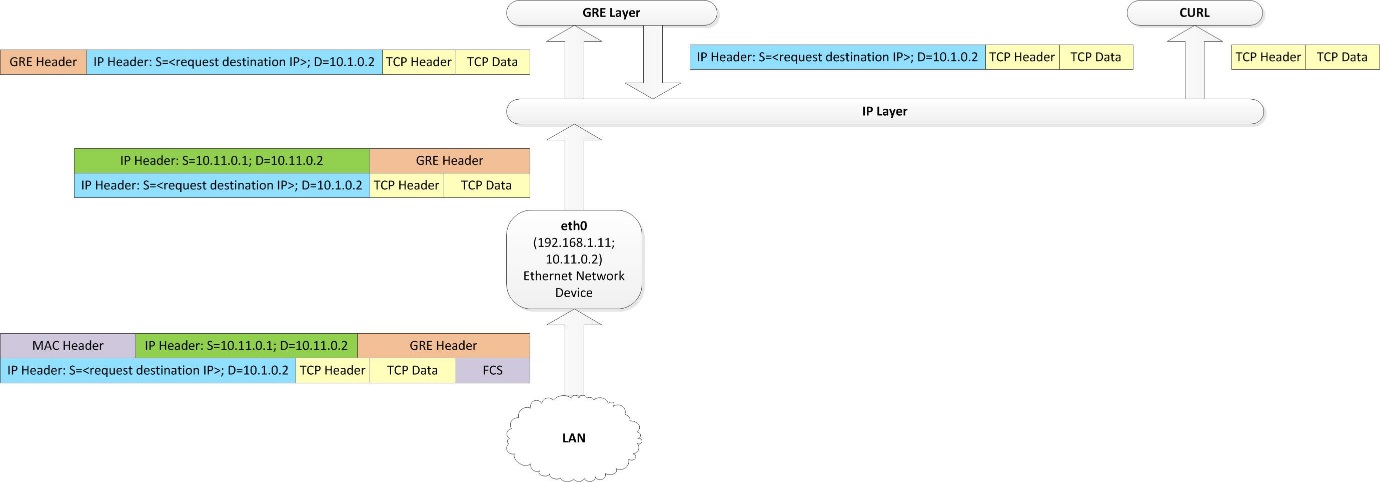
When the destination host in the cloud sends a response IP packet, it arrives at the WIRClient1 Wi-Fi interface. The response IP packet specifies the original source IP as 10.1.0.2, the address of the original GRE tunnel. The WIRClient1 IP layer identifies a rule that routes all packets destined for 10.1.0.2 to gretun0. Similar to earlier, when WIRServer sent a packet through gretun1, the return packet is wrapped in a GRE header by gretun0, and then by the WIRClient1 Ethernet interface.



Receiving response on WIRClient1

#### WIR-REQ-295958/A-Receiving Response On WIRServer Module Example Proposal

Similarly, response packet, when it arrives at the WIRServer interface, the MAC header is stripped by the WIRServer Ethernet interface. The local IP packet is stripped and routed to the GRE layer. The GRE layer strips the GRE header, and sends the encapsulated IP packet back to the IP layer. From there, the response packet is routed down the network stack to the socket being used by the CURL library.



Receiving response on WIRServer

### Use Cases

### White Box View

## WIR-FUN-REQ-295959/A-Diagnostics

### Requirements

#### WIR-REQ-295960/B-Purpose

The purpose of Diagnostics module is to provide WIFI diagnostics and connectivity interfaces log information.

#### WIR-REQ-295961/B-WIFI Diagnostics

Each time the WIFI interface is down, the WIRClient shall log the below information.

* SSID (Xfinity / non Xfinity (customer SSID is private and no need to store this information just classification is fine)
* Connect/Disconnect timestamps of the access point
* Data rate
* Band and channels utilized
* RSSI statistics
* Chipset roles / use
* Statistics about the data transmitted and received if possible
* Reason for disconnect

This information shall be stored by WIRServer for a period of 1 month and shall be sent to the diagnostics framework based on Ondemand request from Cloud via FTCP message.

For the on demand request message from Cloud please refer FTCP protofile.

#### WIR-REQ-295963/D-Diagnostics Information Storage

The WIRServer diagnostics module shall store WIFI diagnostics information and try to upload it to Ford cloud through diagnostics framework via only WiFi connection. Cellular connection is not allowed for the upload. If it can’t be uploaded, WIRServer shall try to upload it until WIR\_WIFI\_DIAGNOSTIC\_STORAGE\_TIME. After WIR\_WIFI\_DIAGNOSTIC\_STORAGE\_TIME, WIRServer shall delete it and as a result, it will not be uploaded to the cloud.

#### WIR-REQ-370232/A-On demand diagnostic request

Ford cloud shall be able to request diagnostic data upload. WIRServer shall upload any pending diagnostic data to cloud immediately when it is requested. WIRServer or diagnostic framework shall try WiFi connection first and fallback to cellular for the upload.

#### WIR-REQ-295964/B-Diagnostics Information Sending To Cloud

The applicable WIRClient diagnostics module shall store the diagnostics information and send the diagnostics data to the WIRServer which in turn sends to cloud via diagnostics framework upon request from diagnostic framework.

#### WIR-REQ-388321/C-Reporting WiFi Connection Event to SDN

WIRServer shall monitor WiFi connectivity events and report it to SDN through FTCP alert for marketing purpose. Also, WIRServer shall report WiFi client connection status change in real time. If FCI connection is not available, WiFi client connection status change report shall be delayed until FCI becomes available again. WIRServer shall not wake up ECG from LPR to send the report.

Marketing report shows how many times WiFi is connected to external AP. WIRServer shall count each connection and report it to SDN every WIFI\_CONNECTION\_ALERT\_INTERVAL day. WIRServer shall use WiFiNetworkUsageAlert FTCP command. The report shall have the following parameters:

* Count: number of times WiFi client connected to external AP
* Duration: number of seconds connected to external AP

SDN shall save WiFiNetworkUsageAlert to SCAV.

An application shall request WiFi client connection status change report by calling WIR API (ex. WifiConnStatusReport(bool enable)). An application shall call the API with enable=TRUE if it requires WiFi client connection status report. If the application doesn’t require the report anymore, it shall call the API with enable=FALSE. The request is valid only until WIRServer power off. After WIRServer powered on again, the application shall call the API again if it requires the report. WIRServer shall reject the WIR API if an application doesn’t have a permission for “WiFi connection change report” on policy table.

WiFi client connection status change shall be reported when the following 3 conditions are met:

* There are one or more application requesting WiFi interface through background intent
* There are one or more application requesting status change report
* WiFi client connection status changed

Also, WiFi client connection status change shall be sent once when report enabled which give current WiFi connection status.

The report shall have following parameters:

* Connection status: connected, disconnected
* Timestamp: time stamp of the eventDuration: number of seconds connected to external AP, valid for disconnected event only

Marketing report shall use WiFiNetworkUsageAlert FTCP message.

Marketing WiFiNetworkUsageAlert report flow diagram:



WiFi client connection status change report flow diagram:



#### WIR-REQ-295965/C-Sending The App Requests And Interfaces Provided To App To WIRServer Central Controller

If in the event the WIRClient provided interfaces to requesting applications without WIRServer central controller involvement, then the WIRClient shall store the details and shall send to WIRServer once WIRServer central controller becomes available.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unique AllocationID and Time stamp** | **Feature ID** | **Interface provided and Token information (This is just an example)** | **Edge Interface** | **Health of the all edge interfaces available** |
| 101 06142017 2.25.11 pm | Online traffic (example 701) | 10.100.10( emer, FCI etc.) and Token Key | TCU cellular | Qdiscs information if available |
| 103 06142017 2.25.25 pm | Parsed  (example 702) | 10.101.14(background generic) | TCU WIFI | Qdiscs information if available |

If Central controller is available and Central controller provides interface in that event the WIRClient shall provide AllocationID, Feature ID to the WIRServer Central controller.

#### WIR-REQ-402383/A-Controlling diagnostic data upload by CCS setting

If bAllow\_FEATURE46\_Diagnostics is set to ON, WIRServer shall upload diagnostic data. Otherwise, WIRServer shall not upload diagnostic data.

### Use Cases

#### WIR-UC-REQ-296112/E-WIR provides diagnostics information to cloud based on request from Cloud

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClients, Ford Cloud |
| **Pre-conditions** | 1. Customer triggers bug report |
| **Scenario Description** | Customer has a connectivity issue and triggers bug report to be sent to the cloud via on demand diagnostics |
| **Post-conditions** | WIR receives the request for WIR diagnostics logs  WIR retrieves the stored logs  WIR sends the logs to diagnostics framework  Diagnostics framework sends the logs to Ford cloud |
| **Interfaces** |  |
| **Notes** |  |

### White Box View

## WIR-FUN-REQ-295966/A-Data Usage

### Requirements

#### WIR-REQ-295967/D-Data usage calculation

The Data usage shall be calculated per application by each WIRClient and WIRServer.

#### WIR-REQ-295968/D-Data usage calculation scope

The Data usage shall be calculated per application in each of the Ethernet connected modules (all WIRClients and WIRServer).

#### WIR-REQ-295969/D-Data usage aggregation per application

WIRClients shall provide calculated data usage to WIRServer where data usage is aggregated.

If the same application resides in WIRClients and WIRServer then data usage shall be calculated collectively and aggregated by WIRServer.

#### WIR-REQ-295970/H-Data usage and edge interface

Data usage for each of the application shall be calculated individually for each of the application and each individual interface and stored in WIRServer module and shall be sent to cloud upon request. The WIRServer shall be responsible for maintaining / aggregating the calculated data usage details. All of the WIRClients shall provide the WIRServer details of data usage calculated per App calculated locally.

Below is an example table which WIRServer maintains.

For example:

Data usage for OTA app for period

OTA app using cellular: 10 MB

OTA app using WIFI: 500 MB

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | WIRClient2 APN1 | | WIRClient2 Wi-Fi | | WIRClient1 Wi-Fi | | WIRClient5 APN1 | |
| Application | Sent | Recv | Sent | Recv | Sent | Recv | Sent | Recv |
| WIRServer OTA |  |  | 2MB | 40MB | 10MB | 100MB | 1MB | 30MB |
| WIRServer FCI | 2.5MB | 2.5MB |  |  |  |  |  |  |
| WIRClient1 Voice Rec | 10MB | 10MB |  |  |  |  |  |  |
| WIRClient1 VICC | 50MB | 100MB |  |  | 10MB | 20MB |  |  |
| On Line Traffic |  |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |  |

For SiriusXM, WIRServer shall report data usage.

Data usage shall be accumulated until key off and door opened. At key off and door opened, WIRServer shall report data usage to Ford backend if there is an SDN connection.

If there is pending data usage report during WIRServer power off, WIRServer shall store it on persistent memory and report it during next report trigger.

#### WIR-REQ-295972/B-Data usage storage

At any point in time WIRServer shall aggregate data usage per app per interface and store the same for the period. WIRClients shall report data usage to WIRServer during key off and door open event. If a WIRClient fails to report to WIRServer the data usage, it shall report at next key off and door open event.

#### WIR-REQ-295975/A-Data usage data transmission to Ford cloud

The WIRServer shall be capable of transmitting the Data usage any time to the cloud upon request from the cloud FTCP.

For more details on the type and content please refer FTCP protofile.

#### WIR-REQ-295976/A-Data usage data transmission format

The WIRServer shall be capable of transmitting data usage for individual application and also as a whole.

#### WIR-REQ-295977/A-Data usage request processing from in vehicle HMI

The WIRServer shall be capable of receiving request to data usage from in vehicle HMI.

#### WIR-REQ-295978/A-Data usage data transmission to in vehicle HMI

The WIRServer shall be capable of transmitting the Data usage any time to the cloud upon request from the in vehicle HMI.

#### WIR-REQ-295979/A-Accuracy of data usage calculation

The Data usage calculation per application shall be 99.9999% accuracy rounded to the nearest Kilobyte.

#### WIR-REQ-295980/A-Data usage timestamp for cellular

The WIRServer shall have a log of data usage per session and associated time stamps for the selected period.

#### WIR-REQ-402384/B-Controlling data usage data upload by CCS setting

If the bAllow for entity Feature94 (ID 94, Type 1) is disabled, the WIRServer shall not share any data consumption information outside the vehicle.

If the bAllow for entity Feature94 (ID 94, Type 1) is enabled, the WIRServer shall be allowed to share data consumption related information outside the vehicle.

Note: the impact of other privacy settings (e.g. vehicle data sharing) on the enablement of the feature is handled through CCS dependency on Policy Table Extension.

WIRServer shall stop collecting data usage when the bAllow for entity Feature94 (ID 94, Type 1) is disabled and resume collecting data usage again when the bAllow for entity Feature94 (ID 94, Type 1) is enabled.

### Use Cases

### White Box View

## WIR-FUN-REQ-295982/A-WIFI Connect Reminders - HMI

### Requirements

#### WIR-REQ-295983/A-WIFI Connect Reminder Popup

SYNC module shall support the WIFI connect reminder pop up.

Text content and pop up shall be finalized and captured in HMI specification.

#### WIR-REQ-295984/C-Trigger

The WIR Server module shall count the number of ignition cycles where Vehicle is not connected to WIFI.

Vehicle WIFI refers to either WIRClient1 WIFI or WIRClient2 WIFI or WIRClient5 WIFI.

If the vehicle fails to connect to WIFI for WIFI\_CONNECT\_REMINDER\_COUNTER number of ignition cycles WIRServer shall notify WIRClient1 to display WIFI Connect reminder pop up.

#### WIR-REQ-295985/A-Trigger Frequency

The trigger frequency is WIFI\_CONNECT\_REMINDER\_COUNTER ignition cycles. The initial value for WIFI\_CONNECT\_REMINDER\_COUNTER is 100.

#### WIR-REQ-295986/B-WIFI Connect Reminder Popup User Action

The user shall have an option to select “No” or “Remind me later” to the WIFI connect reminder pop up.

#### WIR-REQ-295987/B-User Selects OK

If the user selects OK then vehicle shall not display the pop up once again.

WIRClient1 shall notify WIRServer about user selection. WIRServer shall reset the WIFI\_CONNECT\_REMINDER\_COUNTER counter and wait for trigger conditions as stated in Trigger requirement.

#### WIR-REQ-295988/B-User Selects Remind Me Later

If the user selects “REMIND ME LATER” then vehicle shall not display the pop up once again until WIFI\_CONNECT\_REMINDER\_TEMP ignition cycles have been completed.

WIRClient1 shall notify WIRServer module about User selection. WIRServer module shall start another counter WIFI\_CONNECT\_REMINDER\_TEMP and WIFI\_CONNECT\_REMINDER\_COUNTER shall still remain 100. The value of WIFI\_CONNECT\_REMINDER\_TEMP counter shall configurable and shall be set to 100.

#### WIR-REQ-295989/A-WIFI\_CONNECT\_REMINDER Popup Configurable

The WIFI\_CONNECT\_REMINDER pop up shall be a configurable parameter and shall be turned ON / OFF by WIRServer module based on the configuration.

#### WIR-REQ-295990/B-User Selects “No” Check Box

If the user selects “No” again then the pop up is never displayed again until master reset occurs.

WIRClient1 communicates this information to WIRServer which tracks this status and never triggers the pop up.

#### WIR-REQ-295991/B-Master Reset

If a master reset is performed by user then WIRClient1 notifies WIRServer of master reset.

WIRServer will reset the Pop up trigger conditions and over rides any previously selected “No.”

Master reset will reset this and pop up will be brought out again.

### Use Cases

### White Box View

## WIR-FUN-REQ-295996/A-Central Controller

### Requirements

#### WIR-REQ-296012/E-Receive requests from local controller

WIRServer central controller shall receive requests from any WIRClient and WIRServer local controllers.

#### WIR-REQ-296013/C-Process requests from Local controller:

The WIR Server shall process the request specified by the application and provide the appropriate interface

|  |  |
| --- | --- |
| **Type of request** | **Interface to be provided** |
| Foreground | Cellular connection if available  WIFI only if cellular connection is not available  If WIFI preferred is selected by the application then WIFI shall be provided even if cellular connection is available |
| Background | WIFI connection |
| Background guaranteed | Till Timer expiry provide WIFI connection. Once timer expires provide cellular connection |
| Offpeak | Cellular connection |

#### WIR-REQ-296014/A-Store requests from Local controller along with interface provided

The WIRServer shall be capable of storing the intent requests and interfaces provided for diagnostics purpose as described in diagnostics section

#### WIR-REQ-296015/D-Ability to control already provided interface

The WIRServer central controller shall be able to control all applications for which it has provided interfaces via Local controllers (WIRServer and WIRClients) for bandwidth shaping / prioritization using

1. Pause – Applications shall remember the interface but shall not use the interface until it receives resume from central controller
2. Resume – Applications shall start using the interfaces once again
3. Stop – Applications shall stop using the interface and shall request for new interface if needed

#### WIR-REQ-296016/D-Scheduling application requests

The central controller shall schedule application requests until an interface is provided.

Central controller shall queue WIRServer and WIRClient application requests.

#### WIR-REQ-296017/A-Providing interfaces to application requests

The Central controller shall provide interfaces to requesting applications via local controller

#### WIR-REQ-370384/A-Z2 Modem Limitation

The Central controller shall provide interfaces to applications based on policy table. For Z2 applications shall only be privileged to use one of the two modems. The central controller shall not route applications to the modem interfaces an application is not privileged to use.

#### WIR-REQ-296018/B-Removing application requests from scheduling queue

The Central controller shall remove application requests from scheduling queue in the following scenarios

|  |  |
| --- | --- |
| **Type of Intent** | **Removing from scheduling queue** |
| Foreground | 1. Interface provided successfully, or 2. Ignition OFF |
| Background | 1. Interface provided successfully, or 2. Ignition OFF |
| Background guaranteed | 1. Interface provided successfully |
| Off-peak | 1. Interface provided successfully |

#### WIR-REQ-296019/A-Retaining application requests in scheduling queue in Nonvolatile memory

The Central controller shall retain background guaranteed and off-peak requests in Non volatile memory

#### WIR-REQ-296020/A-Removing application requests from scheduling queue in Nonvolatile memory

The Central controller shall remove background guaranteed and off-peak requests from Non volatile memory only after a successful interface is provided back to the application

#### WIR-REQ-296021/A-Scheduling and priority

Before providing an interface to an application the central controller shall take into account the priority of the intent and appropriate traffic classification

For more details please refer Intent and traffic classification section of this document

At all times Central controller shall ensure application requests priority and appropriate traffic classifications are met

#### WIR-REQ-296022/A-Central controller and Off-peak

For off-peak intent based tasks and background guaranteed tasks whose timer has expired the central controller shall coordinate with off-peak applications and handover the tasks to queue

#### WIR-REQ-388353/A-Disabling cellular or WiFi for non-connected vehicle

Non-connected vehicle doesn’t support internet connectivity through cellular or WiFi.

WIRServer shall reject cellular connection request if DID DE00, CONNECTED\_MARKET field is FALSE.

WIRServer shall reject WiFi connection request if the vehicle does not support wifi, i.e. bPAllow of CCS entity EmbeddedWifiConnectivity (ID 23, Type 0) is disabled.

Both CONNECTED\_MARKET and bPAllow of CCS entity EmbeddedWifiConnectivity shall be set to correct value at factory based on connectivity option provided by vehicle target.

#### WIR-REQ-402361/A-Support for China (DuerOS)

WIRServer shall provide the following to DuerOS target:

* An API to read the entire policy table, via ReadPolicyTable
* A NetId setting for the Foreground and Background intents. NetId shall be passed to VNM on DuerOS to configure NetId
* Network interface allocation shall return DNS addresses and Interface IP address

### Use Cases

#### WIR-UC-REQ-296096/D-Central controller provides connection interface to local controller

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClients |
| **Pre-conditions** | 1. Application requests WIR local controller for connection interface |
| **Scenario Description** | Local controller works with central controller to provide interfaces to requesting application |
| **Post-conditions** | Local controller successfully sends the request to the central controller  Central controller processes the request and provides interface back to the local controller |
| **Interfaces** |  |
| **Notes** | Exception use cases  Interface not available |

#### WIR-UC-REQ-296097/D-Interface not available

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClients |
| **Pre-conditions** | 1. Application requests WIR local controller for connection interface |
| **Scenario Description** | Local controller requests central controller to provide interfaces to requesting application. But Interface is not available |
| **Post-conditions** | Central controller notifies the local controller  Once interface becomes available the central controller provides the interface to the local controller which cascades the same to the application |
| **Interfaces** |  |
| **Notes** |  |

### White Box View

## WIR-FUN-REQ-295999/A-Off-Peak Handler

### Requirements

#### WIR-REQ-296024/A-Off-peak and central controller:

WIRServer Off-peak shall receive off-peak requests to queue from central controller

#### WIR-REQ-296025/B-Off-peak determination of off-peak wake up qualification

WIR off-peak application shall determine whether the vehicle can be woken up for off-peak activities by means of checking

1. Whether there are off-peak activity pending interface (Activities selected offpeak and activities selected background guaranteed with offpeak flag enabled and Timer expired)
2. Ignition ON
3. Battery SOC > WIR\_OFF-PEAK\_BATTERY\_SOC

#### WIR-REQ-296026/A-Off-peak request to cloud

Once Off-peak application determines Off-peak qualification off-peak application shall send a FTCP message to cloud requesting wake up.

For more information on the FTCP message details please refer FTCP protofile

#### WIR-REQ-310878/B-Requests during offpeak

During off-peak mode, WIR module shall support all intents from application. There is no separate restriction on intent support during off-peak mode.

#### WIR-REQ-296027/B-Cloud response to off-peak request from vehicle

The cloud shall acknowledge the vehicle request and shall send a confirmation for off-peak wake up.

Only when the vehicle receives the acknowledgement off-peak application shall determine the request was successful.

For more information on the FTCP message details please refer FTCP protofile.

If the off-peak application fails to receive the acknowledgement then off-peak application shall attempt to retry to send the wake up request to the cloud.

#### WIR-REQ-296028/B-Repeated off-peak wake up

Once off-peak application receives successful acknowledgement from cloud regarding off-peak wake up request, then the off-peak application shall not send repeated requests to cloud until off-peak wake up happens or after 5 ignition cycles during which no off-peak wake up request has occurred from cloud.

#### WIR-REQ-296029/C-Off-peak wake up

WIRServer shall receive off-peak wake up FTCP command from Cloud.

If off-peak wake up is successful, off-peak application shall also send an FTCP command response to the Ford cloud indicating that off-peak wake up is successful.

For more information on the FTCP message details please refer FTCP protofile.

#### WIR-REQ-296030/C-Off-peak wake up restrictions

WIRServer shall receive off-peak wake up FTCP command from Cloud. Upon receiving wake up WIRServer shall check if Ignition is ON or KOL = Hibernate or critical battery or Factory or Transport and WIRClient2 and WIRServer shall go to sleep immediately if KOL = Hibernate or critical battery.

Also, WIR off-peak application shall request for off-peak once again when KOL = Normal since Cloud will remove existing off-peak requests upon scheduling.

WIRServer shall send a failure message to cloud indicating vehicle Ignition is ON or vehicle is in Hibernate or Critical battery state so off-peak is not possible as part of off-peak wake up command response.

For more details please refer the FTCP protofile.

#### WIR-REQ-385873/A-VPSM subscription and callback handling during off-peak service

WIRServer shall request a target resource (WIRClient2 cellular) with a severity level which is defined in DID OFFPEAK\_VPSM\_SEVERITY\_LEVEL. OFFPEAK\_VPSM\_SEVERITY\_LEVEL shall be programmed as HIGH at factory. WIRServer shall release all WIR interfaces and prepare for power down if it is notified by VPSM that it’s power request is no longer valid.

#### WIR-REQ-296031/C-OFF-Peak activities

Upon Off-peak wake up WIRServer and WIRClient2 off-peak application shall prioritize tasks which needs off-peak interface and provide interfaces to application via Central controller and local controllers as soon as possible.

#### WIR-REQ-296032/C-Off-peak termination

Off-peak mode shall be terminated in any of the following scenarios:

1. If all the applications using off-peak interface confirms it no longer needs the interface
2. Off-peak allowed time specified in off-peak wake up FTCP command expired
3. Vehicle ignition is turned ON

WIR module shall notify the applications that off-peak mode will be terminated before terminating the off-peak connection.

WIR module shall also notify the cloud that off-peak mode is ended.

If in case off-peak mode is terminated because vehicle ignition is ON then WIR shall queue the off-peak requests and send another wake up request to cloud after terminating the off-peak event.

#### WIR-REQ-310879/B-Off-peak timer

Off-peak application shall maintain a timer to make sure off-peak mode time does not exceed the off-peak allowed time in the FTCP command. The timer value shall match the time specified in off-peak wake up FTCP command.

#### WIR-REQ-296033/B-Off-peak termination notification to applications

Off-peak application shall send a notification to all off-peak applications before terminating off-peak via WIR Central controller and local controllers.

#### WIR-REQ-310880/A-Off-peak completion notification to cloud

Off-peak application shall send a notification to Ford cloud via a FTCP message indicating off-peak completion. For more information on the FTCP message details please refer FTCP protofile.

#### WIR-REQ-296034/B-Off-peak and Ignition

When Ignition ON event happens during off-peak off-peak shall be terminated by off-peak application.

The offpeak application shall send an error message to the cloud indicating Ignition ON event happened and as a result off-peak is not possible.

#### WIR-REQ-296035/A-Off-peak power management

WIR off-peak application shall make sure WIRServer and WIRClient2 are up and running during off-peak time. WIR off-peak application shall communicate this information to the power management module.

WIR off-peak application shall also make sure once off-peak is terminated it shall make sure to let the power mode module of WIRServer and WIRClient2 that off-peak activity is completed and it can go to sleep.

#### WIR-REQ-296036/A-WIR Off-peak

WIR off-peak application shall know whether off-peak is supported or not by the vehicle using the configurable parameter WIR\_OFFPEAK\_SUPPORT.

WIR off-peak application shall send requests to cloud only if WIR\_OFFPEAK\_SUPPORT is enabled.

WIR Client controllers shall reject application requests as unsupported if WIR\_OFFPEAK\_SUPPORT is disabled.

#### WIR-REQ-321008/A-Off-peak error codes

WIR application in case of error shall select the following applicable error code and send to the cloud via FTCP application.

|  |  |
| --- | --- |
| **Off peak Error codes** | **Error message/ Description** |
| IGNITION\_ON | Ignition ON event happens and as a result off-peak is not possible |
| CRITICAL\_BATTERY | Vehicle is in Hibernate or Critical battery state so off-peak is not possible |
| OTHER\_ERRORS | All other Errors - will be defined later |

The error codes shall be sent via off-peak wake up command response or off-peak end alert depending on the state where the error has occurred.

For example if an error has occurred before the off-peak application sends off-peak command response then error shall be included as part of command response.

If error occurs after the command response then off-peak application shall include as part of off-peak finished alert.

### Use Cases

#### WIR-UC-REQ-296098/B-Off peak wake up

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClient2 |
| **Pre-conditions** | 1. WIR off-peak application has sent a request to cloud for off-peak wake up |
| **Scenario Description** | Cloud sends a wake up SMS if required, and off-peak wake up FTCP command to vehicle for off-peak wake up |
| **Post-conditions** | WIRServer and WIRClient2 successfully wakes up  WIR successfully provides off-peak interface to requesting applications  Once off-peak timer has expired or off-peak applications has completed tasks WIRServer and WIRClient go to sleep |
| **Interfaces** |  |
| **Notes** | Exception use cases  Cloud fails to wake up the vehicle  Off-peak timer fails to work  WIRClient2 fails to respond to battery hibernate or critical battery |

#### WIR-UC-REQ-296099/A-Cloud fails to wake up the vehicle

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClient2 |
| **Pre-conditions** | 1. WIR off-peak application has sent a request to cloud for off-peak wake up |
| **Scenario Description** | WIR off-peak requests cloud to wake up vehicle during off-peak hours and cloud fails to wake up |
| **Post-conditions** | WIRClient2 and WIRServer fail to wake up  Off-peak activities cannot be completed |
| **Interfaces** |  |
| **Notes** |  |

#### WIR-UC-REQ-296100/A-Off-peak timer fails to work

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClient2 |
| **Pre-conditions** | 1. WIR off-peak application has sent a request to cloud for off-peak wake up |
| **Scenario Description** | WIR off-peak requests cloud to wake up vehicle during off-peak hours and cloud wakes up vehicle |
| **Post-conditions** | WIRServer and WIRClient2 successfully wakes up  WIR successfully provides off-peak interface to requesting applications  Off-peak timer fails to expire and vehicle fails to go to sleep within off-peak maximum allowed time  Vehicle battery is affected and leads to customer dissatisfaction |
| **Interfaces** |  |
| **Notes** |  |

#### WIR-UC-REQ-296101/A-WIRClient2 fails to respond to battery hibernate or critical battery

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClient2 |
| **Pre-conditions** | 1. WIR off-peak application has sent a request to cloud for off-peak wake up |
| **Scenario Description** | WIR off-peak requests cloud to wake up vehicle during off-peak hours and cloud wakes up vehicle |
| **Post-conditions** | WIRClient2 fails to respond to battery hibernate and wakes up WIRServer  Vehicle battery is affected and leads to customer dissatisfaction |
| **Interfaces** |  |
| **Notes** |  |

### White Box View

## WIR-FUN-REQ-295997/A-Wifi Management

### Requirements

#### WIR-REQ-296038/B-WIRClient1 WIFI status

WIRServer shall aggregate WIFI Chipset availability, WIFI connection and scan status from WIRClient1.

#### WIR-REQ-296039/B-WIRClient2 WIFI status

WIRServer shall aggregate WIFI Chipset availability, WIFI connection and scan status from WIRClient2.

#### WIR-REQ-370385/C-WIRClient5 WIFI status

WIRServer shall aggregate WIFI Chipset availability, WIFI connection and scan status from WIRClient5.

#### WIR-REQ-311551/A-WI-FI ON/OFF settings

WIRServer shall not request WLAN services for any WIFI operation if WI-FI is OFF.

#### WIR-REQ-311552/A-WI-FI Station mode ON/OFF settings

WIRServer shall not request WLAN services for any WIFI operation if WI-FI Station mode is OFF.

#### WIR-REQ-311553/D-User initiated scan

If WIRClient1 WIFI and WIRClient2 WIFI are available and user triggers scan WIRServer shall choose the WIFI chipset and access point with the strongest signal strength and provide the scan results to the user in order to display to the HMI. User initiated HMI scans are controlled by driver restrictions on WIRClient1, no additional restrictions to be imposed by WIR.

#### WIR-REQ-311554/C-User initiated connect to an access point

If user selects an access point and connects WIRServer shall connect to the access point by passing the request to WLAN Services via WIRClient1 or WIRClient2 or WIRClient5 WIFI whichever is active and available.

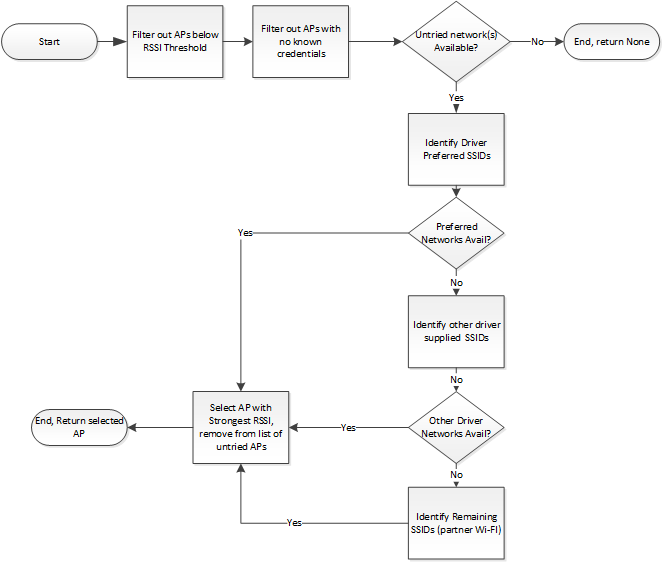
WIRServer shall store this information in non-volatile memory and make sure WIRClient1, WIRClient2 and WIRClient5 also stores this information for Key off operation.

#### WIR-REQ-311555/C-User initiated disconnect from an access point

If a user disconnects from an access point WIRServer shall disconnect from the access point by passing the request to WLAN Services via WIRClient1 or WIRClient2 or WIRClient5 WIFI whichever is active and connected to the access point. After disconnection, WIRServer shall not allow the AP during next scan and connection until a user manually connects to the AP. WIRServer shall not delete password from internal database but use it for next manual connection.

#### WIR-REQ-311556/B-Selecting An Access Point

The WIRServer and WIRClient1 / WIRClient2 / WIRClient5 in case of WIRServer not active follows the below UML Activity diagram when selecting an AP to connect to, from the list provided by the WLAN services scan.



#### WIR-REQ-296040/E-Arbitration between WIRClients WIFI

If WIRClient1, WIRClient2, and WIRClient5 WIFI are available and application request is generic background / background guaranteed / Foreground with WIFI preferred / Foreground but no active cellular network then WIRServer shall choose the WIFI chipset and access point with the strongest signal strength and provide the interface to the requesting applications.

#### WIR-REQ-296042/C-WIFI disconnect

WIRServer shall make sure to connect only one of WIRClient1 or WIRClient2 or WIRClient5 to WIFI access point and disconnect other WIRClient WIFI. This decision shall be based on the intents and priority provided by the applications.

#### WIR-REQ-296043/C-Connecting multiple WIRClients WIFI to same access point

WIRServer shall ensure at no point any combination of WIRClient1 WIFI, WIRClient2 WIFI, and WIRClient5 WIFI are connected to same access point at the same time. The connected WIFI access point shall be displayed when user navigates to the WIFI menu.

#### WIR-REQ-296044/B-WIFI restrictions

WIR central controller shall follow WIFI restrictions as specified in the Intent section before handing out interfaces to applications.

#### WIR-REQ-311557/B-Handling failures

WIRServer shall receive failure connection information from WLAN services and notify the user as appropriate.

#### WIR-REQ-311558/C-WIFI hotspot SSID in HMI display

WIRServer shall receive the WIRClient2’s WIFI hotspot SSID from WLAN services and store the same.

* If DISPLAY\_WIFIHOTSPOT\_CREDENTIALS is disabled then WIRServer shall make sure the WIFI Hotspot SSID is **not displayed** to the end user as part of scan results on WIRClient1 in order to prevent the user from connecting to WIRClient2’s hotspot
* If DISPLAY\_WIFIHOTSPOT\_CREDENTIALS is enabled then WIRServer shall allow the WIFI Hotspot SSID to be displayed to the end user on WIRClient1 to allow the user to connect to WIRClient2’s hotspot.

#### WIR-REQ-311559/A-WIFI Network availability notiification

If Wi-Fi Network Availability Notification feature is ON and the vehicle goes into park mode (for automatic transmission), parking brake is active (for manual transmission), extended play mode, accessory mode or delayed accessory mode, the WIR module shall display the network availability notification icon if

* There is a Wi-Fi network within range
* The system is not already connected to a Wi-Fi network

#### WIR-REQ-311560/A-WIFI network availability notification HMI

If the Wi-Fi network(s) availability notification is ON and all the conditions to trigger the notification are met, the HMI SHALL display a pressable icon (button) in a general location on the screen. Once this button is pressed, the system SHALL display the Wi-Fi menu.

#### WIR-REQ-311561/A-WIFI network availability notification default setting

The Default setting for the Wi-Fi network notification feature SHALL be ON as long as the Wi-Fi feature is ON. Turning the Wi-Fi feature OFF will also turn the Wi-Fi network availability notification feature OFF. WIR module shall ensure the functionality.

#### WIR-REQ-370000/D-WIFI scan when gear moved park and engine is not running

WIRServer shall start scanning available AP when gear is moved to the park position and engine is not running.

WIRServer shall scan for 2.5 minutes after ignition off. If WIRServer didn’t find any available AP, WIRServer shall stop scanning, shall not scan again until ignition start and WLAN shall be in low power mode to save battery.

If first entry of WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL is 0, WIR shall not trigger scan.

If WIRServer finds an available AP and successfully connected to the AP, WIRServer shall maintain the connection for 1 minute even though no application is requesting the WiFi connection. After 1 minute, if there is no pending WiFi connection request from any applications, WIRServer shall disconnect from the AP but remember the AP for later connection.

If any application requesting WiFi interface while WiFi connection is disconnected, WIRServer shall try the AP previously connected. If the AP is not available, WIRServer shall try any other available AP to provide WiFi connection to requesting application.

When last application released WIR WiFi interface, WIRServer shall maintain the WiFi connection for 1 minute and disconnect it if there is no application requesting the connection.

#### WIR-REQ-370001/B-WIFI scan when driver restriction is disabled and engine is running

WIRServer shall start scanning available AP when driver restriction is disabled which is controlled by vehicle speed.

WIRServer shall continue scanning available AP with 7 times according to WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL seconds interval and continue with last entry of WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL seconds.

If first entry of WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL is 0, WIR shall not trigger scan.

WIRServer shall not connect to the found AP unless there is pending WiFi connection request.

During WiFi scan if driver restriction enabled again by vehicle speed faster than threshold, WIRServer shall stop scanning. WIRServer shall start new scan sequence when driver restriction disabled again.

#### WIR-REQ-370002/B-Excluding WIRClient1 projection mode AP from scan result

WIRServer shall detect WIRClient1 WiFi is acting as projection mode AP and exclude it from WiFi scan result. WLAN shall return an error scan result.

#### WIR-REQ-470986/A-Prevent customer to connect to projection mode AP manually

WIRServer shall detect WIRClient1 WiFi is acting as projection mode AP and prevent customer to connect to the AP.

#### WIR-REQ-370266/A-Scan and connecting to an AP while WIRServer or WIRClient2 is not provisioned

WIRServer shall scan and connect an AP regardless of gear position when LifeCycleMode\_D\_Actl = FACTORY.

#### WIR-REQ-370267/D-Factory WiFi network support

WIRServer shall support factory WiFi while LifeCycMde\_D\_Actl = FACTORY mode. Once LifeCycMde\_D\_Actl is changed from FACTORY to TRANSPORT or NORMAL to TRANSPORT, WIRServer shall stop supporting factory WiFi. Master reset shall not re-enable factory WiFi support again.

After OTA update, any vehicle shall disable factory WiFi

1. if LifeCycleMode is in TRANSPORT

OR

2. if factory WiFi is enabled and LifeCycMde\_D\_Actl is not FACTORY mode and

2.a if all ECUs are provisioned or not present state or not configured state

DID D021 == 0x26 (WIRServer provisioned) of AND

DID FD02 has following values

TCU\_Provisioning\_State == 0x32 (Provisioned Mode)

SYNC\_Provisioning\_State == 0x38 (Not Present or Not Configured) OR 0x3B (Present & Provisioned)

DSRC\_Provisioning\_State == 0x38 (Not Present or Not Configured) OR 0x3B (Present & Provisioned)

OBCC\_Provisioning\_State == 0x38 (Not Present or Not Configured) OR 0x3B (Present & Provisioned)

BLEM\_Provisioning\_State == 0x50 (Bluetooth Low Energy Module (BLEM) Not Present) OR 0x54 (Bluetooth Low Energy Module (BLEM) Present & Key Delivered).

WIRServer shall read factory WiFi SSID and password from secured file and shall not save on local NV memory. WIRServer shall treat factory WiFi as a normal WiFi network and provide the network interface to all application requesting WiFi interface if policy table “Factory WiFi” for the application is enabled.

When factory WiFi and other WiFi are available at the same time, WIRServer shall select factory WiFi even factory WiFi signal is not best one.

User initiated HMI scans are controlled by driver restrictions on WIRClient1, no additional restrictions to be imposed by WIR while factory WiFi supported is enabled.

At factory, some vehicles can go through sample test and the tester may connect to other WiFi through WIRClient1 UI. WIRServer shall treat it as normal case and save SSID and password as other WiFi network.

#### WIRv2-REQ-470987/A-Factory WiFi network support

Factory WiFi data (FWD) shall be saved in secure enclave during WIRServer image build. FWD includes SSID, password, auth type and SEED value for encryption. Key Manager shall provide APIs to read and delete FWD. WIRServer shall use the key manager API to read or delete FWD.

WIRServer, WIRClient1 and WIRClient2 shall have internal flag GATE\_CHECK\_DONE.

Internal flag GATE\_CHECK\_DONE shall have 0 (default) as initial value.

At gate check,

* WIRServer shall detect LifeCycMde\_D\_Actl is changed to TRANSPORT mode and
* WIRServer has production build and doesn’t have debug token installed, then
* Delete FWD data and
* Set internal flag GATE\_CHECK\_DONE to 1 for all 3 ECUs

WIRServer shall enable factory WiFi if

* FWD data is available and
* Internal flag GATE\_CHECK\_DONE has 0 (default value) on all 3 ECUs and
* One or more ECUs are not provisioned

Otherwise WIRServer shall

* Disable factory WiFi and disconnect from it if already connected and
* Delete FWD data if it is available and
* Set internal flag GATE\_CHECK\_DONE to 1 on all 3 ECUs

WIRServer shall disable factory WiFi if factory WiFi is enabled and LifeCycMde\_D\_Actl is not FACTORY mode and if all ECUs are provisioned or not present state or not configured state

DID D021 == 0x26 (WIRServer provisioned) of AND

DID FD02 has following values

TCU\_Provisioning\_State == 0x32 (Provisioned Mode)

SYNC\_Provisioning\_State == 0x38 (Not Present or Not Configured) OR 0x3B (Present & Provisioned)

DSRC\_Provisioning\_State == 0x38 (Not Present or Not Configured) OR 0x3B (Present & Provisioned)

OBCC\_Provisioning\_State == 0x38 (Not Present or Not Configured) OR 0x3B (Present & Provisioned)

BLEM\_Provisioning\_State == 0x50 (Bluetooth Low Energy Module (BLEM) Not Present) OR 0x54 (Bluetooth Low Energy Module (BLEM) Present & Key Delivered) AND

all 3 ECU’s GATE\_CHECK\_DONE is set to 0 AND

LifeCycMde\_D\_Actl doesn’t change to TRANSPORT\_MODE within 10 ignition cycles.

WIRServer shall not save FWD on local NVM.

Master reset shall not recover FWD.

Factory WiFi is allowed to any application with “Factory WiFi” field is set on policy table.

When factory WiFi and other WiFi are available at the same time, WIRServer shall select factory WiFi even factory WiFi signal is not best one.

User initiated HMI scans are controlled by driver restrictions on WIRClient1, no additional restrictions to be imposed by WIR while factory WiFi support is enabled.

At factory, some vehicles can go through sample test and the tester may connect to other WiFi through WIRClient1 UI. WIRServer shall treat it as normal case and save SSID and password as other WiFi network

#### WIR-REQ-392769/A-Network connection password failure

WIRServer shall support WFCF-FUR-REQ-025317 of WiFi Configuration Settings Server v2 SPSS.

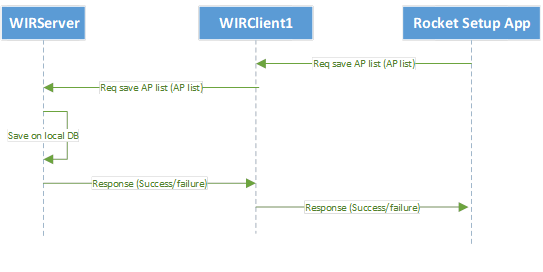
#### WIR-REQ-392787/A-Hidden network display after disconnect

WIRServer shall support [H31a.R416.01].

#### WIR-REQ-402796/A-RocketSetup Support - API to Save WiFi AP list

WIRServer and WIRClient1 shall provide an API to save WiFi AP list from Rocket Setup app which is running on WIRClient1. AP list shall have SSID, password and security type. AP list can have one or more APs on the request. In order to save the AP list on WIRServer local database, Rocket Setup shall call WIRClient1 API and WIRClient1 shall return success or failure result. WIRServer shall mark all AP as provided by Rocket Setup. All AP from Rocket Setup shall be used as other AP on local database.

Following diagram shows call flow:



#### WIR-REQ-402797/A-API to control WIRClient1 WLAN STA mode

WIRServer shall provide 2 APIs to control WIRClient1 WLAN station mode:

* disableSyncWifiStaMode
* enableSyncWifiStaMode

#### WIR-REQ-470988/A-Personal & Portable Profile Support - API to Save WIFI Setup

WIRServer shall provide an API to save the Wifi Setup global settings from Personal & Portable Profile app running in ECG.

In order to save the Wifi Setup global settings on WIRServer local database, PPP app shall call addWlanProfiles from WIRServer API, WlanProfiles shall have SSID, sec and security type.

WIRServer shall return success or failure result. WIRServer shall mark all AP as provided by Rocket Setup.

Following diagram shows call flow:



#### WIR-REQ-470989/A-Supporting same SSID and different security type AP

WIRServer shall support saving same SSID and different security AP and connecting to those Aps.

### Use Cases

#### WIR-UC-REQ-296102/B-Connect to WIFI Access point

|  |  |
| --- | --- |
| **Actors** | WIRClient1, WIRServer, WIRClient2, WIRClient5 |
| **Pre-conditions** | 1. Gear position transitions to Park in automatic gear box system and ignition is ON or 2. Gear position transitions to Neutral in a manual gear box system and ignition is ON or 3. Ignition is Accessory or 4. Delayed accessory is ON and Ignition is OFF or 5. Ignition is OFF and delayed accessory is OFF provided ECU is up and running |
| **Scenario Description** | WIR requests WIFI interface to connect to access point |
| **Post-conditions** | WLAN services successfully connects to access point  WIR provides WIFI interface to requesting application |
| **Interfaces** |  |
| **Notes** | Exception use cases  WIRClient1 WI-FI is not available  WIRClient2 WI-FI is not available  WIRClient5 WI-FI is not available  WIRClient1, WIRClient2 and WIRClient5 access points are not available  Known Access point not available |

#### WIR-UC-REQ-296103/B-WIRClient1 WI-FI is not available

|  |  |
| --- | --- |
| **Actors** | WIRClient1, WIRServer, WIRClient2, WIRClient5 |
| **Pre-conditions** | 1. Gear position transitions to Park in automatic gear box system and ignition is ON or 2. Gear position transitions to Neutral in a manual gear box system and ignition is ON or 3. Ignition is Accessory or 4. Delayed accessory is ON and Ignition is OFF or 5. Ignition is OFF and delayed accessory is OFF provided ECU is up and running |
| **Scenario Description** | WIR requests WIFI interface to connect to access point and WIRClient1 WIFI is not available |
| **Post-conditions** | WLAN services successfully connects to WIRClient2 or WIRClient5 access point  WIR provides WIFI interface to requesting application |
| **Interfaces** |  |
| **Notes** |  |

#### WIR-UC-REQ-296104/B-WIRClient2 WI-FI is not available

|  |  |
| --- | --- |
| **Actors** | WIRClient1, WIRServer, WIRClient2, WIRClient5 |
| **Pre-conditions** | 1. Gear position transitions to Park in automatic gear box system and ignition is ON or 2. Gear position transitions to Neutral in a manual gear box system and ignition is ON or 3. Ignition is Accessory or 4. Delayed accessory is ON and Ignition is OFF or 5. Ignition is OFF and delayed accessory is OFF provided ECU is up and running |
| **Scenario Description** | WIR requests WIFI interface to connect to access point and WIRClient2 WIFI is not available |
| **Post-conditions** | WLAN services successfully connects to WIRClient1 or WIRClient5 access point  WIR provides WIFI interface to requesting application |
| **Interfaces** |  |
| **Notes** |  |

#### WIR-UC-REQ-370386/A-WIRClient5 WI-FI is not available

|  |  |
| --- | --- |
| **Actors** | WIRClient1, WIRServer, WIRClient2, WIRClient5 |
| **Pre-conditions** | 1. Gear position transitions to Park in automatic gear box system and ignition is ON or 2. Gear position transitions to Neutral in a manual gear box system and ignition is ON or 3. Ignition is Accessory or 4. Delayed accessory is ON and Ignition is OFF or 5. Ignition is OFF and delayed accessory is OFF provided ECU is up and running |
| **Scenario Description** | WIR requests WIFI interface to connect to access point and WIRClient2 WIFI is not available |
| **Post-conditions** | WLAN services successfully connects to WIRClient1 or WIRClient2 access point  WIR provides WIFI interface to requesting application |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-296105/B-WIRClient1 and WIRClient2 WI-FI are not available

|  |  |
| --- | --- |
| **Actors** | WIRClient1, WIRServer, WIRClient2, WIRClient5 |
| **Pre-conditions** | 1. Gear position transitions to Park in automatic gear box system and ignition is ON or 2. Gear position transitions to Neutral in a manual gear box system and ignition is ON or 3. Ignition is Accessory or 4. Delayed accessory is ON and Ignition is OFF or 5. Ignition is OFF and delayed accessory is OFF provided ECU is up and running |
| **Scenario Description** | WIR requests WIFI interface to connect to access point and WIRClient1 and WIRClient2 WIFI is not available |
| **Post-conditions** | WLAN services successfully connects to WIRClient5 access point  WIR provides WIFI interface to requesting application |
| **Interfaces** |  |
| **Notes** |  |

#### WIR-UC-REQ-370387/A-WIRClient1 and WIRClient5 WI-FI are not available

|  |  |
| --- | --- |
| **Actors** | WIRClient1, WIRServer, WIRClient2, WIRClient5 |
| **Pre-conditions** | 1. Gear position transitions to Park in automatic gear box system and ignition is ON or 2. Gear position transitions to Neutral in a manual gear box system and ignition is ON or 3. Ignition is Accessory or 4. Delayed accessory is ON and Ignition is OFF or 5. Ignition is OFF and delayed accessory is OFF provided ECU is up and running |
| **Scenario Description** | WIR requests WIFI interface to connect to access point and WIRClient1 and WIRClient5 WIFI is not available |
| **Post-conditions** | WLAN services successfully connects to WIRClient2 access point  WIR provides WIFI interface to requesting application |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-370388/A-WIRClient2 and WIRClient5 WI-FI are not available

|  |  |
| --- | --- |
| **Actors** | WIRClient1, WIRServer, WIRClient2, WIRClient5 |
| **Pre-conditions** | 1. Gear position transitions to Park in automatic gear box system and ignition is ON or 2. Gear position transitions to Neutral in a manual gear box system and ignition is ON or 3. Ignition is Accessory or 4. Delayed accessory is ON and Ignition is OFF or 5. Ignition is OFF and delayed accessory is OFF provided ECU is up and running |
| **Scenario Description** | WIR requests WIFI interface to connect to access point and WIRClient2 and WIRClient5 WIFI is not available |
| **Post-conditions** | WLAN services successfully connects to WIRClient1 access point  WIR provides WIFI interface to requesting application |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-370389/A-WIRClient1, WIRClient2 and WIRClient5 WI-FI are not available

|  |  |
| --- | --- |
| **Actors** | WIRClient1, WIRServer, WIRClient2, WIRClient5 |
| **Pre-conditions** | 1. Gear position transitions to Park in automatic gear box system and ignition is ON or 2. Gear position transitions to Neutral in a manual gear box system and ignition is ON or 3. Ignition is Accessory or 4. Delayed accessory is ON and Ignition is OFF or 5. Ignition is OFF and delayed accessory is OFF provided ECU is up and running |
| **Scenario Description** | WIR requests WIFI interface to connect to access point and WIRClient1, WIRClient2 and WIRClient5 WIFI is not available |
| **Post-conditions** | WLAN services cannot connect WIFI access point  WIR not able to provide WIFI interface to requesting application |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-296106/B-Known access point not available

|  |  |
| --- | --- |
| **Actors** | WIRClient1, WIRServer, WIRClient2, WIRClient5 |
| **Pre-conditions** | 1. Gear position transitions to Park in automatic gear box system and ignition is ON or 2. Gear position transitions to Neutral in a manual gear box system and ignition is ON or 3. Ignition is Accessory or 4. Delayed accessory is ON and Ignition is OFF or 5. Ignition is OFF and delayed accessory is OFF provided ECU is up and running |
| **Scenario Description** | WIR requests WIFI interface to connect to access point and no known access points are available |
| **Post-conditions** | WLAN services cannot connect to any WIFI access point  WIR not able to provide WIFI interface to requesting application |
| **Interfaces** |  |
| **Notes** |  |

#### WIR-UC-REQ-370033/A-Driver restriction is disabled and engine is off

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClient1, WIRClient2, WIRClient5 |
| **Pre-conditions** | Gear position transitions to Park in automatic gear box system and ignition is OFF |
| **Scenario Description** | WIRServer requests WIFI interface to connect to access point with 10, 10, 10, 20, 40, 60 seconds interval |
| **Post-conditions** | WLAN services request AP scan to WIRClient1 WIRClient2 and WIRClient5  WIRClient1, WIRClient2 and WIRClient5 start scanning with the timer interval  After 6 scanning’s, WIRClient1, WIRClient2 and WIRClient5 goes in to sleep mode if WIRServer can’t find available AP from scan list |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-370030/A-WiFi is connected but no application is requesting WiFi connection

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClient1, WIRClient2, WIRClient5 |
| **Pre-conditions** | Gear position transitions to Park in automatic gear box system and ignition is OFF  WiFi is connected to an AP |
| **Scenario Description** | WIRServer disconnects after 1 min if no application requests it |
| **Post-conditions** | WIRServer initiates WiFi connection  WIRClient1, WIRClient2 or WIRClient5 disconnect from the AP  WIRClient1, WIRClient2 and WIRClient5 are in sleep mode  WIRServer remembers last AP connected |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-370031/A-WIRServer disconnected from an AP and an application requests WiFi connection

|  |  |
| --- | --- |
| **Actors** | WIRServer, WIRClient1, WIRClient2, WIRClient5 |
| **Pre-conditions** | WIRServer disconnected from an AP since there is no application requesting WiFi connection |
| **Scenario Description** | WIRServer connects to the AP connected previously |
| **Post-conditions** | WIRServer tries to connect to the previously connected AP  If it fails to connect to the AP, WIR initiates scanning again |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-370032/A-Last application releases WIR WiFi connection

|  |  |
| --- | --- |
| **Actors** | WIRServer |
| **Pre-conditions** | WIRServer connected to an AP  An application is using the WiFi connection released WIR interface |
| **Scenario Description** | WIRServer disconnects from AP when last application released WIR WiFi interface and no application is requesting for 1 minute |
| **Post-conditions** | Last application released WIR WiFi interface  WIRServer maintains connection for 1 minute  WIR server disconnects from the AP if there is no active or pending connection request |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

### White Box View

## WIR-FUN-REQ-296074/B-Performance and General Requirements

### Requirements

#### WIR-REQ-296075/D-Boot up

WIR module in WIRServer and all WIRClients should be treated as a high priority system service and shall be booted up and ready before connectivity applications are booted up.

#### WIR-REQ-296076/A-Tunnel setup

WIR module shall make sure tunnels are setup before any connectivity needed applications are booted up so WIR local and central controllers can process application requests as soon as application sends requests

#### WIR-REQ-296077/D-WIRServer and WIRClient interfaces

Any inter ECU interfaces between WIRServer and WIRClients shall be treated with high priority and processing shall be instantaneous.

#### WIR-REQ-296078/B-Providing interfaces

If an interface is available or becomes available WIR shall provide the interface instantly to requesting applications without any time delay.

#### WIR-REQ-296079/A-Requests from FCI / WIFI Hotspot and high priority applications

Requests from FCI / WIFI hotspot and applications with high priority intents shall be treated with the highest priority and shall be serviced immediately

#### WIR-REQ-310882/B-Response to privacy mode settings

WIR shall process CCS privacy mode setting and shall immediately disconnect connectivity interfaces or reconnect to network interfaces as stated in requirements from the CCS specification. Currently WLAN is triggering connection change.

Disconnect from network: REQ-309217

Reconnect to network: REQ-309233

#### WIR-REQ-310883/C-eCall / Emergency assistance / ERA-GLONASS / Emergency Connect related requirements

WIR module shall support implementing of any eCall / Emergency assistance / ERA-GLONASS / Emergency Connect related network / WIFI disconnect events. Such disconnect requirements, if any, shall be captured in eCall / Emergency assistance / ERA-GLONASS / Emergency Connect specifications. Those are implemented by WIRClient2 SW by disconnecting APNs.

### Use Cases

### White Box View

## WIR-FUN-REQ-470990/A-Phoenix Support

### Requirements

#### WIR-REQ-470991/A-Support Android OS

WIRServer shall support WIRClient1 Android OS release 12.

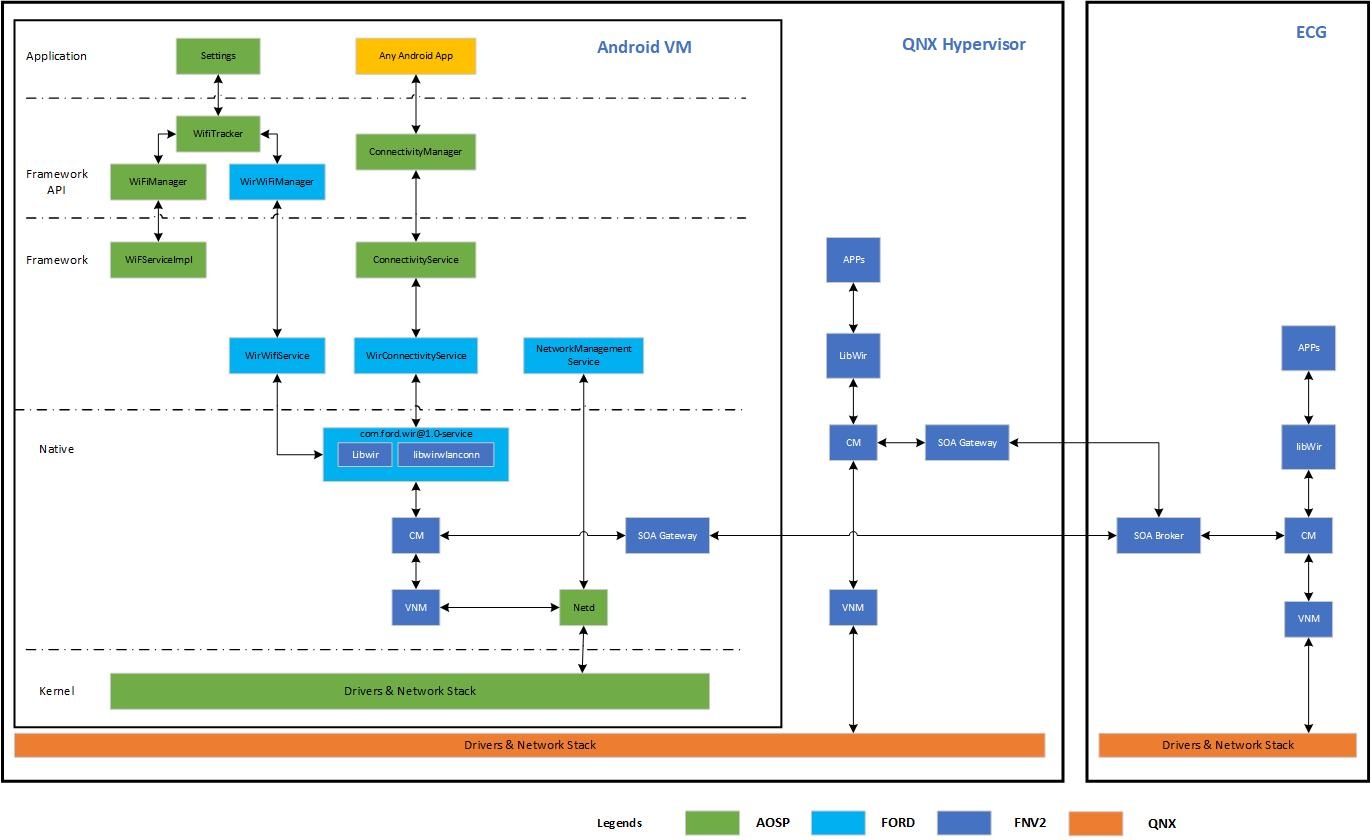
#### WIR-REQ-470992/A-FNV2 CM Client

An FNV2 client shall exist on QNX hypervisor. No direct communication shall exist between QNX WIR CM client and WIR CM on Android. All communication between QNX Hypervisor client and Android client shall go through WIRServer.

#### WIR-REQ-470993/A-WIR Implementation

WIR central controller and local controller shall be implemented as native C/C++ libraries.

* WirConnectivityService: Implements the network Interfaces required in the WIR.
* WirWifiService shall implement basic operations on Wi-Fi.
* WirWifiManager: shall manages requests and responses on WirWiFiServices.
* com.ford.wir@1.0-service: Interfaces provided by libwir and libwirwlanconn libraries shall be embedded here.



#### WIR-REQ-470994/A-Link Bandwidth Support

WIRServer shall use mLinkUpBandwidthKbps and mLinkDownBandwidthKbps of NetworkCapabilities to select target cellular modem if allowed based on policy table.

#### WIR-REQ-470995/A-WIRClient2 WLAN support

WIRServer shall send all WLAN command to both WIRClient2 and WIRClient1 WLAN.

#### WIR-REQ-470996/A-Data Usage Per Application

WIRServer shall support android API’s for collect and report data usage. APIs shall be available to WIRClient1 upon request.

#### WIR-REQ-470997/A-BSSID support on WiFi configuration

WIRServer central controller shall save BSSID as part of WiFi configuration. If the configuration storage reaches maximum size, it shall retain max entries in LRU order.

#### WIR-REQ-470998/A-Android Settings Items for ConnectivityManager

Android Settings application shall be allowed to have full control on Wi-Fi functionality. All or part of items shall be defined as DID.

#### WIR-REQ-470999/A-Wifi Configuration Items

WIRServer shall store following configuration items.

* SSID
* BSSID
* Security type
* Security authentication algorithm
* Frequency band
* Hidden SSID flag
* Rocket setup provided profile
* MAC address

#### WIR-REQ-471000/A-WiFi Auto Connect Support

WIRServer shall support WiFi auto connect. WiFi auto connect settings shall be applied when WIRClient1 is turned off.

#### WIR-REQ-471001/A-MAC Randomization Support

WIRServer shall support MAC randomization via WLAN. MAC randomization is enabled by default for client mode

#### WIR-REQ-471002/A-Wi-Fi Preferred Network Offload (PNO) Scan Support

WIRServer shall support Wi-Fi Preferred Network offload.

PNO scan - Scan schedule during screen off and WiFi disconnected state

#### WIR-REQ-471003/A-Dual WLAN Support

WIRServer shall not support both WIRClient1 and WIRClient2 WLAN at the same time as a STA mode.

#### WIR-REQ-471004/A-WLAN Command and Response Routing

WIRServer shall detect current WLAN chipset target and send all WLAN commands to target ECU’s supplicant (WIRClient1 or WIRClient2).

#### WIR-REQ-471005/A-WiFi Scoring Algorithm Support

WIRServer shall support android wifi scoring algorithm for wifi network selection.

#### WIR-REQ-471006/A-WiFi Health Monitoring

Android records and detects WiFi health issue after daily and after SW upgrade. If it finds any issues, it reports to Server. WIRServer shall support same feature when WIRClient1 is not running. Wi-Fi health monitor report shall be sent to Ford server.

#### WIR-REQ-471007/A-WiFi Throughput Estimation Support

WIRServer shall support Wifi throughput estimation regardless WIRClient1 is running.

Android estimates WiFi throughput based on standard (802-a, b, …), RSSI, channel utilization, velocity.

Velocity based throughput estimation shall utilize RSSI, Kalman filtering, number of TX success, TX bad, TX retry, RX success, RX bad and RX retry packets.

#### WIR-REQ-471008/A-WPA3 Support

WIRServer shall support WPA3.

#### WIR-REQ-471009/A-Networking and Device Discovery

WIRServer SHALL support the relevant parts of the following RFCs:

* RFC 1213: Management InformationBase for Network Management of TCP/IP based internets:MIB-II”, Mar. 1991
* RFC 791: Internet Protocol (IP), DARPA Internet Program, protocol specification, Sept 1981
* RFC 2460: Internet Protocol, Version 6 (IPv6), Dec 1998
* RFC 4193:  Unique Local IPv6 Unicast Addresses, Oct 2005RFC
* RFC 793: Transmission Control Protocol (TCP), Sept 1981
* RFC 768: User Datagram Protocol (UDP), Aug. 1980
* RFC 826: An Ethernet Address Resolution Protocol (ARP),Nov. 1982
* RFC 4861: Neighbor Discovery for IP version 6 (IPv6), Sept 2007
* RFC 2131: Dynamic Host Configuration Protocol (DHCP), Mar. 1997
* RFC 792: Internet Control Message Protocol (ICMP), DARPA Internet Program, Protocol specification, Sept. 1981
* RFC 4443: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, Mar 2006
* RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, Dec. 1998
* RFC 3966: The tel URI for Telephone Numbers, Dec. 2004
* RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification, June 2010
* RFC 1398: Definitions of Managed Objects for the Ethernet-like Interface Types, Jan. 1993

## WIR-FUN-REQ-370034/A-Captive Portal Check

### Requirements

#### WIR-REQ-370003/B-Requesting WLAN connection

WIRServer shall request connection to external WiFi Hotspot to applicable WIRClient. WIRServer shall send the request with SSID and password for the WiFi Hotspot.

WIRServer shall receive the connection result from applicable WIRClient.

#### WIR-REQ-370004/B-Caching WiFi Hotspot password

WIRServer shall save SSID and password pair when applicable WIRClient is connected to Wi-Fi Hotspot.

When connected Wi-Fi Hotspot is open network, WIRServer shall save the password as open network.

#### WIR-REQ-370005/B-Retrieving saved WiFi Hotspot password

WIRServer shall retrieve saved Wi-Fi Hotspot password based on SSID value and provide SSID and password pair to applicable WIRClients when requesting WLAN connection to Wi-Fi Hotspot.

#### WIR-REQ-370006/B-Common WiFi Hotspot SSID/password database

WIRServer shall maintain one SSID/password database for applicable WIRClients.

#### WIR-REQ-370007/B-Captive portal check request

Once WIRClient WLAN is connected to a Wi-Fi Hotspot, WIRServer shall start captive portal check request to CPC. CPC run captive portal check procedure and returns either internet connectivity success or failure.

#### WIR-REQ-370008/B-CPC procedure

WIRServer CM shall check CONNECTIVITY\_CHECK\_URL\_AVAILABLITY DID. If the DID is set to ENABLED, WIRServer CM shall enable CPC feature. If it is set to FALSE, WIRServer CM shall disable CPC feature and return internet connectivity is available when WLAN is connected to WiFi Hotspot. Factory default value shall be enabled.

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| CONNECTIVITY\_CHECK\_URL\_AVAILABLITY | Boolean | FALSE – Disable CPC feature  TRUE – Enable CPC feature |

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| CPC\_MAX\_CHECK | Integer | Maximum CPC check times |
| CPC\_CHECK\_INTERVAL | Integer | Interval in seconds between CPC check |

After received a captive portal check request, CPC shall wait CPC\_CHECK\_INTERVAL seconds. After the wait, CPC shall send HTTP request to CONNECTIVITY\_CHECK\_URL.

If the HTTP response is 2XX success, CPC determines internet connection through the WLAN is ready and return success.

If the HTTP response is any other response or there is no response, CPC shall wait CPC\_CHECK\_INTERVAL seconds and send next HTTP request until it tried CPC\_MAX\_CHECK times.

If the HTTP response 3XX redirection, CPC shall send WEB engine display request with URI and local IP address.

After CPC\_MAX\_CHECK times trials, if CPC receives 3XX, 4XX or 5XX HTTP response or there is no response, CPC shall report internet connection failure to WIRServer Central controller.

If the response is a redirection response (3XX Redirection), CPC shall send the redirected URI and local IP address assigned by WLAN to WEB engine. The WEB engine shall display the HTML page and when HTTP procedure triggered by customer input, it shall use supplied local IP address for HTTP connection until the WEB engine is closed.

DID CONNECTIVITY\_CHECK\_URL is a WEB URL to check captive portal status. CONNECTIVITY\_CHECK\_URL shall be programmed at factory.

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| CONNECTIVITY\_CHECK\_URL | String, 128 bytes | URL (ex., [www.google.com](http://www.google.com))  China – [www.baidu.com](http://www.baidu.com)  EU – per country |

For EU, if some country limits HTTP request to inside of country, CONNECTIVITY\_CHECK\_URL shall be programmed at EOL to web site which belongs to the country.

#### WIR-REQ-370009/B-Central controller

WIRServer central controller receives WIR allocation request from local controller of WIRServer, WIRClient1 or WIRClient2. After receiving a notification WLAN is connected to Wi-Fi Hotspot, WIRServer central controller shall send CPC check request to CPC.

After CPC determined internet connectivity status through WLAN, CPC reports connectivity status to WIRServer central controller.

If WIRServer central controller receives an internet connection failure report from CPC, WIRServer central controller shall initiate WLAN disconnection from the Wi-Fi Hotspot and WIRServer shall not automatically connect to the AP until next ignition cycle but user initiated connection shall be allowed.

#### WIR-REQ-471010/A-Not supporting HTTP captive portal

WIRServer shall check 3XX response and determine the redirection is to HTTP or HTTPS server. WIRServer shall disconnect from the AP if the redirection is to HTTP server and display message insecure captive portal is not supported. WIRServer shall not auto connect to the AP until WIRServer power down.

#### WIR-REQ-370010/A-Request WEB engine to display landing page

WIRServer CPC sends a redirected URL which is specified on Location field of 3XX response and local IP address to WEB engine to be displayed on WIRClient1UI.

Following shows an example 302 HTTP response.

HTTP/1.1 302 Found

Server: Apache/2.4.23 (Unix)

Location: /nbrd/visit/counter.html

Content-Type: text/html

Content-Length: 0

#### WIR-REQ-370011/A-WLAN internet connection lost

After Wi-Fi Hotspot is connected and internet service provided, some Wi-Fi service provider may deactivate internet service and start to redirect all HTTP request to service provider’s captive portal again after predefined time limit.

WIR application shall detect a socket layer error (ex., Destination not reachable) and report it to WIR when the application receives the error 5 times in consecutive. WIR shall send a network down event to all WIR application which are using the WLAN interface. Then WIR application shall request WIR connection request again if it requires internet connection.

#### WIR-REQ-370012/A-CPC check procedure

Following diagram shows CPC check procedure flow.



### Use Cases

#### WIR-UC-REQ-369972/A-WIRClient1 Wi-Fi connected to a Wi-Fi Hotspot first time and redirected to landing page

|  |  |
| --- | --- |
| **Actors** | WIRClient1 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1 Wi-Fi turned on  Customer initiates Wi-Fi scan  WIRClient1 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | WIRClient1 Wi-Fi finds available Wi-Fi Hotspot and makes a connection to it  Wi-Fi Hotspot requires landing page loading and accepts T&C by a customer |
| **Post-conditions** | WIRClient1 Wi-Fi finds available SSID  Customer selects a Wi-Fi Hotspot to connect from WIRClient1 UI and enters password  WIRClient1 Wi-Fi connects to the Wi-Fi Hotspot  CPC waits for CPC\_CHECK\_INTERVAL seconds  CPC sends HTTP GET request to connectivity test URL  CPC receives HTTP 3XX redirection to landing page response  CPC sends redirected URI and local IP address to WEB engine  CPC start a wait timer for CPC\_CHECK\_INTERVAL seconds  WEB engine displays the landing page  Customer clicks accept button  WEB engine binds local IP address, creates HTTP connection and sends HTTP GET request for home page  Wi-Fi Hotspot sends home page and WLAN routes it to WEB engine  WEB engine displays the home page  The wait timer expires  CPC sends HTTP GET request to check captive portal stage is done  CPC receives HTTP 2XX Success response  CPC notifies WIRServer central controller internet connectivity is available  WIR saves SSID and password pair for future use |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369973/A-WIRClient1 Wi-Fi connected to an Wi-Fi Hotspot first time and Wi-Fi Hotspot doesn’t have landing page

|  |  |
| --- | --- |
| **Actors** | WIRClient1 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1 Wi-Fi turned on  WIRClient1 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | WIRClient1 Wi-Fi finds available Wi-Fi Hotspot and makes a connection to it.  Wi-Fi Hotspt provides internet connection without landing page redirection |
| **Post-conditions** | WIRClient1 Wi-Fi finds available SSID  Customer selects a Wi-Fi Hotspot to connect from WIRClient1 UI  WIRClient1 Wi-Fi connects to the Wi-Fi Hotspot  CPC sends HTTP GET request to connectivity test URL  CPC receives HTTP 2XX Success response  CPC ignores the HTTP response  CPC notifies WIRServer central controller internet connectivity is available  WIR saves SSID and password pair for future use |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369974/B-WIRClient Wi-Fi connected to a Wi-Fi Hotspot connected previously

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi turned on  WIRClient1/WIRClient2/WIRClient5 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds available open Wi-Fi Hotspot and makes a connection to it.  WIR provides Wi-Fi Hotspot internet connection without user interaction |
| **Post-conditions** | WIR sends WLAN connection request  WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds an available SSID  WIRClient1/WIRClient2/WIRClient5 Wi-Fi connects to the Wi-Fi Hotspot  CPC sends HTTP GET request to connectivity test URL  CPC receives 2XX OK response  CPC notifies WIR internet connection success  WIR gets internet connection through Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369975/B-WIRClient Wi-Fi connected to an open Wi-Fi Hotspot connected previously and redirected to landing page

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi turned on  WIRClient1/WIRClient2/WIRClient5 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds available open Wi-Fi Hotspot and makes a connection to it.  Wi-Fi Hotspt redirect to landing page |
| **Post-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds available SSID  Customer selects a Wi-Fi Hotspot to connect from WIRClient1 UI and enters password  WIRClient1 Wi-Fi connects to the Wi-Fi Hotspot  CPC waits for CPC\_CHECK\_INTERVAL seconds  CPC sends HTTP GET request to connectivity test URL  CPC receives HTTP 3XX redirection to landing page response  CPC sends redirected URI and local IP address to WEB engine  CPC start a wait timer for CPC\_CHECK\_INTERVAL seconds  WEB engine displays the landing page  Customer clicks accept button  WEB engine binds local IP address, creates HTTP connection and sends HTTP GET request for home page  Wi-Fi Hotspot sends home page and WLAN routes it to WEB engine  WEB engine displays the home page  The wait timer expires  CPC sends HTTP GET request to check captive portal stage is done  CPC receives HTTP 2XX Success response  CPC notifies WIRServer central controller internet connectivity is available  WIR saves SSID and password pair for future use |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369976/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot first time

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi turned on  WIRClient1/WIRClient2/WIRClient5 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | Customer enters password and WIRClient1/WIRClient2 Wi-Fi gets an internet connection |
| **Post-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds available password protected Wi-Fi Hotspot  Customer selects the Wi-Fi Hotspot to connect from WIRClient1 UI  WIRClient1/WIRClient2/WIRClient5 Wi-Fi makes a connection to it.  Wi-Fi Hotspot password enter screen is shown on WIRClient1 display  Customer enters password  WIRClient1/WIRClient2/WIRClient5 Wi-Fi finishes authentication procedure  Customer opens WEB browser  WIR saves SSID and password for future use  CPC sends HTTP GET request to connectivity test URL  CPC receives 2XX OK response  CPC notifies WIR internet connection success  WIR gets internet connection through Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369977/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters verification code recei

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi turned on  WIRClient1/WIRClient2/WIRClient5 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | Customer enters password and WIRClient1/WIRClient2 Wi-Fi gets internet connection |
| **Post-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds available password protected Wi-Fi Hotspot  Customer selects the Wi-Fi Hotspot to connect from WIRClient1 UI  WIRClient1/WIRClient2/WIRClient5 Wi-Fi makes a connection to it.  Wi-Fi Hotspot password enter screen is shown on WIRClient1 display  Customer enters password  WIRClient1/WIRClient2/WIRClient5 Wi-Fi finishes authentication procedure  Customer opens WEB browser  WIR saves SSID and password for future use  Customer enters his Smartphone number on the WEB page  Wi-Fi Hotspot provider sends verification code to customer’s Smartphone through MT-SMS  Customer enters verification code on the WEB page  Wi-Fi Hotspot provider verifies password and send home page response  CPC sends HTTP GET request to connectivity test URL  CPC receives 2XX OK response  CPC notifies WIR internet connection success  WIR has internet connection through Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369978/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot first time and lading page displayed

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi turned on  WIRClient1/WIRClient2/WIRClient5 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | Customer enters password and WIRClient1/WIRClient2 Wi-Fi gets internet connection  Landing page displayed after connection |
| **Post-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds available password protected Wi-Fi Hotspot  Customer selects the Wi-Fi Hotspot to connect from WIRClient1 UI  WIRClient1/WIRClient2/WIRClient5 Wi-Fi makes a connection to it.  Wi-Fi Hotspot password enter screen is shown on WIRClient1 display  Customer enters password  WIRClient1/WIRClient2/WIRClient5 Wi-Fi finishes authentication procedure  Customer opens WEB browser  WIR saves SSID and password for future use  CPC sends HTTP GET request for test URL  CPC receives 3XX redirection response  CPC sends redirected URI and local IP address to WEB engine  Landing pages displayed on WEB engine  Customer accepts T&Cs  WEB engine make a HTTP connection with supplied local IP address and sends HTTP GET request for Wi-Fi Hotspot home page  WEB engine receives home page response and display it on WIRClient1 screen  CPC sends HTTP GET request to connectivity test URL  CPC receives 2XX OK response  CPC notifies WIR internet connection success  WIR has internet connection through Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369979/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot first time and customer enters password with on-screen

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/ WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi turned on  WIRClient1/WIRClient2/WIRClient5 Wi-Fi is connecting to Wi-Fi Hotspot and customer is asked to enter password |
| **Scenario Description** | Customer enters password with on-screen keyboard |
| **Post-conditions** | WIRClient1 UI displays password enter screen  Customer enters password  WIRClient1/WIRClient2//WIRClient5 WLAN starts connection procedure to Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369980/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot again

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi turned on  WIRClient1/WIRClient2/WIRClient5 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | WIR uses saved SSID/password for connection |
| **Post-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds available password protected Wi-Fi Hotspot  WIRServer central controller find saved SSID/password and provides it to WIRClient1/WIRClient2/WIRClient5 Wi-Fi when request to connect  WIRClient1/WIRClient2/WIRClient5 Wi-Fi finishes connection procedure  CPC sends HTTP GET request to connectivity test URL  CPC receives 2XX OK response  CPC notifies WIR internet connection success  WIR has internet connection through Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369981/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot again and landing page displayed

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi turned on  WIRClient1/WIRClient2/WIRClient5 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | WIR uses saved SSID/password for connection |
| **Post-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds available password protected Wi-Fi Hotspot  Customer makes a connection to it through WIRClient1 UI  WIR found saved SSID/password and provides it to WIRClient1/WIRClient2/WIRClient5 Wi-Fi  WIRClient1/WIRClient2/WIRClient5 Wi-Fi finishes authentication procedure  CPC sends HTTP GET request for connectivity test URL  CPC receives 3XX redirection response  CPC sends redirected URI and local IP address  Landing pages displayed on WEB engine  Customer accepts T&Cs  WEB engine make a HTTP connection with supplied local IP address and sends HTTP GET request for Wi-Fi Hotspot home page  WEB engine receives home page response and display it on screen  CPC sends HTTP GET request to connectivity test URL  CPC receives 2XX OK response  CPC notifies WIR internet connection success  WIR has internet connection through Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369982/B-WIRClient Wi-Fi connected to password protected Wi-Fi Hotspot again but authentication fails

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi turned on  WIRClient1/WIRClient2/WIRClient5 Wi-Fi is scanning available Wi-Fi Hotspot signal |
| **Scenario Description** | Saved password is not valid and customer enters password |
| **Post-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds an available password protected Wi-Fi Hotspot  WIR found saved SSID/password and provides it to WIRClient1/WIRClient2/WIRClient5 Wi-Fi  WIRClient1/WIRClient2/WIRClient5 Wi-Fi fails authentication procedure due to password changed since last connection  Wi-Fi Hotspot password enter screen is shown on WIRClient1 display  Customer enters password  WIRClient1/WIRClient2/WIRClient5 Wi-Fi client finishes authentication procedure  CPC sends HTTP GET request to connectivity test URL  CPC receives 2XX OK response  CPC notifies WIR internet connection success  WIR has internet connection through Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369983/B-Wi-Fi connection lost after Wi-Fi Hotspot timeout

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi is connected to Wi-Fi Hotspot  WIR has internet connection through WIRClient1/WIRClient2 Wi-Fi |
| **Scenario Description** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi connection lost after timeout |
| **Post-conditions** | Wi-Fi Hotspot connection timer expired  Wi-Fi Hotspot redirect all HTTP request and ignore all other internet traffic  WIR application receives a socket error (ex. destination unreachable) for 5 times consecutively  WIR application reports the socket error to WIR  WIR sends network interface down notification to all WIR app which is using the WLAN |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369984/B-Wi-Fi connection lost due to Wi-Fi signal becomes too weak

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi is connected to Wi-Fi Hotspot  WIR has internet connection through WIRClient1/WIRClient2/WIRClient5 Wi-Fi |
| **Scenario Description** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi lost connection when signal becomes weak |
| **Post-conditions** | Customer drive vehicle  Wi-Fi connection lost  WIRClient1/WIRClient2/WIRClient5 WLAN notifies WIR connection lost event  WIR detects Wi-Fi connection lost  WIR notifies WIR app network interface down event |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369985/B-Wi-Fi connection lost during connection setup due to Wi-Fi signal becomes weak

|  |  |
| --- | --- |
| **Actors** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi, Wi-Fi Hotspot, customer, WIR |
| **Pre-conditions** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi finds available Wi-Fi Hotspot |
| **Scenario Description** | WIRClient1/WIRClient2/WIRClient5 Wi-Fi lost connection during connection setup when signal becomes weak |
| **Post-conditions** | WIR finds Wi-Fi Hotspot  Optionally customer enters password if it is not saved and password protected  WIRClient1/WIRClient2/WIRClient5 initiates connection procedure to Wi-Fi Hotspot  During connection procedure Wi-Fi Hotspot signal becomes too weak and connection lost.  WIR notifies WIR app allocation failure |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369986/B-Customer selects another screen while landing page displayed

|  |  |
| --- | --- |
| **Actors** | WIRClient1 UI, customer |
| **Pre-conditions** | WIRClient1 UI displays landing page |
| **Scenario Description** | Customer selects another screen to do something |
| **Post-conditions** | Customer selects another screen from WIRClient1 UI  Customer continues to use WIRClient1 application while landing page display staying on background  WEB engine instance is killed and customer can’t access landing page again |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369987/A-Other screen overrides while landing page displayed

|  |  |
| --- | --- |
| **Actors** | WIRClient1 UI, CPC, customer, WIR |
| **Pre-conditions** | WIRClient1 UI displays landing page |
| **Scenario Description** | Other screen override landing page screen |
| **Post-conditions** | Other screen (rear view camera or popup) displayed on WIRClient1 screen by some event  Other screen finished  Landing screen displayed again  Customer continues accepting T&C  CPC sends HTTP GET request to connectivity test URL  CPC receives 2XX OK response  CPC notifies WIR internet connection success  WIR has internet connection through Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369988/A-WEB engine supports different WIRClient1 screen size

|  |  |
| --- | --- |
| **Actors** | WEB engine, CPC |
| **Pre-conditions** | CPC requests WEB engine to display WEB page |
| **Scenario Description** | WEB engine formats WEB page based on WIRClient1 screen size |
| **Post-conditions** | CPC receives redirection WEB page  CPC requests WEB engine to display the WEB page with URI and local IP address  WEB engine properly displays the WEB page based on WIRClient1 screen size  WEB engine handles user input correctly |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369989/A-CPC get internet connection success after multiple try

|  |  |
| --- | --- |
| **Actors** | CPC, WIR |
| **Pre-conditions** | CPC received HTTP 3XX redirection  CPC passed redirected URI and local IP address to WEB engine  WEB engine displayed the page on WIRClient1 screen  Customer is interacting with the page |
| **Scenario Description** | CPC checks internet connectivity and gets success after multiple try |
| **Post-conditions** | CPC sends HTTP GET request to connectivity test URL  CPC receives 3XX redirection response  CPC waits CPC\_CHECK\_INTERVAL seconds  CPC sends HTTP GET request to connectivity test URL  CPC receives 3XX redirection response  CPC waits CPC\_CHECK\_INTERVAL seconds  CPC sends HTTP GET request to connectivity test URL  CPC receives 2XX response  CPC notifies WIR internet connection success  WIR has internet connection through Wi-Fi Hotspot |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369990/A-CPC get internet connection failure after max try

|  |  |
| --- | --- |
| **Actors** | CPC, WIR |
| **Pre-conditions** | CPC received HTTP 3XX redirection  CPC passed redirected URI and local IP address  WEB engine displayed the page on WIRClient1 screen  Customer is interacting with the page |
| **Scenario Description** | CPC checks internet connectivity and gets failure after max try |
| **Post-conditions** | Customer ignores landing page OR closes WEB engine  CPC sends HTTP GET request to connectivity test URL  CPC receives 3XX redirection response  CPC waits CPC\_CHECK\_INTERVAL seconds  CPC sends HTTP GET request to connectivity test URL  CPC receives 3XX redirection response  CPC waits CPC\_CHECK\_INTERVAL seconds  CPC sends HTTP GET request to connectivity test URL  CPC receives 3XX redirection response  CPC tried all CPC\_MAX\_CHECK times  CPC notifies WIR internet connection failure  WIR doesn’t have internet connection through Wi-Fi Hotspot  WIR notifies WIR app allocation failure |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369991/A-Password enter screen interrupted by other screen

|  |  |
| --- | --- |
| **Actors** | WIRClient1 UI, customer |
| **Pre-conditions** | Password enter screen is displayed |
| **Scenario Description** | Password enter screen is displayed but override by other screen  Customer switch password screen and continue entering screen |
| **Post-conditions** | Customer is entering password  WIRClient1 screen is switched to other screen by some event  Customer switches to password screen  Customer continues entering password |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369992/A-WEB engine fails to display landing page

|  |  |
| --- | --- |
| **Actors** | CPC, WEB engine, diagnostic agent, WIR |
| **Pre-conditions** | WEB engine received landing page display request from CPC |
| **Scenario Description** | WEB engine fails to display landing page and CPC returns internet connection failure to WIR |
| **Post-conditions** | WEB engine fails to display the landing page  Diagnostic agent detects the failure and creates a log entry  CPC tries CPC\_MAX\_CHECK but receives 3XX redirection response  CPC notifies WIR internet connection failure  WIR doesn’t have internet connection through Wi-Fi Hotspot  WIR notifies WIR app allocation failure |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

#### WIR-UC-REQ-369993/A-WEB engine opens 2nd WEB page while landing page is displayed---TBD

|  |  |
| --- | --- |
| **Actors** | WEB engine, customer |
| **Pre-conditions** | WEB engine displayed landing page on WIRClient1 screen |
| **Scenario Description** | 2nd WEB engine override landing page screen |
| **Post-conditions** | WEB engine receives another WEB page display request  WEB engine displays 2nd WEB page  Customer switch to landing page screen  Customer continue to interact with the landing page |
| **List of Exception Use Cases** |  |
| **Interfaces** |  |

### White Box View

## WIR-FUN-REQ-296080/A-WIRServer Configuration

### Requirements

#### WIR-REQ-296095/G-Configuration Parameters

The list of config parameters for WIRServer are identified in table below. These parameters shall be updatable from cloud as well as from EOL.

| **Parameter** | **Description** | **Default value** | **Metric** | **Resolution** | **Tuning Range** | **DID** |
| --- | --- | --- | --- | --- | --- | --- |
| WIR\_LATENCY\_  CHECK | Whether to check Latency of WIFI connection | 1 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| WIR\_LOW\_COST\_WIFI\_PARTNER\_AVAILABILITY | Whether Ford has a low cost partnered WIFI | 1 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| WIR\_WIFI\_PARTNER\_CREDENTIALS\_SSID | Partnered WIFI SSID credentials | Null | String | NA | 0 to 255 characters | FD18 |
| WIR\_WIFI\_PARTNER\_CREDENTIALS\_PASSWORD | Partnered WIFI password credentials | Null | String | NA | 0 to 255 characters | FD18 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_1 | WIFI scan interval 1st | 10 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_2 | WIFI scan interval 2nd | 10 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_3 | WIFI scan interval 3rd | 10 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_4 | WIFI scan interval 4th | 20 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_5 | WIFI scan interval 5th | 40 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_6 | WIFI scan interval 6th | 60 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_REPEAT | WIFI scan interval repeat | 120 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_OFFPEAK\_  SUPPORT | Whether to check Offpeak support is available or not | 1 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| WIR\_OFF-PEAK\_  BATTERY\_SOC | Minimum Battery off-peak SOC for off-peak qualification | 70 | Percentage | 1 | 0 to 100 | FD03 |
| WIR\_WIFI\_  DIAGNOSTIC\_  STORAGE\_TIME | Time interval for which to store the WIFI data in non volatile memory | 30 | Ignition cycles | 1 | 0 to 255 | FD03 |
| WIR\_GENERAL\_  DIAGNOSTIC\_  STORAGE\_TIME | Time interval for which to store the WIFI data in non volatile memory | 30 | Ignition cycles | 1 | 0 to 255 | FD03 |
| WIFI\_CONNECT\_  REMINDER\_COUNTER | Number of ignition cycles which the user did not connect WIFI.  Resets when user connects to WIFI | 100 | Ignition cycles | 1 | 0 to 512 | FD03 |
| WIFI\_CONNECT\_  REMINDER\_TEMP | Number of ignition cycles which the user did not connect WIFI after user selected remind me later  Resets when user connects to WIFI | 5 | Ignition cycles | 1 | 0 to 255 | FD03 |
| WIR\_WIFI\_PARTNER\_CREDENTIALS\_DHCP | WIFI partner DHCP data | NULL | String | NA | 0 to 255 characters | FD18 |
| WIR\_WIFI\_PARTNER\_CREDENTIALS\_SECURITY\_WEP\_WPA | The type of security used by the WIFI partner | 0 | Integer | 0 – WPA  1- WEP | 0-7 | DE0E |
| CONNECTIVITY\_CHECK\_URL | URL to check connectivity | NULL | String | NA | 0 to 50 characters | DE13 |
| CONNECTIVITY\_CHECK\_URL\_AVAILABLITY | Whether connectivity URL is available or not | 1 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| DISPLAY\_WIFIHOTSPOT\_CREDENTIALS | Whether to display WIFI hotspot credentials in WIRClient1 HMI for user to connect | 0 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| CPC\_MAX\_CHECK | Max CPC request times | 5 | Integer | Times | 0 to 255 | FD03 |
| CPC\_CHECK\_INTERVAL | Wait time between CPC request | 30 | Integer | Seconds | 0 to 255 | FD03 |
| FACTORY\_WIFI\_VALID\_DURATION | Valid duration of factory WiFi after production | 30 | Integer | Days | 0 to 255 | FD03 |
| OFFPEAK\_VPSM\_SEVERITY\_LEVEL | Severity level for offpeak subscription to VPSM | 2 | Integer | 1 | 1 to 4 | FD28 |
| WIFI\_CONNECTION\_ALERT\_INTERVAL | Days to report WiFi connection alert | 30 | Integer | Days | 0 to 255 | FD28 |
| POLICY\_PULL\_IGN\_CYCLE | Number of ignition cycles before sending policy table pull request | 100 | Integer | Number of ignition cycle | 1 to 255 | FD28 |

#### WIRv2-REQ-471011/A-Configuration Parameters

The list of config parameters for WIRServer are identified in table below. These parameters shall be updatable from cloud as well as from EOL.

| **Parameter** | **Description** | **Default value** | **Metric** | **Resolution** | **Tuning Range** | **DID** |
| --- | --- | --- | --- | --- | --- | --- |
| WIR\_LATENCY\_  CHECK | Whether to check Latency of WIFI connection | 1 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| WIR\_ANDROID\_DEFAULT\_MODEM | Default WIRClient2 for android applications | 1 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable |  |
| WIR\_LOW\_COST\_WIFI\_PARTNER\_AVAILABILITY | Whether Ford has a low cost partnered WIFI | 1 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| WIR\_WIFI\_PARTNER\_CREDENTIALS\_SSID | Partnered WIFI SSID credentials | Null | String | NA | 0 to 255 characters | FD18 |
| WIR\_WIFI\_PARTNER\_CREDENTIALS\_PASSWORD | Partnered WIFI password credentials | Null | String | NA | 0 to 255 characters | FD18 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_1 | WIFI scan interval 1st | 10 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_2 | WIFI scan interval 2nd | 10 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_3 | WIFI scan interval 3rd | 10 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_4 | WIFI scan interval 4th | 20 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_5 | WIFI scan interval 5th | 40 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_6 | WIFI scan interval 6th | 60 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_WIFI\_PERIODIC\_SCAN\_INTERVAL\_REPEAT | WIFI scan interval repeat | 120 | Integer | Seconds | 0 – Disable periodic scan  1 to 255 – scan interval | FD03 |
| WIR\_OFFPEAK\_  SUPPORT | Whether to check Offpeak support is available or not | 1 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| WIR\_OFF-PEAK\_  BATTERY\_SOC | Minimum Battery off-peak SOC for off-peak qualification | 70 | Percentage | 1 | 0 to 100 | FD03 |
| WIR\_WIFI\_  DIAGNOSTIC\_  STORAGE\_TIME | Time interval for which to store the WIFI data in non volatile memory | 30 | Ignition cycles | 1 | 0 to 255 | FD03 |
| WIR\_GENERAL\_  DIAGNOSTIC\_  STORAGE\_TIME | Time interval for which to store the WIFI data in non volatile memory | 30 | Ignition cycles | 1 | 0 to 255 | FD03 |
| WIFI\_CONNECT\_  REMINDER\_COUNTER | Number of ignition cycles which the user did not connect WIFI.  Resets when user connects to WIFI | 100 | Ignition cycles | 1 | 0 to 512 | FD03 |
| WIFI\_CONNECT\_  REMINDER\_TEMP | Number of ignition cycles which the user did not connect WIFI after user selected remind me later  Resets when user connects to WIFI | 5 | Ignition cycles | 1 | 0 to 255 | FD03 |
| WIR\_WIFI\_PARTNER\_CREDENTIALS\_DHCP | WIFI partner DHCP data | NULL | String | NA | 0 to 255 characters | FD18 |
| WIR\_WIFI\_PARTNER\_CREDENTIALS\_SECURITY\_WEP\_WPA | The type of security used by the WIFI partner | 0 | Integer | 0 – WPA  1- WEP | 0-7 | DE0E |
| CONNECTIVITY\_CHECK\_URL | URL to check connectivity | NULL | String | NA | 0 to 50 characters | DE13 |
| CONNECTIVITY\_CHECK\_URL\_AVAILABLITY | Whether connectivity URL is available or not | 1 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| DISPLAY\_WIFIHOTSPOT\_CREDENTIALS | Whether to display WIFI hotspot credentials in WIRClient1 HMI for user to connect | 0 | ENABLE / DISABLE (Boolean) | 1 | 0 – Disable  1 – Enable | DE0E |
| CPC\_MAX\_CHECK | Max CPC request times | 5 | Integer | Times | 0 to 255 | FD03 |
| CPC\_CHECK\_INTERVAL | Wait time between CPC request | 30 | Integer | Seconds | 0 to 255 | FD03 |
| FACTORY\_WIFI\_VALID\_DURATION | Valid duration of factory WiFi after production | 30 | Integer | Days | 0 to 255 | FD03 |
| OFFPEAK\_VPSM\_SEVERITY\_LEVEL | Severity level for offpeak subscription to VPSM | 2 | Integer | 1 | 1 to 4 | FD28 |
| WIFI\_CONNECTION\_ALERT\_INTERVAL | Days to report WiFi connection alert | 30 | Integer | Days | 0 to 255 | FD28 |
| POLICY\_PULL\_IGN\_CYCLE | Number of ignition cycles before sending policy table pull request | 100 | Integer | Number of ignition cycle | 1 to 255 | FD28 |

#### WIR-REQ-370035/A-Master Reset and Configuration Parameters

When master reset triggered, all configuration parameters shall be restored to factory default values.

### Use Cases

### White Box View

# Appendix: Reference Documents

|  |  |
| --- | --- |
| Reference # | Document Title |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |