Accessory Interface Specification

Release R22

Developer

Contents

1. Introduction 50
1.1 Purpose of This Specification 50
1.2 Requirements, Recommendations, and Permissions 50
1.3 Applicability 51
1.4 Terminology 52
1.4.1 Accessory, Device, and Product 52
1.4.2 Authentication Coprocessor 52
1.4.3 I2C Bus 52
1.4.4 Challenge 52
1.4.5 Challenge Response 52
1.4.6 X.509 Certificate 53
1.4.7 Component 53
1.4.8 Feature 53
1.4.9 USB Device and Host Mode 53
1.4.10 iAP 53
1.4.11 Direct User Action 54

2. General Requirements and Recommendations 55
2.1 Minimum Apple Device Compatibility 55
2.2 Development Tools and Emulators 56
2.3 Reference Designs & Development Kits 57
2.4 Accessory Authentication and Accessory Identification 57
2.5 iAP 57
2.6 Connector Assemblies 58
2.7 Adapters and Proxies 58
2.8 Mixed 30-pin and Lightning Connectors 58
2.9 Mixed Headset Jack and Lightning Connectors 60
2.10 Apple USB Power Adapters 61
2.11 Apple Device Detection 61
2.12 Multiple Simultaneous iAP2 Connections 61
2.13 Presentation of Apple Device Updates 61
2.14 Relationships Between Multiple Accessories 62
2.15 iBeacon 62
2.16 Feature Duplication 62

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2.17 Temperature Range 63
2.18 Magnetic Fields 63
2.19 Cables with USB Connectors 63
2.20 Cables with Non-USB Connectors 63
2.21 Integrated USB Receptacles 63
2.22 Integrated Non-USB Receptacles 64
2.23 User Supplied Cables and Power Supplies 64
2.24 Removable Storage 65
2.25 RF Transmission and Reception 65
2.26 TDMA Noise 65

3. Apple Authentication Coprocessor 2.0C 66
3.1 Coprocessor 2.0C Overview 66
3.2 Coprocessor 2.0C Authentication Protocol 66
3.3 Coprocessor 2.0C Signals and Pinouts 66
3.4 Coprocessor 2.0C Address Selection 67
3.5 Coprocessor 2.0C Reference Circuit 68
3.6 Coprocessor 2.0C System Voltage 68
3.7 Coprocessor 2.0C I2C Interface 69
3.7.1 I2C Startup On Power On 69
3.7.2 I2C Startup On Warm Reset 70
3.7.3 I2C Communications Process 71
3.7.4 I2C Sleep Mode 72
3.8 Coprocessor 2.0C Registers 72
3.8.1 Register Addresses 72
3.8.2 Register Descriptions 75
3.9 Coprocessor 2.0C I2C Protocol 82
3.9.1 Slave Selection and Reset 82
3.9.2 Coprocessor Busy 82
3.9.3 Writing to the Coprocessor 82
3.9.4 Reading from the Coprocessor 82
3.10 Coprocessor 2.0C Device Characteristics 83
3.10.1 Physical Configuration 83
3.10.2 Maximum Environmental Conditions 90
3.10.3 Recommended Operating Conditions 90
3.10.4 I2C Interface Characteristics 90
3.10.5 DC Electrical Characteristics 91
3.10.6 Timing Characteristics 91

4. Apple Lightning Audio Module 93

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

4.1 Overview 95
4.2 Firmware Updates 95
4.3 Authentication 95
4.4 Lightning Connector 95
4.5 Mechanical 95
4.6 Pad Layout and Assignments 98
4.7 Input/Output Electrical Characteristics 100
4.8 Cable 102
4.9 Power 103
4.9.1 Power States 103
4.9.2 Module Power 104
4.9.3 Accessory Power 104
4.9.4 Device Power 105
4.10 Audio 106
4.10.1 I2S 106
4.11 Control 108
4.11.1 S0-S2 Headset Remote Inputs 108
4.11.2 HID Headset Remote 108
4.11.3 Serial 108
4.12 Serial Protocol 109
4.12.1 Packet Format 109
4.12.2 Initialization Packets 110
4.12.3 Configuration Packets 110
4.12.4 Operation Packets 116
4.12.5 Examples 123
4.13 Test Procedures 125
4.13.1 Connectivity 126
4.13.2 Audio Quality 126
4.13.3 Power 126
4.13.4 Charge/Sync 126
4.13.5 Headset Jack 127
4.13.6 Firmware Update 127

5. Apple Lightning Connector 128
5.1 Comparison with the 30-pin Connector 128
5.1.1 Dimensions 128
5.1.2 Active Component 128
5.1.3 Reversability 129
5.1.4 Accessory Identify 129

Copyright © 2015 Apple Inc. All Rights Reserved.

5.1.5 Accessory Detect 129
5.1.6 Apple Device Detect 129
5.1.7 Analog Audio 129
5.1.8 Analog Video 129
5.1.9 DisplayPort Video 129
5.2 Connector Versions 129
5.3 Connector Pad/Pin Configuration 139
5.4 Connector Mechanical Requirements 145
5.4.1 All Accessories 145
5.4.2 Cable Accessories 147
5.4.3 Dock Accessories 158
5.4.4 Dongle Accessories 167
5.4.5 Form-Fitting Accessories 167
5.5 Connector Electrical Requirements 168
5.5.1 Connector Ground Pad/Pin Connection Requirements 168
5.5.2 Connector Shielding Requirements 168
5.5.3 Connector ESD Protection Requirements 168
5.5.4 Connector Signal Integrity Requirements 168
5.5.5 Lightning (C48)/Lightning (C68) Accessory Testing Requirements 169
5.6 Test Procedures 169
5.6.1 All Accessories 169
5.6.2 Lightning (C48)/Lightning (C68) Accessories 170
5.6.3 Mixed 30-pin/Lightning Connector Accessories 172
5.6.4 Cable Accessories 173
5.6.5 Dock Accessories 179
5.6.6 Dongle Accessories 184

6. Apple Lightning Receptacle 186
6.1 Lightning Receptacle Requirements 187
6.2 Lightning Receptacle Mechanical Requirements 188
6.2.1 Mounting 190
6.2.2 Receptacle Keepout Area 190
6.2.3 Receptacle Label 190
6.2.4 Receptacle Encapsulation 191
6.3 Lightning Receptacle Electrical Requirements 192
6.3.1 Identifying Power Supplies Using Resistor Networks 193
6.3.2 Pins and Assignments 194
6.4 Test Procedures 195
6.4.1 Static Receptacle Opening 196

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

6.4.2 Receptacle Mounting Under Insertion Force 197
6.4.3 Receptacle Mounting Under Extraction Force 197
6.4.4 Bend Test 197
6.4.5 Passthrough USB Signal Integrity 198
6.4.6 T37 USB Power Source Identification Tester for C37 201
6.4.7 Power Supply Resistor Network Detection Test 203

7. Apple Lightning Receptacle Controller 208
7.1 Mechanical 208
7.1.1 Physical Configuration 209
7.1.2 Recommended Handling 211
7.1.3 Recommended Operating Conditions 212
7.2 Electrical 212
7.2.1 Reference Circuit 213
7.2.2 Input/Output Electrical Characteristics 215

8. Apple Magnetic Charging Module 218
8.1 Magnetic Charging Module Requirements 218
8.2 Magnetic Charging Module Mechanical Requirements 219
8.3 Magnetic Charging Module Electrical Requirements 221
8.3.1 Power 221
8.3.2 Pins and Assignments 223
8.3.3 EMC 224
8.4 Factory Configuration 225
8.4.1 Set Vendor Name 226
8.4.2 Set Product Name 226
8.4.3 Set Model Number 227
8.4.4 Set Serial Number 227
8.4.5 Set USB Vendor ID 227
8.4.6 Set USB Product ID 228
8.4.7 Lock Configuration 228
8.4.8 Example Sequence 229
8.4.9 Verifying Configuration 229
8.5 USB Virtual COM Port 230
8.6 Test Procedures 230
8.6.1 Mechanical 230
8.6.2 Electrical 231

9. Apple 30-pin Connector 234
9.1 Connector Versions 234

Copyright © 2015 Apple Inc. All Rights Reserved.

9.1.1 Mid-Plane Plug 234
9.1.2 Mid-Plane Plug with Friction Latches 234
9.1.3 Mid-Plane Plug with Friction Latches, Power Only 235
9.1.4 Mid-Plane Plug (Top Shell) 235
9.1.5 Mid-Plane Plug (Bottom Shell) 235
9.1.6 Cable-End Plug 235
9.1.7 Cable-End Plug, Power Only 235
9.1.8 Cable-End Plug (Top Shell) 236
9.1.9 Cable-End Plug (Bottom Shell) 236
9.1.10 Vertical Dock Plug 236
9.1.11 Angled Dock Plug 236
9.1.12 Receptacle 236
9.2 Connector Dimensional Drawings 238

10. Apple Smart Connector 251
10.1 Reversibility 251
10.2 Smart Connector Pad Configuration 251
10.3 Mechanical Requirements 251
10.4 Electrical Requirements 252

11. Apple Smart Connector Module 253
11.1 Overview 254
11.2 Firmware Updates 254
11.3 Authentication 254
11.4 Smart Connector 255
11.5 Mechanical 255
11.6 Pad Layout and Assignments 257
11.7 Input/Output Electrical Characteristics 259

12. Accessory Authentication 261
12.1 Accessory Authentication Requirements 261
12.1.1 Accessory Authentication over iAP2 Requirements 261
12.2 Accessory Authentication Usage 262
12.2.1 Accessory Authentication over iAP2 Usage 262
12.3 Accessory Authentication Examples 263
12.3.1 Typical Accessory Authentication over iAP2 263
12.3.2 Accessory Authentication over iAP2 Failure Due To Invalid Certificate 264
12.3.3 Accessory Authentication over iAP2 Failure Due To Invalid Response 264

13. Accessory Identification 265

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

13.1 Accessory Identification Requirements 265
13.2 Accessory Identification Usage 265
13.2.1 Accessory Identification of Manufacturing Information 266
13.2.2 Accessory Identification of Sent/Received iAP2 Control Session Messages 267
13.2.3 Accessory Identification of Power Capabilities 267
13.2.4 Accessory Identification of iAP2 Transport Components 267
13.2.5 Additional Accessory Identification Parameters 268
13.3 Accessory Identification Examples 268
13.3.1 Typical Accessory Identification 268
13.3.2 Successful Accessory Identification With Two Tries 268
13.3.3 Unsuccessful Accessory Identification After Two Tries 269
13.3.4 Accessory Identification With Information Update 269
13.4 Test Procedures 269

14. AirPlay 271
14.1 Introduction 271
14.2 User requirements for using AirPlay 271
14.2.1 Hardware 271
14.2.2 Network 271
14.3 AirPlay Accessory Requirements 272
14.3.1 Accessory Features 272
14.3.2 Network Requirements 272
14.3.3 Implementation Requirements 273
14.3.4 Certification Requirements 273
14.4 AirPlay Setup Experience 273
14.4.1 Network configuration example for Wi-Fi networks 274
14.4.2 Network configuration example for wired networks 274
14.5 Power States and Accessory Availability 274
14.6 Bonjour 275
14.7 Web Interface 276
14.8 Firmware Update 276
14.9 AirPlay Accessory User Interface 277
14.9.1 Required Accessory user interface 277
14.9.2 Metadata Presentation 278
14.9.3 Transport controls 279
14.9.4 Shuffle and Repeat State Controls 280
14.9.5 Volume Controls 280
14.9.6 Accessories with multiple inputs 281
14.10 User Documentation 282

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.11 Special Considerations for Adapter Accessories 282
14.12 Accessory Compliance Test Plan for Audio Streaming Devices 283
14.12.1 Web Based Firmware Upgrade Test 287
14.12.2 Generic Firmware Upgrade Test 288
14.12.3 Link Layer Connection Auto Negotiation for Wired Network 288
14.12.4 Wi-Fi 288
14.12.5 Security Mode Verification 290
14.12.6 IPv4 Connectivity 293
14.12.7 IPv6 Connectivity 293
14.12.8 PHY: Wired Performance 294
14.12.9 PHY: Wireless Performance (100 ft) 294
14.12.10 Bonjour TXT Records Tests 295
14.12.11 Bonjour ADD and RMV via browsing 295
14.12.12 Bonjour TXT Record Lookup 296
14.12.13 Bonjour IP Lookup 297
14.12.14 Bonjour TXT Record Change based on Speaker Name Change 298
14.12.15 Bonjour TXT Record Change based on change Password Protection 299
14.12.16 Single Audio Stream (iTunes) Tests 301
14.12.17 Single Audio Stream (iOS) Tests 301
14.12.18 Single Audio Stream to Multiple Devices (iTunes) Tests 302
14.12.19 Devices with Multiple Inputs Tests 302
14.12.20 Metadata publishing Tests 309
14.12.21 AirPlay Password Tests 315
14.12.22 Device Commands (DACP) (iTunes) 320
14.12.23 Device Commands (DACP) (iOS) 325
14.12.24 Standby State Tests 329
14.12.25 Wake on LAN Tests 332
14.12.26 Wake on WLAN Tests 332
14.12.27 Web Configuration Interface 333
14.12.28 Audio Delay 336
14.12.29 Certification Procedure 337

15. App Launch 338
15.1 App Launch Requirements 338
15.2 App Launch Usage 339

16. App Match 340
16.1 App Match Requirements 340
16.1.1 External Accessory Protocol 341
16.1.2 Team ID 341

16.2 App Match Usage 341
16.2.1 External Accessory Protocol 341
16.2.2 Team ID 341

17. AssistiveTouch 342
17.1 AssistiveTouch Requirements 342
17.2 AssistiveTouch Usage 343

18. Bluetooth 344
18.1 Conformity With Bluetooth Specifications 344
18.1.1 Enhanced Data Rate 344
18.1.2 Adaptive Frequency Hopping 344
18.1.3 Sniff Mode for Low Power Consumption 344
18.1.4 Role and Topology Management 345
18.1.5 Extended Inquiry Response 346
18.1.6 Secure Simple Pairing 346
18.2 Profiles 347
18.2.1 Device ID Profile (DID) 347
18.2.2 Hands-Free Profile (HFP) 348
18.2.3 Message Access Profile (MAP) 350
18.2.4 Audio/Video Remote Control Profile (AVRCP) 350
18.2.5 Advanced Audio Distribution Profile (A2DP) 353
18.3 Audio Routing 354
18.3.1 Audio Data Received via HFP Profile 355
18.3.2 Audio Data Received via A2DP Profile 355
18.4 iAP2 357
18.5 Test Procedures 358
18.5.1 Pairing and Connection Establishment 358
18.5.2 Reconnection 358
18.5.3 Indicators 359
18.5.4 Outgoing Call 359
18.5.5 Incoming Call 360
18.5.6 FaceTime Audio 361
18.5.7 Three Way Call 361
18.5.8 Enhanced Call Control 362
18.5.9 Visual Voice Mail 362
18.5.10 Siri 363
18.5.11 A2DP 363
18.5.12 AVRCP 364
18.5.13 Turn-by-turn Navigation 365

Copyright © 2015 Apple Inc. All Rights Reserved.

18.5.14 Phonebook 365
18.5.15 iAP 366

19. Bluetooth Accessory Identification 370
19.1 HFP Command AT+XAPL 370

20. Bluetooth Connection 372
20.1 Bluetooth Connection Requirements 372
20.2 Bluetooth Connection Usage 373
20.3 Test Procedures 373
20.3.1 iAP2 Tests 373

21. Bluetooth Headset Battery Level Indication 374
21.1 HFP Command AT+IPHONEACCEV 374

22. Bluetooth Low Energy 375
22.1 Role 375
22.2 Advertising Channels 375
22.3 Advertising PDU 375
22.4 Advertising Data 375
22.5 Advertising Interval 376
22.6 Connection Parameters 377
22.7 Privacy 377
22.8 Permissions 377
22.9 Pairing 378
22.10 MTU Size 378
22.11 Services 378
22.11.1 Generic Access Profile Service 378
22.11.2 Generic Attribute Profile Service 378
22.11.3 Device Information Service 379
22.11.4 Available Services 379
22.12 GATT Server 379

23. CarPlay 381
23.1 Additional Specifications 381
23.2 General Requirements 383
23.2.1 Overview 383
23.2.2 Hardware Requirements 383
23.2.3 Architecture 386
23.2.4 CarPlay over USB 388

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.2.5 CarPlay over Wireless 395
23.2.6 Transitioning Between Wireless and USB 408
23.2.7 Software Clients 409
23.2.8 Media Types and Formats 411
23.3 CarPlay Communication Protocol 424
23.3.1 Discovery 424
23.3.2 Setup and Control 427
23.3.3 Resource Management 448
23.3.4 Modes 451
23.3.5 User Input 456
23.3.6 Siri User Input 475
23.3.7 Commands 477
23.3.8 CarPlay Communication Plug-in 485
23.4 Test Procedures 486

24. Cases 487
24.1 Product Design 487
24.1.1 Device Layouts and Dimensions for Apple Devices 487
24.1.2 Access to Controls 487
24.1.3 Access to the Headset Jack and 30-pin or Lightning Connector 488
24.1.4 Access to the Smart Connector 488
24.1.5 Device Protection 488
24.1.6 Cover Glass Contact 488
24.1.7 Dock Compatibility 489
24.1.8 Touchscreen 489
24.2 Acoustics 489
24.2.1 Speaker and Microphone Openings 489
24.2.2 Speaker to Microphone Coupling 490
24.2.3 Call Quality 490
24.3 Sensors 490
24.3.1 Ambient Light and Proximity Sensor Interference 490
24.3.2 Magnetic Interference 490
24.3.3 Touch ID Sensor 491
24.4 Camera 491
24.4.1 Geometry 491
24.4.2 Color 491
24.4.3 Surface Finish 492
24.4.4 Image Degradation Examples 492
24.5 Reliability 494

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.5.1 Device Insertion and Removal 494
24.5.2 Colorfastness 494
24.6 Environmental 494
24.7 RF 495
24.7.1 Materials and Coatings 495
24.7.2 Near Field Communication (NFC) 495
24.8 Touchscreen 495
24.8.1 Overlay 495
24.8.2 Edge Swipe Gestures 496
24.8.3 Edge Press Gestures 496
24.9 Test Procedures 496
24.9.1 Apple Device Compatibility 496
24.9.2 Apple Device Configurations 500
24.9.3 Product Design 501
24.9.4 RF (OTA) 504

25. Communications 517
25.1 Communications Requirements 517
25.2 Communications Usage 518
25.2.1 Call State Updates 519
25.2.2 Communications Updates 519
25.2.3 Call Controls 520
25.2.4 List Updates 520
25.2.5 Support for Apple devices running iOS 8.2 or older 521
25.3 Test Procedures 521
25.3.1 iAP2 Tests 521

26. Device Authentication 523
26.1 Device Authentication Requirements 523
26.2 Device Authentication Usage 523
26.3 Device Authentication Examples 524
26.3.1 Typical Device Authentication 524
26.3.2 Device Authentication Failure Due To Invalid Certificate 525
26.3.3 Device Authentication Failure Due To Invalid Response 525
26.4 Test Procedures 525
26.4.1 iAP2 525

27. Device Notifications 526
27.1 Device Notifications Requirements 526
27.2 Device Notifications Usage 526

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

28. Digital Audio 528
28.1 Digital Audio Requirements 528
28.1.1 Readiness for Audio Streaming 528
28.1.2 Multiple Audio Connections 528
28.1.3 Audio Input Source Switching 529
28.1.4 Copy Protection of Digital Audio Output 530
28.1.5 USB Host Mode Audio 530
28.1.6 USB Device Mode Audio 531
28.2 Digital Audio Usage 532
28.2.1 USB Host Mode Audio 532
28.2.2 USB Device Mode Audio Usage 532
28.3 Digital Audio Examples 532
28.3.1 Typical USB Device Mode Digital Audio 532
28.4 Test Procedures 533
28.4.1 Equipment Required 533
28.4.2 Test Cases 533

29. External Accessory Protocol 535
29.1 External Accessory Protocol Requirements 535
29.1.1 iAP2 EA Session Requirements 536
29.1.2 EA Native Transport (USB Host Mode) Requirements 536
29.2 External Accessory Protocol Usage 538
29.2.1 iAP2 EA Session Usage 538
29.2.2 EA Native Transport (USB Host Mode) Usage 538
29.3 External Accessory Protocol Examples 539
29.3.1 iAP2 EA Session Example 539
29.3.2 EA Native Transport (USB Host Mode) Example 540
29.4 Test Procedures 540
29.4.1 SDK Apps 540
29.4.2 External Accessory Protocol over iAP2 541
29.4.3 External Accessory Protocol over Native 542

30. Game Controller Module 543
30.1 Overview 544
30.2 Firmware Updates 544
30.3 Authentication 545
30.4 Bluetooth 545
30.5 Lightning Receptacle 545
30.6 Mechanical 545
30.7 Electrical 546

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

30.7.1 Battery 547
30.7.2 I/O 547
30.7.3 Pressure Sensitive Switches and Position Encoders 548
30.7.4 Joysticks 548

31. Headsets 549
31.1 Product Design 549
31.2 Audio and Data Interfaces 549
31.3 Remote Controls 550
31.4 Native Voice Recognition 550
31.5 Extension Cables and Adapters 550
31.6 Test Procedures 552
31.6.1 Product Design 552
31.6.2 Headset Jack 552
31.6.3 Lightning 552
31.6.4 Bluetooth 553
31.6.5 Controls 553
31.6.6 Native Voice Recognition 554

32. Headset Plug (3.5 mm) 555
32.1 Pin Assignments 556
32.2 Example Circuitry 557
32.3 Test Procedures 558

33. Headset Remote and Mic (3.5 mm) 559
33.1 Headset Remote and Mic Requirements 559
33.2 Headset Remote Transmitter Chip Usage 563
33.2.1 Transmitter Chip Pin Assignments and Physical Packaging 564
33.2.2 Transmitter Chip Maximum Voltage and Current Ratings 566
33.2.3 Transmitter Chip Thermal Impedance 566
33.2.4 Transmitter Chip Moisture Sensitivity 567
33.2.5 Transmitter Chip Electrical Characteristics 567
33.2.6 Transmitter Chip Theory of Operation 569
33.2.7 Transmitter Chip Button Mode 570
33.2.8 Transmitter Chip Tone Mode 571
33.3 Button Detection Circuitry Usage 574
33.3.1 Button Detection Circuitry Adjustments 576
33.4 Test Procedures 577
33.4.1 Product Design 577
33.4.2 Electrical 578

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

34. HID (Human Interface Device) 580
34.1 HID Requirements 580
34.1.1 HID Report Descriptor Requirements 581
34.1.2 HID over iAP2 Requirements 582
34.1.3 HID Native Transport (USB Host Mode) Requirements 582
34.1.4 HID Native Transport (Bluetooth) Requirements 582
34.2 HID Usage 583
34.2.1 HID over iAP2 Usage 583
34.2.2 Native Bluetooth HID Usage 583
34.3 Test Procedures 583
34.3.1 iAP2 583
34.3.2 HID over Native 584

35. HID Assistive Switch Control 585
35.1 HID Assistive Switch Control Requirements 585
35.2 HID Assistive Switch Control Usage 586
35.3 HID Assistive Switch Control Examples 586
35.3.1 Assistive Switch Control HID Report Descriptor 586
35.3.2 Assistive Switch Control Usage Example 587
35.4 Test Procedures 587
35.4.1 Tests Cases 587

36. HID AssistiveTouch Pointer 588
36.1 HID AssistiveTouch Pointer Requirements 588
36.2 HID AssistiveTouch Pointer Examples 589
36.2.1 AssistiveTouch HID Report Descriptor 589
36.2.2 AssistiveTouch Usage Example 590
36.3 Test Procedures 590
36.3.1 Test Cases 590

37. HID Game Controller 591
37.1 HID Game Controller Requirements 591
37.1.1 Gamepad 595
37.1.2 Switches and Position Encoders 597
37.1.3 Product Design 598
37.1.4 Menu Button 599
37.1.5 Face Button Group 599
37.1.6 Directional Pad 601
37.1.7 Shoulder Buttons 602
37.1.8 Joysticks 602

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

37.1.9 LED Array 603
37.1.10 Hold Switch 604
37.1.11 Reset Button 604
37.1.12 Bluetooth 605
37.1.13 Bluetooth Button 606
37.1.14 App Match for Controller-Enabled Games 606
37.2 HID Game Controller Examples 607
37.2.1 Gamepad Example HID Report Descriptor 607
37.3 Test Procedures 609
37.3.1 General Requirements 609
37.3.2 Bluetooth Controllers 609
37.3.3 Form-Fitting Controllers 610
37.3.4 Non Form-Fitting Controllers 610
37.3.5 Control Surface Layout and Labeling 611
37.3.6 Control Surfaces 614

38. HID Headset Remote 616
38.1 HID Headset Remote Requirements 616
38.2 HID Headset Remote Examples 617
38.2.1 Headset Remote Example HID Report Descriptor (Telephony) 617
38.2.2 Headset Remote Example HID Report Descriptor (Media Playback) 618
38.2.3 Headset Remote Example HID Report Descriptor (Telephony and Media Playback) 619

39. HID Keyboard 621
39.1 HID Keyboard Requirements 621
39.2 HID Keyboard Examples 627
39.2.1 Keyboard Example HID Report Descriptor 627
39.2.2 Keyboard Usage Example 629
39.3 Test Procedures 629
39.3.1 General Requirements 629

40. HID Media Playback Remote 631
40.1 HID Media Playback Remote Requirements 631
40.2 HID Media Playback Remote Examples 632
40.2.1 Media Playback Remote Example HID Report Descriptor 632
40.2.2 Media Playback Remote Example Usage 633
40.3 Test Procedures 634
40.3.1 Equipment Required 634
40.3.2 Tests Cases 634
40.3.3 iAP2 636

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

41. Location Information 637
41.1 Location Information Requirements 637
41.1.1 Global Navigation Satellite System (GNSS) Mode 638
41.1.2 Sensors Mode 639
41.1.3 NMEA Sentence Fields 639
41.2 Location Information Usage 645
41.3 Location Information Examples 646
41.3.1 GPGGA Sentence Example 646
41.3.2 GPRMC Sentence Example 646
41.3.3 GPGSV Sentence Example 646
41.3.4 GPHDT Sentence Example 646
41.3.5 PASCD Sentence Examples 647
41.3.6 PAGCD Sentence Examples 647
41.3.7 PAACD Sentence Example 648

42. Media Library Access 649
42.1 Media Library Access Requirements 649
42.1.1 Media Library Information Requirements 649
42.1.2 Media Library Updates Requirements 649
42.1.3 Media Library Playback Requirements 650
42.2 Media Library Access Usage 651
42.2.1 Media Library Information Usage 651
42.2.2 Media Library Updates Usage 651
42.2.3 Media Library Playback Usage 653
42.3 Test Procedures 653
42.3.1 Media Library Info 653
42.3.2 Media Library Updates 654
42.3.3 Media Library Playback 654

43. Musical Instrument Digital Interface (MIDI) 656

44. Now Playing Updates 657
44.1 Now Playing Updates Requirements 657
44.2 Now Playing Updates Usage 658
44.3 Test Procedures 658
44.3.1 Now Playing Updates 658

45. Power 660
45.1 Power Requirements 660
45.1.1 Neither Providing Power nor Drawing Power 661

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

45.1.2 Providing Power to the Apple Device 661
45.1.3 Drawing Power From the Apple Device 667
45.1.4 Multiple Power States 669
45.2 Power Usage 669
45.2.1 Providing Power to the Apple Device 669
45.2.2 Drawing Power from the Apple Device 670
45.3 Test Procedures 671
45.3.1 Accessory Power Sources 671
45.3.2 Power Rules over iAP2 675
45.3.3 Accessories That Draw Power 675
45.3.4 Battery Pack Power 678
45.3.5 Charging 679

46. Serial 680

47. Siri 682
47.1 Enabling Custom Siri Commands 682
47.2 Obtaining Siri Availability Information 682
47.2.1 Obtaining Status Information at Connection 682
47.2.2 Receiving Siri Availability Updates from the Apple Device 683
47.3 Initiating a Siri Session 683
47.3.1 Initiating a Session from the Accessory 684
47.3.2 Initiating a Session from the Apple Device 685
47.3.3 Ending a Session from the Accessory 685
47.4 Siri Eyes Free Mode 686
47.4.1 HFP Command AT+APLEFM 686
47.5 Improving Voice Recognition 687
47.5.1 Wide Band Speech Support 688
47.6 Optimizing the Siri Experience 688
47.7 Common Siri Applications 688
47.7.1 Initialization Procedure After Connection is Established 688
47.7.2 Phone Dialing Using Siri 689
47.7.3 Audio Routing and Media Playback Using Siri 689
47.7.4 Turn-By-Turn Directions Using Siri 690
47.8 User Interaction with Siri Eyes Free in a Vehicle 690
47.9 Enabling/Disabling Siri from the Apple Device 692
47.10 Test Procedures 693
47.10.1 Siri Eyes Free 693

48. USB 698

Copyright © 2015 Apple Inc. All Rights Reserved.

48.1 USB Host Mode Passthroughs 698
48.2 USB Signal Integrity 698
48.3 Test Procedures 698
48.3.1 Signal Integrity 698
48.3.2 High-Speed USB 699
48.3.3 Full-Speed USB 703

49. USB Device Mode 706
49.1 USB Embedded Host Implementation 706
49.2 Power 706
49.3 Enumeration 706
49.4 Connecting Multiple Apple Devices to a Mac 707
49.5 iAP2 Configuration 707
49.6 HID Interface 709
49.7 Test Procedures 711
49.7.1 iAP2 Audio Tests 711

50. USB Host Mode 712
50.1 iAP2 Interface Descriptor 713
50.2 iAP2 Data Transfers 713
50.3 iAP2 Performance Optimization 713

51. USB Role Switch 714
51.1 USB Role Switch Requirements 714
51.2 USB Role Switch Usage 714
51.3 Test Procedures 717
51.3.1 USB Role Switch 717

52. Vehicle Status 718
52.1 Vehicle Status Requirements 718
52.2 Vehicle Status Usage 718

53. VoiceOver 720
53.1 VoiceOver Requirements 720
53.2 VoiceOver Usage 721
53.3 Test Procedures 722
53.3.1 iAP2 Tests 722

54. Wi-Fi 723
54.1 Overview 723

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54.2 Test Procedures for the Consumers 723
54.2.1 Association Tests 723
54.2.2 Performance Tests 725
54.2.3 Other Scenarios 726
54.2.4 Secondary Tests 727
54.3 Test Procedures for the Enterprises 728
54.3.1 Association Tests 728
54.3.2 Performance Tests 730
54.3.3 Functional Tests 731
54.3.4 Additional Tests 735

55. Wi-Fi Accessory Configuration 737
55.1 Introduction 737
55.2 Hardware Requirements 737
55.3 Network Requirements 737
55.4 Wi-Fi required Accessory features 738
55.5 Wi-Fi Network requirements 738
55.6 Wi-Fi Implementation requirements 739
55.7 Wi-Fi Accessory behavioral requirements 739
55.8 Wi-Fi Certification requirements 740
55.9 Wi-Fi Accessory Configuration Setup Experience 740
55.10 Bonjour 742
55.11 Apple Device Information Element (IE) 743
55.11.1 General Usage 743
55.11.2 Structure 743
55.11.3 Payload 745
55.12 Accessory Compliance Test Plan 747
55.12.1 Wi-Fi Accessory Association Verification Tests 749
55.12.2 2.4 GHz vs 5 GHz Beaconing Tests 754
55.12.3 Security Mode Verification Tests for WPA2 Personal 755
55.12.4 IP Connectivity Tests 756
55.12.5 Bonjour TXT Records Tests for ADD and RMV 759
55.12.6 Certification Procedure 760

56. Wi-Fi Information Sharing 761
56.1 Wi-Fi Information Sharing Requirements 762
56.1.1 Apple Device to Accessory 762
56.1.2 Accessory to Apple Device 762
56.2 Wi-Fi Information Sharing Usage 762
56.2.1 Apple Device to Accessory 762

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

56.2.2 Accessory to Apple Device 763
56.3 Test Procedures 763
56.3.1 iAP2 Tests 763

57. iAP2 Link 764
57.1 Packet Structure 764
57.1.1 Start of Packet 765
57.1.2 Packet Length 765
57.1.3 Control Byte 765
57.1.4 Packet Sequence Number 766
57.1.5 Packet Acknowledgement Number 766
57.1.6 Session Identifier 767
57.1.7 Header Checksum 767
57.1.8 Payload Data 767
57.1.9 Payload Checksum 767
57.1.10 Checksum Calculation 768
57.2 Link Synchronization Payload 768
57.2.1 Link Version 769
57.2.2 Maximum Number of Outstanding Packets 769
57.2.3 Maximum Received Packet Length 769
57.2.4 Retransmission Timeout 770
57.2.5 Cumulative Acknowledgement Timeout 770
57.2.6 Maximum Number of Retransmissions 770
57.2.7 Maximum Cumulative Acknowledgements 770
57.2.8 iAP2 Sessions 771
57.3 iAP2 Session Payload 771
57.4 Extended Acknowledgement Payload 771
57.5 Reset 771
57.6 Sleep 772
57.7 Operation 772
57.7.1 Record 772
57.7.2 Initialization 773
57.7.3 Synchronization 773
57.7.4 Acknowledgements 776
57.7.5 Retransmissions 777
57.7.6 Flow Control 777
57.7.7 Reset 778
57.8 Examples 778
57.8.1 Typical Link Initialization 778

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

57.8.2 Connection Initialization When Device is Busy 779
57.8.3 Connection Requiring Multiple Negotiation Attempts 779
57.8.4 Connection With Failed Negotiation 780
57.8.5 Normal Connection Traffic 781
57.8.6 Device Reset of Transport Connection 782
57.8.7 Cumulative Ack Timeout Expired 782
57.8.8 Continuous Data Transmission with ACKs 783
57.8.9 Resend of missing packets using EAK 783
57.8.10 Receiving Packets Out Of Order 784
57.8.11 iAP2 Link Packet Structure Example 785
57.8.12 iAP2 Link Synchronization Payload Example 785
57.9 Test Procedures 786
57.9.1 Link Layer (v1) 786

58. iAP2 Sessions 788
58.1 Attributes 788
58.1.1 Type 788
58.1.2 Version 788
58.2 Control Session 789
58.2.1 Message Structure 789
58.2.2 Message Parsing 791
58.2.3 Parameter Types 791
58.2.4 Message Example 793
58.3 File Transfer Session 795
58.3.1 TransferIdentifier 795
58.3.2 Setup Datagram 796
58.3.3 Start 797
58.3.4 FirstData 797
58.3.5 FirstAndOnlyData 797
58.3.6 Data 798
58.3.7 LastData 798
58.3.8 Cancel 799
58.3.9 Pause 799
58.3.10 Success 799
58.3.11 Failure 800
58.3.12 Examples 800
58.4 External Accessory Session 802
58.4.1 Setup 803
58.4.2 ExternalAccessorySession Datagram 803

Copyright © 2015 Apple Inc. All Rights Reserved.

58.5 Test Procedures 803
58.5.1 Control Session 803

59. iAP2 Control Session Messages 805
59.1 Accessory Authentication 805
59.1.1 RequestAuthenticationCertificate 805
59.1.2 AuthenticationCertificate 805
59.1.3 RequestAuthenticationChallengeResponse 805
59.1.4 AuthenticationResponse 806
59.1.5 AuthenticationFailed 806
59.1.6 AuthenticationSucceeded 806
59.1.7 AccessoryAuthenticationSerialNumber 807
59.2 Accessory Identification 807
59.2.1 StartIdentification 807
59.2.2 IdentificationInformation 807
59.2.3 IdentificationAccepted 816
59.2.4 IdentificationRejected 817
59.2.5 CancelIdentification 819
59.2.6 IdentificationInformationUpdate 819
59.3 App Launch 819
59.3.1 RequestAppLaunch 820
59.4 AssistiveTouch 820
59.4.1 StartAssistiveTouch 820
59.4.2 StopAssistiveTouch 821
59.4.3 StartAssistiveTouchInformation 821
59.4.4 AssistiveTouchInformation 821
59.4.5 StopAssistiveTouchInformation 821
59.5 Bluetooth Connection 822
59.5.1 BluetoothComponentInformation 822
59.5.2 StartBluetoothConnectionUpdates 822
59.5.3 BluetoothConnectionUpdate 823
59.5.4 StopBluetoothConnectionUpdates 824
59.6 Communications 824
59.6.1 StartCallStateUpdates 824
59.6.2 CallStateUpdate 825
59.6.3 StopCallStateUpdates 827
59.6.4 StartCommunicationsUpdates 827
59.6.5 CommunicationsUpdate 828
59.6.6 StopCommunicationsUpdates 830

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.6.7 InitiateCall 831
59.6.8 AcceptCall 832
59.6.9 EndCall 832
59.6.10 SwapCalls 833
59.6.11 MergeCalls 833
59.6.12 HoldStatusUpdate 833
59.6.13 MuteStatusUpdate 834
59.6.14 SendDTMF 834
59.6.15 StartListUpdates 835
59.6.16 ListUpdate 836
59.6.17 StopListUpdates 838
59.7 Device Authentication 839
59.7.1 RequestDeviceAuthenticationCertificate 839
59.7.2 DeviceAuthenticationCertificate 839
59.7.3 RequestDeviceAuthenticationChallengeResponse 839
59.7.4 DeviceAuthenticationResponse 840
59.7.5 DeviceAuthenticationFailed 840
59.7.6 DeviceAuthenticationSucceeded 840
59.8 Device Notifications 840
59.8.1 DeviceInformationUpdate 841
59.8.2 DeviceLanguageUpdate 841
59.8.3 DeviceTimeUpdate 841
59.8.4 DeviceUUIDUpdate 842
59.8.5 WirelessCarPlayUpdate 842
59.9 External Accessory Protocol 843
59.9.1 StartExternalAccessoryProtocolSession 843
59.9.2 StopExternalAccessoryProtocolSession 843
59.9.3 StatusExternalAccessoryProtocolSession 843
59.10 Human Interface Device 844
59.10.1 StartHID 844
59.10.2 DeviceHIDReport 845
59.10.3 AccessoryHIDReport 845
59.10.4 StopHID 846
59.10.5 StartNativeHID 846
59.11 Location 846
59.11.1 StartLocationInformation 846
59.11.2 LocationInformation 847
59.11.3 StopLocationInformation 848
59.12 Media Library Access 848

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.12.1 StartMediaLibraryInformation 848
59.12.2 MediaLibraryInformation 849
59.12.3 StopMediaLibraryInformation 849
59.12.4 StartMediaLibraryUpdates 850
59.12.5 MediaLibraryUpdate 852
59.12.6 StopMediaLibraryUpdates 856
59.12.7 PlayMediaLibraryCurrentSelection 856
59.12.8 PlayMediaLibraryItems 856
59.12.9 PlayMediaLibraryCollection 857
59.12.10 PlayMediaLibrarySpecial 858
59.13 Now Playing 858
59.13.1 StartNowPlayingUpdates 858
59.13.2 NowPlayingUpdate 860
59.13.3 StopNowPlayingUpdates 863
59.13.4 SetNowPlayingInformation 863
59.14 Power 864
59.14.1 StartPowerUpdates 864
59.14.2 PowerUpdate 864
59.14.3 StopPowerUpdates 866
59.14.4 PowerSourceUpdate 866
59.15 USB Device Mode Audio 866
59.15.1 StartUSBDeviceModeAudio 866
59.15.2 USBDeviceModeAudioInformation 867
59.15.3 StopUSBDeviceModeAudio 867
59.16 Vehicle Status 867
59.16.1 StartVehicleStatusUpdates 868
59.16.2 VehicleStatusUpdate 868
59.16.3 StopVehicleStatusUpdates 868
59.17 VoiceOver 869
59.17.1 StartVoiceOver 869
59.17.2 StopVoiceOver 869
59.17.3 RequestVoiceOverMoveCursor 870
59.17.4 RequestVoiceOverActivateCursor 870
59.17.5 RequestVoiceOverScrollPage 870
59.17.6 RequestVoiceOverSpeakText 871
59.17.7 RequestVoiceOverPauseText 871
59.17.8 RequestVoiceOverResumeText 872
59.17.9 StartVoiceOverUpdates 872
59.17.10 VoiceOverUpdate 872

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.17.11 StopVoiceOverUpdates 873
59.17.12 RequestVoiceOverConfiguration 873
59.17.13 StartVoiceOverCursorUpdates 874
59.17.14 VoiceOverCursorUpdate 874
59.17.15 StopVoiceOverCursorUpdates 875
59.18 Wi-Fi Information Sharing 876
59.18.1 RequestWiFiInformation 876
59.18.2 WiFiInformation 876
59.18.3 RequestAccessoryWiFiConfigurationInformation 877
59.18.4 AccessoryWiFiConfigurationInformation 877

60. Accessory Test System 879

61. Device Dimensional Drawings 880
61.1 Apple Watch 38 mm 883
61.2 Apple Watch 42 mm 884
61.3 iPhone 6s Plus 885
61.4 iPhone 6s 886
61.5 iPhone 6 Plus 887
61.6 iPhone 6 888
61.7 iPhone 5s 889
61.8 iPhone 5c 890
61.9 iPhone 5 891
61.10 iPhone 4s 892
61.11 iPhone 4 (CDMA model) 893
61.12 iPhone 4 (GSM model) 894
61.13 iPhone 3G and iPhone 3GS 895
61.14 iPhone 896
61.15 iPad Pro Wi-Fi 897
61.16 iPad Pro Wi-Fi + Cellular 898
61.17 iPad mini 4 Wi-Fi 899
61.18 iPad mini 4 Wi-Fi + Cellular 900
61.19 iPad Air 2 Wi-Fi 901
61.20 iPad Air 2 Wi-Fi + Cellular 902
61.21 iPad mini 2 & 3 Wi-Fi 903
61.22 iPad mini 2 & 3 Wi-Fi + Cellular 904
61.23 iPad Air Wi-Fi 905
61.24 iPad Air Wi-Fi + Cellular 906
61.25 iPad mini with Wi-Fi 907
61.26 iPad mini with Wi-Fi + Cellular 908

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

61.27 iPad with Wi-Fi (4th generation) 909
61.28 iPad with Wi-Fi + Cellular (4th generation) 910
61.29 iPad with Wi-Fi (3rd generation) 911
61.30 iPad with Wi-Fi + 4G (3rd generation) 912
61.31 iPad 2 with Wi-Fi 913
61.32 iPad 2 with Wi-Fi + 3G 914
61.33 iPad with Wi-Fi 915
61.34 iPad with Wi-Fi + 3G 916
61.35 iPod touch (6th generation) 917
61.36 iPod touch (5th generation) 918
61.37 iPod touch (4th generation) 919
61.38 iPod touch (3rd generation) 920
61.39 iPod touch (2nd generation) 921
61.40 iPod touch 922
61.41 iPod nano (7th generation) 923
61.42 iPod nano (6th generation) 924
61.43 iPod nano (5th generation) 925
61.44 iPod nano (4th generation) 926
61.45 iPod nano (3rd generation) 927
61.46 iPod nano (2nd generation) 928
61.47 iPod nano 929
61.48 iPod classic 160GB 930
61.49 iPod classic 80GB 931
61.50 iPod (5th generation) 60GB/80GB 932
61.51 iPod (5th generation) 30GB 933
61.52 iPod (4th generation) 934
61.53 iPod (3rd generation) 935
61.54 iPod photo 30GB/60GB 936
61.55 iPod photo 937
61.56 iPod shuffle (4th generation) 938
61.57 iPod shuffle (3rd generation) 939
61.58 iPod shuffle (2nd generation) 940
61.59 iPod shuffle 941
61.60 iPod mini 943

62. Revision History 944

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figures and Tables

2. General Requirements and Recommendations 55

Figure 2-1 Mixed 30-pin and Lightning Connector USB Accessory 59
Figure 2-2 Mixed 30-pin and Lightning Connector USB Accessory with Multiplexer 59
Figure 2-3 Mixed 30-pin and Lightning Connector Serial Accessory 60
Figure 2-4 Mixed 30-pin and Lightning Connector Serial Accessory with Multiplexer 60

3. Apple Authentication Coprocessor 2.0C 66

Figure 3-1 Coprocessor pinout, top view 67
Figure 3-2 Coprocessor reference circuit diagram 68
Figure 3-3 Coprocessor I2C power on timing 70
Figure 3-4 Coprocessor I2C warm reset timing 71
Figure 3-5 Coprocessor I2C slave write address 71
Figure 3-6 Coprocessor I2C slave read address 71
Figure 3-7 Coprocessor Authentication Control and Status register, read-only bits 76
Figure 3-8 Coprocessor Authentication Control and Status register, write-only bits 77
Figure 3-9 Coprocessor Self-Test Control and Status register, write-only bits 80
Figure 3-10 Coprocessor Self-Test Control and Status register, read-only bits 80
Figure 3-11 Coprocessor package 84
Figure 3-12 Coprocessor Packing Carrier Tape 85
Figure 3-13 Coprocessor Packing Reel (1,000 unit) 86
Figure 3-14 Coprocessor Packing Reel (1,000 unit) details 87
Figure 3-15 Coprocessor Packing Reel (10,000 unit) 88
Figure 3-16 Coprocessor Packing Protective Band 89
Figure 3-17 Coprocessor typical I/O port input waveform 92
Table 3-1 Coprocessor signals 67
Table 3-2 Coprocessor address selection signals 67
Table 3-3 Coprocessor register map 72
Table 3-4 Coprocessor error codes 76
Table 3-5 Coprocessor Authentication ERR_SET values 77
Table 3-6 Coprocessor Authentication PROC_RESULTS values 77
Table 3-7 Coprocessor Authentication PROC_CONTROL values 78
Table 3-8 Coprocessor Self-Test PROC_CONTROL values 80
Table 3-9 Coprocessor Self-Test Results bits 80
Table 3-10 Coprocessor maximum electrical and temperature ranges 90

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 3-11 Coprocessor maximum electrical and temperature ranges 90
Table 3-12 Coprocessor I2C interface characteristics 90
Table 3-13 Coprocessor supply current into V_{CC}, excluding external current 91
Table 3-14 Coprocessor inputs 91
Table 3-15 Coprocessor outputs 91
Table 3-16 Coprocessor Values for typical I/O port input waveform 92

4. Apple Lightning Audio Module 93
Figure 4-1 LAM Accessory without Power 94
Figure 4-2 Lightning Audio Module Mechanical Package 97
Figure 4-3 Lightning Audio Module pickup offsets (bottom view) 98
Figure 4-4 Lightning Audio Module pad layout (top view) 99
Figure 4-5 I/O VREF RC circuit 102
Figure 4-6 LAM Accessory with Power 106
Figure 4-7 Lightning Audio Module I2S Format 107
Table 4-1 Lightning Audio Module pad assignments 99
Table 4-2 Lightning Audio Module I2S pad electrical characteristics 100
Table 4-3 Lightning Audio Module I2S electrical timing characteristics 101
Table 4-4 Lightning Audio Module I/O pad (except I2S) electrical characteristics 101
Table 4-5 Lightning Audio Module Serial RX electrical characteristics 102
Table 4-6 Lightning Audio Module serial protocol packet 109
Table 4-7 Lightning Audio Module accConfigurationInformation packet payload 111
Table 4-8 Lightning Audio Module accNameInformation packet payload 112
Table 4-9 Lightning Audio Module accManufacturerInformation packet payload 112
Table 4-10 Lightning Audio Module accModelInformation packet payload 112
Table 4-11 Lightning Audio Module accSerialNumberInformation packet payload 112
Table 4-12 Lightning Audio Module accPreferredAppBundleIdentifierInformation packet payload 113
Table 4-13 Lightning Audio Module accExternalAccessoryProtocolNameInformation packet payload 113
Table 4-14 Lightning Audio Module accHIDComponentInformationInformation packet payload 113
Table 4-15 Lightning Audio Module accAudioTerminalInformation packet payload 114
Table 4-16 Lightning Audio Module Audio Output Terminal Types 115
Table 4-17 Lightning Audio Module Audio Input Terminal Types 115
Table 4-18 Lightning Audio Module Audio Terminal Gain/Attenuation Step Size Values 115
Table 4-19 Preferred Apple Device Audio Processing Options 116
Table 4-20 Lightning Audio Module lamInformation packet payload 117
Table 4-21 Lightning Audio Module Device Power State Values 117
Table 4-22 Lightning Audio Module Audio Stream Signal Processing Mode Values 118
Table 4-23 Lightning Audio Module lamInformation Audio Sample Rate Values 118
Table 4-24 Lightning Audio Module lamAudioTerminalInformation packet payload 119

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 4-25 Lightning Audio Module lamExternalAccessoryProtocolSessionData packet payload 120
Table 4-26 Lightning Audio Module lamHIDReport packet payload 120
Table 4-27 Lightning Audio Module accExternalAccessoryProtocolSessionData packet payload 120
Table 4-28 Lightning Audio Module accHIDReport packet payload 120
Table 4-29 Lightning Audio Module accAudioTerminalStateInformation packet payload 121
Table 4-30 Lightning Audio Module accPassthroughPowerSourceInformation packet payload 122
Table 4-31 Lightning Audio Module lamVersionInformation packet payload 122

5. Apple Lightning Connector 128
Figure 5-1 Comparison of 30-pin connector and Lightning connector dimensions 128
Figure 5-2 Lightning (C10) 131
Figure 5-3 Lightning (C11) 131
Figure 5-4 Lightning (C12) 132
Figure 5-5 Lightning (C48) 132
Figure 5-6 Lightning (C68) 133
Figure 5-7 Lightning (C10) dimensions 134
Figure 5-8 Lightning (C11) dimensions 135
Figure 5-9 Lightning (C12) dimensions 136
Figure 5-10 Lightning (C48) dimensions 137
Figure 5-11 Lightning (C68) dimensions 138
Figure 5-12 C10/C11 Connector Pads 140
Figure 5-13 C12 Connector Pins 140
Figure 5-14 C48 Connector Pads 141
Figure 5-15 C68 Connector Pads 142
Figure 5-16 Connector critical areas 146
Figure 5-17 Connector plug engagement 146
Figure 5-18 Connector plug exposure 147
Figure 5-19 Connector cable enclosure 147
Figure 5-20 Connector cable enclosure with right angle 148
Figure 5-21 Allowable USB-A Lightning Cable Integrations 149
Figure 5-22 Allowable USB-B Lightning Cable Integrations 149
Figure 5-23 Example Non-USB Lightning Cable Integration 149
Figure 5-24 Cable EMC Diagram 150
Figure 5-25 Lightning (C48) Cable Encapsulation Diagram 152
Figure 5-26 Lightning (C48) Cable Shielding Diagram 152
Figure 5-27 Lightning (C48) Cable Shield Laser Welding Diagram 153
Figure 5-28 Lightning (C48) Recommended Cable Shield, Crimp 154
Figure 5-29 Lightning (C48) Recommended Cable Shield, No Crimp 155
Figure 5-30 Lightning (C68) Recommended Cable Shield, Crimp 156

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-31 Lightning (C68) Recommended Cable Shield, No Crimp 157
Figure 5-32 iPhone/iPod touch/iPod nano backstop 159
Figure 5-33 iPhone/iPod touch flexible mechanism 159
Figure 5-34 iPhone/iPod touch contact 160
Figure 5-35 iPhone/iPod touch sideways angle 160
Figure 5-36 iPad backstop 161
Figure 5-37 iPad flexible mechanism 162
Figure 5-38 iPod nano flexible mechanism 163
Figure 5-39 Lightning (C48) Dock Shielding Diagram 163
Figure 5-40 Lightning (C48) Dock Shield Laser Welding Diagram 164
Figure 5-41 Lightning (C48) Recommended Dock Shield Top 165
Figure 5-42 Lightning (C48) Recommended Dock Shield Bottom 166
Figure 5-43 Dongle Maximum Allowable Force 167
Figure 5-44 Identifying Lightning (C10), Lightning (C11), or Lightning (C12) connector configuration 169
Figure 5-45 Signal Integrity Test Setup for USB cables 173
Figure 5-46 Differential impedance graph with limit lines 174
Figure 5-47 Differential insertion loss graph with limit lines 175
Figure 5-48 DCR Ground in Parallel with Shield Conductor Test Setup ($R_{GND||SHIELD}$) 176
Figure 5-49 DCR VBUS Conductor Test Setup (R_{VBUS}) 176
Figure 5-50 DCR Ground Conductor Test Setup (R_{GND}) 176
Figure 5-51 Bend Test Setup for Lightning Cable Shield 178
Figure 5-52 Bend Test Setup for Lightning Cables 178
Figure 5-53 Tester Setup for Lightning Cables 179
Figure 5-54 180° Cable Bend Test Setup for Lightning Cables 179
Figure 5-55 360° Cable Bend Test Setup for Lightning Cables 179
Figure 5-56 Bend Test Clamp Region for Lightning Dock Shield 180
Figure 5-57 Bend Test Setup for Lightning Dock Shield 180
Figure 5-58 Fitting mechanism to enable testing of certain docks that limit device access 182
Figure 5-59 Dongle testing using a force gauge 185
Table 5-1 Lightning connector modules 130
Table 5-2 Lightning (C10) connector configurations 143
Table 5-3 Lightning (C11) connector configurations 143
Table 5-4 Lightning (C12) connector configurations 144
Table 5-5 Lightning (C48) connector configurations 144
Table 5-6 Lightning (C68) connector configurations 144

6. Apple Lightning Receptacle 186
Figure 6-1 Lightning Receptacle Top View 186
Figure 6-2 Lightning Receptacle Bottom View 187

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 6-3 Lightning Receptacle Dimensions 189
Figure 6-4 Lightning Receptacle Keepout 190
Figure 6-5 Lightning Receptacle Icon 191
Figure 6-6 Lightning Receptacle Bottom View with Pin Assignments 194
Figure 6-7 Lightning Receptacle Mechanical Test Plug 196
Figure 6-8 Passthrough USB Signal Integrity Test Environment 198
Figure 6-9 High-Speed Eye Diagram Template at the SMA test fixture 199
Figure 6-10 Full-Speed Eye Diagram Template at the SMA test fixture 201
Figure 6-11 T37 USB Power Source Identification Tester 203
Table 6-1 USB D+/D- resistor values for power-providing accessory connectors that do not implement iAP2 193
Table 6-2 Lightning Receptacle pins 195
Table 6-3 High-Speed Eye Diagram Template Specifications 199
Table 6-4 Lightning Receptacle Tester VBUS Output Voltages 201
Table 6-5 Lightning Receptacle Tester D+/- Output Voltages 201
Table 6-6 Test Results Matrix 204

7. Apple Lightning Receptacle Controller 208
Figure 7-1 Lightning Receptacle Controller package 209
Figure 7-2 Lightning Receptacle Controller packing tape and reel 210
Figure 7-3 Lightning Receptacle Controller packing tape 211
Figure 7-4 Lightning Receptacle Controller reference circuit 213
Table 7-1 Lightning Receptacle Controller package measurements 209
Table 7-2 Lightning Receptacle Controller packing tape dimensions 211
Table 7-3 Lightning Receptacle Controller pad assignments 214
Table 7-4 Input/Output Electrical Limiting Values 216
Table 7-5 Input/Output Electrical Operating Conditions 216
Table 7-6 Receptacle power and LDO AC/DC characteristics 216

8. Apple Magnetic Charging Module 218
Figure 8-1 Magnetic Charging Module 218
Figure 8-2 Magnetic Charging Module Dimensions 219
Figure 8-3 Magnetic Charging Module Charging Arm Clearance 220
Table 8-1 Magnetic Charging Module Pins 223

9. Apple 30-pin Connector 234
Figure 9-1 Mid-Plane Plug 238
Figure 9-2 Mid-Plane Plug, with Friction Latches 239
Figure 9-3 Mid-Plane Plug with Friction Latches, Power Only 240
Figure 9-4 Top Shell for Mid-Plane Plugs 241

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 9-5 Bottom Shell for Mid-Plane Plugs 242
Figure 9-6 Cable-End Plug 243
Figure 9-7 Cable-End Plug, Power Only 244
Figure 9-8 Top Shell for Cable End Plugs 245
Figure 9-9 Bottom Shell for Cable End Plugs 246
Figure 9-10 Vertical Dock Plug 248
Figure 9-11 Angled Dock Plug 249
Figure 9-12 Receptacle 250

10. Apple Smart Connector 251
Figure 10-1 Smart Connector Module Mounting Bracket 252

11. Apple Smart Connector Module 253
Figure 11-1 Smart Connector Module Accessory (Power Providing) Example 254
Figure 11-2 Smart Connector Module Mechanical Package 256
Figure 11-3 Smart Connector Module pickup offsets (bottom view) 257
Figure 11-4 Smart Connector Module pad layout (top view looking through module) 258
Table 11-1 Smart Connector Module pad assignments 258
Table 11-2 Smart Connector Module Input/Output Characteristics 259
Table 11-3 Smart Connector Module I/O and I2C pad voltage characteristics 259
Table 11-4 Smart Connector Module I/O and I2C pad current characteristics 260

14. AirPlay 271
Figure 14-1 Test bed 1 285
Figure 14-2 Test bed 2 285
Figure 14-3 Test bed 3 286
Figure 14-4 Test bed 4 286
Figure 14-5 Test bed 5 286
Figure 14-6 Test bed 6 287
Table 14-1 UI Status Indicators 277
Table 14-2 Suggested Text Indicators 278
Table 14-3 Metadata field 278
Table 14-4 Terms and Definitions 283
Table 14-5 Status Indicator 291

15. App Launch 338
Figure 15-1 App Launch Alert 338

16. App Match 340
Figure 16-1 App Match Alert 340

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

17. AssistiveTouch 342
Figure 17-1 AssistiveTouch Pointer 342

18. Bluetooth 344
Figure 18-1 Initiate Audio Playback (e.g. music) 356
Figure 18-2 Initiate System Sound (e.g. turn-by-turn directions) 356
Figure 18-3 RF Interference Test Setup 367
Table 18-1 SubBand Codec Information Elements for Apple products 353
Table 18-2 MPEG-2/4 AAC Codec Information Elements for Apple devices 353
Table 18-3 AAC audio packet for Apple devices 354

23. CarPlay 381
Figure 23-1 Topology of an automotive head unit using CarPlay 387
Figure 23-2 Process for establishing a CarPlay session 389
Figure 23-3 Initial Connection and Pairing using Bluetooth and iAP2 404
Figure 23-4 Reconnect using Bluetooth 405
Figure 23-5 Telephony Audio Round-Trip Path 416
Figure 23-6 Audio Tolerance Mask For Send Frequency Response (Sensitivity (dB) vs. Frequency (Hz)) 419
Figure 23-7 Event timeline for audio ducking/unducking 424
Figure 23-8 Mixing CarPlay audio 449
Figure 23-9 Touchscreen 457
Figure 23-10 Touchpad 458
Figure 23-11 Example Input Report Layout for Single-Touch Screen 461
Figure 23-12 Example Input Report Layout for Single-Touch Touchpad with Basic Character Gesture Recognition 464
Figure 23-13 Example Input Report Layout for Single-Touch Touchpad with Character Gesture Recognition with Alternate Interpretations 468
Figure 23-14 Multi-axis Controller 469
Figure 23-15 Example Input Report Layout for Multi-Axis Controller 471
Figure 23-16 Example Input Report Layout for Simple Media Buttons 474
Figure 23-17 Example Input Report Layout for Simple Telephone Buttons 475
Figure 23-18 CarPlay Communication Plug-in Reference Architecture 486
Table 23-1 USB NCM Control Interface Descriptor 390
Table 23-2 USB NCM Communication Interface Descriptor Requirements 390
Table 23-3 USB NCM Data Interface Descriptor 391
Table 23-4 Operational Channels for 2.4 GHz Wi-Fi 398
Table 23-5 Operational Channels for 5 GHz Wi-Fi 398
Table 23-6 Expected Throughput for IEEE 802.11 MAC/PHY Modes 401
Table 23-7 Features flag settings 402
Table 23-8 Operational Channels for Cellular Coexistence 407

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-9 CarPlay H.264 Video Stream Standard Resolutions 412
Table 23-10 CarPlay H.264 Levels 412
Table 23-11 Entertainment Stream Requirements 415
Table 23-12 Telephony Audio Stream Requirements 415
Table 23-13 FaceTime Audio Stream Requirements 417
Table 23-14 Audio Tolerance Mask Limits For Send Frequency Response 418
Table 23-15 Siri Stream Requirements 419
Table 23-16 Alert Stream Requirements 420
Table 23-17 Default Audio Stream Requirements 421
Table 23-18 Main High Audio Stream Requirements 422
Table 23-19 Alternate Audio Stream Requirements 422
Table 23-20 CarPlay Control Bonjour Service Keys 425
Table 23-21 CarPlay Control Commands 426
Table 23-22 CarPlay Bonjour Service Keys 427
Table 23-23 Status flags bitfield enum values 427
Table 23-24 Encryption type enum values 430
Table 23-25 Info Message request keys 431
Table 23-26 Info Message response keys 431
Table 23-27 Audio format keys 435
Table 23-28 Audio formats bitfield enum values 435
Table 23-29 Audio latencies keys 437
Table 23-30 Display keys 437
Table 23-31 Display features bitfield enum values 438
Table 23-32 Primary input device enum values 438
Table 23-33 Extended features string values 439
Table 23-34 Limited UI elements string values 439
Table 23-35 Icon keys 440
Table 23-36 Initial Setup request keys 441
Table 23-37 Initial Setup response keys 441
Table 23-38 Setup request keys 442
Table 23-39 Setup response keys 442
Table 23-40 Main/alternate audio stream descriptors request keys 443
Table 23-41 Vocoder Information keys 443
Table 23-42 Main/alternate audio stream descriptors response keys 443
Table 23-43 Screen stream descriptors request keys 444
Table 23-44 Screen stream descriptors response keys 444
Table 23-45 Stream ID enum values 444
Table 23-46 Audio type string values 445
Table 23-47 Channels 445

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-48 Feedback response keys 446
Table 23-49 Feedback stream keys 446
Table 23-50 URLs for CarPlay 446
Table 23-51 Entity enum values 452
Table 23-52 Resource ID enum values 452
Table 23-53 Resource constraint enum values 452
Table 23-54 Resource ownership transfer type enum values 452
Table 23-55 Resource transfer priority enum values 453
Table 23-56 App state enum values 453
Table 23-57 Speech mode enum values 453
Table 23-58 HID device description keys 456
Table 23-59 Digitizer Support HID Usages 458
Table 23-60 Character Input Gesture Support HID Usages 461
Table 23-61 Knob Support HID Usages 469
Table 23-62 Button Support HID Usages 471
Table 23-63 Siri actions enum values 475
Table 23-64 duckAudio request keys 478
Table 23-65 unduckAudio request keys 478
Table 23-66 disableBluetooth request keys 479
Table 23-67 changeModes request keys 479
Table 23-68 appState keys 479
Table 23-69 resource keys 479
Table 23-70 changeModes response keys 480
Table 23-71 modesChanged request keys 480
Table 23-72 appState keys 481
Table 23-73 resource keys 481
Table 23-74 hidSendReport request keys 481
Table 23-75 hidSetInputMode request keys 482
Table 23-76 HID input modes enum values 482
Table 23-77 requestSiri request keys 483
Table 23-78 Accessory to Apple device requestUI request keys 483
Table 23-79 Apple device to accessory requestUI request keys 484
Table 23-80 setNightMode request keys 484
Table 23-81 setLimitedUI request keys 484
Table 23-82 iAPSendMessage request keys 485

24. Cases 487
Figure 24-1 Touchscreen Keep-out Area 489
Figure 24-2 Image degradation by color shifting 492

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 24-3 Image degradation by decrease of contrast 493
Figure 24-4 Image degradation by blocking 493
Figure 24-5 Image degradation by flash interference 493
Figure 24-6 Apple Device Proudness Test 503
Figure 24-7 Apple Device Gap Test 503
Figure 24-8 Apple Device Touchscreen Keep-out Test 504
Figure 24-9 Apple Device Setup 505
Figure 24-10 Generic Setup for TRP/TIS(Free Space) 505
Figure 24-11 iPad with Post-it Notes 506
Table 24-1 Required device configurations for accessory case test procedures 500
Table 24-2 EIRP/EIS Measurement Matrix for case OTA testing procedure 507
Table 24-3 TRP/TIS Measurement Matrix for case OTA testing procedure 508
Table 24-4 Required Apple device models for testing of specific OTA bands 510

28. Digital Audio 528
Table 28-1 Audio Transport Connection Actions 529

29. External Accessory Protocol 535
Table 29-1 USB Interface Descriptor (Alternate Setting 1 - Active Session) for External Accessory Native Transport (USB Host Mode) 537
Table 29-2 USB Interface Descriptor (Alternate Setting 0 - Zero Bandwidth) for External Accessory Native Transport (USB Host Mode) 537

30. Game Controller Module 543
Figure 30-1 Game Controller Module 543
Figure 30-2 Game Controller Module Dimensions 546

31. Headsets 549
Figure 31-1 Double Mono Airline Adapter 551
Figure 31-2 3.5 mm to 6.35 mm (1/4 in) Adapter 551
Figure 31-3 Example Headset Extension Cable 552

32. Headset Plug (3.5 mm) 555
Figure 32-1 Headset Plug Dimensional and Flange Details 555
Figure 32-2 Headset Plug Cable Shell Dimensions 555
Figure 32-3 Typical circuitry in a headset accessory 557
Table 32-1 Headset Plug Pin Assignments 556
Table 32-2 Recommended component values for typical circuitry in a headset accessory 557

33. Headset Remote and Mic (3.5 mm) 559

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 33-1 Headset Remote and Mic Example A 560
Figure 33-2 Headset Remote and Mic Example B 561
Figure 33-3 Headset Remote and Mic Example C 562
Figure 33-4 Headset Remote and Mic Example D 563
Figure 33-5 Transmitter Chip Package 565
Figure 33-6 Transmitter Chip Block Diagram 570
Figure 33-7 Transmitter Chip Startup Timing 572
Figure 33-8 Tone Mode ACK Sequence 573
Figure 33-9 Tone Transmit/Decode Method 573
Figure 33-10 Transmitter Circuit 575
Figure 33-11 Microphone Circuit 575
Figure 33-12 Headsets Looped Around Ear Test Setup 578
Figure 33-13 Headset Remote and Mic Test Setup 579
Table 33-1 Transmitter Chip Part Numbers 564
Table 33-2 Transmitter Chip Pin Assignments 564
Table 33-3 Transmitter Chip Package Dimensions 565
Table 33-4 Transmitter Chip Maximum Voltage and Current Ratings 566
Table 33-5 Transmitter Chip Electrical Characteristics (General) 567
Table 33-6 Transmitter Chip Electrical Characteristics (Tone Mode) 568
Table 33-7 Transmitter Chip Electrical Characteristics (Button Mode) 569
Table 33-8 Transmitter Chip DETECT Pin Voltages 570
Table 33-9 Transmitter Circuit Components 576
Table 33-10 Approved transmitter circuit MEMS digital microphone components 577
Table 33-11 Headset Remote and Mic Expected mic_bias Voltages in Button Mode 579

35. HID Assistive Switch Control 585
Figure 35-1 HID Assistive Switch Control 585

37. HID Game Controller 591
Figure 37-1 Form-Fitting Gamepad Sample 592
Figure 37-2 Non Form-Fitting Gamepad Sample 593
Figure 37-3 iPad mini Form-Fitting Gamepad Sample 594
Figure 37-4 HID Gamepad Control Layout 597
Figure 37-5 HID game controller menu button 599
Figure 37-6 HID game controller face button group layout 600
Figure 37-7 HID game controller face button group layout (alternate) 601
Figure 37-8 HID gamepad directional pad layout 602
Figure 37-9 HID game controller joystick 603
Table 37-1 HID Game Controller pressure sensitive switch mechanical requirements 598
Table 37-2 HID Game Controller face button group color definitions 600

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 37-3 HID game controller LED states 604
Table 37-4 HID Game Controller App Match Parameters 606
Table 37-5 Required Game Controller Control Surfaces 611

38. HID Headset Remote 616
Table 38-1 HID Consumer Page controls for use by headset remotes 616
Table 38-2 HID Telephony Page controls for use by headset remotes 617

39. HID Keyboard 621
Table 39-1 Required HID Keyboard/Keypad Page controls for use by keyboard components 622
Table 39-2 Required HID Keyboard/Keypad Page controls for use by JIS keyboard components 625
Table 39-3 Optional HID Keyboard/Keypad Page controls for use by keyboard components 625
Table 39-4 HID Consumer Page controls for use by keyboard components 626

40. HID Media Playback Remote 631
Table 40-1 HID Consumer Page controls for use by media playback remote components 631

41. Location Information 637
Table 41-1 Supported NMEA Sentences and Maximum/Recommended Rates 637
Table 41-2 GPGGA Sentence Fields 640
Table 41-3 GPRMC Sentence Fields 640
Table 41-4 GPGSV Sentence Fields 641
Table 41-5 GPHDT Sentence Fields 642
Table 41-6 PASCD Sentence Fields 643
Table 41-7 PAGCD Sentence Fields 644
Table 41-8 PAACD Sentence Fields 644

45. Power 660
Figure 45-1 Typical AC adapter diode bridge circuit 663
Figure 45-2 USB D+/D- resistor networks for power-providing accessory connectors that do not implement iAP2 665
Table 45-1 Typical component values for an AC adapter diode bridge circuit 663
Table 45-2 USB cable maximum DC resistances 665
Table 45-3 USB D+/D- resistor values for power-providing accessory connectors that do not implement iAP2 666
Table 45-4 USB Vendor Request for accessory USB Embedded Host connector that does not implement iAP2 to communicate available power 666
Table 45-5 Maximum allowable Low Power Mode current draw 668
Table 45-6 USB Events that permit Intermittent High Power Mode 670
Table 45-7 USB Events that exit Intermittent High Power Mode 671

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 45-8 Power Supply Voltage Outputs for Lightning Accessories 671
Table 45-9 Power Supply Voltage Outputs for non-Lightning Accessories 672
Table 45-10 Maximum allowable Low Power Mode current draw 678

46. Serial 680
Table 46-1 Serial transport mark and space levels for Apple devices 680

47. Siri 682
Figure 47-1 Siri is Disabled/Enabled from the Apple Device's Settings 683
Figure 47-2 Initiating a Siri Session from the Accessory 684
Figure 47-3 Initiating a Siri Session from the Apple Device 685
Figure 47-4 Ending a Siri Session from the Accessory 686
Figure 47-5 Siri Initialization Procedure 689
Figure 47-6 Siri Initialization Procedure with Siri Eyes Free 689
Figure 47-7 Siri Eyes Free User Interaction 691
Figure 47-8 Siri is Deactivated - Launching Voice Control 692
Figure 47-9 Siri is Deactivated - Displaying a Warning Message 692

48. USB 698
Figure 48-1 High-Speed Test Environment 699
Figure 48-2 USB Test Points 700
Figure 48-3 Eye Diagram Template at TP2 700
Figure 48-4 TP2 Eye Diagram Mask 703
Figure 48-5 Full-Speed Test Environment 704
Table 48-1 Eye Diagram Template Specifications 701
Table 48-2 Pass/Fail Criteria for Host Under Test 701
Table 48-3 Test Packet Layout 701
Table 48-4 Full-Speed Signal Quality Requirements 704

49. USB Device Mode 706
Figure 49-1 USB Device Mode Interface Descriptor 708
Figure 49-2 USB Device Mode Interface HID Report 709
Figure 49-3 USB Device Mode Interface Report Packing 710
Table 49-1 Link control byte usage 710

50. USB Host Mode 712
Table 50-1 USB Host Mode iAP2 interface descriptor 713

51. USB Role Switch 714

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 51-1 USB Role Switch Custom Vendor Request (Apple device is USB Device, Accessory is USB Host) 715
Table 51-1 USB Vendor Request for Apple Device to Get Supported Capabilities 715
Table 51-2 USB Vendor Request for Apple Device to Host Mode Switch: wValues (bitfield) 716
Table 51-3 USB Vendor Request for Apple Device to Host Mode Switch 716

55. Wi-Fi Accessory Configuration 737
Figure 55-1 Test bed 748
Table 55-1 Configuration TLVs 741
Table 55-2 mfi config tcp TXT record keys 742
Table 55-3 MFi Configuration Feature Flags 743
Table 55-4 MFi Configuration Status Flag 743
Table 55-5 Apple Device IE overall structure 744
Table 55-6 Apple Device IE element structure 744
Table 55-7 Apple Device IE elements 745
Table 55-8 Flags 746
Table 55-9 Terms And Definitions 747

56. Wi-Fi Information Sharing 761
Figure 56-1 Wi-Fi Information Sharing Alert 761

57. iAP2 Link 764
Table 57-1 iAP2 Link Packet Structure 764
Table 57-2 iAP2 Link Control Byte Bits 766
Table 57-3 Link Synchronization Payload (Version 1) 768
Table 57-4 EAK Packet Payload (Link v1) 771
Table 57-5 iAP2 Link Operation Record Variables 772
Table 57-6 Default link parameters during synchronization 774
Table 57-7 Suggested link parameters for USB Host Mode transport (Full Speed) 775
Table 57-8 Suggested link parameters for USB Device Mode transport (Full Speed) 775
Table 57-9 Suggested link parameters for Bluetooth transport 775
Table 57-10 Suggested link parameters for 57.6 kbps serial transport 776
Table 57-11 iAP2 Link Packet Structure Example - Accessory SYN Packet 785
Table 57-12 Link Synchronization Payload Example 785

58. iAP2 Sessions 788
Figure 58-1 Control Session Message Structure 790
Figure 58-2 Control Session Message Parameter Structure 791
Figure 58-3 Group Parameter Structure 793
Figure 58-4 StartExternalAccessoryProtocolSession Control Session Message Example 794

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 58-5 ExternalAccessoryProtocolIdentifier Parameter 794
Figure 58-6 ExternalAccessoryProtocolSessionIdentifier Parameter 794
Figure 58-7 StartNowPlayingUpdates Control Session Message Example 795
Figure 58-8 PlaybackAttributes Parameter Group 795
Table 58-1 iAP2 Session Types 788
Table 58-2 File Transfer Session Setup Datagram 796
Table 58-3 File Transfer Session Start Datagram 797
Table 58-4 File Transfer Session FirstData Datagram 797
Table 58-5 File Transfer Session FirstAndOnlyData Datagram 798
Table 58-6 File Transfer Session Data Datagram 798
Table 58-7 File Transfer Session LastData Datagram 799
Table 58-8 File Transfer Session Cancel Datagram 799
Table 58-9 File Transfer Session Pause Datagram 799
Table 58-10 File Transfer Session Success Datagram 800
Table 58-11 File Transfer Session Failure Datagram 800
Table 58-12 ExternalAccessorySession Datagram 803

59. iAP2 Control Session Messages 805
Table 59-1 `RequestAuthenticationCertificate` message parameters 805
Table 59-2 `AuthenticationCertificate` message parameters 805
Table 59-3 `RequestAuthenticationChallengeResponse` message parameters 806
Table 59-4 `AuthenticationResponse` message parameters 806
Table 59-5 `AuthenticationFailed` message parameters 806
Table 59-6 `AuthenticationResponseSucceeded` message parameters 806
Table 59-7 `AccessoryAuthenticationSerialNumber` message parameters 807
Table 59-8 `StartIdentification` message parameters 807
Table 59-9 `IdentificationInformation` message parameters 808
Table 59-10 `PowerProvidingCapability` enum 810
Table 59-11 `ExternalAccessoryProtocol` parameter group 810
Table 59-12 `MatchAction` enum 810
Table 59-13 `USBDeviceTransportComponent` parameter group 811
Table 59-14 `USBDeviceModeAudioSampleRate` enum 811
Table 59-15 `USBHostTransportComponent` parameter group 812
Table 59-16 `SerialTransportComponent` parameter group 812
Table 59-17 `BluetoothTransportComponent` parameter group 812
Table 59-18 `iAP2HIDComponent` parameter group 813
Table 59-19 `VehicleInformationComponent` parameter group 813
Table 59-20 `EngineTypes` enum 813
Table 59-21 `VehicleStatusComponent` parameter group 813

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-22 LocationInformationComponent parameter group 814
Table 59-23 USBHostHIDComponent parameter group 815
Table 59-24 HIDComponentFunction enum 815
Table 59-25 WirelessCarPlayTransportComponent parameter group 816
Table 59-26 BluetoothHIDComponent parameter group 816
Table 59-27 IdentificationAccepted message parameters 816
Table 59-28 IdentificationRejected message parameters 817
Table 59-29 CancelIdentification message parameters 819
Table 59-30 IdentificationInformationUpdate message parameters 819
Table 59-31 RequestAppLaunch message parameters 820
Table 59-32 AppLaunchMethod enum 820
Table 59-33 StartAssistiveTouch message parameters 820
Table 59-34 StopAssistiveTouch message parameters 821
Table 59-35 StartAssistiveTouchInformation message parameters 821
Table 59-36 AssistiveTouchInformation message parameters 821
Table 59-37 StopAssistiveTouchInformation message parameters 822
Table 59-38 BluetoothComponentInformation message parameters 822
Table 59-39 BluetoothComponentStatus parameter group 822
Table 59-40 StartBluetoothConnectionUpdates message parameters 823
Table 59-41 BluetoothConnectionUpdate message parameters 823
Table 59-42 BluetoothComponentProfiles parameter group 823
Table 59-43 StopBluetoothConnectionUpdates message parameters 824
Table 59-44 StartCallStateUpdates message parameters 824
Table 59-45 CallStateUpdate message parameters 825
Table 59-46 CallStateUpdateStatus enum 826
Table 59-47 CallStateUpdateDirection enum 826
Table 59-48 CallStateUpdateService enum 826
Table 59-49 CallStateUpdateDisconnectReason enum 826
Table 59-50 CallStateUpdateStatusLegacy enum 827
Table 59-51 CallStateUpdateDirectionLegacy enum 827
Table 59-52 StopCallStateUpdates message parameters 827
Table 59-53 StartCommunicationsUpdates message parameters 828
Table 59-54 CommunicationsUpdate message parameters 829
Table 59-55 CommunicationsUpdateSignalStrength enum 829
Table 59-56 CommunicationsUpdateRegistrationStatus enum 830
Table 59-57 StopCommunicationsUpdates message parameters 830
Table 59-58 InitiateCall message parameters 831
Table 59-59 InitiateCallType enum 831
Table 59-60 InitiateCallService enum 831

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-61 AcceptCall message parameters 832
Table 59-62 AcceptCallAcceptAction enum 832
Table 59-63 EndCall message parameters 832
Table 59-64 EndCallEndAction enum 832
Table 59-65 SwapCalls message parameters 833
Table 59-66 MergeCalls message parameters 833
Table 59-67 HoldStatusUpdate message parameters 833
Table 59-68 MuteStatusUpdate message parameters 834
Table 59-69 SendDTMF message parameters 834
Table 59-70 SendDTMFTone enum 834
Table 59-71 StartListUpdates message parameters 835
Table 59-72 RecentsListProperties parameter group 835
Table 59-73 FavoritesListProperties parameter group 836
Table 59-74 ListUpdate message parameters 836
Table 59-75 RecentsList parameter group 837
Table 59-76 FavoritesList parameter group 837
Table 59-77 ListUpdateService enum 838
Table 59-78 ListUpdateRecentsListType enum 838
Table 59-79 StopListUpdates message parameters 838
Table 59-80 RequestDeviceAuthenticationCertificate message parameters 839
Table 59-81 DeviceAuthenticationCertificate message parameters 839
Table 59-82 RequestDeviceAuthenticationChallengeResponse message parameters 839
Table 59-83 DeviceAuthenticationResponse message parameters 840
Table 59-84 DeviceAuthenticationFailed message parameters 840
Table 59-85 DeviceAuthenticationResponseSucceeded message parameters 840
Table 59-86 DeviceInformationUpdate message parameters 841
Table 59-87 DeviceLanguageUpdate message parameters 841
Table 59-88 DeviceTimeUpdate message parameters 841
Table 59-89 DeviceUUIDUpdate message parameters 842
Table 59-90 WirelessCarPlayUpdate message parameters 842
Table 59-91 WirelessCarPlayUpdateStatus enum 842
Table 59-92 StartExternalAccessoryProtocolSession message parameters 843
Table 59-93 StopExternalAccessoryProtocolSession message parameters 843
Table 59-94 StatusExternalAccessoryProtocolSession message parameters 843
Table 59-95 ExternalAccessoryProtocolSessionStatus enum 844
Table 59-96 StartHID message parameters 844
Table 59-97 DeviceHIDReport message parameters 845
Table 59-98 AccessoryHIDReport message parameters 845
Table 59-99 StopHID message parameters 846

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-100 StartNativeHID message parameters 846
Table 59-101 StartLocationInformation message parameters 847
Table 59-102 LocationInformation message parameters 848
Table 59-103 StopLocationInformation message parameters 848
Table 59-104 StartMediaLibraryInformation message parameters 848
Table 59-105 MediaLibraryInformation message parameters 849
Table 59-106 MediaLibraryInformation parameter group 849
Table 59-107 MediaLibraryType enum 849
Table 59-108 StopMediaLibraryInformation message parameters 849
Table 59-109 StartMediaLibraryUpdates message parameters 850
Table 59-110 MediaItemProperties parameter group 850
Table 59-111 MediaPlaylistProperties parameter group 851
Table 59-112 MediaLibraryUpdate message parameters 852
Table 59-113 MediaItem parameter group 853
Table 59-114 MediaPlaylist parameter group 855
Table 59-115 MediaType enum 855
Table 59-116 StopMediaLibraryUpdate message parameters 856
Table 59-117 PlayMediaLibraryCurrentSelection message parameters 856
Table 59-118 PlayMediaLibraryItems message parameters 856
Table 59-119 PlayMediaLibraryCollection message parameters 857
Table 59-120 MediaLibraryCollectionType enum 857
Table 59-121 PlayMediaLibrarySpecial message parameters 858
Table 59-122 StartNowPlayingUpdates message parameters 858
Table 59-123 StartNowPlayingMediaItemAttributes parameter group 859
Table 59-124 StartNowPlayingPlaybackAttributes parameter group 860
Table 59-125 NowPlayingUpdate message parameters 861
Table 59-126 PlaybackAttributes parameter group 861
Table 59-127 PlaybackStatus enum 862
Table 59-128 PlaybackShuffle enum 862
Table 59-129 PlaybackRepeat enum 863
Table 59-130 StopNowPlayingUpdates message parameters 863
Table 59-131 SetNowPlayingInformation message parameters 863
Table 59-132 StartPowerUpdates message parameters 864
Table 59-133 PowerUpdate message parameters 865
Table 59-134 AccessoryPowerModes enum 865
Table 59-135 BatteryChargingState enum 865
Table 59-136 StopPowerUpdates message parameters 866
Table 59-137 PowerSourceUpdate message parameters 866
Table 59-138 StartUSBDeviceModeAudio message parameters 867

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-139 `USBDeviceModeAudioInformation` message parameters 867
Table 59-140 `StopUSBDeviceModeAudio` message parameters 867
Table 59-141 `StartVehicleStatusUpdates` message parameters 868
Table 59-142 `VehicleStatusUpdate` message parameters 868
Table 59-143 `StopVehicleStatusUpdates` message parameters 869
Table 59-144 `StartVoiceOver` message parameters 869
Table 59-145 `StopVoiceOver` message parameters 869
Table 59-146 `RequestVoiceOverMoveCursor` message parameters 870
Table 59-147 `VoiceOverCursorDirection` enum 870
Table 59-148 `RequestVoiceOverActivateCursor` message parameters 870
Table 59-149 `RequestVoiceOverScrollPage` message parameters 871
Table 59-150 `VoiceOverScrollDirection` enum 871
Table 59-151 `RequestVoiceOverSpeakText` message parameters 871
Table 59-152 `RequestVoiceOverPauseText` message parameters 872
Table 59-153 `RequestVoiceOverResumeText` message parameters 872
Table 59-154 `StartVoiceOverUpdates` message parameters 872
Table 59-155 `VoiceOverUpdate` message parameters 873
Table 59-156 `StopVoiceOverUpdates` message parameters 873
Table 59-157 `RequestVoiceOverConfiguration` message parameters 873
Table 59-158 `StartVoiceOverCursorUpdates` message parameters 874
Table 59-159 `VoiceOverCursorUpdate` message parameters 874
Table 59-160 VoiceOverCursorUpdate Traits 875
Table 59-161 `StopVoiceOverCursorUpdates` message parameters 876
Table 59-162 `RequestWiFiInformation` message parameters 876
Table 59-163 `WiFiInformation` message parameters 876
Table 59-164 `WiFiRequestStatus` enum 877
Table 59-165 `RequestAccessoryWiFiConfigurationInformation` message parameters 877
Table 59-166 `AccessoryWiFiConfigurationInformation` message parameters 877
Table 59-167 `WiFiSecurityType` enum 878

61. Device Dimensional Drawings 880
Figure 61-1 Apple Watch 38 mm Dimensional Drawing 883
Figure 61-2 Apple Watch 42 mm Dimensional Drawing 884
Figure 61-3 iPhone 6s Plus Dimensional Drawing 885
Figure 61-4 iPhone 6s Dimensional Drawing 886
Figure 61-5 iPhone 6 Plus Dimensional Drawing 887
Figure 61-6 iPhone 6 Dimensional Drawing 888
Figure 61-7 iPhone 5s Dimensional Drawing 889
Figure 61-8 iPhone 5c Dimensional Drawing 890

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 61-9 iPhone 5 Dimensional Drawing 891
Figure 61-10 iPhone 4s Dimensional Drawing 892
Figure 61-11 iPhone 4 CDMA Dimensional Drawing 893
Figure 61-12 iPhone 4 GSM Dimensional Drawing 894
Figure 61-13 iPhone 3G and iPhone 3GS Dimensional Drawing 895
Figure 61-14 iPhone Dimensional Drawing 896
Figure 61-15 iPad Pro Wi-Fi Dimensional Drawing 897
Figure 61-16 iPad Pro Wi-Fi + Cellular Dimensional Drawing 898
Figure 61-17 iPad mini 4 Wi-Fi Dimensional Drawing 899
Figure 61-18 iPad mini 4 Wi-Fi + Cellular Dimensional Drawing 900
Figure 61-19 iPad Air 2 Wi-Fi Dimensional Drawing 901
Figure 61-20 iPad Air 2 Wi-Fi + Cellular Dimensional Drawing 902
Figure 61-21 iPad mini 2 & iPad mini 3 Wi-Fi Dimensional Drawing 903
Figure 61-22 iPad mini 2 & iPad mini 3 Wi-Fi + Cellular Dimensional Drawing 904
Figure 61-23 iPad Air Wi-Fi Dimensional Drawing 905
Figure 61-24 iPad Air Wi-Fi + Cellular Dimensional Drawing 906
Figure 61-25 iPad mini with Wi-Fi Dimensional Drawing 907
Figure 61-26 iPad mini Wi-Fi + Cellular Dimensional Drawing 908
Figure 61-27 iPad Wi-Fi (4th generation) Dimensional Drawing 909
Figure 61-28 iPad Wi-Fi + Cellular (4th generation) Dimensional Drawing 910
Figure 61-29 iPad Wi-Fi (3rd Generation) Dimensional Drawing 911
Figure 61-30 iPad Wi-Fi + 4G (3rd Generation) Dimensional Drawing 912
Figure 61-31 iPad 2 Wi-Fi Dimensional Drawing 913
Figure 61-32 iPad 2 Wi-Fi + 3G Dimensional Drawing 914
Figure 61-33 iPad Wi-Fi Dimensional Drawing 915
Figure 61-34 iPad Wi-Fi + 3G Dimensional Drawing 916
Figure 61-35 iPod touch 6th gen. Dimensional Drawing 917
Figure 61-36 iPod touch 5th gen. Dimensional Drawing 918
Figure 61-37 iPod touch 4th gen. Dimensional Drawing 919
Figure 61-38 iPod touch 3rd gen. Fall '09 32GB and 64GB Dimensional Drawing 920
Figure 61-39 iPod touch 2nd gen. 8GB, 16GB, 32GB Dimensional Drawing 921
Figure 61-40 iPod touch Dimensional Drawing 922
Figure 61-41 iPod nano 7th gen. Dimensional Drawing 923
Figure 61-42 iPod nano 6th gen. Dimensional Drawing 924
Figure 61-43 iPod nano 5th gen. Dimensional Drawing 925
Figure 61-44 iPod nano 4th gen. Dimensional Drawing 926
Figure 61-45 iPod nano 3rd gen. Dimensional Drawing 927
Figure 61-46 iPod nano 2nd gen. Dimensional Drawing 928
Figure 61-47 iPod nano Dimensional Drawing 929

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 61-48 iPod classic 160GB Dimensional Drawing 930
Figure 61-49 iPod classic 80GB Dimensional Drawing 931
Figure 61-50 iPod 5th gen. 60GB/80GB Dimensional Drawing 932
Figure 61-51 iPod 5th gen. 30GB Dimensional Drawing 933
Figure 61-52 iPod 4th gen. Dimensional Drawing 934
Figure 61-53 iPod 3rd gen. Dimensional Drawing 935
Figure 61-54 iPod photo 30/60GB Dimensional Drawing 936
Figure 61-55 iPod photo Dimensional Drawing 937
Figure 61-56 iPod shuffle 4th gen. Dimensional Drawing 938
Figure 61-57 iPod shuffle 3rd gen. Dimensional Drawing 939
Figure 61-58 iPod shuffle 2nd gen. Dimensional Drawing 940
Figure 61-59 iPod shuffle Dimensional Drawing (1 of 2) 941
Figure 61-60 iPod shuffle Dimensional Drawing (2 of 2) 942
Figure 61-61 iPod mini Dimensional Drawing 943

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1. Introduction

NOTICE OF PROPRIETARY PROPERTY: THE INFORMATION CONTAINED HEREIN IS THE PROPRIETARY PROPERTY OF APPLE INC. THE POSSESSOR AGREES TO THE FOLLOWING: (I) TO MAINTAIN THIS DOCUMENT IN CONFIDENCE, (II) NOT TO REPRODUCE OR COPY IT, (III) NOT TO REVEAL OR PUBLISH IT IN WHOLE OR IN PART, (IV) ALL RIGHTS RESERVED.

ACCESS TO THIS DOCUMENT AND THE INFORMATION CONTAINED THEREIN IS GOVERNED BY THE TERMS OF THE MFI LICENSE AGREEMENT AND/OR THE IPOD-IPHONE AIS EVALUATION AGREEMENT. ALL OTHER USE SHALL BE AT APPLE'S SOLE DISCRETION.

1.1 Purpose of This Specification

This specification details requirements and recommendations for accessories that interface with Apple devices that have the Apple Lightning™ connector and Apple TV.

1.2 Requirements, Recommendations, and Permissions

This specification contains statements that are incorporated by reference into legal agreements between Apple and its licensees. The use of the words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *not recommended*, *may*, *optional*, and *deprecated* in a statement have the following meanings:

- *must*, *shall*, or *required* means the statement is an absolute requirement.
- *must not*, *shall not* or *prohibited* means the statement is an absolute prohibition.
- *should* or *recommended* means the full implications must be understood before choosing a different course.
- *should not* or *not recommended* means the full implications must be understood before choosing this course.
- *may* or *optional* means the statement is truly optional, and its presence or absence cannot be assumed.
- *deprecated* means the statement is provided for historical purposes only and is equivalent to 'must not'.

The absence of requirements, recommendations, or permissions for a specific accessory design in this specification must not be interpreted as implied approval of that design. Developers are strongly encouraged to ask Apple for feedback on accessory designs that are not explicitly mentioned in this specification.

1.3 Applicability

This specification covers the accessory interface exposed by Apple devices that have the Lightning connector and Apple TV. At time of release, this includes the following Apple devices:

- iPhone 6s Plus
- iPhone 6s
- iPhone 6 Plus
- iPhone 6
- iPhone 5s
- iPhone 5c
- iPhone 5
- iPad Pro
- iPad Air 2
- iPad Air
- iPad (4th generation)
- iPad mini 4
- iPad mini 3
- iPad mini 2
- iPad mini
- iPod touch (6th generation)
- iPod touch (5th generation)
- iPod nano (7th generation)
- Apple TV (4th generation)

This specification also covers portions of the accessory interface exposed by Apple devices that have the 30-pin connector. More content can be found in the following specifications:

- *MFi Accessory Firmware Specification R46*
- *MFi Accessory Hardware Specification R9*

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Any conflicts between the *Accessory Interface Specification* and *MFi Accessory Firmware/Hardware Specifications* must be resolved in favor of the *Accessory Interface Specification*.

1.4 Terminology

1.4.1 Accessory, Device, and Product

Throughout this specification:

- The term *device* is used to refer to an Apple iPhone, iPad, or iPod (typically running iOS, Apple's mobile operating system).
- The term *accessory* is used to refer to any product intended to interface with a device via the means described in this specification.
- The term *product* is used to refer generically to either a Mac (Apple computers that run OS X) or an aforementioned *device*.

1.4.2 Authentication Coprocessor

An accessory hardware component that provides Apple device-related digital signature creation and verification services.

1.4.3 I2C Bus

A 2-wire serial bus designed by Philips to allow easy communication between components that reside on the same circuit board. The I2C specification is located at http://www.nxp.com/documents/user_manual/UM10204.pdf.

1.4.4 Challenge

A random number sent from an Apple device to an accessory, or vice versa. The device/accessory being challenged must perform a challenge response computation on the offered challenge and return the resulting Challenge Response (page 52) to the challenging device for verification.

1.4.5 Challenge Response

The result obtained by performing a challenge response process on an offered Challenge (page 52).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1.4.6 X.509 Certificate

A standard defined by the International Telecommunications Union (ITU) that governs the format of certificates used for authentication and sender identity verification in public-key cryptography. X.509 certificates contain the public keys used in the Apple device's accessory authentication process.

1.4.7 Component

An accessory is defined as a collection of functional units called *components*. Examples of a component include, but are not limited to, the following:

- Data transport
- Power source
- Human Interface Device (HID) control set

1.4.8 Feature

All accessories must support one or more accessory interface *features*. Each feature may have associated accessory design requirements and recommendations; an accessory must comply with all feature-specific requirements to properly support the feature. Some features require other features to be implemented.

Accessory design must take into account the possibility that an Apple device may not support all of the accessory's implemented features and react appropriately.

1.4.9 USB Device and Host Mode

Apple devices are capable of taking on both USB Host and USB Device roles when connecting to accessories. In this specification, *USB Host Mode* is always used when the Apple device is the USB host and the accessory is a USB device. *USB Device Mode* is always used when the Apple device is the USB device and the accessory is a USB host.

1.4.10 iAP

There are two iAP protocols - *iAP1* and *iAP2*. iAP1 is the *iPod Accessory Protocol* as specified in *MFi Accessory Firmware Specification R46*. iAP2 is a complete replacement for iAP1 and is not backward compatible; it is specified in this specification.

An iAP2 *connection* is composed of an iAP2 *transport*, iAP2 *link*, and one or more iAP2 *sessions*.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1.4.11 Direct User Action

Some accessory interface features or iAP2 messages have a direct user action requirement. When this requirement is specified, the accessory must not autonomously use the feature or send the message without direct action taken by the user, such as pressing a physical button on the accessory or attaching the accessory to the Apple device's Lightning receptacle. This extends to resends or retries of iAP2 control session messages; sending such a message more than once in response to one direct user action is explicitly prohibited. Retries of iAP2 link packets are not affected.

Failure to observe this requirement when specified will always result in failure to pass self certification.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. General Requirements and Recommendations

The requirements in this section apply to all accessories regardless of their feature sets.

2.1 Minimum Apple Device Compatibility

> **Note:** Apple will use an accessory's claimed device compatibility list to prioritize future development and bug fixes. Devices that currently work with an accessory, but are not on the accessory's claimed compatibility list, will receive a lower priority.

Accessories that claim compatibility with iPhone and/or iPod should claim compatibility with at least the set of iPhones and iPods running iOS 9.2 or later:

- iPhone 6s Plus
- iPhone 6s
- iPhone 6 Plus
- iPhone 6
- iPhone 5s
- iPhone 5c
- iPhone 5
- iPhone 4s
- iPod touch (6th generation)
- iPod touch (5th generation)

Accessories that claim compatibility with iPad should claim compatibility with at least the set of iPads running iOS 9.2 or later:

- iPad Pro
- iPad Air 2
- iPad Air

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- iPad (4th generation)
- iPad (3rd generation)
- iPad 2
- iPad mini 4
- iPad mini 3
- iPad mini 2
- iPad mini

Accessories may exclude Apple devices from their compatibility list if they:

- Cannot physically connect to the Apple device.
- Implement a feature that is not supported by the Apple device.
- Implement a feature in a specified manner that is not compatible with the Apple device.

For example, accessories that:

- Connect to Apple devices via Bluetooth, Wi-Fi, USB-A to Lightning cable, or USB-A to 30-pin cable must not exclude any devices unless they meet one of the exclusion criteria
- Comply with dock requirements for iPhone/iPod and cannot be reasonably connected to any iPad (see iPhone and iPod touch Dock Requirements (page 158)) may exclude iPad devices
- Comply with power source requirements for iPhone/iPod (see Lightning Connectors on Power-Providing Accessories (page 664)) may exclude iPad devices
- Implement only iAP2 may exclude iPhone 4s
- Implement both iAP1 and iAP2 must claim compatibility with iPhone 4s
- Connect to Apple devices only via an integrated Lightning connector may exclude iPhone 4s

2.2 Development Tools and Emulators

Use of this specification to create accessory development tools, such as protocol analyzers, intended for use by other accessory developers or manufacturers is prohibited unless explicitly authorized by Apple.

Similarly, this specification must not be used to create accessories that emulate the accessory interface exposed by Apple devices.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2.3 Reference Designs & Development Kits

Reference designs and development kits may be developed using this specification to assist with the design and prototyping of accessories. These may consist of any of the following:

- Mechanical designs or components.
- Electrical designs or components.
- Firmware or application software (libraries, frameworks, sample code, etc.).
- Development tools.

Reference designs and development kits must treat specification recommendations as requirements and pass certification audits with no warnings.

A datasheet for the reference design or development kit listing all supported accessory interface features and expected accessory design/prototyping approaches must be submitted for certification audit.

2.4 Accessory Authentication and Accessory Identification

All accessories other than simple chargers or charge/sync cables must authenticate and identify their feature sets to the Apple device.

2.5 iAP

All accessories that support iAP must implement iAP2.

Accessories may support both iAP1 and iAP2. This specification details how an accessory may detect whether the connected Apple device supports iAP2 in Initialization (page 773). If the Apple device does not support iAP2, the accessory may fall back to iAP1.

If an accessory implements support for both iAP1 and iAP2, it must:

- use iAP2 when connected to an Apple device that supports iAP2.
- present the exact same Accessory Identification information (see Accessory Identification (page 265)) over both iAP1 and iAP2.

Accessories that claim compatibility with the iPod nano (7th generation) must support iAP1.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2.6 Connector Assemblies

Connectors as defined in Apple Lightning Connector (page 128) must not be modified in assemblies.

Docks, dongles, and form fitting accessories as defined in Apple Lightning Connector (page 128) must not implement user detachable 30-pin/Lightning connector assemblies.

2.7 Adapters and Proxies

Accessories must not directly or indirectly facilitate the connection of other products that are not compliant with this specification to Apple devices.

2.8 Mixed 30-pin and Lightning Connectors

Simple charger accessories may incorporate both a Power/Sync Only 30-pin connector and a B configuration (USB Device Mode) Lightning connector. See Multiple Connectors (page 667) for additional requirements.

Accessories that both provide power and exchange data with an Apple device may incorporate both a 30-pin connector and a Lightning connector under the following conditions:

- The accessories must use the same data transport (serial/USB Host/USB Device) to communicate with both connectors unless they implement the Digital Audio (page 528) feature on both connectors. In that situation, the accessory may use the USB Host Mode transport on the Lightning connector and USB Device Mode transport on the 30-pin connector.
- If the USB data transport is used, both connectors must meet the signal integrity requirements set forth in Connector Signal Integrity Requirements (page 168). Possible implementation approaches include a microcontroller with sufficient I/O capability to maintain two separate data transport paths or a multiplexer chip such as the Intersil ISL54233.
- All connectors that are physically accessible must provide power even when no Apple device is attached.
- If the accessory is intended to work with only one Apple device at any given time, it must switch to the last attached Apple device by looking for Accessory Power to go high on a connector.
- The accessory must use iAP1 or iAP2 to inform the device of available power.

The following diagrams illustrate these requirements.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 2-1 Mixed 30-pin and Lightning Connector USB Accessory

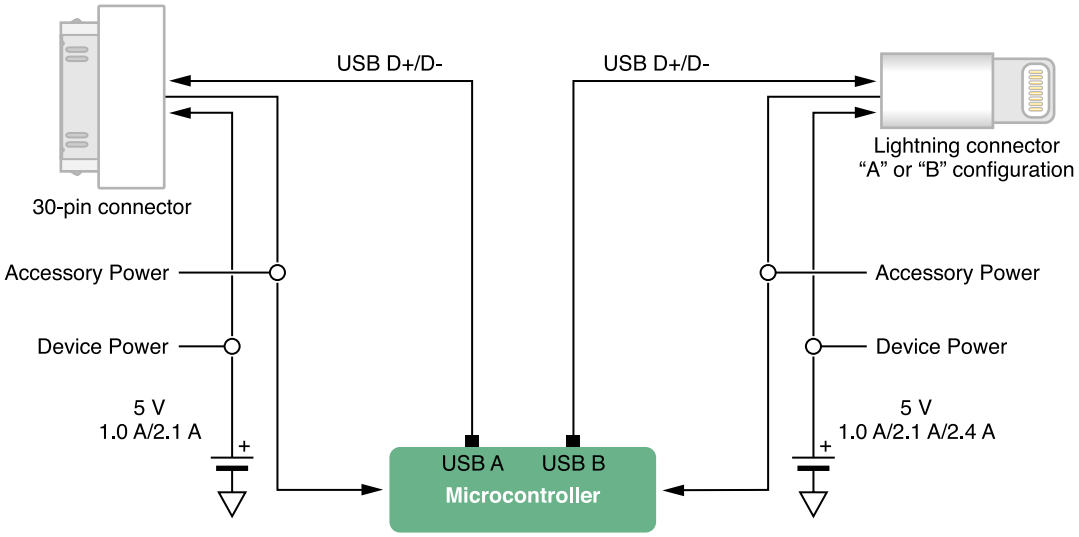


Figure 2-2 Mixed 30-pin and Lightning Connector USB Accessory with Multiplexer

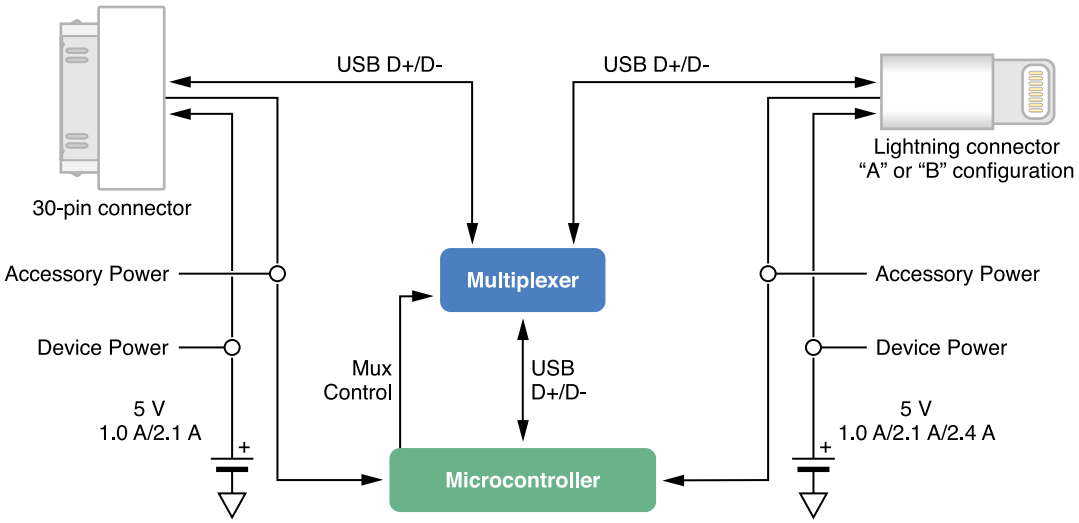


Figure 2-3 Mixed 30-pin and Lightning Connector Serial Accessory

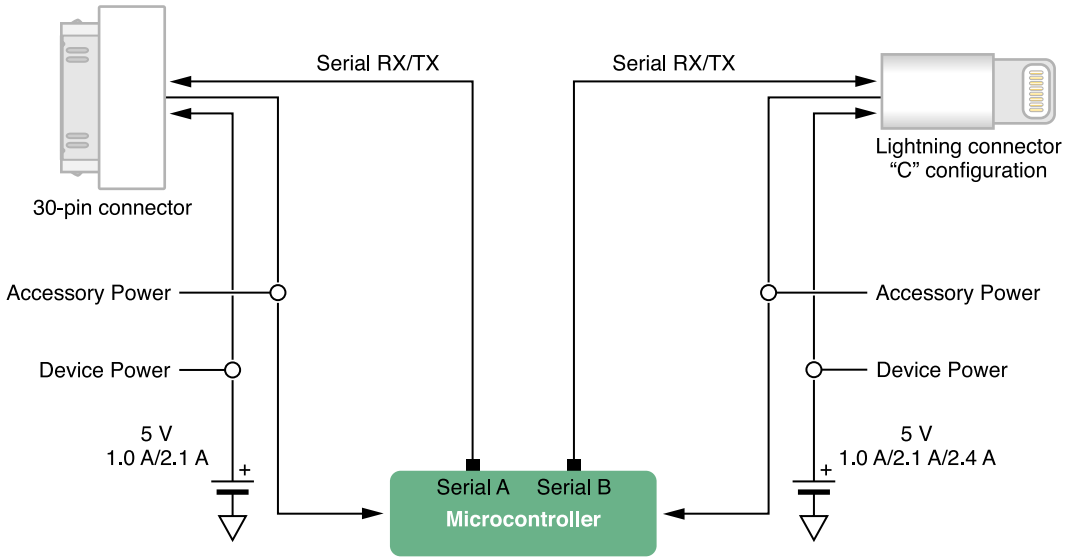
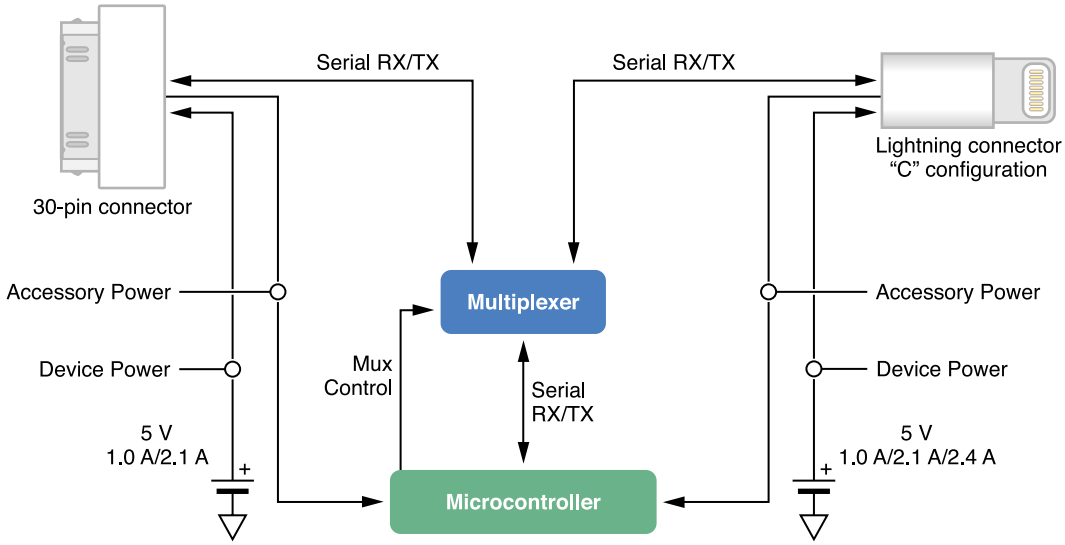


Figure 2-4 Mixed 30-pin and Lightning Connector Serial Accessory with Multiplexer



All other accessories must not integrate both 30-pin and Lightning connectors.

2.9 Mixed Headset Jack and Lightning Connectors

Accessories must not attach to both the headset jack and the Lightning connector on an Apple device.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2.10 Apple USB Power Adapters

Accessories must not use Apple USB power adapters to support the weight of an Apple device while plugged into a wall outlet.

2.11 Apple Device Detection

Accessories that implement iAP must not assume that an Apple device is present until the Apple device responds to the accessory's StartIDPS command (iAP1) or Link Initialization Byte Sequence (iAP2).

Accessories that both implement iAP and integrate a Lightning connector that offers Accessory Power (see Connector Pad/Pin Configuration (page 139)) may choose to wait until Accessory Power is detected before sending the StartIDPS command or Link Initialization Byte Sequence.

Accessories that do not implement iAP must assume that an Apple device is present at all times.

2.12 Multiple Simultaneous iAP2 Connections

Accessories may implement iAP2 connections on multiple transports simultaneously, for example wired and wireless, until a determination can be made that the connections are to the same Apple device. Accessories must not continue using multiple iAP2 connections to the same Apple device unless otherwise specified by a specific feature.

An accessory must use available means such as DeviceUUIDUpdate (page 842) or BluetoothConnectionUpdate (page 823) (comparing the BluetoothTransportComponent's BluetoothTransportMediaAccessControlAddress) to determine whether the connections are to the same Apple device.

Alternatively, an accessory may shut down the iAP2 connection on one transport and then start up an iAP2 connection on a different transport, but it must change its identification information accordingly.

If both wired (e.g., Lightning) and wireless (e.g., Bluetooth) transports are simultaneously available to the same Apple device, the accessory must use the wired transport unless otherwise specified by a specific feature.

2.13 Presentation of Apple Device Updates

Accessories must not present the state of an Apple device to a user before being informed of that state by the Apple device. For example, if the user presses a 'Shuffle' button on an accessory and causes a corresponding media remote control HID usage to be sent to the Apple device, the accessory must not inform the user of a

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

change in shuffle state until the Apple device has confirmed the state change. Assuming that the state change will happen will cause problems if the state change does not occur, such as when a third party app is running and does not respond to the user input.

Similarly, accessories must not cache Apple device information unless explicitly specified otherwise.

> **Note:** Any accessory control, whether physical or virtual, that is an analogue to a corresponding HID usage specified in HID Requirements (page 580) must send that exact HID usage to the Apple device when the user takes direct action to actuate that control. For example, use of a 'Play/Pause' button to programmatically send either the 'Play' or 'Pause' HID usage based on cached Apple device state is prohibited; a 'Play/Pause' HID usage must be sent.

2.14 Relationships Between Multiple Accessories

All accessories must not require that another accessory be connected to the same Apple device in order to function. Additionally, accessories must not take action on behalf of another accessory. In all such situations, one accessory with multiple components must be designed and implemented, and the accessory is solely responsible for managing its overall connection state with the Apple device when individual components are active or inactive.

2.15 iBeacon

MFi accessories must not support the iBeacon feature.

2.16 Feature Duplication

Accessories must not implement functionality that overlaps with a specified accessory interface feature without also offering that same functionality via the specified interface.

For example, every accessory that provides or receives any kind of MIDI data to or from an Apple device must support the MIDI feature in addition to any other mechanism it may implement, such as an External Accessory Protocol.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2.17 Temperature Range

The temperature range of the accessory must be greater than or equal to the published temperature ranges of all the Apple devices with which it claims compatibility.

2.18 Magnetic Fields

All accessories that claim to be compatible with Apple devices that contain digital compasses must minimize interference with the digital compass and must not repeatedly trigger compass recalibration.

2.19 Cables with USB Connectors

All cables included and/or permanently tethered to or attached to an accessory that terminate in at least one USB plug must meet or exceed USB-IF specifications and ECNs.

Accessories must use the conductors in standard USB plugs as defined in the USB-IF specifications and Engineering Change Notices.

For cables incorporating a Lightning connector, see Apple Lightning Connector (page 128).

2.20 Cables with Non-USB Connectors

Accessory cables that incorporate a non-USB connector must integrate the non-USB connector in accordance with any specifications or requirements that apply to the non-USB connector. For instance:

- Cables that terminate in a 2.5 mm or 3.5 mm audio plug must only carry analog audio signals.
- Cables that terminate in a HDMI, DisplayPort, FireWire, or Thunderbolt connector must use the connector in a manner that is compliant with their respective specifications.

For cables incorporating a Lightning connector, see Apple Lightning Connector (page 128).

2.21 Integrated USB Receptacles

Accessories that incorporate a USB receptacle for the purpose of drawing power from an external USB power supply must comply with the following requirements:

- They must support the USB Dedicated Charging Port (DCP) as defined in the USB Battery Charging 1.2 specification.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- They must not attempt to detect Apple branded or MFi power supplies that connect the D+/D- pins to resistor networks as specified in Connectors That Do Not Implement iAP2 (page 665).
- If the accessory can also be enumerated as a USB device via the USB receptacle, it must comply with the USB 2.0 specification and not draw more than 100 mA of current until it has been successfully enumerated by the external USB host.

2.22 Integrated Non-USB Receptacles

Accessories that integrate non-USB receptacles must use them in accordance with any specifications or requirements that apply to the non-USB receptacle. For instance:

- 2.5 mm and 3.5 mm audio plug receptacles must not carry power or digital data.
- HDMI, DisplayPort, FireWire, or Thunderbolt receptacles must be implemented in a manner that is compliant with their respective specifications.

Accessory non-USB receptacles that are intended for an included accessory cable that incorporates both a Lightning connector and non-USB connector may be labeled using the compatibility icon for an iPhone 5 (without any accompanying text) to highlight to the user that the cable should be plugged into that receptacle.

2.23 User Supplied Cables and Power Supplies

Accessories that can reasonably be used with user-supplied cables and/or power supplies must be designed to work with any cable or power supply that is compliant with this specification, including Apple branded cables and power supplies. Such accessories must not declare compatibility only with Apple branded USB cables or USB power supplies.

This compatibility requirement applies to all aspects of user-supplied cables and power supplies. For example:

- Connector receptacles on accessories must accommodate all spec-compliant connector overmolds, and any accessory opening surrounding the Lightning connector on an Apple device must provide sufficient clearance for spec-compliant connector overmolds.
- Accessories must work with all cables that comply with the specification with regards to electrical DCR and SI.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: Such accessories must be tested with a wide variety of spec-compliant cables (including various lengths of the same cable if applicable) and power supplies during accessory development, in addition to Apple branded cables and power supplies.

2.24 Removable Storage

Accessories must not integrate removable storage components, such as SD card readers, that transfer data to or from an Apple device via means other than an External Accessory Protocol (see External Accessory Protocol (page 535)).

2.25 RF Transmission and Reception

Accessories must avoid degrading the wireless performance of Apple devices. Device antenna and sensor locations must be taken into consideration during the design process. Specifically, the following wireless performance characteristics must be considered:

- **Reduction of the device's RF/antenna efficiency:** Accessories must minimize decreases in the device's total radiated power (TRP). This can be quantified by measuring TRP across all of the device's operating bands.
- **Desense of the device's RF reception:** Accessories must minimize decreases in the device's effective isotropic sensitivity (EIS). This can be quantified by measuring EIS across all of the device's operating bands.

See RF (OTA) (page 504).

2.26 TDMA Noise

GSM phones emit radiated and conducted RF noise, which can produce time division multiple access (TDMA) sounds from audio outputs. Accessories must minimize coupling of audible interference from the Apple device (commonly known as *TDMA noise* or *chopper noise*) into an accessory's electronics.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Apple Authentication Coprocessor 2.0C

3.1 Coprocessor 2.0C Overview

An Apple device verifies whether a third-party accessory attached to it is authorized for use with the Apple device by issuing an authentication challenge to the accessory. The accessory must respond to the Apple device's challenge, and it can do so only with the assistance of an Apple Authentication Coprocessor (CP) chip located in the accessory. Conversely, the accessory can use its CP chip to authenticate the Apple device.

Earlier versions of the Apple Authentication Coprocessor (1.0, 2.0A, and 2.0B) were implemented in QFN-40, QFN-20, and SOP-8 packages. The current version, 2.0C, is supplied in a smaller and more efficient PG-USON-8-1 package.

The Export Classification Control Number (ECCN) of the Apple Authentication Coprocessor 2.0C is 5A992 NLR.

3.2 Coprocessor 2.0C Authentication Protocol

The authentication protocol supported by the Apple Authentication Coprocessor 2.0C is based on standard X.509 version 3 certification. Each certificate is generated and signed by a recognized certificate authority and has a unique serial number. Information about the X.509 standard can be found at the IETF website http://tools.ietf.org/html/3280.

For information about accessory authentication of the Apple device, refer to Accessory Authentication (page 261).

For information about Apple device authentication of the accessory, refer to Device Authentication (page 523).

3.3 Coprocessor 2.0C Signals and Pinouts

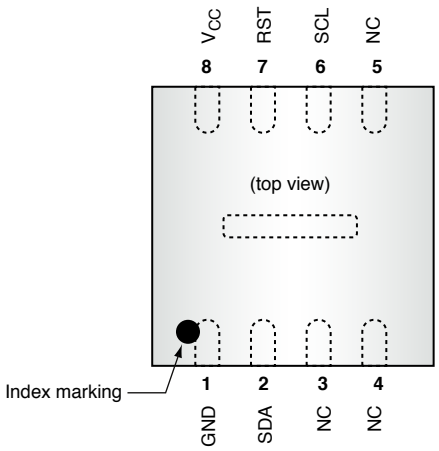
The 2.0C CP chip signal descriptions are given in Table 3-1 (page 67) and its pinouts are shown in Figure 3-1 (page 67).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 3-1 Coprocessor signals

Signal name	Pin	I/O	Description
GND	1		Supply voltage, negative terminal
SDA	2	I/O	I2C data
NC	3-5		Must not be connected
SCL	6	I	I2C clock
RST	7	I	At reset: selects I2C slave address. During operation: CP warm reset
V_{CC}	8		Supply voltage, positive terminal

Figure 3-1 Coprocessor pinout, top view



PG-USON-8-1 package

The thermal pad on the bottom of the CP may be left unconnected or connected to GND.

3.4 Coprocessor 2.0C Address Selection

After power-up or in response to a warm reset, the state of RST is used to select the CP's I2C slave addresses, as shown in Table 3-2 (page 67).

Table 3-2 Coprocessor address selection signals

RST state	I2C write address	I2C read address
0	0x20	0x21

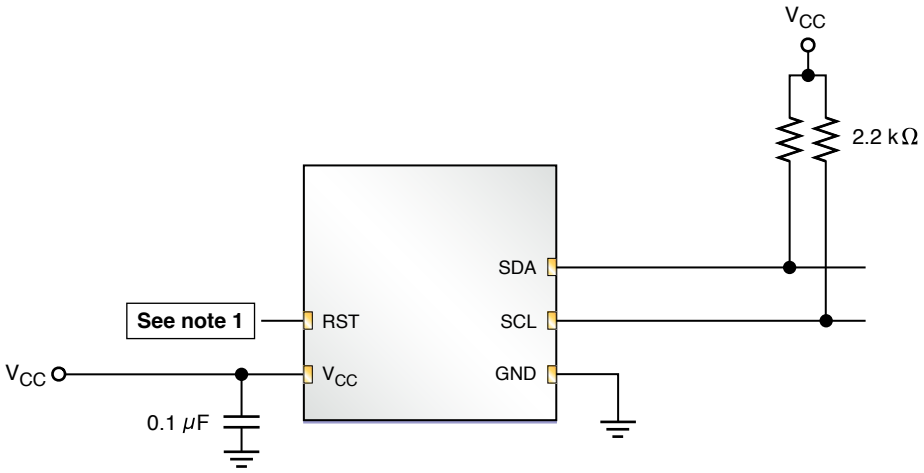
RST state	I2C write address	I2C read address
1	0x22	0x23

See I2C Communications Process (page 71) for the interface requirements of the CP's I2C slave communication transport.

3.5 Coprocessor 2.0C Reference Circuit

The reference circuit for I2C operation of the CP is shown in Figure 3-2 (page 68).

Figure 3-2 Coprocessor reference circuit diagram



Note: If the CP's warm reset function is not needed, RST can be tied to either V_{CC} or GND, depending on which of the addressing modes shown in Table 3-2 (page 67) is used. If the warm reset function is needed, RST should be connected to a general-purpose I/O line on the accessory's controller. For further details, see I2C Startup On Warm Reset (page 70).

3.6 Coprocessor 2.0C System Voltage

The 2.0C CP may be used either in an accessory powered by an attached Apple device or in an accessory that has its own power source.

3.7 Coprocessor 2.0C I2C Interface

The CP's I2C communication interface can be started up either by supplying power to the V_{CC} line or by performing a warm reset on the RST line. The accessory actions required for these two procedures are specified in the next sections.

3.7.1 I2C Startup On Power On

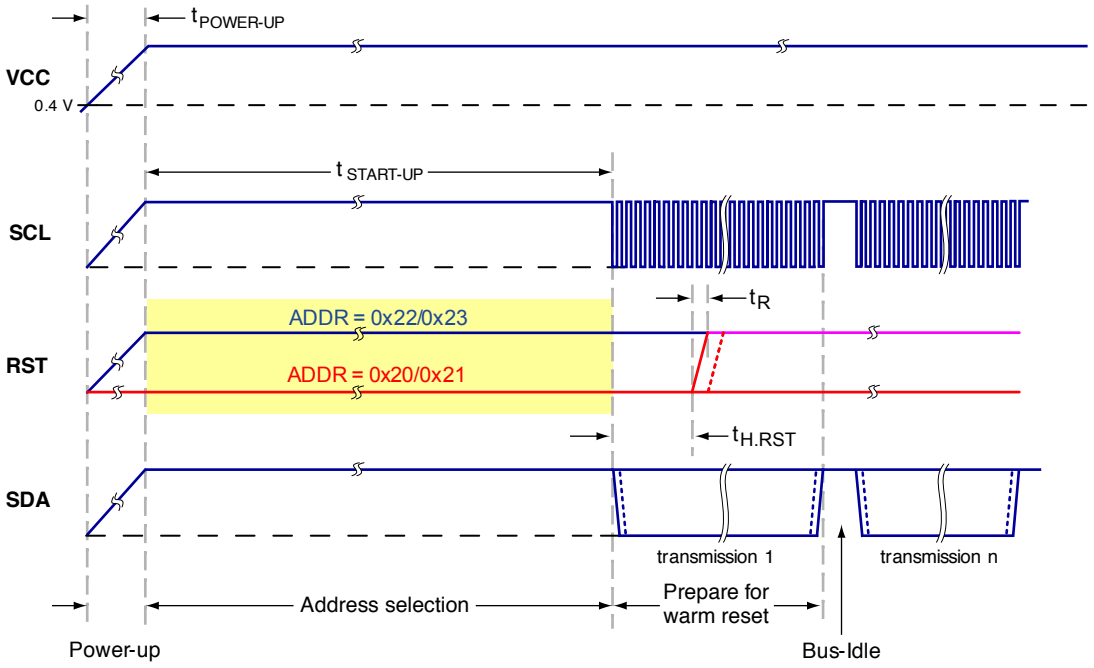
To activate the I2C interface by supplying power to the V_{CC} line, the accessory must perform the following startup procedure. This procedure is required both to support address selection by means of the RST line and to support the option of a warm reset later.

- The V_{CC} line must be supplied with power and the SDA and SCL lines must be set high during the entire procedure.
- The RST line, used to select addressing as described in Coprocessor 2.0C Address Selection (page 67), must be kept either low or high during the entire procedure. If the RST line is kept low during the procedure, the accessory must set it high not earlier than 10 *ms* after the V_{CC} line goes high but before the first data transmission occurs.
- The first data transmission may start not earlier than 10 *ms* after the V_{CC} line goes high. If the RST line has been kept low and the accessory might need to perform a warm reset of the CP chip later, the accessory must set the RST line high not earlier that 1 *ms* after the start of the first data transmission ($t_{H.RST}$ interval in Figure 3-3 (page 70)). If the option of a warm reset later is not needed, the accessory may tie RST directly to either V_{CC} or GND.

Figure 3-3 (page 70) diagrams the timing of the I2C interface when it is started up by turning power on.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 3-3 Coprocessor I2C power on timing



In this diagram, $t_{STARTUP} \geq 10\ ms$ and $t_{H.RST} \geq 1\ ms$. The rise time of t_R and the fall time of t_F are both $< 1\ \mu s$ from 10% to 90% of the signal amplitude, and $t_{POWER\text{-}UP} < 200\ \mu s$ from $V_{CC} = 0.4$ V until $V_{CC} = 90\%$ of the target supply voltage.

3.7.2 I2C Startup On Warm Reset

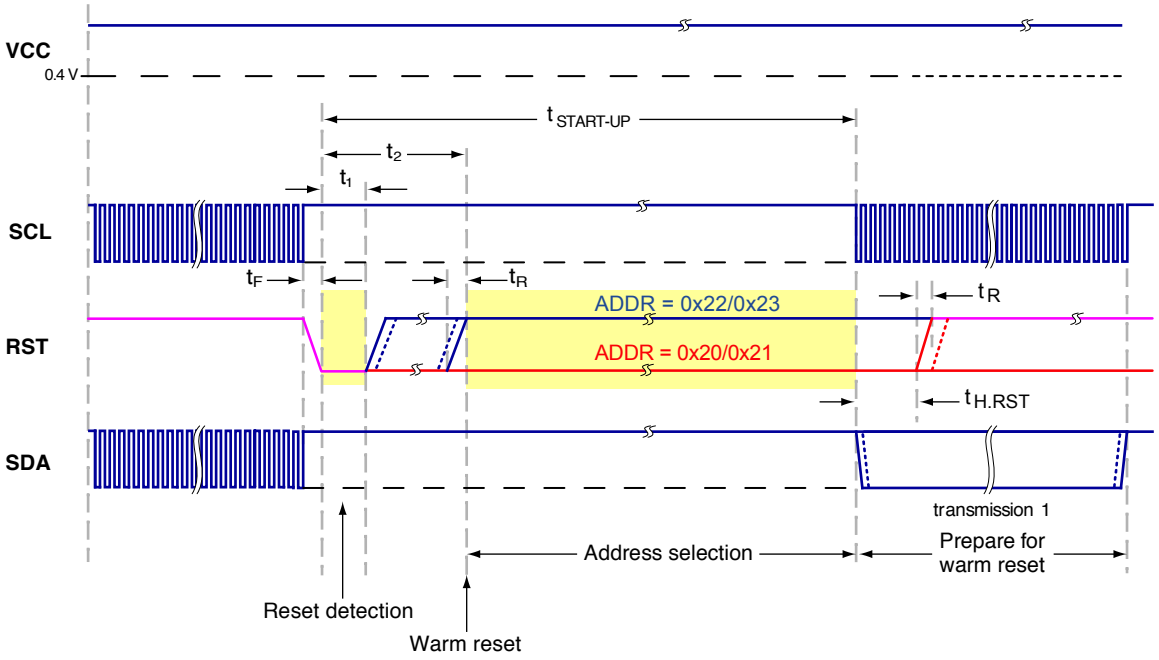
To reset the I2C interface through the RST line, after activating it through power-up as specified in I2C Startup On Power On (page 69), the accessory must perform the following procedure:

- The V_{CC} line must be supplied with power during the entire procedure.
- The terminal must halt I2C communication, and the SDA and SCL lines must be set high before the RST line is set low.
- The RST line must be set low and kept low for at least 10 μs. Not later than 1 *ms* after the falling edge of the RST signal, the accessory must finish its I2C address selection by either keeping RST low or driving RST high. If the RST line is kept low during the warm reset, the accessory must set it high not earlier than 10 *ms* after the V_{CC} line goes high but before the first data transmission occurs.
- The first data transmission may start not earlier than 10 *ms* after the falling edge of the RST signal. If the RST line has been kept low during the warm reset, the accessory must set it high not earlier that 1 *ms* after the start of the first data transmission ($t_{H.RST}$ interval in Figure 3-4 (page 71)).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 3-4 (page 71) diagrams the timing of the I2C interface when it is reset by software toggling the RST line without a power off/on cycle.

Figure 3-4 Coprocessor I2C warm reset timing



In this diagram, $t_{STARTUP} \geq 10$ ms, $t_{H.RST} \geq 1$ ms, $t_1 > 10\ \mu s$, and $t_2 = 3\ ms$. The rise time of t_R and the fall time of t_F are both $< 1\ \mu s$ from 10% to 90% of the signal amplitude.

3.7.3 I2C Communications Process

When the CP is addressed using I2C, it acts as a standard 7-bit I2C slave. The I2C slave address is configured upon reset and is based on the RST input. The I2C effective slave address for writing is shown in Figure 3-5 (page 71) and the corresponding read address in Figure 3-6 (page 71).

Figure 3-5 Coprocessor I2C slave write address

A6	A5	A4	A3	A2	A1	A0	R/nW
0	0	1	0	0	0	RST	0

Figure 3-6 Coprocessor I2C slave read address

A6	A5	A4	A3	A2	A1	A0	R/nW
0	0	1	0	0	0	RST	1

Copyright © 2015 Apple Inc. All Rights Reserved.

In I2C mode, the CP has both a write address and a read address, as is typical for an I2C device. The I2C write address of the CP consists of the seven bits [A6:A0] followed by 0 for the R/nW bit. The I2C read address of the CP consists of the seven bits [A6:A0] followed by 1 for the R/nW bit. If the RST input is connected to ground, the write and read addresses of the CP are 0x20 and 0x21 respectively; if it is pulled high, the write and read addresses of the CP are 0x22 and 0x23.

3.7.4 I2C Sleep Mode

The 2.0C CP conserves power by entering a Sleep mode. It enters this mode automatically and cannot be forced to it externally. When in Sleep mode, the CP automatically wakes in response to any I2C communications sent to its address.

3.8 Coprocessor 2.0C Registers

Registers within the Apple Authentication Coprocessor 2.0C (CP) are accessed via I2C transport, as described in I2C Communications Process (page 71). This section specifies the CP's register addressing details and telegram formats.

3.8.1 Register Addresses

Registers and their addresses in the CP are listed in Table 3-3 (page 72). Each register is discussed in the sections that follow.

> **Note:** Registers in the same block with consecutive addresses may be read from sequentially in increasing numerical order. Registers must not be written to sequentially except as noted. Multibyte numeric values are stored in big-endian order; for example, the first byte in a two-byte register is the MSB of the stored value and the second byte is its LSB.

Table 3-3 Coprocessor register map

Address	Block	Name	Bytes	Reset Value	Access
0x00	0	Device Version	1	0x05	Read-only
0x01	0	Firmware Version	1	0x01	Read-only
0x02	0	Authentication Protocol Major Version	1	0x02	Read-only

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Address	Block	Name	Bytes	Reset Value	Access
0x03	0	Authentication Protocol Minor Version	1	0x00	Read-only
0x04	0	Device ID	4	0x00000200	Read-only
0x05	0	Error Code	1	0x00	Read-only
0x10	1	Authentication Control and Status	1	0x00	Read/write
0x11	1	Challenge Response Data Length	2	128	Read/write
0x12	1	Challenge Response Data	128	Undefined	Read/write
0x20	2	Challenge Data Length	2	20	Read/write
0x21	2	Challenge Data	128	Undefined	Read/write
0x30	3	Accessory Certificate Data Length	2	≤ 1280	Read-only
0x31	3	Accessory Certificate Data (Part 1)	128	Certificate	Read-only
0x32	3	Accessory Certificate Data (Part 2)	128	Certificate	Read-only
0x33	3	Accessory Certificate Data (Part 3)	128	Certificate	Read-only
0x34	3	Accessory Certificate Data (Part 4)	128	Certificate	Read-only
0x35	3	Accessory Certificate Data (Part 5)	128	Certificate	Read-only
0x36	3	Accessory Certificate Data (Part 6)	128	Certificate	Read-only
0x37	3	Accessory Certificate Data (Part 7)	128	Certificate	Read-only
0x38	3	Accessory Certificate Data (Part 8)	128	Certificate	Read-only

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Address	Block	Name	Bytes	Reset Value	Access
0x39	3	Accessory Certificate Data (Part 9)	128	Certificate	Read-only
0x3A	3	Accessory Certificate Data (Part 10)	128	Certificate	Read-only
0x40	4	Self-Test Control and Status	1	0x00	Read/write
0x41-0x4C	4	Reserved			
0x4D	4	System Event Counter (SEC)	1	Undefined	Read-only
0x4E	4	Accessory Certificate Serial Number	31	Null-terminated string	Read-only
0x50	5	Apple Device Certificate Data Length	2	0x0000	Read-only
0x51	5	Apple Device Certificate Data (Part 1)	128	Undefined	Read/write
0x52	5	Apple Device Certificate Data (Part 2)	128	Undefined	Read/write
0x53	5	Apple Device Certificate Data (Part 3)	128	Undefined	Read/write
0x54	5	Apple Device Certificate Data (Part 4)	128	Undefined	Read/write
0x55	5	Apple Device Certificate Data (Part 5)	128	Undefined	Read/write
0x56	5	Apple Device Certificate Data (Part 6)	128	Undefined	Read/write
0x57	5	Apple Device Certificate Data (Part 7)	128	Undefined	Read/write
0x58	5	Apple Device Certificate Data (Part 8)	128	Undefined	Read/write

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: Normally, reading register 0x05 will clear it. However, register 0x05 can be read sequentially only as part of a sequence that begins with a register in the range 0x00-0x04, in which case the read operation does not clear it.

Note: Registers 0x11 and 0x20 may each be written sequentially with registers 0x12 and 0x21, respectively.

3.8.2 Register Descriptions

This section describes the ways that the CP registers listed in Table 3-3 (page 72) are used.

3.8.2.1 Device Version

The Device Version read-only register contains the version number of the coprocessor device. The current Authentication 2.0C coprocessor is designated as device version 0x05.

3.8.2.2 Firmware Version

The Firmware Version read-only register contains the version number of the coprocessor firmware. Firmware version numbers advance by whole integers.

3.8.2.3 Authentication Protocol Major and Minor Versions

The Authentication Protocol Major Version and Authentication Protocol Minor Version read-only registers provide the version number of the authentication protocol that the CP supports. This information is accessed by the iAP command `RetDevAuthenticationInfo` during accessory authentication.

3.8.2.4 Device ID

The Device ID read-only register is not used by accessories that implement iAP2.

3.8.2.5 Error Code

The Error Code read-only register stores the most recent communication or authentication process error code generated since the register was last cleared. The error code register is cleared after it is read. The possible error codes are listed in Table 3-4 (page 76).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

If a single communication operation happens to produce multiple errors (for example, by writing an invalid challenge response length during a multiregister write that also attempts to continue past the end of the corresponding block) then only the highest-numbered error code is stored.

Table 3-4 Coprocessor error codes

Error Code	Description
0x00	No error
0x01	Invalid register for read
0x02	Invalid register for write
0x03	Invalid challenge response length
0x04	Invalid challenge length
0x05	Invalid certificate length
0x06	Internal process error during challenge response generation
0x07	Internal process error during challenge generation
0x08	Internal process error during challenge response verification
0x09	Internal process error during certificate validation
0x0A	Invalid process control
0x0B	Process control out of sequence
0x0C-0xFF	Reserved

3.8.2.6 Authentication Control and Status

The Authentication Control and Status read/write register provides control and status information for the CP's authentication processes.

When read from, the Authentication Control and Status register provides the status of the most recently requested CP process, as shown in Figure 3-7 (page 76), Table 3-5 (page 77) and Table 3-6 (page 77).

Figure 3-7 Coprocessor Authentication Control and Status register, read-only bits



Table 3-5 Coprocessor Authentication ERR_SET values

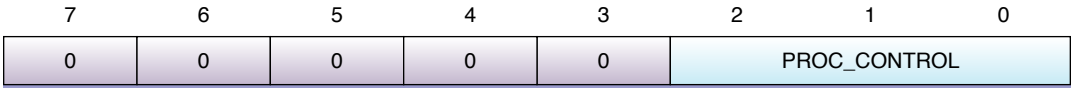
ERR_SET Value	Description
0	The Error Code register does not contain a code generated by the most recent command execution. However, it may still contain the most recent error code (greater than 0x00) generated by an earlier command.
1	The Error Code register contains the most recent process or communication error. Both this bit and the Error Code register contents are cleared after the Error Code register is next read. This bit is also cleared after every successful command execution.

Table 3-6 Coprocessor Authentication PROC_RESULTS values

PROC_RESULTS Value	Description
0	Most recent process did not produce valid results.
1	Accessory challenge response successfully generated.
2	Challenge successfully generated.
3	Apple device challenge response successfully verified.
4	Apple device certificate successfully validated.
5-7	Reserved.

When written to, the Authentication Control and Status register controls the start of CP processes, as shown in Figure 3-8 (page 77) and Table 3-7 (page 78).

Figure 3-8 Coprocessor Authentication Control and Status register, write-only bits



Note: Attempts to write to other bits in the Coprocessor Authentication Control and Status register are ignored.

Table 3-7 Coprocessor Authentication PROC_CONTROL values

PROC_CONTROL Value	Description	Notes
0	No operation	This control does nothing and always reports success (ERR_SET = 0; PROC_RESULTS = 0).
1	Start new challenge response-generation process	
2	Start new challenge-generation process	The length of the challenge to be generated is defined by the Challenge Data Length register and ranges from 1 to 128 bytes.
3	Start new challenge response-verification process	
4	Start new certificate-validation process	Do not attempt to read the accessory certificate after writing the Apple device certificate but before validating it by this control.
5	No operation	This control does nothing and always reports success (ERR_SET = 0; PROC_RESULTS = 0).
6-7	Reserved.	

3.8.2.7 Challenge Response Data Length

The Challenge Response Data Length read/write register holds the length in bytes of the results of the most recent challenge response-generation process (if the Apple device is authenticating an accessory) or challenge response-verification process (if the accessory is authenticating the Apple device).

Before a challenge response-generation process begins, this register should contain 0x80, the maximum allowable challenge response length. After completion of the challenge response-generation process, the CP updates this register to contain the actual length of the generated challenge response. This updated value should be read in order to determine how much of the Challenge Response Data register contains valid challenge response bytes.

Before a challenge response-verification process begins, this register should hold the actual length of the challenge response being verified.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3.8.2.8 Challenge Response Data

In the case of a challenge response generation process, the Challenge Response Data register holds the newly-generated data. In the case of a challenge response verification process, it holds the challenge response to be verified.

3.8.2.9 Challenge Data Length

The Challenge Data Length read/write register holds the length, in bytes, of the current challenge. This challenge may either be written into the CP, during Apple device authentication of an accessory, or generated by the CP during accessory authentication of an Apple device.

Before starting a challenge response-generation process on the current challenge during Apple device authentication of an accessory, this register must contain the length of the challenge.

Before starting a new challenge-generation process during accessory authentication of an Apple device, this register should contain the requested challenge length. The length must be in the range of 1 to 128 bytes, thus writing any other value will cause an error.

The required length of an authentication challenge is 20 bytes.

3.8.2.10 Challenge Data

The Challenge Data register holds the current challenge data. This data is either written into the CP or generated by the CP depending on the specific operation. The number of bytes used or generated is determined by the value of the Challenge Length Data register.

3.8.2.11 Accessory Certificate Data Length

The Accessory Certificate Data Length read-only register holds the length of the X.509 certificate that the Apple device uses to authenticate an accessory. The length of a certificate varies but is always less than or equal to 1280 bytes. This length limit may not hold for future versions of the authentication protocol.

3.8.2.12 Accessory Certificate Data

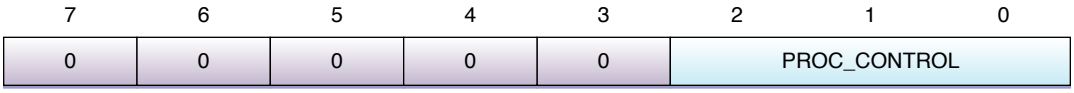
The Accessory Certificate Data read-only register holds the PKCS#7-wrapped X.509 certificate that the Apple device uses to authenticate an accessory. The Accessory Certificate may be read from the coprocessor in 128-byte pages starting at any Accessory Certificate Data Page address, or it may be read in a continuous stream starting at Page 1. Since the length of the Accessory Certificate varies, fewer than all of the pages may be used. The Accessory Certificate Data Length value can be read to determine which Accessory Certificate Data Pages contain the certificate data.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3.8.2.13 Self-Test Control and Status

The Self-Test Control and Status read/write register provides access to the built-in self-test functions of the coprocessor. When it is set to a value of 1, the Self-Test Control and Status register initiates a self-test process, as shown in Figure 3-9 (page 80) and Table 3-8 (page 80).

Figure 3-9 Coprocessor Self-Test Control and Status register, write-only bits



Note: Attempts to write other bits are ignored.

Table 3-8 Coprocessor Self-Test PROC_CONTROL values

PROC_CONTROL Value	Description
0	None
1	Run X.509 certificate and private key tests
2-7	Reserved

When read from, bits 7-4 of the Self-Test Control and Status register report the results of the X.509 certificate and private key tests, as shown in Figure 3-10 (page 80) and Table 3-9 (page 80). The CP detects a read cycle and resets the Control and Status register to 0x00 after it; hence bits 7-4 must all be retrieved in one operation.

Figure 3-10 Coprocessor Self-Test Control and Status register, read-only bits



Table 3-9 Coprocessor Self-Test Results bits

Self-Test Results Bit	Test	Meaning If 0	Meaning If 1
7	X.509 Certificate	Certificate not found	Certificate found in memory
6	Private key	Private key not found	Private key found in memory
5-4	Reserved		

Note: The X.509 and private key tests only verify that these elements are present in Flash memory; no authentication is performed.

3.8.2.14 System Event Counter

The System Event Counter (SEC) is a non-volatile register that holds the current value of the CP's event counter. The event counter automatically decrements one count per second while the CP is powered, stopping at 0. If the accessory controls power to the CP, it must wait until the SEC has decremented to 0 before removing power.

3.8.2.15 Accessory Certificate Serial Number

The Accessory Certificate Serial Number register holds the serial number of the X.509 certificate that the Apple device uses to authenticate an accessory. The certificate serial number is a null-terminated string, with a maximum length of 31 bytes (inclusive of the null character).

3.8.2.16 Apple Device Certificate Data Length

The Apple Device Certificate Data Length register holds the length of the X.509 certificate supplied by the attached Apple device. An accessory uses this certificate to authenticate an Apple device in both the certificate validation and challenge response verification processes. The length of an Apple device certificate varies but is always less than or equal to 1024 bytes. This length limit may not hold for future versions of the authentication protocol.

Writing a value in a range greater than 0 and less than or equal to 1024 will cause the CP to validate the data contained in the iPod Certificate Data registers. If the CP invalidates the iPod Certificate Data, it sets this register to 0.

3.8.2.17 Apple Device Certificate Data

The Apple Device Certificate Data register holds the X.509 Certificate that an accessory uses to authenticate an Apple device in both the certificate validation and challenge response verification processes. The Apple Device Certificate may be written to the coprocessor in 128-byte pages starting at any Apple Device Certificate Data Page address, but it may not be written in a multipage stream. Since the length of the Apple Device Certificate varies, not all of the pages need to be used. The Apple Device Certificate Data Length value determines which Apple Device Certificate Data Pages contain valid certificate data.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3.9 Coprocessor 2.0C I2C Protocol

The Apple Authentication Coprocessor (CP) supports the I2C communication protocol, acting as an I2C slave. Its SCL signal is the I2C clock line and is driven by the accessory. Its SDA signal is the I2C data line and is driven by whichever device is currently sending data.

Unlike the Apple Authentication Coprocessor version 2.0B, CP 2.0C does not perform clock synchronization by stretching SCL. It may, however, not-acknowledge (NACK) a requested register operation if busy, so the I2C master should expect retry operations as a normal part of CP 2.0C use.

The maximum supported I2C clock rate is 400 kHz.

3.9.1 Slave Selection and Reset

During reset, the RST signal must specify the CP's I2C slave address and must be held stable for at least 10 *ms* after power-up or reset, as described in I2C Communications Process (page 71). As an I2C slave, the CP is then selected in-band via its I2C address. The least significant bit of the I2C slave address controls whether a write or a read operation is to be performed, as described in Coprocessor 2.0C Address Selection (page 67).

3.9.2 Coprocessor Busy

When the CP is busy processing it is unable to handle incoming communication attempts. If the coprocessor does not ACK its slave address during an attempted I2C communication, then the coprocessor is busy. The accessory must repeatedly attempt communication until the coprocessor sends an ACK after receiving its slave address.

3.9.3 Writing to the Coprocessor

To write data to the coprocessor, follow these steps:

1. Send the I2C start sequence.
2. Send the I2C write address of the CP.
3. Check for an ACK from the slave; if a NACK is received, wait 500 µ*s* and then loop back to Step 1.
4. Send the register address at which to begin writing.
5. Send the data bytes.
6. Send the I2C stop sequence.

3.9.4 Reading from the Coprocessor

To read data from the coprocessor, follow these steps:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1. Send the I2C start sequence.
2. Send the I2C write address of the CP.
3. Check for an ACK from the slave; if a NACK is received, wait 500 µs and then loop back to Step 1.
4. Send the register address at which to begin reading.
5. Optional: send the I2C stop sequence.
6. Send the I2C start sequence.
7. Send the I2C read address of the CP.
8. Check for an ACK from the slave; if a NACK is received, wait 500 µs and then loop back to Step 6.
9. Read the data bytes.
10. Send the I2C stop sequence.

Any additional reads after an I2C read stop sequence continue with the byte following the previous byte read until an invalid register address or an end of block is reached, at which point the slave returns 0xFF in response to all further reads.

3.10 Coprocessor 2.0C Device Characteristics

This section provides technical details and tolerances for the Apple Authentication Coprocessor 2.0C (CP) chip.

3.10.1 Physical Configuration

Figure 3-11 (page 84) shows the CP package's layout, pin locations, and dimensional tolerances.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 3-11 Coprocessor package

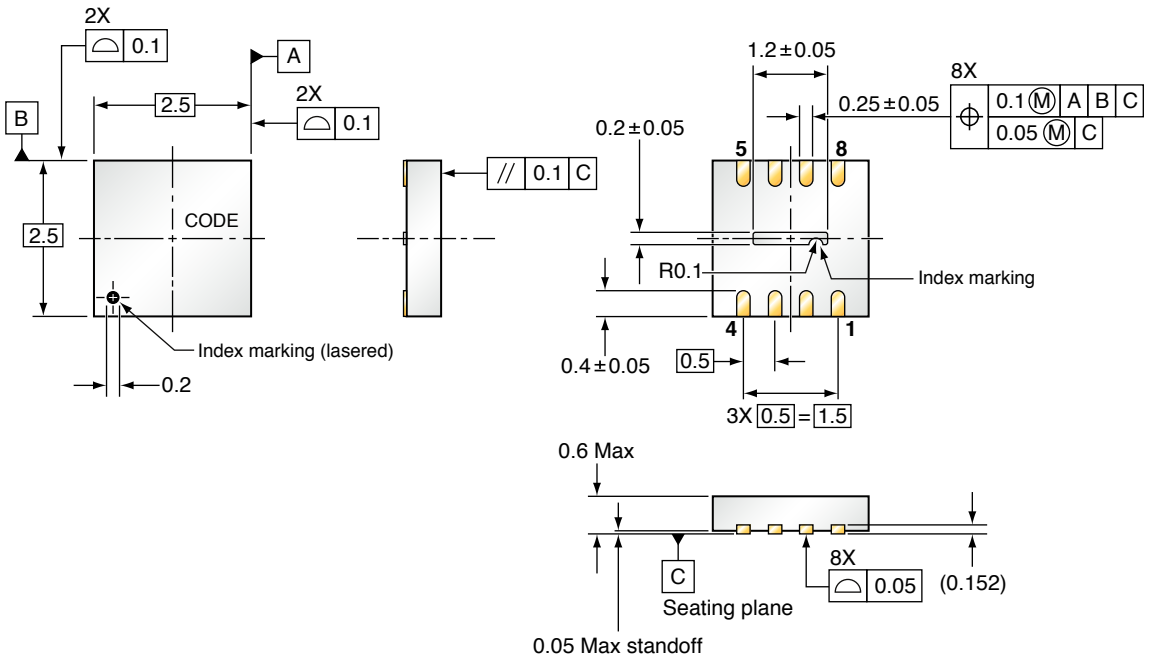


Figure 3-12 (page 85), Figure 3-13 (page 86), Figure 3-14 (page 87), Figure 3-15 (page 88), and Figure 3-16 (page 89) describe how the CP is delivered to accessory manufacturing facilities.

Figure 3-12 Coprocessor Packing Carrier Tape

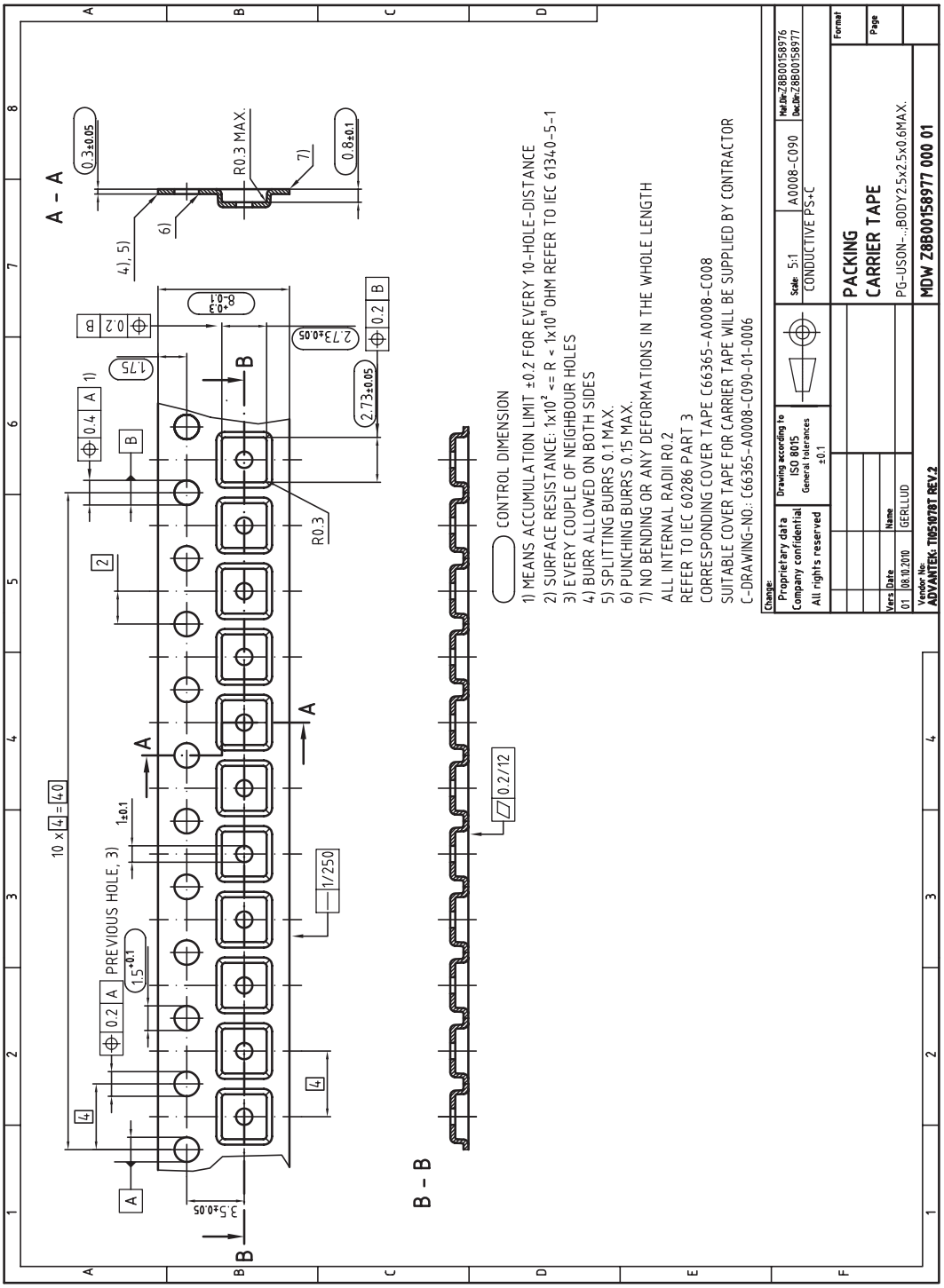


Figure 3-13 Coprocessor Packing Reel (1,000 unit)

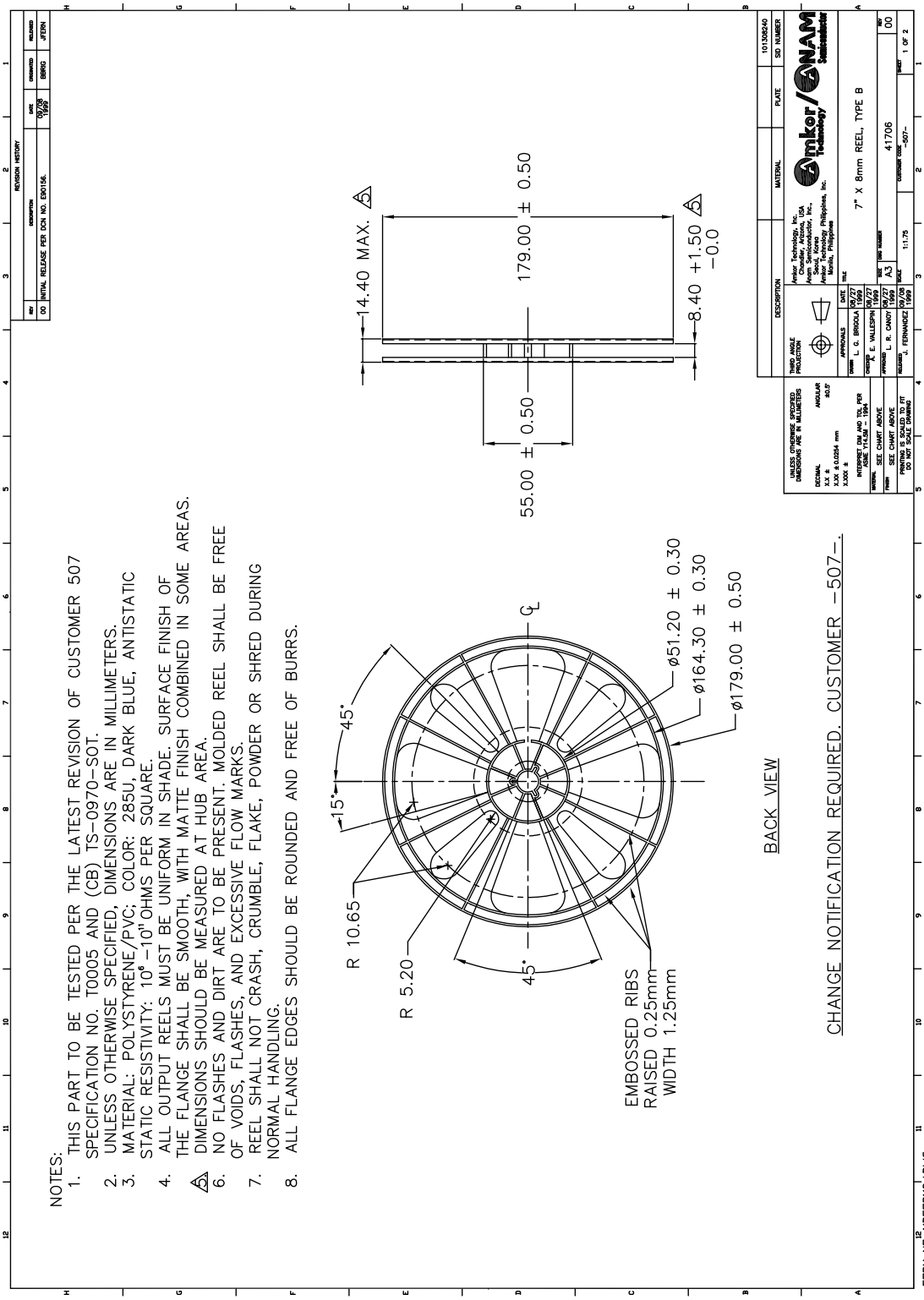


Figure 3-14 Coprocessor Packing Reel (1,000 unit) details

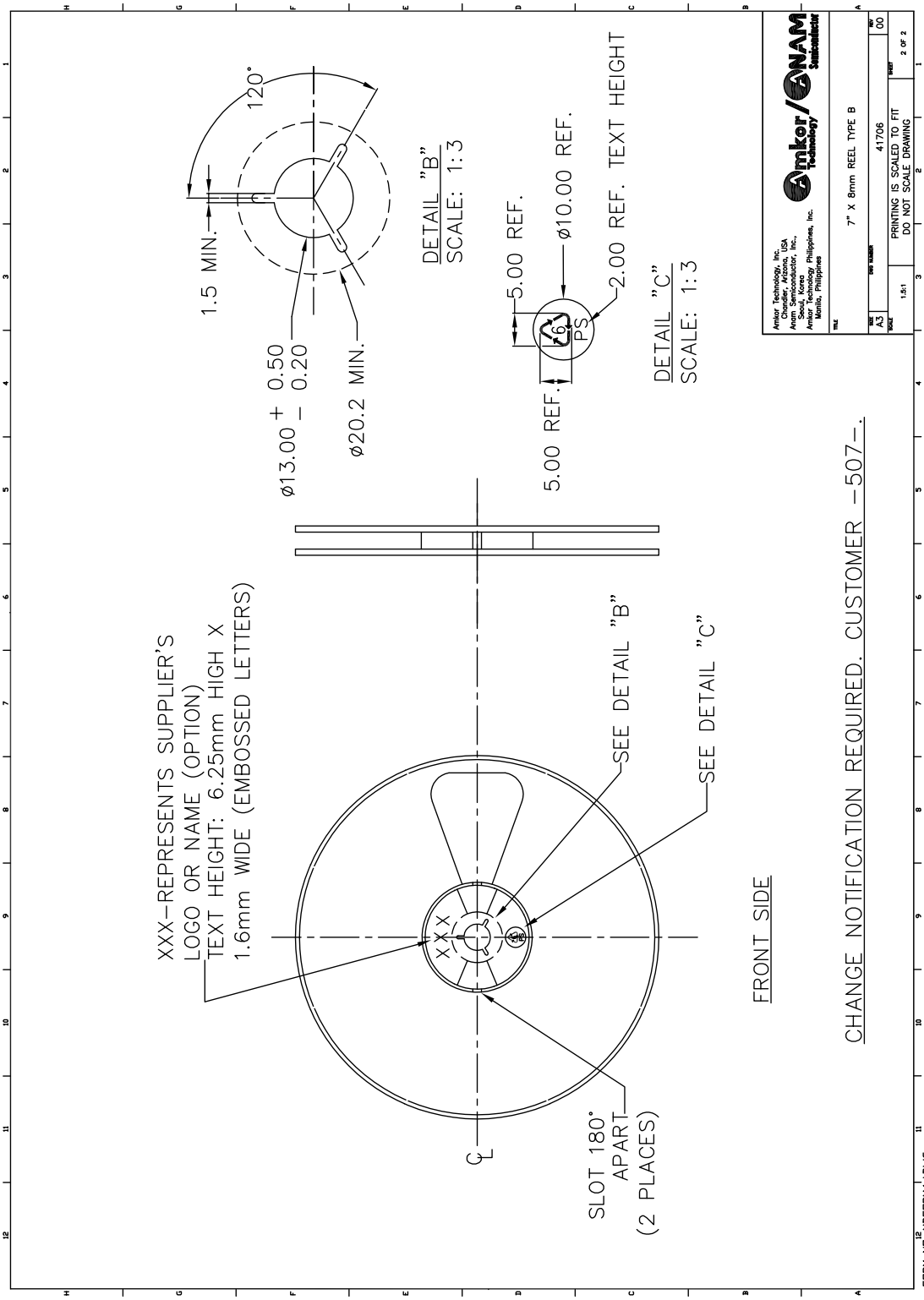


Figure 3-15 Coprocessor Packing Reel (10,000 unit)

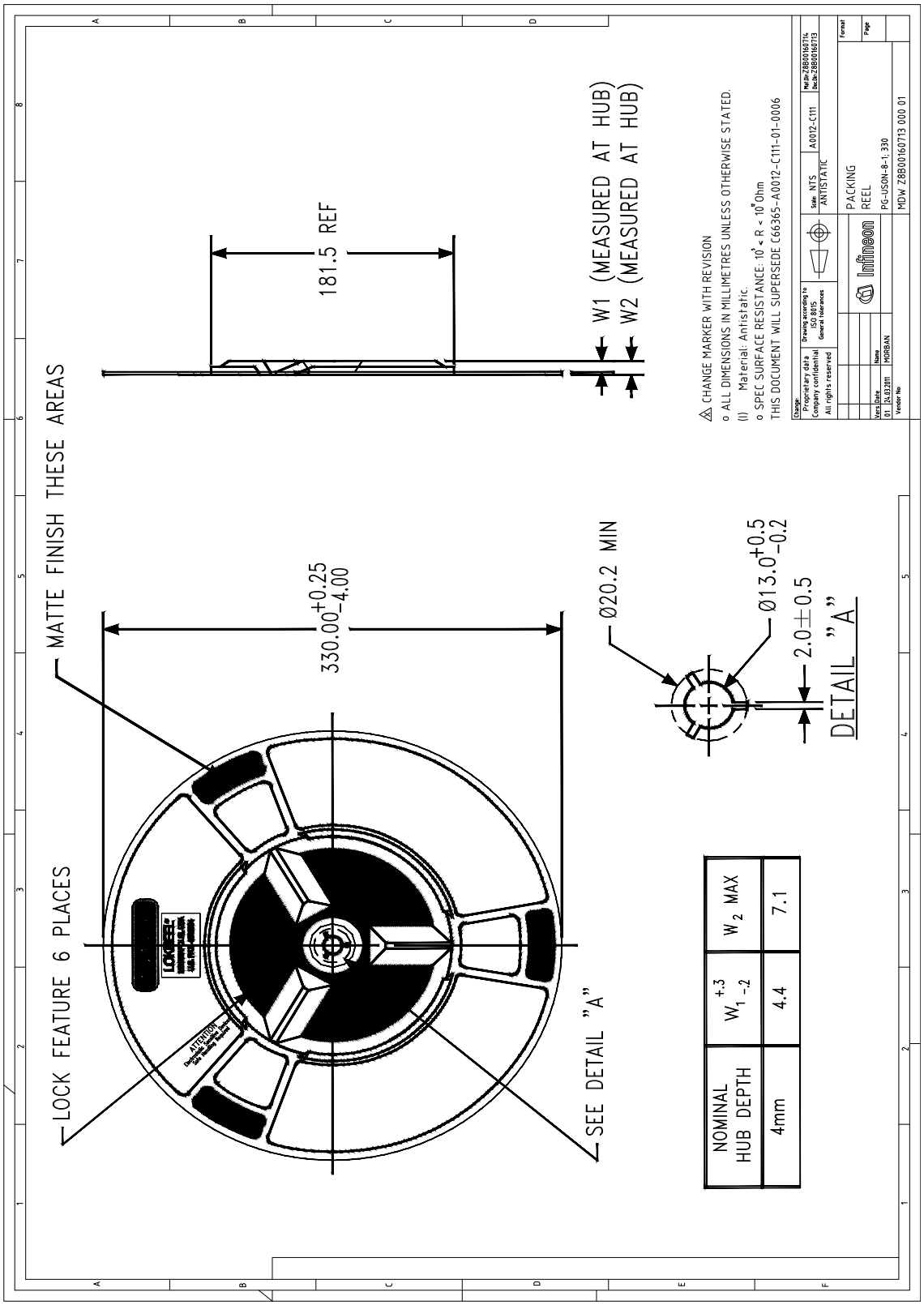
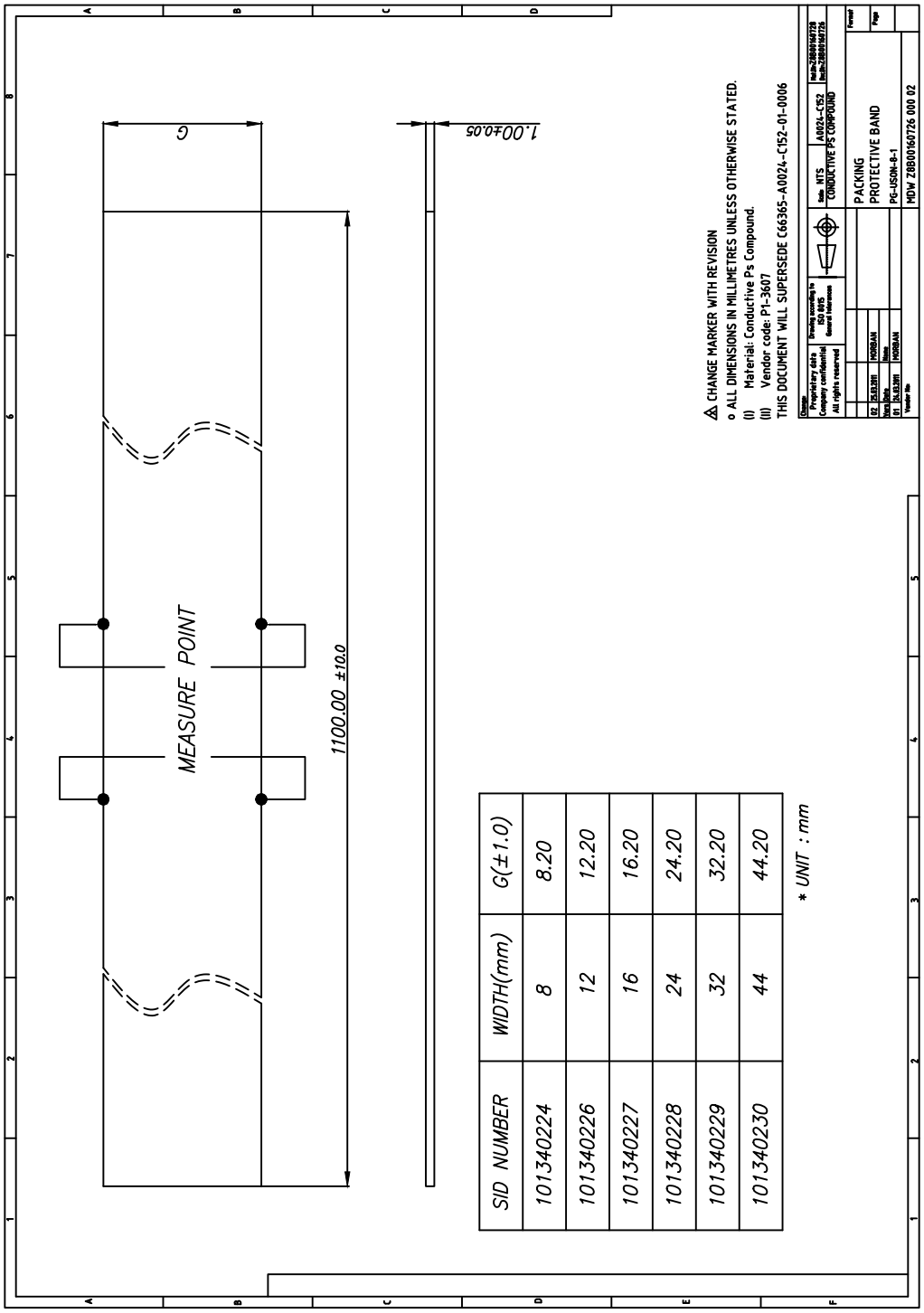


Figure 3-16 Coprocessor Packing Protective Band



SID NUMBER	WIDTH(mm)	G(±1.0)
101340224	8	8.20
101340226	12	12.20
101340227	16	16.20
101340228	24	24.20
101340229	32	32.20
101340230	44	44.20

3.10.2 Maximum Environmental Conditions

Table 3-10 (page 90) lists the CP's absolute maximum electrical and free-air temperature ranges. Stresses to the CP chip beyond the ranges listed in Table 3-10 (page 90) may cause permanent damage. Exposure to either end of any range for extended periods may affect device reliability.

Table 3-10 Coprocessor maximum electrical and temperature ranges

Condition	Maximum Range
Voltage applied at V_{CC} relative to V_{SS}	-0.3 V to +7.0 V
Voltage applied to any pin	-0.3 V to V_{CC} + 0.3 V
Storage temperature	-40 to +125 °C

3.10.3 Recommended Operating Conditions

The CP is available only in a standard temperature range configuration. Internal sensors may force it to its reset state if any of the conditions listed in Table 3-11 (page 90) are exceeded. Attempting to operate the CP in this state is not recommended and may lead to device failure or unreliability.

Table 3-11 Coprocessor maximum electrical and temperature ranges

Condition	Recommended Range
Operating free-air temperature	-25 to +85 °C
Supply voltage during program execution	1.62 to 5.5 V

3.10.4 I2C Interface Characteristics

Table 3-12 (page 90) specifies the limits of the I2C interface between the CP and other components.

Table 3-12 Coprocessor I2C interface characteristics

Parameter	Required Range
SCL clock frequency (f_{SCL})	10 to 400 kHz
External bus capacitance to ground (C_b)	Maximum 100 pF

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3.10.5 DC Electrical Characteristics

Table 3-13 (page 91), Table 3-14 (page 91) and Table 3-15 (page 91) show the DC electrical characteristics of the CP chip over its recommended voltage and temperature ranges. Unless otherwise specified in these tables, V_{CC} = 1.62 to 5.5 V and T_A = -25 to +85 °C.

Table 3-13 Coprocessor supply current into V_{CC}, excluding external current

Parameter	Test Conditions	Minimum	Typical	Maximum	Unit
$I_{(AM)}$ Active mode (authentication process running)			6	7.5	mA
$I_{(sleep)}$ Sleep mode	T_A = 25 °C		35	80	µA

Table 3-14 Coprocessor inputs

Symbol	Parameter	Minimum	Typical	Maximum	Unit
V_{IH}	High-level input voltage	$V_{CC} \times 0.7$		$V_{CC} + 0.3$	V
V_{IL}	Low-level input voltage	-0.3		$V_{CC} \times 0.2$	V
I_i	Input current	-10	1	10	µA

Table 3-15 Coprocessor outputs

Symbol	Parameter	Test Conditions	Minimum	Maximum	Unit
V_{OL}	Low-level output voltage	$I_{OL(max)}$ = -1 mA	V_{SS}	V_{SS} + 0.4	V

3.10.6 Timing Characteristics

When power is turned on to the CP, V_{CC} must reach 90% of the CP's target supply voltage within 200 µs after it exceeds 400 mV.

Figure 3-17 (page 92) illustrates the CP's typical I/O port input signal timing and voltage limits. Table 3-16 (page 92) lists the parameter values in Figure 3-17 (page 92).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 3-17 Coprocessor typical I/O port input waveform

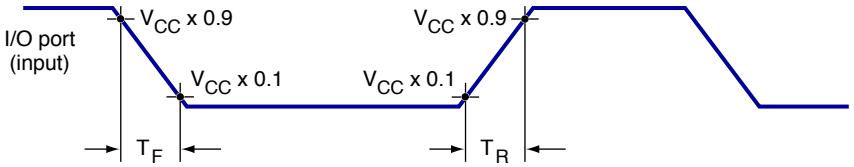


Table 3-16 Coprocessor Values for typical I/O port input waveform

Symbol	Description	Maximum Value
T_F	Fall time	1.0 µs
T_R	Rise time	1.0 µs

4. Apple Lightning Audio Module

Accessories may make use of the Apple Lightning Audio Module (LAM) to create the following types of accessories that plug into the Lightning connector on Apple devices running iOS 7.1 or later:

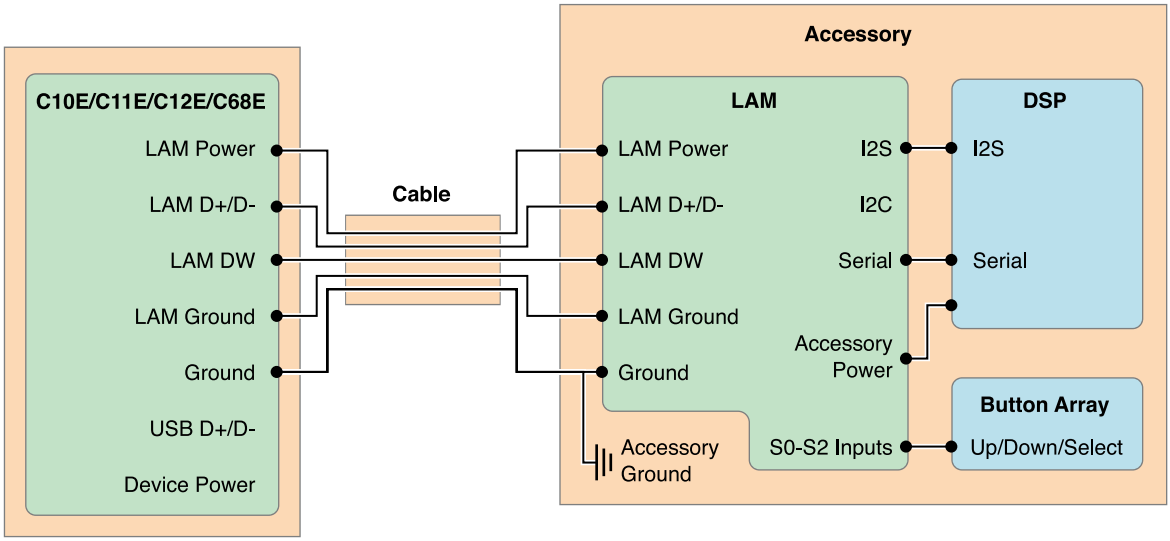
- Headsets (see Headsets (page 549)).
- Accessories with an integrated 3.5 mm headset jack, but not integrated speakers.
- Non-audio accessories.

Accessories using the LAM module may:

- Draw power from the Apple device, even when asleep, and eliminate the need for an internal battery.
- Receive lossless stereo 48 kHz digital audio output from the Apple device.
- Send mono 48 kHz digital audio input to the Apple device (see I2S (page 106)).
- Provide power to the Apple device from an internal battery and/or a pass-through external power source.
- Implement Apple Headset Remote Volume Up/Volume Down/Center buttons.
- Implement additional buttons that work with iTunes Radio and other iOS media playback features.
- Declare a preferred iOS app that works with the accessory.
- Communicate with the preferred iOS app via an External Accessory Protocol.
- Request that the preferred iOS app be launched on the Apple device.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 4-1 LAM Accessory without Power



Accessory designs that implement an integrated 3.5 mm headset jack but do not include integrated speakers may also use the Lightning Audio Module to pass analog audio to and from the Apple device. These accessories must comply with the following requirements:

- They may physically block or occlude the Apple device's headset jack.
- They must use a 4-pole 3.5 mm headset jack.
- They must support both audio input and output.
- They must not physically block or occlude the Apple device's built-in microphones or speakers.
- They must set the Audio Output Terminal Type and Audio Input Terminal Type to Headset Jack. See accAudioTerminalInformation (0x0B) (page 114).
- They must use the accAudioTerminalStateInformation (0x23) (page 121) packet to indicate when headsets/microphone are plugged into or unplugged from the accessory headset jack.

Accessory designs that do not integrate audio features may also use the Lightning Audio Module. These accessories must comply with the following requirements:

- They must not physically block or occlude the Apple device's built-in microphones, speakers, or headset jack.
- They must set the Audio Output Terminal Type and Audio Input Terminal Type to None. See accAudioTerminalInformation (0x0B) (page 114).
- They must set the Audio Output Terminal State and Audio Input Terminal State to Disabled. See accAudioTerminalStateInformation (0x23) (page 121).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

4.1 Overview

Key accessory-facing interfaces of the Lightning Audio Module are:

- I2S audio (Host or Slave)
- Serial
- 3 digital inputs (S0, S1, and S2 buttons)
- Accessory Power

4.2 Firmware Updates

LAM accessories must be capable of field-deployable firmware updates from Apple devices running iOS and/or Macintosh computers running OS X.

4.3 Authentication

Accessories that integrate the Lightning Audio Module do not need to integrate an Apple Authentication Coprocessor.

4.4 Lightning Connector

Accessories that integrate the Lightning Audio Module must attach to the Apple device using either a Lightning (C10E), Lightning (C11E), Lightning (C12E) or Lightning (C68E) connector (see Apple Lightning Connector (page 128)).

4.5 Mechanical

The Lightning Audio Module has the following mechanical characteristics:

- 7.6 mm x 7.6 mm LGA package on tape and reel
- Not encapsulated in order to maximize compatibility with pick and place operations
- Pickup is offset to the CSP surface
- A1 pad has a fiducial mark that may be used as a reference
- FR4 PCB material
- SAC305 solder is used. Solder reflow operations must take this into consideration

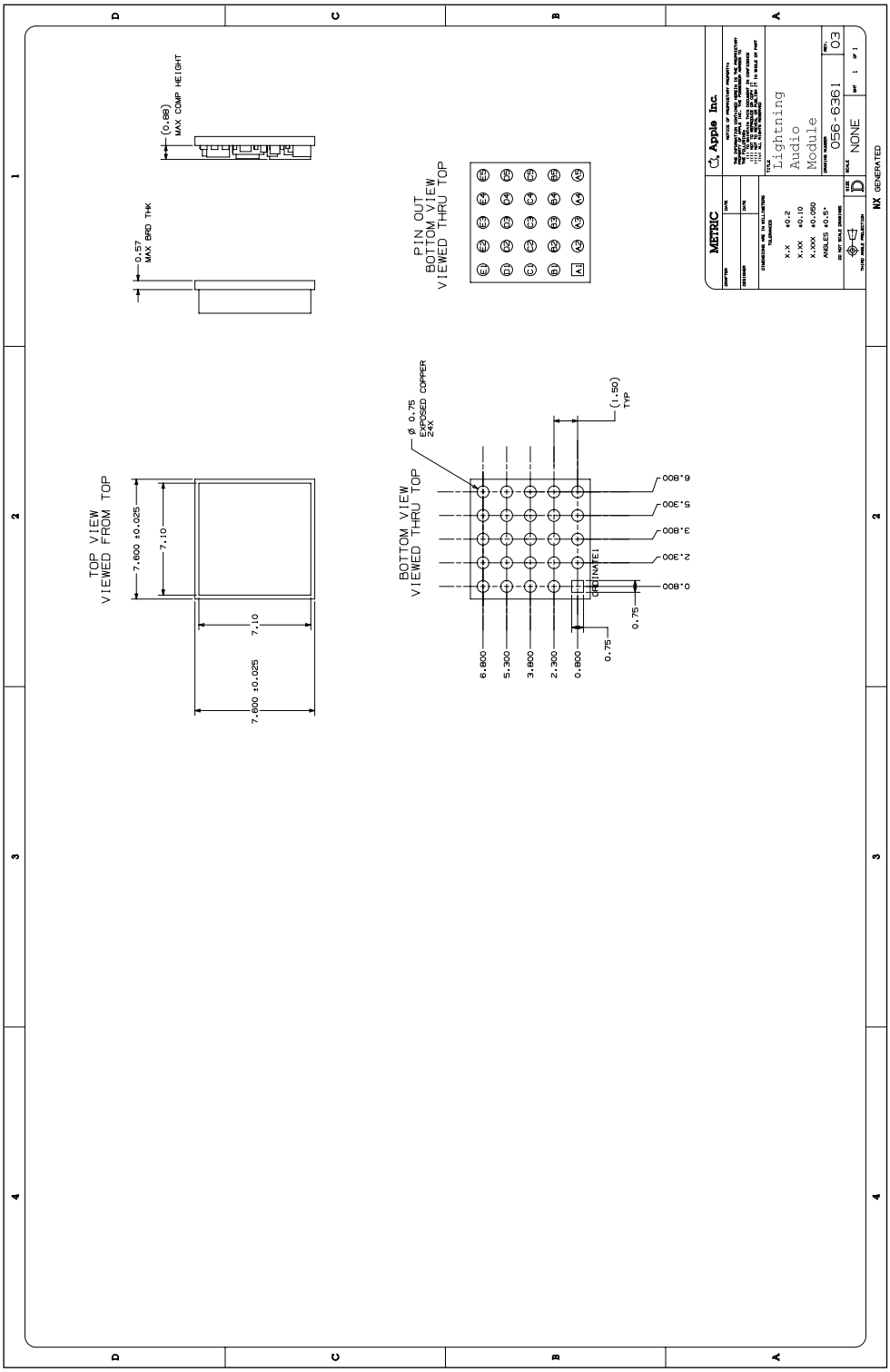
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- 0-70 ℃ working temperature range

> **Note:** Encapsulation of the Lightning Audio Module is recommended to pass salt spray environmental testing.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 4-2 Lightning Audio Module Mechanical Package

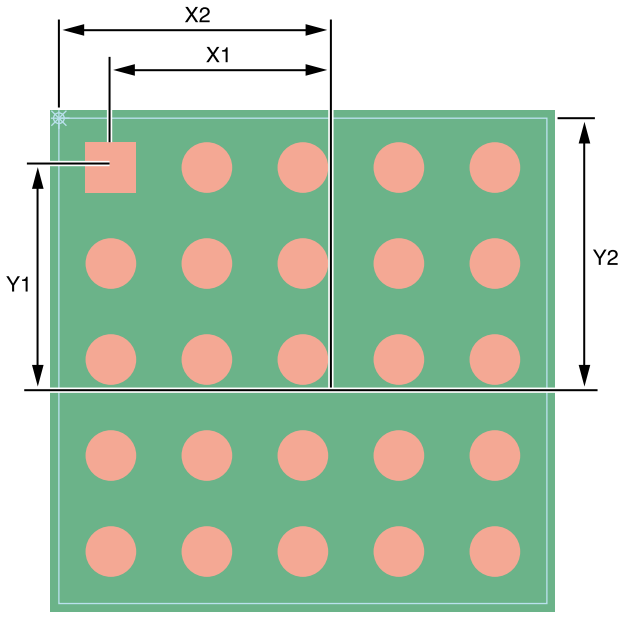


Pickup offsets for the Lightning Audio Module (see Figure 4-3 (page 98)) are:

- (X1) 3.4635075 mm

- (Y1) 3.513225 mm
- (X2) 4.258955 mm
- (Y2) 4.30867 mm

Figure 4-3 Lightning Audio Module pickup offsets (bottom view)



4.6 Pad Layout and Assignments

The following diagrams detail the layout, names, and description of the Lightning Audio Module LGA pads. All Reserved pads must be soldered but left Not Connected (NC).

Figure 4-4 Lightning Audio Module pad layout (top view)

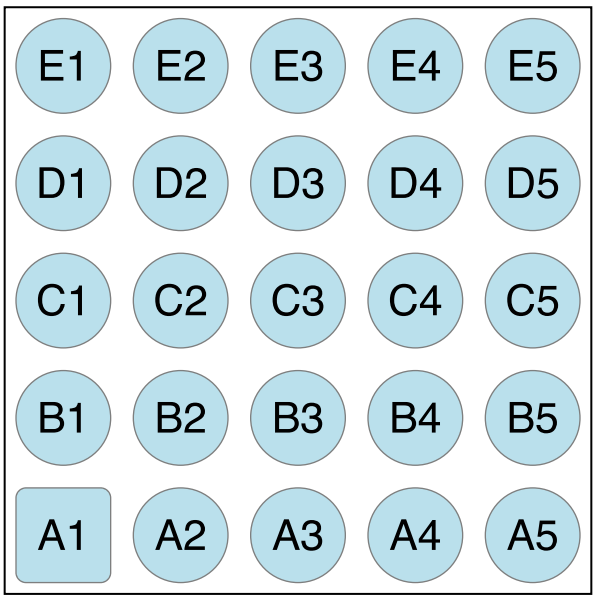


Table 4-1 Lightning Audio Module pad assignments

Pad	Name	Direction	Assignment
A1	Ground	Power Out	Cable (page 102)
A2	Reserved	N/A	
A3	S2	Input	S0-S2 Headset Remote Inputs (page 108)
A4	Reserved	N/A	
A5	LAM Power	Power In	Cable (page 102)
B1	Accessory Power	Power Out	Accessory Power (page 104)
B2	I2S MCLK	Output	I2S (page 106)
B3	S1	Input	S0-S2 Headset Remote Inputs (page 108)
B4	I/O VREF	Output	Input/Output Electrical Characteristics (page 100)
B5	LAM D+	Bi-Direction	Cable (page 102)
C1	Serial TX	Output	Serial (page 108)
C2	I2S SCLK	Output	I2S (page 106)
C3	S0	Input	S0-S2 Headset Remote Inputs (page 108)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Pad	Name	Direction	Assignment
C4	Reserved	N/A	
C5	LAM D-	Bi-Direction	Cable (page 102)
D1	Serial RX	Input	Serial (page 108)
D2	I2S LRCLK	Output	I2S (page 106)
D3	Reserved	N/A	must be connected to ground via a 1 kΩ -10 kΩ resistor.
D4	Reserved	N/A	
D5	LAM Ground	Power In	Cable (page 102)
E1	I2S SDOUT	Output	I2S (page 106)
E2	I2S SDIN	Input	I2S (page 106)
E3	I2S ERROR	Output	I2S (page 106)
E4	LAM DW	Input	Cable (page 102)
E5	Ground	Power Out	Cable (page 102)

4.7 Input/Output Electrical Characteristics

The following electrical characteristics apply to the Lightning Audio Module I2S pads (with V_{DD} = 1.85 V. $I_{output\text{-}high}$ = -100 uA and $I_{output\text{-}low}$ = 100 uA, $T_{ambient}$ = 25 °C):

Table 4-2 Lightning Audio Module I2S pad electrical characteristics

Characteristic	Min	Max
High-level input	1.26 V	2.10 V
Low-level input	-0.3 V	0.54 V
High-level output	1.60 V	
Low-level output		0.20 V

The following electrical timing characteristics apply to the Lightning Audio Module I2S pads:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 4-3 Lightning Audio Module I2S electrical timing characteristics

Characteristic	Min	Max
MCLK frequency	6.114 MHz - 80 ppm	6.114 MHz + 80 ppm
MCLK duty cycle	45%	55%
MCLK peak-to-peak absolute jitter		1 ns
Input sample rate (LRCK)	47.52 kHz	48.48 kHz
LRCK duty cycle	45%	55%
SCLK frequency		3.072 MHz
SCLK duty cycle	45%	55%
SCLK rising edge to LRCK edge	10 ns	
LRCK setup time before SCLK rising edge	40 ns	
SDOUT setup time before SCLK rising edge	20 ns	
SDOUT hold time after SCLK rising edge	30 ns	
SDIN setup time before SCLK rising edge	20 ns	
SDIN hold time after SCLK rising edge	20 ns	

The following electrical characteristics apply to all other Lightning Audio Module input/output pads:

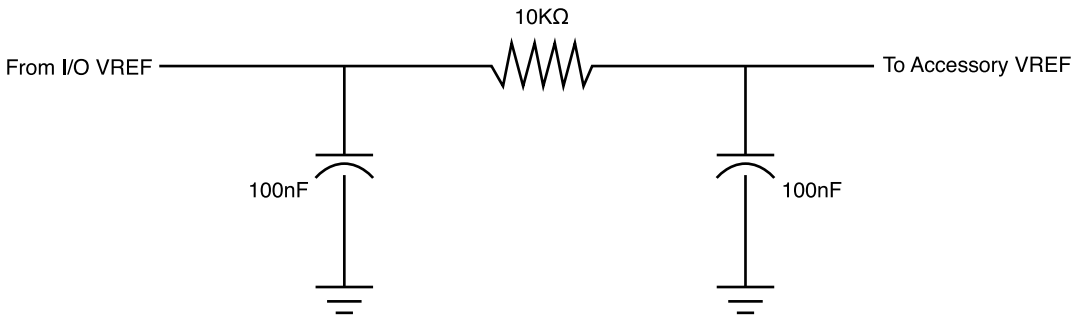
Table 4-4 Lightning Audio Module I/O pad (except I2S) electrical characteristics

Characteristic	Min	Max	Notes
High-level input	1.26 V	2.10 V	
Low-level input	-0.30 V	0.54 V	
High-level output	1.35 V		Max 1 mA, V_{DD} = 1.8 V
Low-level output		0.45 V	Max 2 mA, V_{DD} = 1.8 V

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

I/O VREF is a 1.8 V output and may be used as a voltage reference for the I/O level when the accessory's circuits operate with voltage levels different from the CMOS 1.8 V expected by the Lightning Audio Module. The accessory must not draw power from this pad. A RC filter as shown in Figure 4-5 (page 102) must be added before connecting it to the accessory's reference input.

Figure 4-5 I/O VREF RC circuit



The Serial RX input has the following additional characteristics:

Table 4-5 Lightning Audio Module Serial RX electrical characteristics

Characteristic	Typical	Worst Case
Capacitance	12 pF	15 pF
Input leakage	300 nA	1 uA

4.8 Cable

Accessory cables or interconnects that connect a Lightning (C10E) or Lightning (C68E) connector to a Lightning Audio Module must comply with the requirements in this section.

The following signals must be connected from the Lightning Audio Module to the Lightning (C10E) or Lightning (C68E) connector using separate conductors.

- LAM Power
- LAM D+
- LAM D-
- LAM Ground
- LAM DW
- Ground

Copyright © 2015 Apple Inc. All Rights Reserved.

There is one exception. If all of the following are true:

- The accessory does not provide power to the Apple device (see Device Power (page 105)).
- Ground DCR ≤ 0.4 Ω .
- The maximum current through this path is 100 mA.

Then,

- LAM Ground and Ground conductors may be omitted from the cable or interconnect.
- LAM Ground and Ground pads must be connected to a cable shield drain wire.

LAM Ground must only be used as a non-current-carrying ground reference.

The following requirements also apply:

- LAM D+ and LAM D- must be a twisted pair and shielded using LAM Ground or Ground (if LAM Ground is not present).
- LAM D+ and LAM D- differential impedance nominally 90 Ω (85 Ω min, 95 Ω max).
- LAM Power DCR ≤ 1.0 Ω .

4.9 Power

4.9.1 Power States

The Lightning Audio Module will mirror Apple device power states with the following exceptions:

- If configured to do so, the Lightning Audio Module will continue to provide power to the Lightning accessory even when the Lightning Audio Module and Apple device are sleeping.
- Direct user action on button S0 will wake the Lightning Audio Module and Apple device from a sleep state.
- Incoming packets from a LAM accessory via the serial protocol (see Serial Protocol (page 109)) will wake the Lightning Audio Module, but not the Apple device, from a sleep state.
- accHIDReport serial protocol packets from a LAM accessory (see accHIDReport (0x21) (page 120)) will wake the Lightning Audio Module and Apple device from a sleep state.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: Lightning accessories must not force the Apple device to wake from a sleep state autonomously or prevent it from entering a sleep state. This is grounds for failure to pass self certification.

4.9.2 Module Power

The Lightning Audio Module is powered from the Lightning (C10E) or Lightning (C68E) connector. Lightning accessories do not need to provide power to the Lightning Audio Module.

4.9.3 Accessory Power

Accessories may draw power from the Apple device via the Lightning Audio Module's Accessory Power pad. Additionally, they may configure the Lightning Audio Module to provide constant Accessory Power even when the Apple device is in a sleep state. This may enable them to implement advanced features such as Active Noise Cancellation (ANC) without incorporating an internal power source. The Accessory Power pad must not be used to charge an accessory's internal battery.

Note: Accessories must not draw power directly from the LAM Power pad on the Lightning (C10E) or Lightning (C68E) connector; all power must be drawn from the Accessory Power pin on the Lightning Audio Module.

Apple strongly recommends keeping the power draw as low as possible to maximize the lifetime of the Apple device's internal battery. If the accessory does not perform any user-visible or user-audible function (such as ANC) when the Apple device is asleep, it must not configure the Lightning Audio Module to provide constant Accessory Power. Instead, the Lightning Audio Module will stop providing Accessory Power when audio streaming has stopped or the Apple device enters a sleep state. It will re-enable Accessory Power when audio streaming restarts.

Accessories that configure the Lightning Audio Module to provide constant accessory power should implement the following steps to minimize power draw:

- When the Audio Output Stream Status field in the lamInformation packet (see lamInformation (0x10) (page 117)) is 0, the accessory should power down amplifiers, DACs, and other audio processing circuits but leave its UART operational and ready to process incoming packets.
- The accessory should monitor the SDA pad as specified in I2S (page 106). If SDA is low for more than 500 ms, the accessory may enter a deep sleep state, and should exit that state when SDA is subsequently driven high.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The voltage supply range from the Accessory Power pad is 2.5 V-4.7 V (3.3 V typical).

- Average current draw from the Accessory Power pad must not exceed 50 mA.
- Peak current draw from the Accessory Power pad must not exceed 100 mA with one exception. When the Lightning Audio Module enables Accessory Power, the accessory may exceed the 100 mA current limit until 5 ms has elapsed or 100 uF of capacitance has been charged, whichever comes first.

I2S MCLK will stay active as long as the Lightning Audio Module provides Accessory Power.

> **Note:** To minimize the risk of audible artifacts, Lightning accessories must immediately mute all audio interfaces when they detect a loss of Accessory Power.

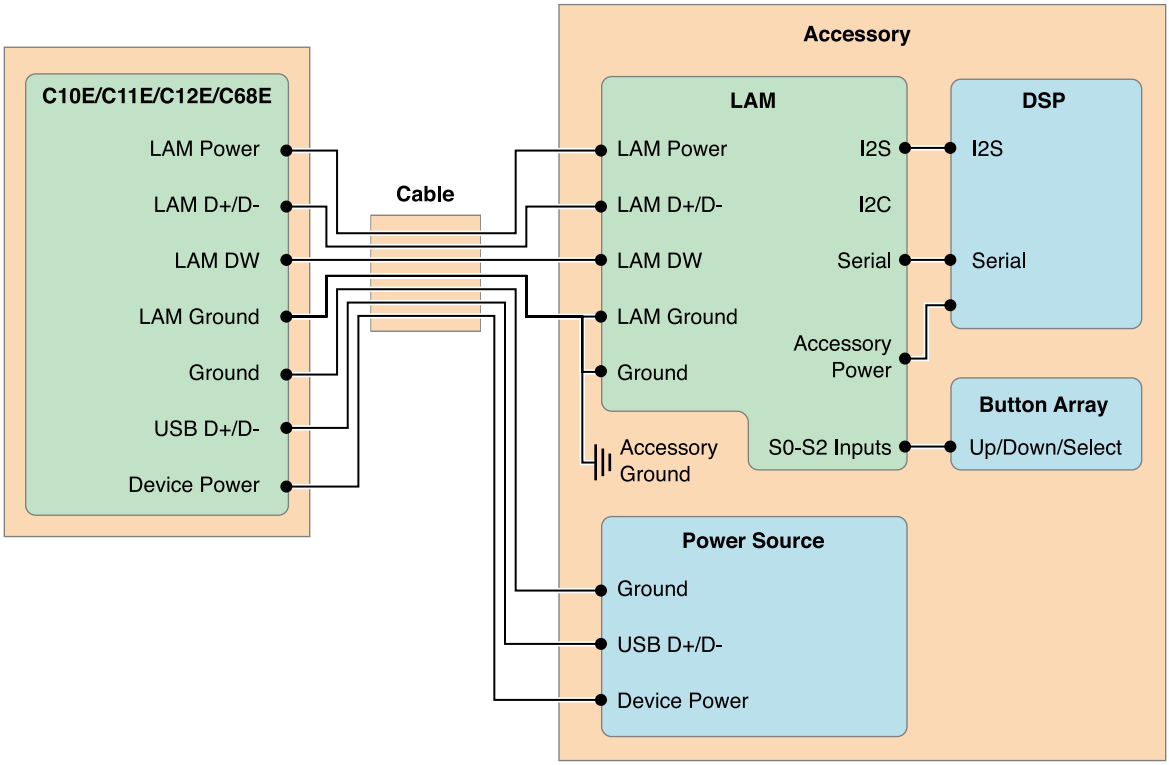
4.9.4 Device Power

Lightning Audio Module accessories may optionally provide power to the Apple device via the Lightning (C10E) or Lightning (C68E)'s Device Power, USB D+, USB D-, and Ground pads from an internal or external USB power source. The power source must communicate how much power is available to the Apple device using D+/D- resistors.

Additionally, the power source's ground must be directly connected to the Ground pad on the Lightning (C10E) or Lightning (C68E) connector and isolated from LAM Ground as shown:

Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 4-6 LAM Accessory with Power



If a LAM accessory has an internal battery, it may draw some current from an external power source to recharge that battery while the Apple device is also charging, but must inform the Apple device of the current draw using accPassthroughPowerSourceInformation (0x24) (page 122). Additionally, the amount of current that may be drawn depends on the Apple device's power state as specified in lamInformation (0x10) (page 117).

4.10 Audio

4.10.1 I2S

LAM accessories may consume audio from the Lightning Audio Module via I2S. The Lightning Audio Module can either be a I2S host or I2S slave. I2S slave mode is the default.

The I2S audio interface consists of five signals:

- *MCLK*: Master clock
- *SCLK*: Serial data shift clock
- *LRCLK*: Left/right clock
  - Identifies where each channel (left or right) is located within the data word

- Toggles at the audio sampling rate

- *SDOUT*: Serial data output
- *SDIN*: Serial data input

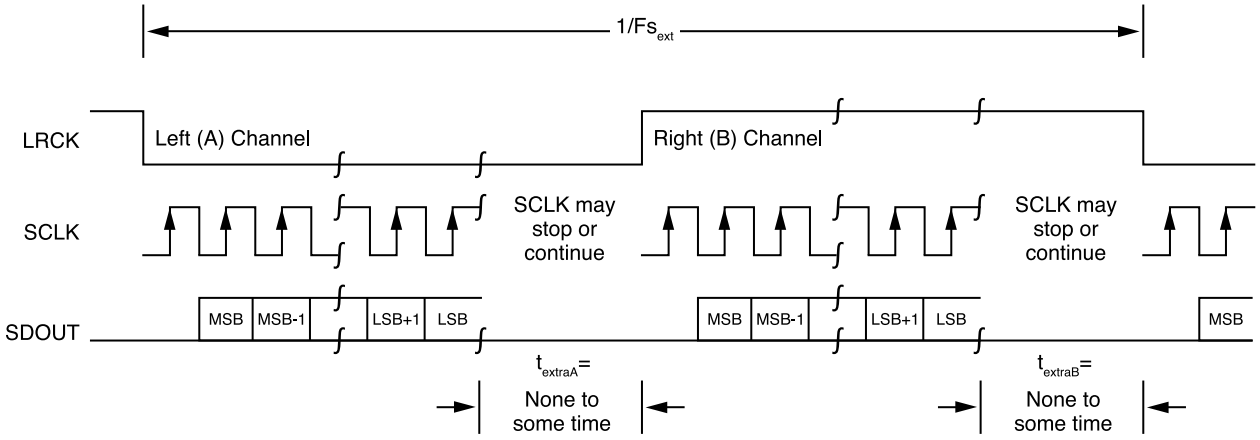
When the I2S audio interface is operating in slave mode, SCLK, and LRCK are inputs. Conversely, SCLK and LRCLK are outputs when the I2S audio interface is operating in host mode. LRCLK is 48 kHz. MCLK is always an output regardless of slave or host mode and is 6.144 MHz ±80 ppm.

The I2S interface only supports a 48 kHz sample rate.

The I2S data stream has the following characteristics:

- Up to 24 bits/sample of stereo data may be transported.
- LRCK identifies the start of a new sample word and the active stereo channel (A or B).
- Data is clocked out of the SDOUT output using the falling edge of SCLK.
- Bit order is MSB to LSB.

Figure 4-7 Lightning Audio Module I2S Format



LAM accessories may treat a low level on the I2S_ERROR pad to indicate possible errors in the I2S data stream. Apple recommends that this signal be used to mute the audio DAC. The signal will remain low until the I2S data stream is deemed free of errors by the Lightning Audio Module. An external pull-up resistor must be attached to the I2S_ERROR pad if it is used in this manner. A resistor value of 10 kΩ - 47 kΩ is recommended. If the I2S_ERROR pad is not used in this manner, it must be connected to ground via a 1 kΩ -10 kΩ resistor.

If the accessory does not contain a microphone, it must connect SDIN to ground. If it does contain a microphone, the I2S audio input stream must comply with all of the following requirements:

- The accessory must natively sample audio at a rate of at least 48 kHz.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Each sample must have at least 16 bits of data. 24 is recommended.
- The data on left and right channels must be identical.

4.11 Control

Lightning accessories must take one of two possible approaches for implementing user controls.

4.11.1 S0-S2 Headset Remote Inputs

Lightning accessories that make use of the S0-S2 headset remote inputs must connect to the following Lightning Audio Module pads:

- *S0*: Center
- *S1*: Volume Down
- *S2*: Volume Up

The S0-S2 inputs default to a high impedance input "1". A button press corresponds to "0". All S0-S2 activity must occur as a result of direct user action.

The Lightning Audio Module provides internal pull-ups for S0, S1, and S2. No external pull-ups are required.

4.11.2 HID Headset Remote

LAM accessories that implement a HID Headset Remote component (see HID Headset Remote (page 616)) must register a HID descriptor with the Lightning Audio Module and send HID usage reports to Apple devices using the serial protocol (see accHIDComponentInformation (0x0A) (page 113) and accHIDReport (0x21) (page 120)). All HID usage reports must be generated as a result of direct user action.

When the Apple device is in a sleep state, LAM accessories must only send an accHIDReport packet if the user takes direct action to press the Center button on the headset remote. All other button presses must be ignored and accHIDReport packets must not be sent.

4.11.3 Serial

LAM accessories must use the Serial interface.

The serial interface consists of two signals:

- *RX*: Receive
- *TX*: Transmit

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The serial interface operates at either 57600 bps or 115200 bps. By default, the interface is configured to operate at 57600 bps. All Lightning Audio Module serial communications use 8 data bits, no parity bits, and one stop bit (8-N-1). The Lightning Audio Module does not support serial hardware flow controls (RTS/CTS and DTR/DSR).

The serial interface protocol offers the following features:

- Configuration of Accessory Identification information (only during manufacturing or once after firmware update)
  - Manufacturer, Model, Serial Number, and HW/FW Version
  - Preferred Team ID
  - External Accessory Protocol name
  - Accessory Power Mode
- Lightning Audio Module status updates
- External Accessory Protocol data transfer
- HID Media Remote Control user inputs
- Request Preferred App launch on Apple device
- Inform Apple device of changes in accessory configuration

4.12 Serial Protocol

4.12.1 Packet Format

The format for Lightning Audio Module serial interface packets is as follows:

Table 4-6 Lightning Audio Module serial protocol packet

Field	Bytes	Description
Wake	1	Optional. If present, must be `0xFF`
Start of Packet	1	Must be `0x55`
Length	1	Number of bytes in Type and Payload fields. Must be 129 or less.
Type	1	Packet type
Payload	0-128	Packet payload data, dependent on packet type
Checksum	1	Checksum byte

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Accessories should send the Wake byte only if the accessory is self powered or has configured the Lightning Audio Module for constant Accessory Power. In that situation, the Lightning Audio Module will make use of the Wake byte to recover from a sleep state and correctly parse the rest of the packet.

The Checksum field is calculated by adding together the values of the Length, Type, and Payload bytes as signed 8-bit values (discarding any signed 8-bit overflow) and then negating that sum to create the signed 8-bit checksum byte. The sum of all bytes in the following packet fields must add up to 0x00:

- Length
- Type
- Payload
- Checksum

All packets with an invalid checksum are presumed invalid and must be ignored by both the Lightning Audio Module and the Lightning accessory.

4.12.2 Initialization Packets

These packets confirm mutual readiness to initiate communication between the Lightning Audio Module and the Lightning accessory.

4.12.2.1 accReady (0x01)

No payload.

Before sending any other packets, the Lightning accessory must confirm the operating state of the Lightning Audio Module by sending this packet once a second until a lamReady (0x02) (page 110) packet is received in response.

4.12.2.2 lamReady (0x02)

No payload.

The Lightning Audio Module will send this packet when it has received a accReady (0x01) (page 110) packet or a configuration packet (see Configuration Packets (page 110)) from the Lightning accessory and is ready to start accepting other packets.

4.12.3 Configuration Packets

Configuration packets must only be sent during factory configuration or immediately after a firmware update.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Accessories must receive a lamReady packet from the Lightning Audio Module after every configuration packet before sending other packets.

To pass certification audit, the model name, serial number, etc. must be configured properly.

4.12.3.1 accConfigurationInformation (0x03)

Table 4-7 Lightning Audio Module accConfigurationInformation packet payload

Field	Bytes	Factory Default	Description
LAM Protocol Version	1	1	Must be set to 1
Accessory Power Mode	1	1	0 = Intermittent, 1 = Constant
I2S Role	1	0	0 = Slave, 1 = Host
Reserved	1	1	Must be set to 0
UART Baud Rate	1	0	0 = 57600 bps, 1 = 115200 bps
Hardware Major Version	1	0	
Hardware Minor Version	1	0	
Hardware Revision	1	0	
Firmware Major Version	1	0	
Firmware Minor Version	1	0	
Firmware Revision	1	0	

Changes to the UART Baud Rate will only take effect after the Lightning Audio Module has been power cycled. Accessories must receive a lamReadyPacket from the Lightning Audio Module before cycling power.

Actual data transfer rates will vary based on the packet size and configured baud rate. Generally, higher baud rates and larger packets are recommended for best results.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

4.12.3.2 accNameInformation (0x04)

Table 4-8 Lightning Audio Module accNameInformation packet payload

Field	Bytes	Factory Default	Description
Name	1-n	null	UTF-8 string

The accessory must provide a value for the Name parameter that matches the accessory's markings and packaging.

4.12.3.3 accManufacturerInformation (0x05)

Table 4-9 Lightning Audio Module accManufacturerInformation packet payload

Field	Bytes	Factory Default	Description
Manufacturer	1-n	null	UTF-8 string

The accessory must provide a value for the Manufacturer parameter that matches the accessory's markings and packaging.

4.12.3.4 accModelInformation (0x06)

Table 4-10 Lightning Audio Module accModelInformation packet payload

Field	Bytes	Factory Default	Description
Model	1-n	null	UTF-8 string

The accessory must provide a value for the Model parameter that matches the accessory's markings and packaging.

4.12.3.5 accSerialNumberInformation (0x07)

Table 4-11 Lightning Audio Module accSerialNumberInformation packet payload

Field	Bytes	Factory Default	Description
SerialNumber	1-n	null	UTF-8 string

Copyright © 2015 Apple Inc. All Rights Reserved.

The accessory must provide a value for the SerialNumber parameter that matches the accessory's markings and packaging.

> **Note:** The SerialNumber parameter must be unique for each accessory (i.e. serialized).

4.12.3.6 accPreferredAppBundleIdentifierInformation (0x08)

Table 4-12 Lightning Audio Module accPreferredAppBundleIdentifierInformation packet payload

Field	Bytes	Factory Default	Description
PreferredAppBundleIdentifier	1-n	null	UTF-8 string

If the accessory has a preferred app that communicates with the accessory using the External Accessory Protocol, it must set the PreferredAppBundleIdentifier parameter accordingly. Otherwise, the parameter must be set to a null string.

4.12.3.7 accExternalAccessoryProtocolNameInformation (0x09)

Table 4-13 Lightning Audio Module accExternalAccessoryProtocolNameInformation packet payload

Field	Bytes	Factory Default	Description
ExternalAccessoryProtocolName	1-n	null	UTF-8 string

If the accessory has a preferred app that communicates with the accessory using the External Accessory Protocol, it must set the ExternalAccessoryProtocolName parameter accordingly. Otherwise, the parameter must be set to a null string.

4.12.3.8 accHIDComponentInformation (0x0A)

Table 4-14 Lightning Audio Module accHIDComponentInformationInformation packet payload

Field	Bytes	Factory Default	Description
HID Component Function	1	0	Must be set to 8
HID Report Descriptor	2-n	null	

Copyright © 2015 Apple Inc. All Rights Reserved.

If the accessory intends to send accHIDReport packets (see accHIDReport (0x21) (page 120)), then it must set its HID component information using accHIDComponentInformation.

The HID Component Function parameter must be set to 8 during factory configuration.

4.12.3.9 accAudioTerminalInformation (0x0B)

Table 4-15 Lightning Audio Module accAudioTerminalInformation packet payload

Field	Bytes	Description
Audio Output Terminal Type	1	See Table 4-16 (page 115)
Audio Output Terminal Latency	2	Number of audio samples as uint16 (Byte 0 = bits[15:8], Byte 1 = bits[7:0])
Audio Output Terminal Gain/Attenuation Step Size	1	See Table 4-18 (page 115)
Audio Output Terminal Gain Steps	1	Number of steps - 1 (0 = 1 step, 255 = 256 steps)
Audio Output Terminal Attenuation Steps	1	Number of steps - 1 (0 = 1 step, 255 = 256 steps)
Audio Input Terminal Type	1	See Table 4-17 (page 115)
Audio Input Terminal Latency	2	Number of audio samples as uint16 (Byte 0 = bits[15:8], Byte 1 = bits[7:0])
Audio Input Terminal Gain/Attenuation Step Size	1	See Table 4-18 (page 115)
Audio Input Terminal Gain Steps	1	Number of steps - 1 (0 = 1 step, 255 = 256 steps)
Audio Input Terminal Attenuation Steps	1	Number of steps - 1 (0 = 1 step, 255 = 256 steps)
Preferred Apple Device Audio Processing	1	See Table 4-19 (page 116)

The accAudioTerminalInformation packet is used to inform the Lightning Audio Module of the accessory's audio input/output terminal configuration.

All other parameters must be set to 0 if the Terminal Type parameter is None.

Copyright © 2015 Apple Inc. All Rights Reserved.

Accessories must accurately communicate their audio processing latency. Apple devices will use these parameters to help maintain audio/video sync across different terminals for a better user experience.

Accessories must accurately communicate their ability to set gain/attenuation in decibels (dB) using the Step Size and Steps parameters. Apple devices will use these parameters to normalize audio volume settings across different terminals for a better user experience. The total number of gain steps (both Gain and Attenuation) must not exceed 255.

At this time, the Preferred Apple Device Audio Processing parameter must be set to 0.

Table 4-16 Lightning Audio Module Audio Output Terminal Types

Value	Type	Notes
0	None	Must be used if accessory has no ability to render audio from Apple device
1	Headset	Speaker(s) must be positioned near user's ears
2	Headset Jack	Integrated 3.5 mm headset jack
3-255	Reserved	

Table 4-17 Lightning Audio Module Audio Input Terminal Types

Value	Type	Notes
0	None	Must be used if accessory has no ability to record audio to Apple device
1	Headset Microphone	Microphone must be positioned near the user's mouth
2	Headset Jack	Integrated 3.5 mm headset jack
3-255	Reserved	

Table 4-18 Lightning Audio Module Audio Terminal Gain/Attenuation Step Size Values

Value	Step Size (dB)
0	0.5
1	1.0
2	1.5
3	2.0

Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Step Size (dB)
4	2.5
5	3.0
6	3.5
7	4.0
8	4.5
9	5.0
10	5.5
11	6.0
12	6.5
13	7.0
14	7.5
15	8.0
16-255	Reserved

Table 4-19 Preferred Apple Device Audio Processing Options

Value	Type	Notes
0	None	Apple device should not perform any audio processing
1-255	Reserved	

4.12.4 Operation Packets

These packets may be sent or received anytime after protocol initialization has occurred.

Copyright © 2015 Apple Inc. All Rights Reserved.

4.12.4.1 lamInformation (0x10)

Table 4-20 Lightning Audio Module lamInformation packet payload

Field	Bytes	Description
External Accessory Protocol Session	1	0 = Session Closed, 1 = Session Opened
Device Power State	1	See Table 4-21 (page 117)
Device Battery Level	1	0-255, 0 = Empty, 255 = Full
Audio Output Stream Status	1	0 = Not Playing, 1 = Playing
Audio Input Stream Status	1	0 = Not Recording, 1 = Recording
Audio Signal Processing Mode	1	See Table 4-22 (page 118).
Audio Sample Rate	1	Sample rate truncated to kHz. See Table 4-23 (page 118).

The External Accessory Protocol Session parameter indicates whether an iOS app on the device has opened an EASession to the accessory. Accessories must not send accExternalAccessoryProtocolSessionData packets (see accExternalAccessoryProtocolSessionData (0x20) (page 120)) until a lamInformation packet has been received with this parameter set to 1.

Accessories that both provide power to the Apple device via the Lightning Audio Module and reserve some current to charge their own internal battery using accPassthroughPowerSourceInformation packets (see accPassthroughPowerSourceInformation (0x24) (page 122)) must monitor the Device Power State and Device Battery Level parameters to implement correct charging behavior. The Device Battery Level parameter is only valid when the Device Power State has a value of 0 or 1.

Table 4-21 Lightning Audio Module Device Power State Values

Value	Description
0-1	Device is drawing from internal battery
2-3	Device is connected to external power and not charging its internal battery
4	Device is connected to external power and charging its internal battery
5	Device is connected to external power and its internal battery is fully charged
6-255	Reserved

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The Audio Input/Output Stream Status parameters may be used to optimize accessory power consumption. For instance, headset amplifiers may be placed into a lower power state while the Output Stream Status is Not Playing, although they must remain able to render audio immediately if the status changes to Playing.

Accessories must use the Audio Signal Processing Mode parameter to trigger content-specific audio processing modes as defined in Table 4-22 (page 118). The signal processing mode applies to both the input and output streams. If an accessory receives a stream type of Reserved, it must treat it as if the stream type was set to Default.

Table 4-22 Lightning Audio Module Audio Stream Signal Processing Mode Values

Value	Name	Accessory Processing Requirements
0	Default	Accessory may apply any processing.
1	Measurement	Non-linear and time varying processing must be disabled.
2	Telephony	Processing appropriate for telephony should be applied.
3	Speech Recognition	Processing appropriate for speech recognition should be applied.
4-255	Reserved	Must be treated the same as Default

The Audio Sample Rate parameter indicates the maximum sample rate of the content within the 48 kHz stream. This information may be used to filter microphone input or otherwise optimize signal processing performed within the accessory.

Table 4-23 Lightning Audio Module lamInformation Audio Sample Rate Values

Value	Sample Rate
8	8.0 kHz
11	11.025 kHz
12	12.0 kHz
16	16.0 kHz
22	22.050 kHz
24	24.0 kHz
32	32.0 kHz
44	44.1 kHz

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Sample Rate
48	48.0 kHz

4.12.4.2 lamAudioTerminalInformation (0x11)

Table 4-24 Lightning Audio Module lamAudioTerminalInformation packet payload

Field	Bytes	Description
Audio Output Terminal Mute	1	0 = Not Muted, 1 = Muted
Audio Output Terminal Gain Steps	1	Signed byte expressing number of gain steps above zero (positive values) or attenuation steps below zero (negative values).
Audio Input Terminal Mute	1	0 = Not Muted, 1 = Muted
Audio Input Terminal Gain Steps	1	Signed byte expressing number of gain steps above zero (positive values) or attenuation steps below zero (negative values).

If an audio terminal is Muted, the accessory must mute its corresponding audio interface. The accessory may apply a ramp of its choosing while transitioning to a Muted state.

If an audio terminal is Not Muted, the accessory must un-mute the audio interface. The accessory may apply a ramp of its choosing while transitioning to a Not Muted state.

Gain values will be set based on information provided by the accessory in an accAudioTerminalInformation packet (see accAudioTerminalInformation (0x0B) (page 114)). Changes in the Gain of an audio terminal while Not Muted must be applied without any ramp by the accessory. The accessory also must not remap or scale the requested Gain value from the Apple device.

The Lightning Audio Module will send a lamAudioTerminalInformation packet to the accessory before starting to stream or record audio for the first time after accessory connect. The accessory must not independently remember and apply its own previous audio terminal volume settings.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

4.12.4.3 lamExternalAccessoryProtocolSessionData (0x12)

Table 4-25 Lightning Audio Module lamExternalAccessoryProtocolSessionData packet payload

Field	Bytes	Description
Data	1-n	External Accessory Protocol data

4.12.4.4 lamHIDReport (0x13)

Table 4-26 Lightning Audio Module lamHIDReport packet payload

Field	Bytes	Description
HID Report	1-n	HID report

4.12.4.5 accVersionInformation (0x14)

No payload.

This packet will request that the LAM provide its version information, see lamVersionInformation (0x25) (page 122).

4.12.4.6 accExternalAccessoryProtocolSessionData (0x20)

Table 4-27 Lightning Audio Module accExternalAccessoryProtocolSessionData packet payload

Field	Bytes	Description
Data	1-n	External Accessory Protocol session data

4.12.4.7 accHIDReport (0x21)

Table 4-28 Lightning Audio Module accHIDReport packet payload

Field	Bytes	Description
HID Report	1-n	HID report

4.12.4.8 accRequestAppLaunch (0x22)

No payload.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

This packet will request that the Apple device launch the app specified by accPreferredAppBundleIdentifierInformation (see accPreferredAppBundleIdentifierInformation (0x08) (page 113)) on the accessory's behalf. The accessory must not assume that the app has launched, because the request may be denied by iOS or the user. A lamInformation (see lamInformation (0x10) (page 117)) packet must be received indicating that the app has successfully opened an External Accessory Protocol session before the accessory may presume that it is now communicating with the app.

> **Note:** This packet must only be sent in response to direct user action, such as pushing a button on the accessory. Autonomous launch of apps is explicitly prohibited.

4.12.4.9 accAudioTerminalStateInformation (0x23)

Table 4-29 Lightning Audio Module accAudioTerminalStateInformation packet payload

Field	Bytes	Description
Audio Output Terminal State	1	0 = Disabled, 1 = Enabled
Audio Input Terminal State	1	0 = Disabled, 1 = Enabled

If the accessory must accurately report terminal state as Enabled or Disabled to the Lightning Audio Module when initially connected and whenever it changes. Apple devices will use this information to adjust audio routing for the appropriate user experience. The Lightning Audio Module will assume that all accessory audio terminals are Disabled until the accessory sends an accAudioTerminalStateInformation packet indicating otherwise. The accessory should report the Enabled state as early as possible to minimize audio routing time.

> **Note:** Audio terminal state must be reported as Enabled if and only if the terminal is fully capable of rendering or recording audio in a manner that is perceptible to the end user. Failure to comply with this requirement may cause a poor user experience in the form of truncated audio output or input and is grounds for failure to pass self certification.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

4.12.4.10 accPassthroughPowerSourceInformation (0x24)

Table 4-30 Lightning Audio Module accPassthroughPowerSourceInformation packet payload

Field	Bytes	Description
Reserved Current For Accessory	2	Number of milliamps expressed as uint16 (Byte 0 = bits[15:8], Byte 1 = bits[7:0]

If the accessory has an internal battery, it must use this packet to communicate how much current it will draw from the pass-through power source to recharge the internal battery before passing the rest on to the device. The Apple device will adjust its current draw from the pass-through power source accordingly. This packet must be sent whenever the accessory's current draw from the pass-through power source changes.

The accessory must not reserve more than 500 mA of current until the Apple device's battery is fully charged as indicated by the Device Power State parameter in a lamInformation packet (see lamInformation (0x10) (page 117)).

4.12.4.11 lamVersionInformation (0x25)

Table 4-31 Lightning Audio Module lamVersionInformation packet payload

Field	Bytes	Description
LAM Hardware Major Version	1	
LAM Hardware Minor Version	1	
LAM Hardware Revision	1	
LAM Firmware Major Version	1	
LAM Firmware Minor Version	1	
LAM Firmware Revision	1	

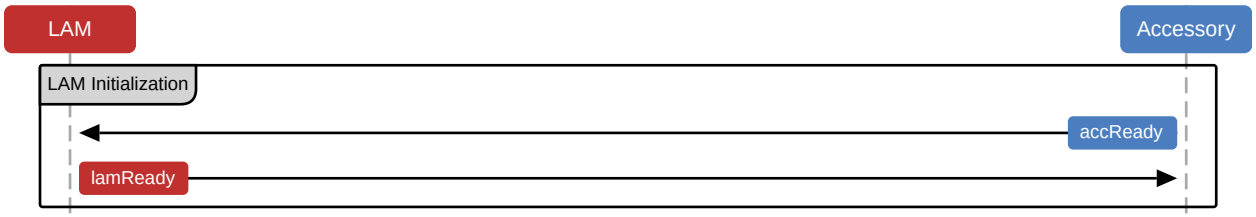
This packet provides information about the hardware and firmware versions of the Lightning Audio Module. The accessory may request this information by sending an accVersionInformation packet, see accVersionInformation (0x14) (page 120).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

4.12.5 Examples

4.12.5.1 Initialization

The following example demonstrates how to initialize communications with a Lightning Audio Module.

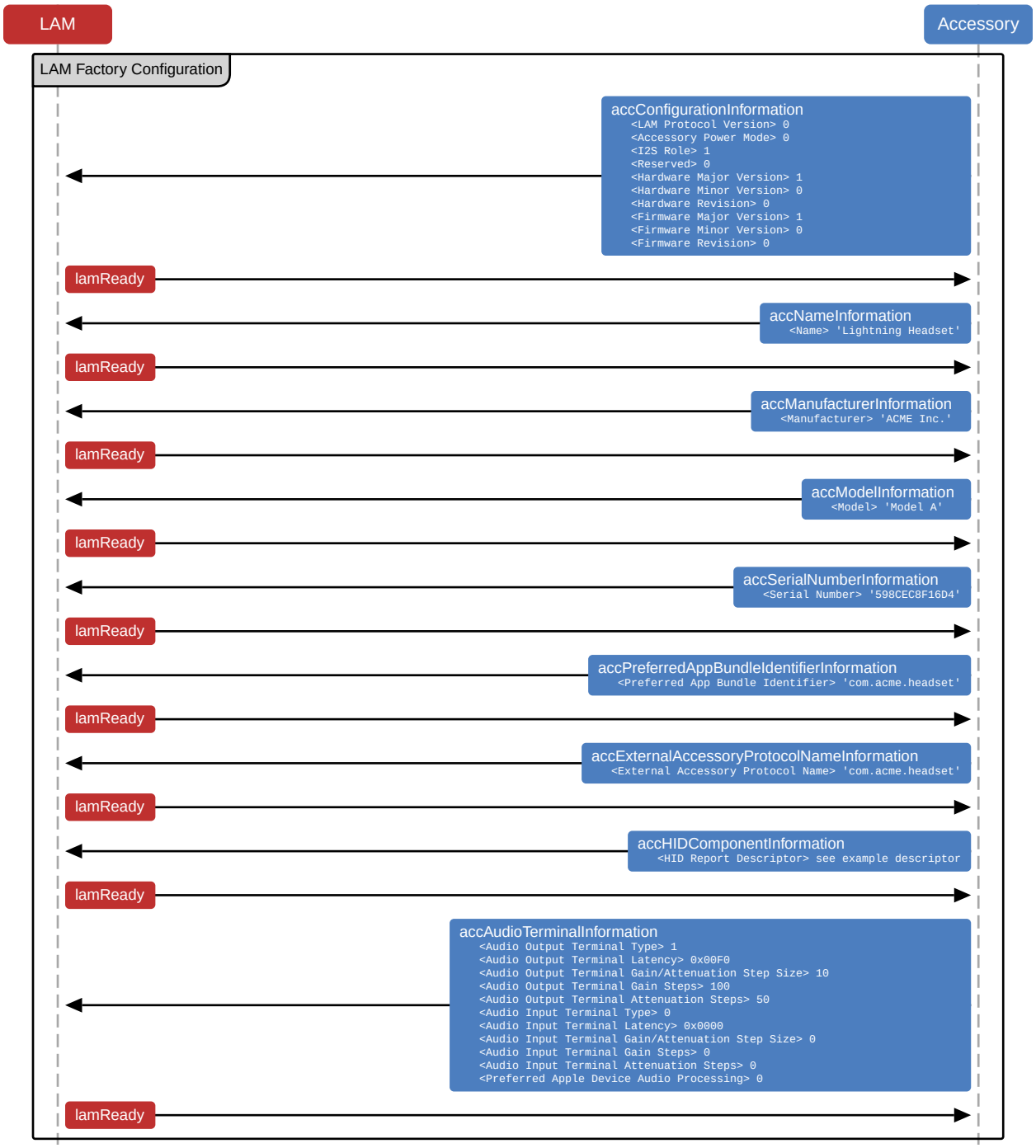


4.12.5.2 LAM Accessory Factory Configuration

The following example demonstrates how to configure the Lightning Audio Module for use in an accessory that:

- Performs active audio processing, but only when audio is streaming from the Apple device
- Communicates with an iOS app that configures EQ settings and downloads firmware updates
- Implements user button controls for Volume Up, Center, and Volume Down

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

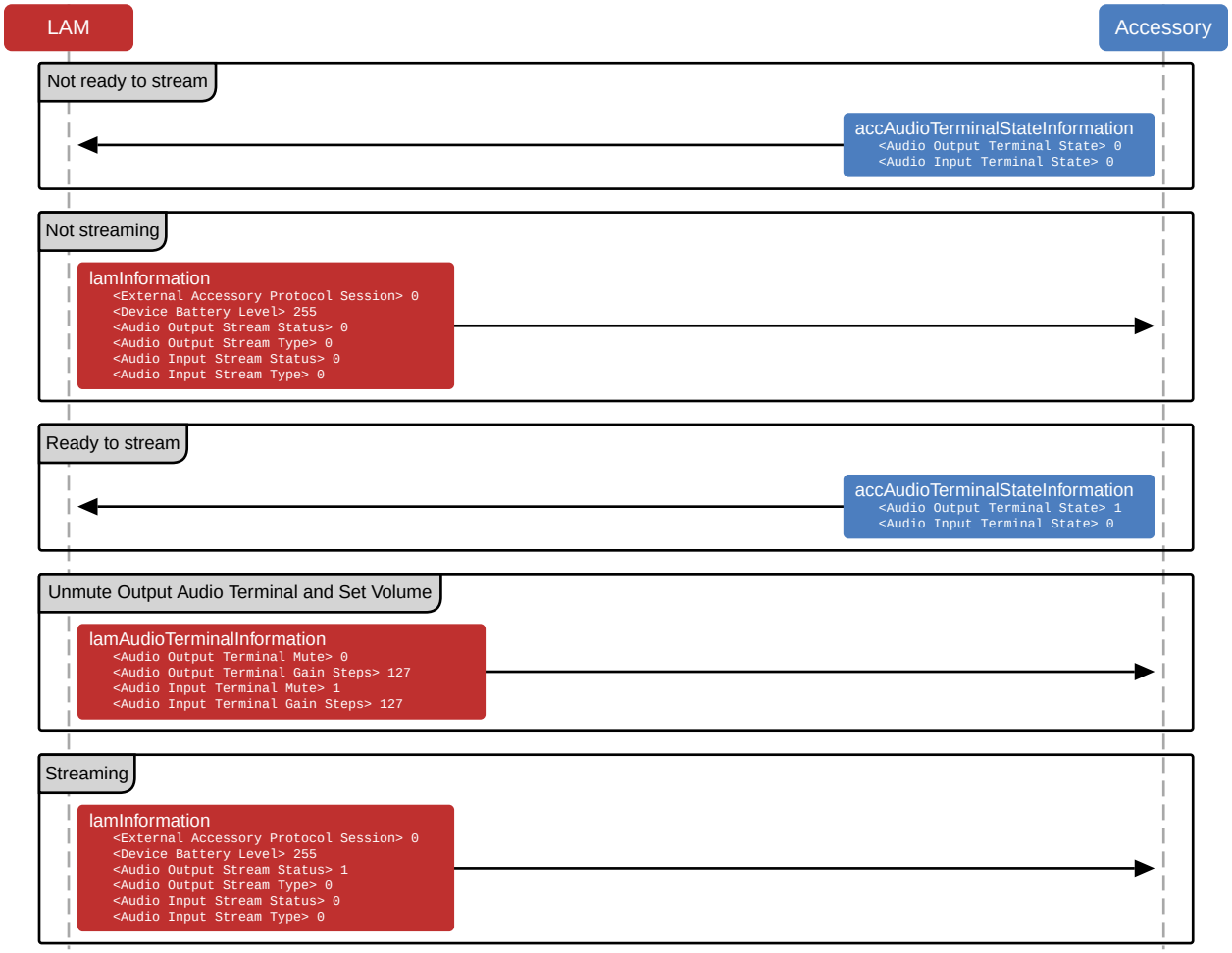


4.12.5.3 Typical Usage

The following example demonstrates typical traffic exchanged between the Lightning Audio Module and an accessory that:

- Communicates its initial audio terminal information (not ready to stream)

- Receives initial audio terminal information from the Lightning Audio Module (not streaming)
- Communicates that it is ready to accept audio output from the Lightning Audio Module
- Receives notification of incoming audio streaming data from the Lightning Audio Module



4.13 Test Procedures

1. Verify that the accessory does not block or occlude the Apple device's built-in microphones or speakers.
2. Verify that the accessory does not have speakers integrated.
3. Verify that accessories using Lightning Audio Module use a cable that integrates a Lightning (C10E) or Lightning (C68E) connector.
4. Verify that the Lightning Audio Module does not force an Apple device to wake from sleep without direct user input.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

4.13.1 Connectivity

While the accessory is attached to the Apple device, verify that:

1. The accessory is visible in the Setting app (Settings: General: About) on the Apple device and that it identifies with the correct information strings.
2. The Apple device will enter a sleep state after a period of inactivity while the accessory is still attached.
3. Pressing the Center button on the headset remote will wake the Apple device from a sleep state.
4. Pressing any other button on the headset remote will not wake the Apple device from a sleep state.
5. After the Apple device hibernates, launch Siri through pressing and holding either home button or headset Center button. Verify that the Siri double beep prompt is heard in full.
6. After the Apple device hibernates, send an iMessage to the hibernating device to trigger notification. Confirm notification sound is heard in full.

4.13.2 Audio Quality

Verify that:

1. When the Apple device is not playing music, the accessory's output stream is devoid of pops/clicks/artifacts at all times, especially during connection or disconnection.
2. When the Apple device is playing music, the accessory's output stream is devoid of pops/clicks/artifacts at all times.

4.13.3 Power

1. If applicable, verify that the accessory battery is not charged from the Apple device.
2. Verify that peak current draw from the Apple device does not exceed 100 mA except for the first 5 ms after the Lightning Audio Module enables Accessory Power.
3. Verify that the average current draw from the Accessory Power pad does not exceed 50 mA.
4. Immediately after a power loss to the accessory, verify that the accessory mutes all audio.

4.13.4 Charge/Sync

If the accessory supports pass-through charge/sync of the Apple device from an external USB host or power supply, verify that:

1. The Apple device charges when the accessory is connected to an external USB power supply, such as an Apple USB power adapter.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. The Apple device is recognized and can sync/restore when the accessory is connected via USB to a host computer.
3. If the accessory has an internal battery that can also recharge from an external USB power supply, the accessory's internal battery does not draw more than 500 mA until the Apple device's battery is fully charged.

4.13.5 Headset Jack

If the accessory has an integrated 3.5 mm headset jack, verify the following:

1. The accessory may physically block or occlude the Apple device's headset jack.
2. The accessory does not physically block or occlude the Apple device's built-in microphones or speakers.

4.13.6 Firmware Update

Verify that the accessory's firmware can be updated by an end user from an Apple product (running iOS or OS X).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

5. Apple Lightning Connector

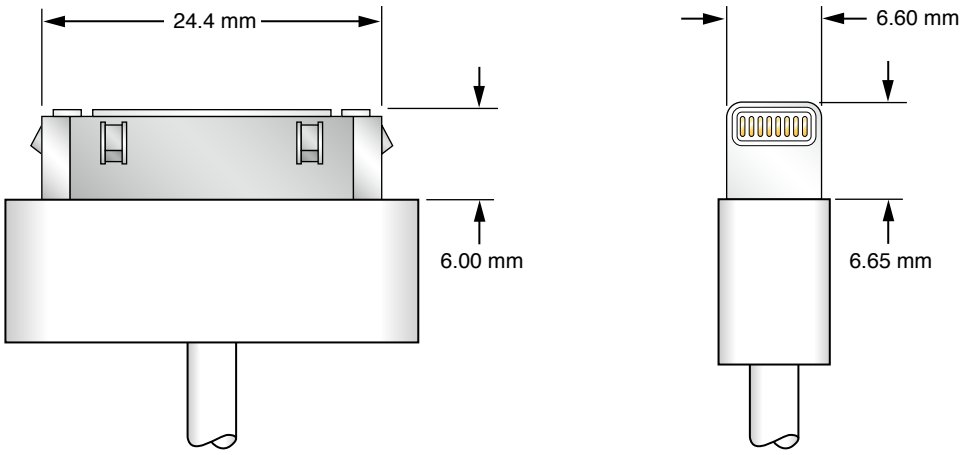
Accessories may incorporate a Lightning connector to establish mechanical and electrical connections with an Apple device and enable other accessory interface features.

This chapter covers the Lightning connector. See *MFi Accessory Hardware Specification R9* for details of the 30-pin.

5.1 Comparison with the 30-pin Connector

5.1.1 Dimensions

Figure 5-1 Comparison of 30-pin connector and Lightning connector dimensions



5.1.2 Active Component

Unlike the 30-pin connector, which is a passive component, the Lightning connector is an active component. Accessories that integrate a Lightning connector must treat it accordingly during both manufacture and field deployment.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

5.1.3 Reversability

The Lightning connector engages with Apple devices in either orientation. Accessories do not need to manage this capability. The connector pad/pin configuration always remains constant from the accessory's point of view, because the connector automatically routes power and data to and from the accessory appropriately after determining its insertion orientation.

5.1.4 Accessory Identify

Accessories do not need to provide an Accessory Identify resistor to indicate choice of iAP transport. The Lightning connector contains all needed configuration information internally and handles interface setup with the Apple device.

5.1.5 Accessory Detect

Accessories do not need to manage the state of an Accessory Detect pin. The Lightning connector handles this task.

5.1.6 Apple Device Detect

There is no Apple Device Detect (pin 30 on the 30-pin connector) equivalent. See Apple Device Detection (page 61) for information on detecting the presence of an Apple device.

5.1.7 Analog Audio

There is no provision for analog audio output via the Lightning connector. See Digital Audio (page 528) for information on digital audio input and output.

5.1.8 Analog Video

There is no provision for analog video output.

5.1.9 DisplayPort Video

There is no provision for DisplayPort video output.

5.2 Connector Versions

There are 5 versions of the Lightning connector:

- Lightning (C10) and Lightning (C11) connectors have 9 pads on both sides of an exposed PCB.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Lightning (C12) has 9 pins located at the end of the connector.
- Lightning (C48) has 4 pads on one side of an exposed PCB.
- Lightning (C68) has 9 pads on both sides of an exposed PCB.

All references to the Lightning connector in this chapter are applicable to all versions unless explicitly specified otherwise.

Accessory developers must take the following recommendations and requirements into consideration when deciding which version to use for a particular accessory design:

Table 5-1 Lightning connector modules

Module	Cable	Dock	Dongle	Form-Fitting
Lightning (C10)	Optional	Prohibited	Prohibited (see note)	Optional
Lightning (C11)	Prohibited	Recommended	Optional	Optional
Lightning (C12)	Prohibited	Prohibited	Optional	Recommended
Lightning (C48)	Optional	Optional	Optional	Optional
Lightning (C68)	Optional	Optional	Optional	Optional

Note: See Dongle Accessories (page 167) for one exception to the prohibition on use of Lightning (C10) in accessory dongle integrations.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-2 Lightning (C10)

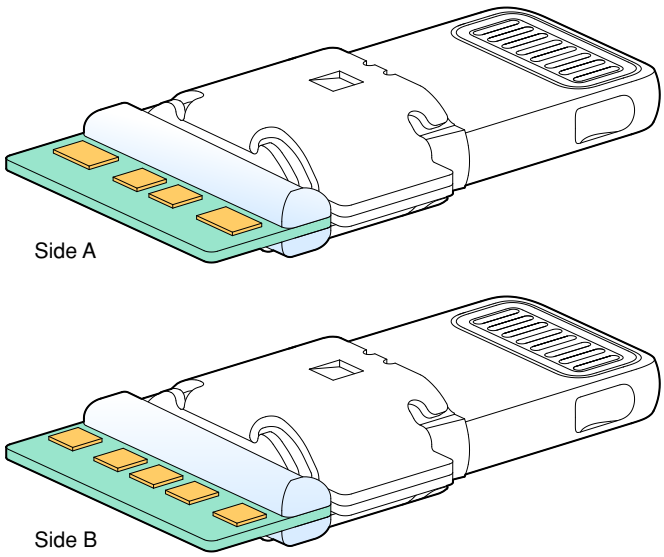
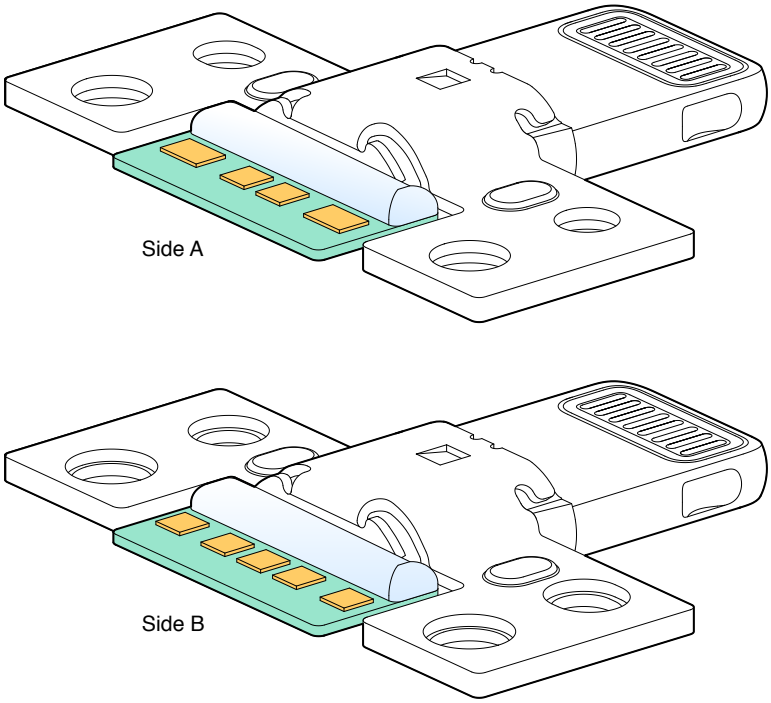


Figure 5-3 Lightning (C11)



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-4 Lightning (C12)

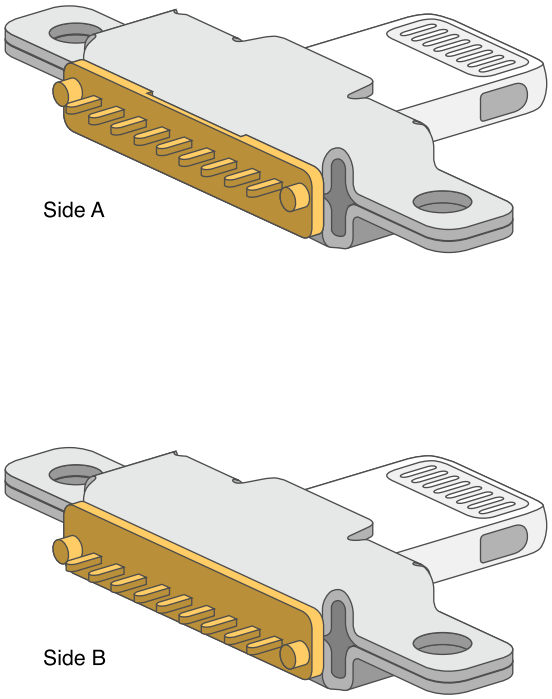
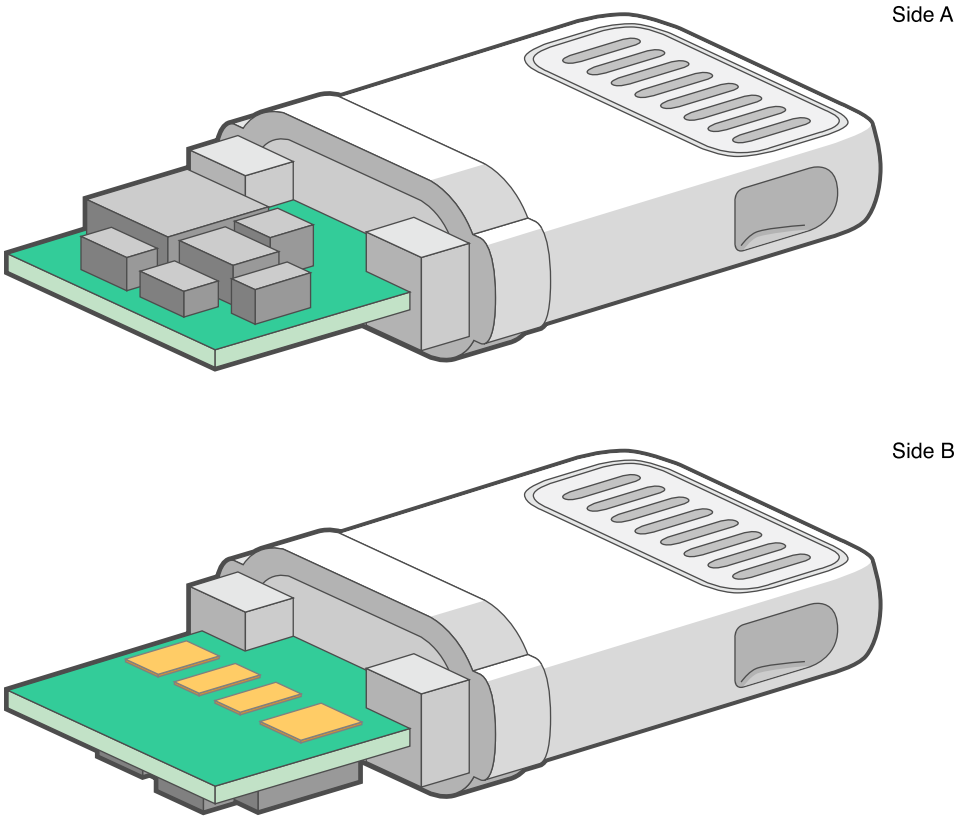


Figure 5-5 Lightning (C48)



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-6 Lightning (C68)

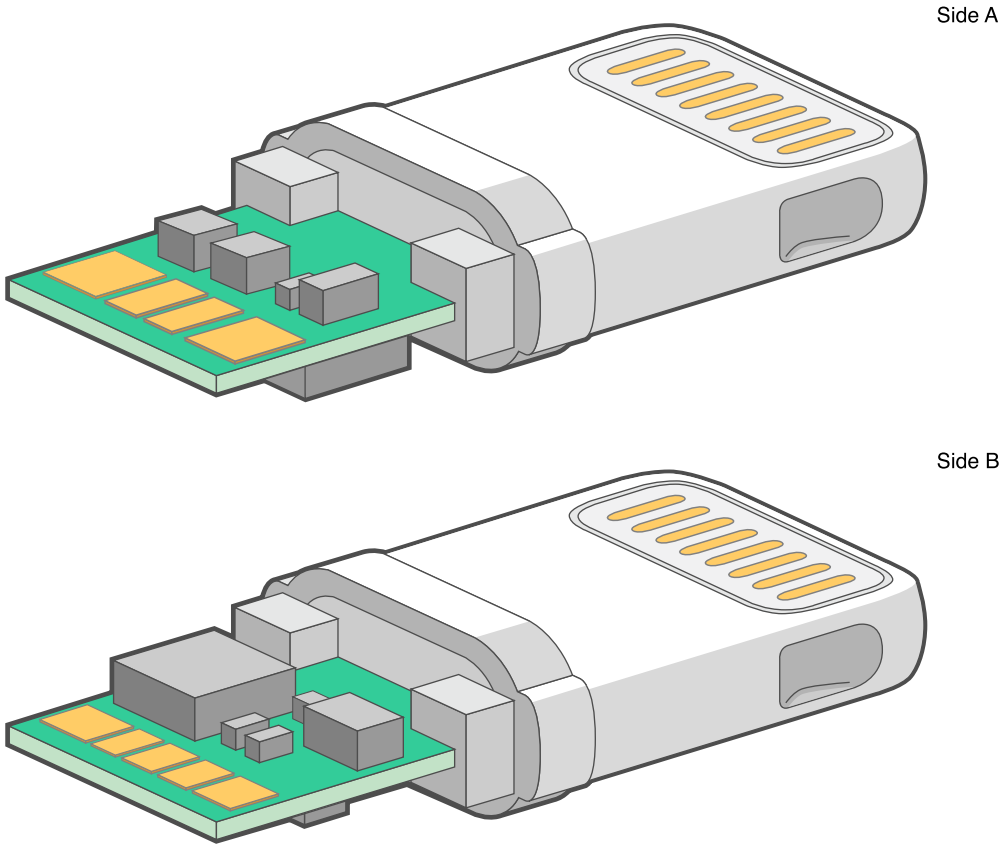


Figure 5-7 Lightning (C10) dimensions

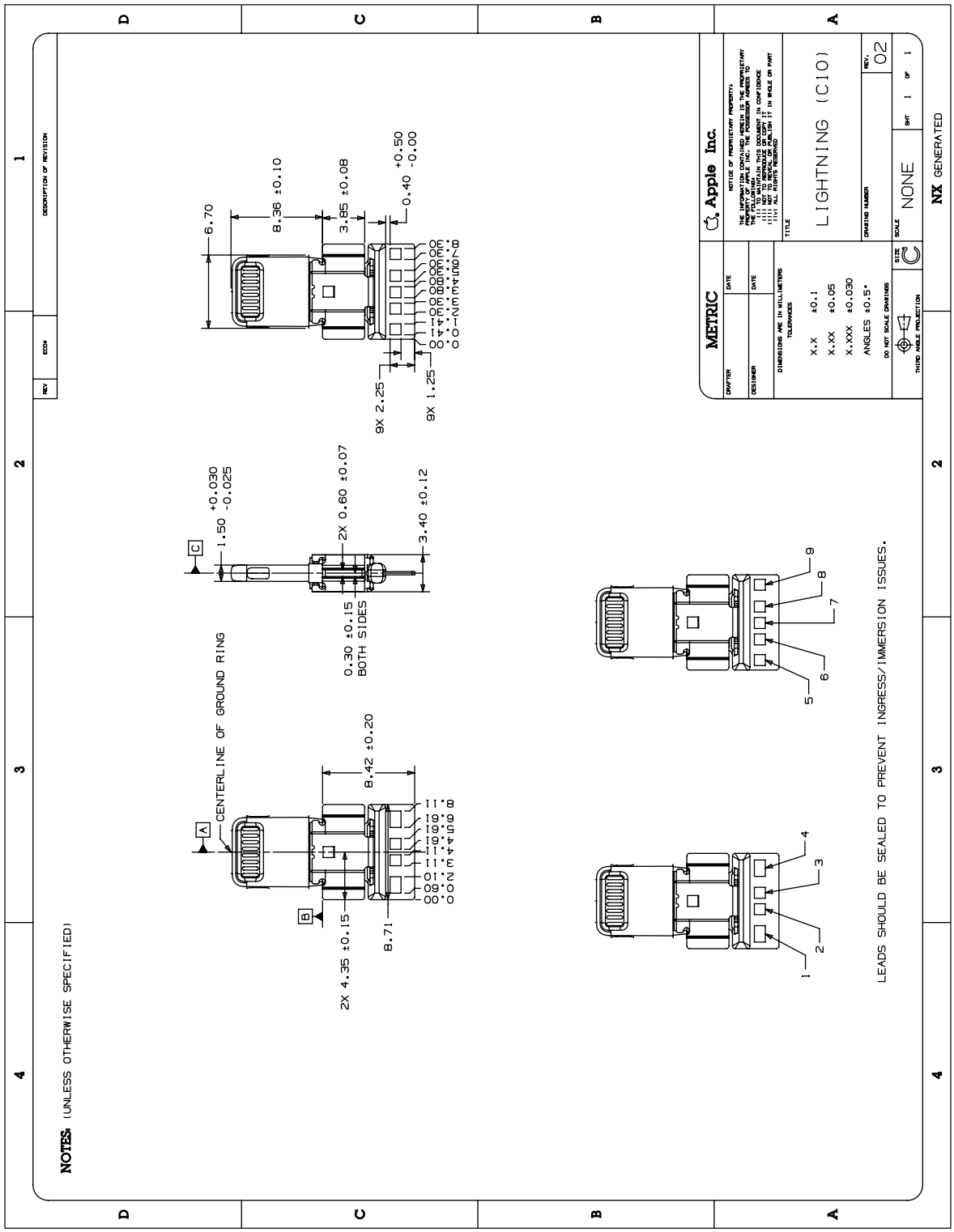


Figure 5-8 Lightning (C11) dimensions

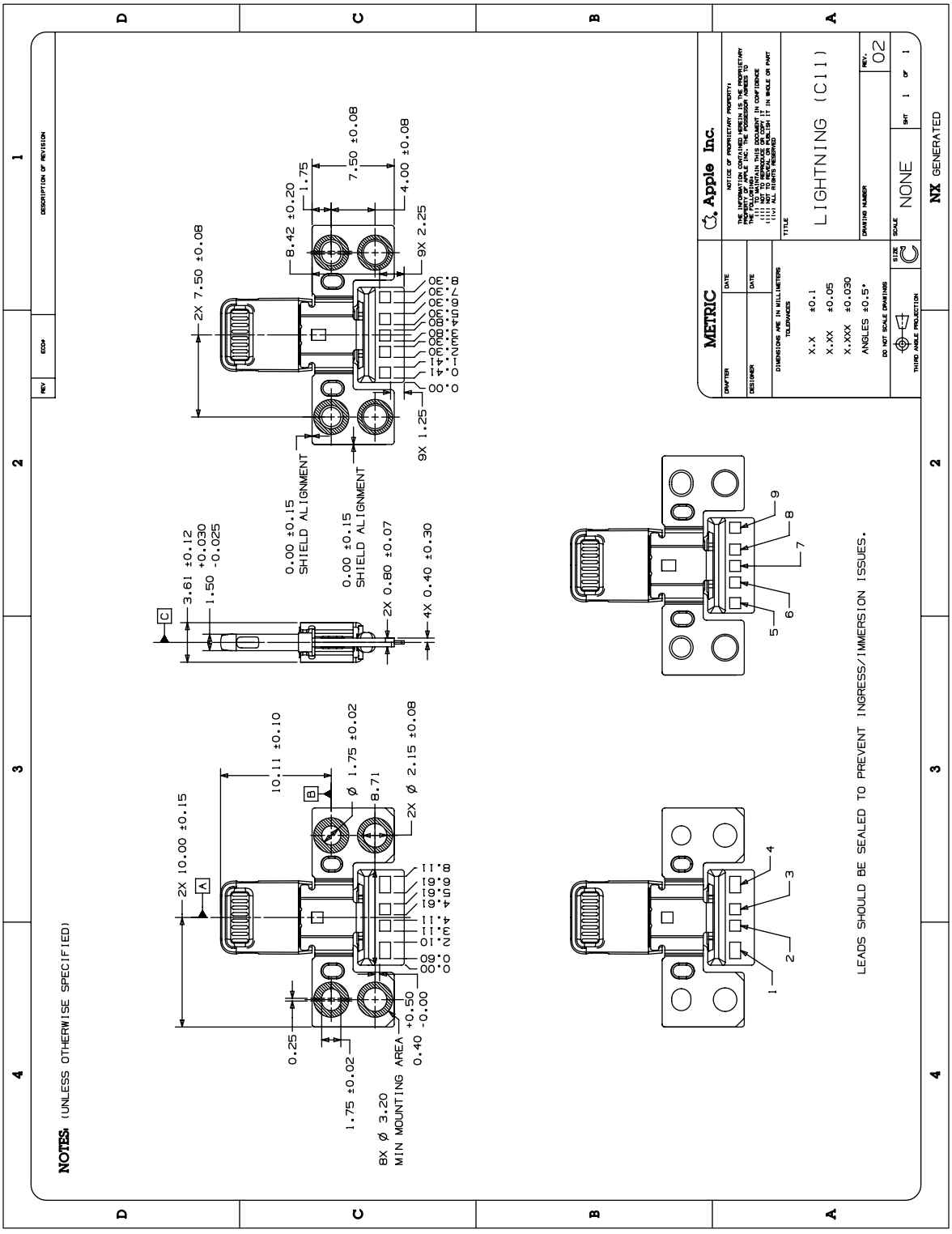


Figure 5-9 Lightning (C12) dimensions

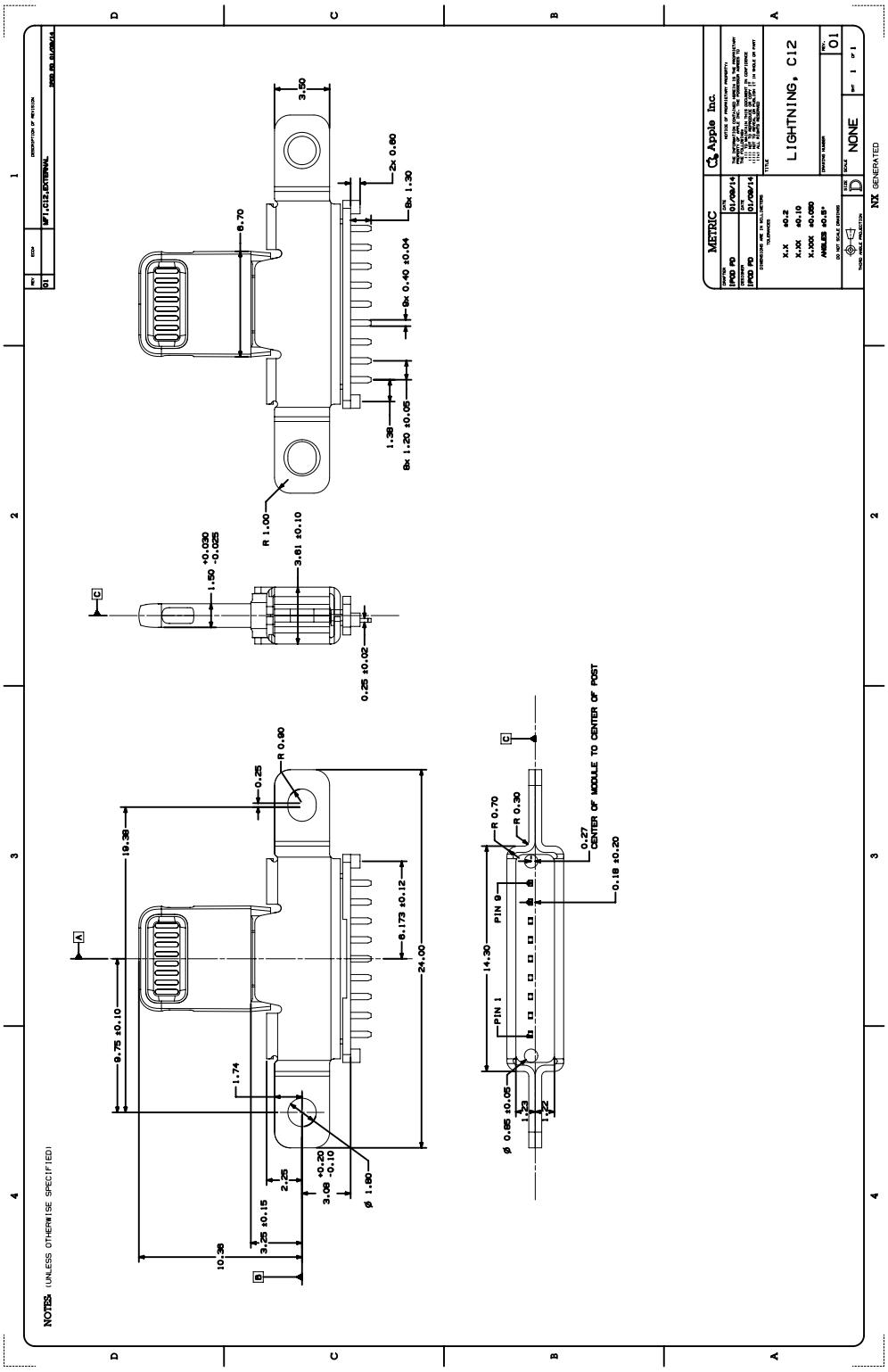


Figure 5-10 Lightning (C48) dimensions

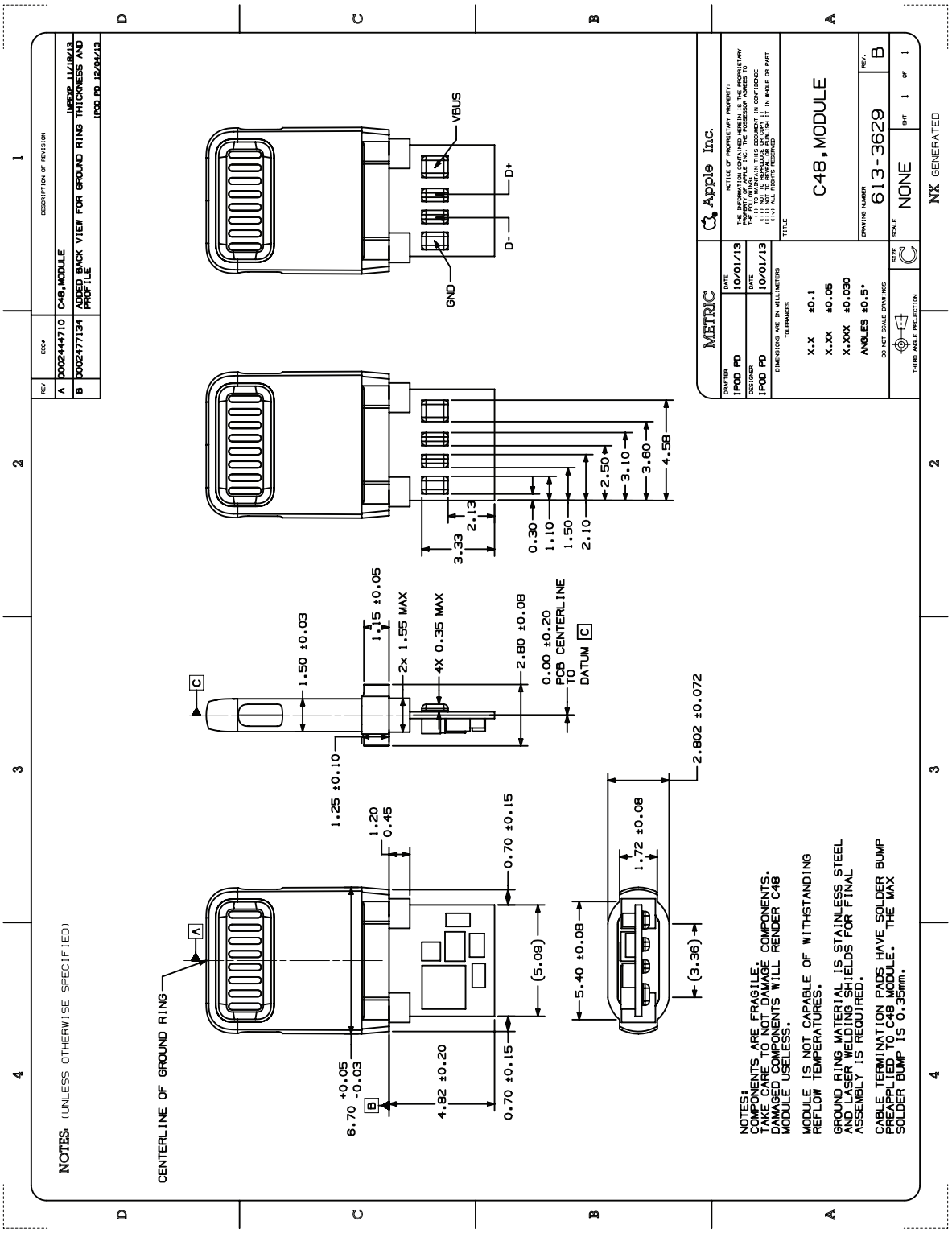
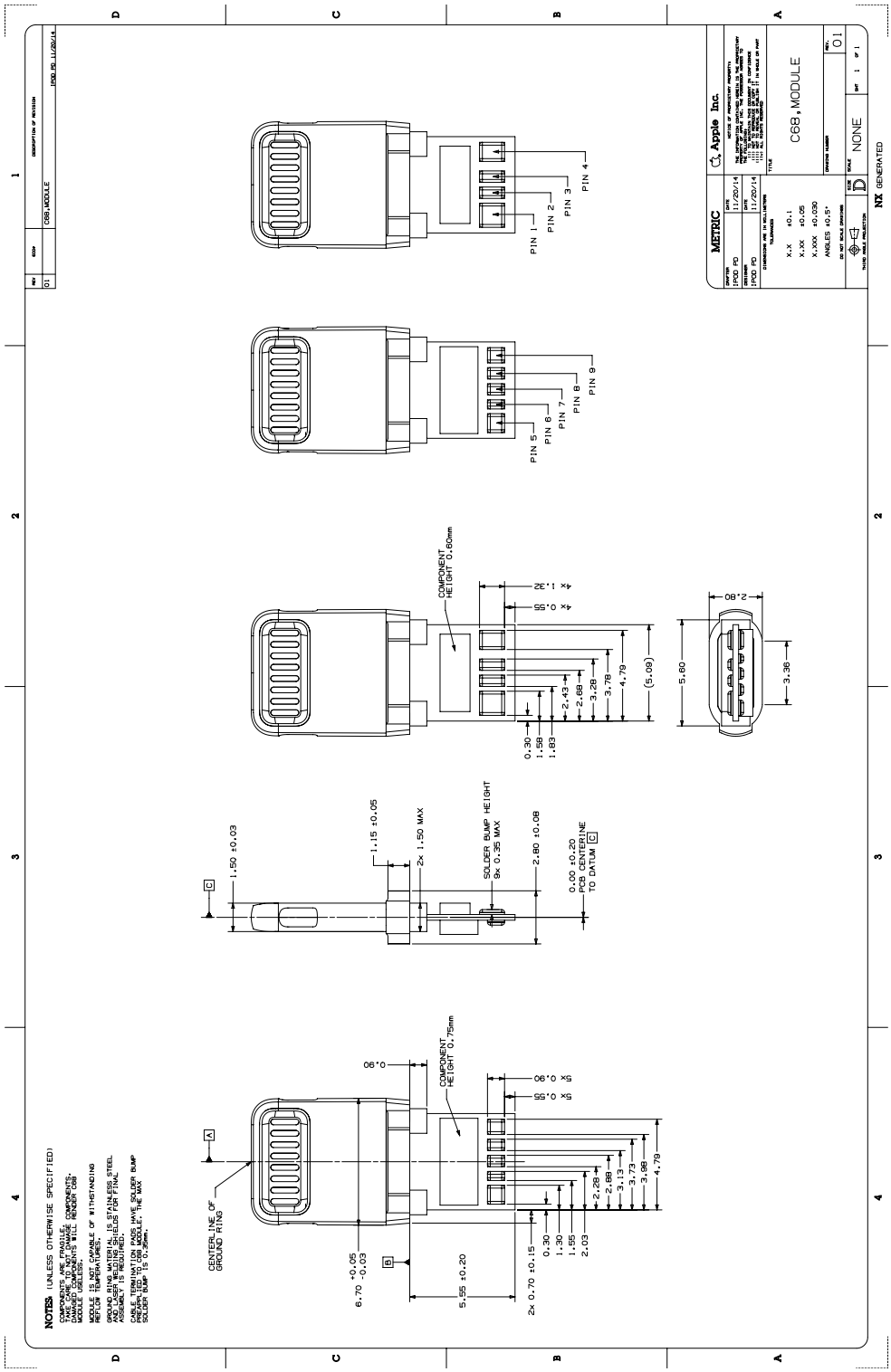


Figure 5-11 Lightning (C68) dimensions



5.3 Connector Pad/Pin Configuration

Aside from power and grounds, the pad/pin assignments are not fixed. Instead, the configuration of the other pads/pins is negotiated between the device and the accessory. Accessory developers must choose a configuration and order pre-configured Lightning connectors from Apple for use in final accessory assembly. There are three standard configurations (A/B/C), one for each of the available data transports (USB Host Mode, USB Device Mode, and Serial), and one configuration (E) that is specific to the Lightning Audio Module (see Apple Lightning Audio Module (page 93)).

> **Note:** All accessories that do not have the potential for data communication with the Apple device (Power Only or Power/Sync) accessories must use the B configuration. The D configuration (Power Only) has been deprecated and must not be used in new accessory designs.

More information on the data transports supported by the Lightning connector can be found in the following chapters:

- Serial (see Serial (page 680))
- USB Device Mode (see USB Device Mode (page 706))
- USB Host Mode (see USB Host Mode (page 712))

To refer to a specific connector version and configuration, the connector version is immediately followed by the configuration. For example, Lightning (C10B) refers to a Lightning (C10) connector with the B, or USB Device Mode, configuration.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-12 C10/C11 Connector Pads

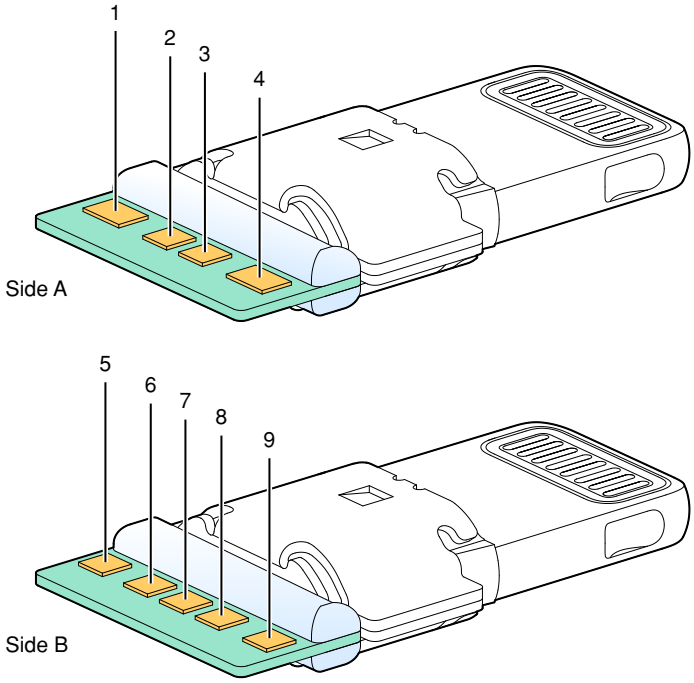


Figure 5-13 C12 Connector Pins

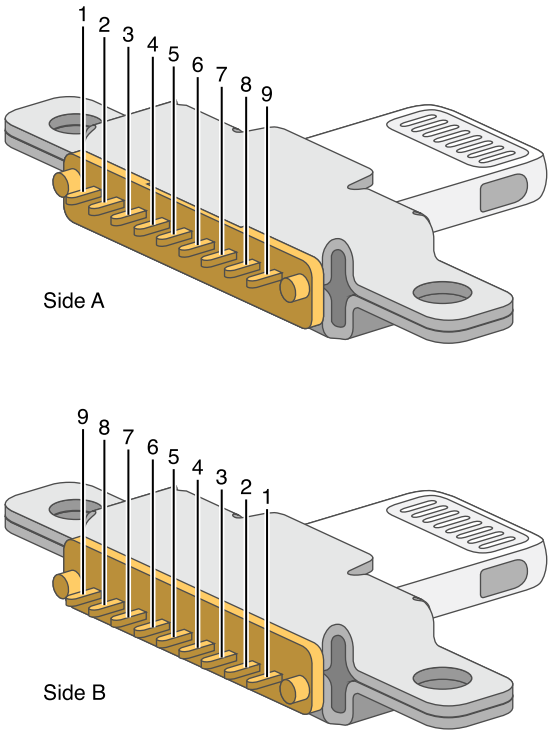


Figure 5-14 C48 Connector Pads

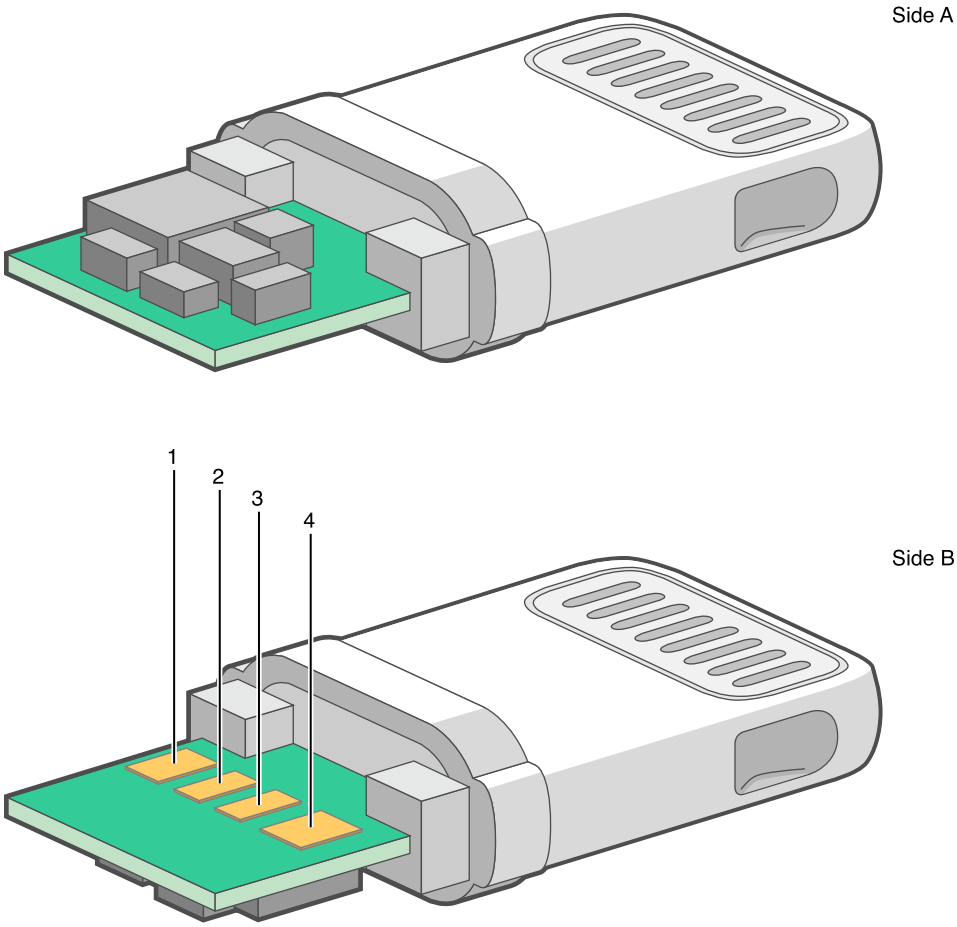
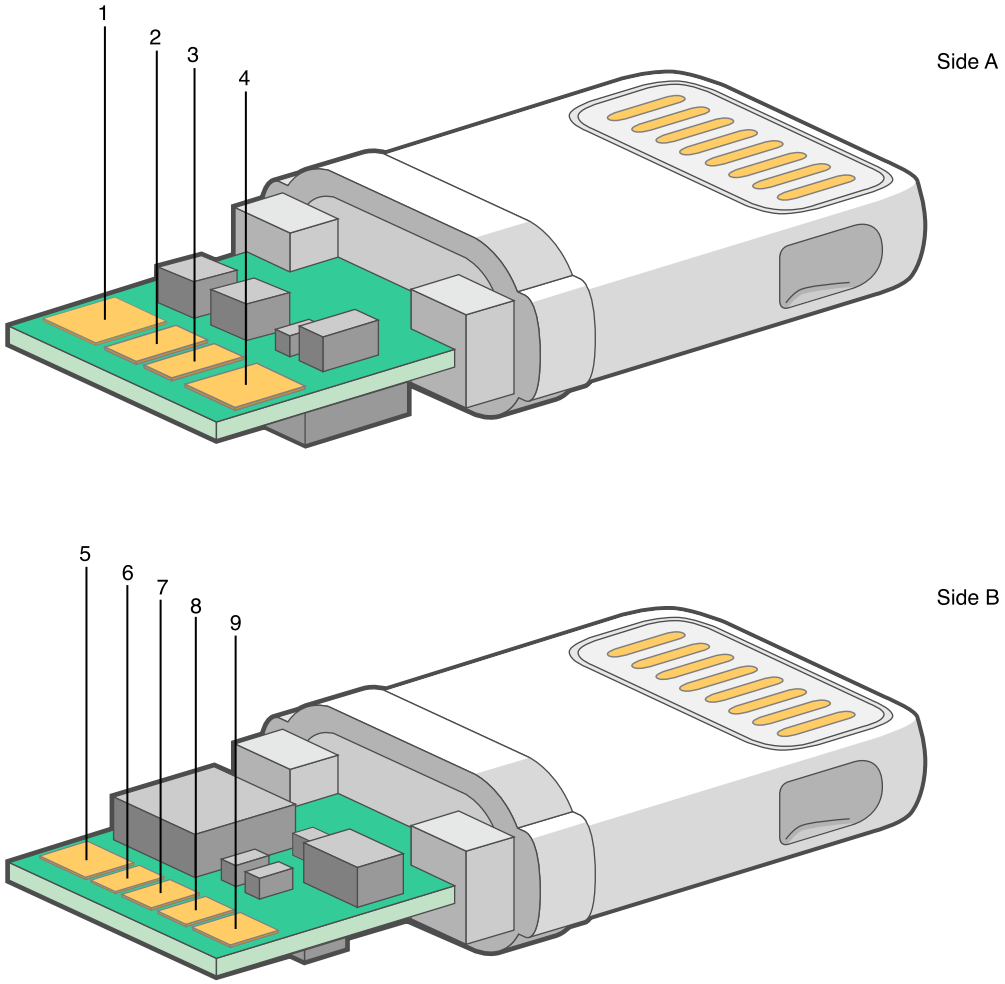


Figure 5-15 C68 Connector Pads



Lightning (C10), Lightning (C11), Lightning (C12), and Lightning (C68) connector configurations that exchange data with the Apple device offer accessories the ability to provide power to, and draw power from, the Apple device via the Device Power and Accessory Power pads/pins respectively. Accessories that do not draw power from the Apple device may leave the Accessory Power pad/pin unconnected (NC). This may ease accessory assembly operations.

Lightning (C48) connectors do not allow accessories to draw power from the Apple device, but do allow them to provide power to the Apple device.

Accessories must use the B configuration for a connector (USB Device Mode) if there is no potential for data communication with the Apple device.

Accessories that incorporate a Lightning Audio Module (see Apple Lightning Audio Module (page 93)) must use a Lightning (C10E), Lightning (C11E), Lightning (C12E), or Lightning (C68E) connector. Conversely, accessories that do not incorporate a Lightning Audio Module must not use those connectors.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 5-2 Lightning (C10) connector configurations

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
1	Ground 1	Ground 1	Ground 1	Ground 1
2	USB D-	USB D-	Accessory Serial TX	USB D-
3	USB D+	USB D+	Device Serial TX	USB D+
4	Device Power	Device Power	Device Power	Device Power
5	Accessory Power	Accessory Power	Accessory Power	LAM Power
6	Not Connected	Not Connected	Not Connected	LAM DW
7	Not Connected	Not Connected	Not Connected	LAM D-
8	Not Connected	Not Connected	Not Connected	LAM D+
9	Ground 2	Ground 2	Ground 2	LAM Ground

Table 5-3 Lightning (C11) connector configurations

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
1	Ground 1	Ground 1	Ground 1	Ground 1
2	USB D-	USB D-	Accessory Serial TX	USB D-
3	USB D+	USB D+	Device Serial TX	USB D+
4	Device Power	Device Power	Device Power	Device Power
5	Accessory Power	Accessory Power	Accessory Power	LAM Power
6	Not Connected	Not Connected	Not Connected	LAM DW
7	Not Connected	Not Connected	Not Connected	LAM D-
8	Not Connected	Not Connected	Not Connected	LAM D+
9	Ground 2	Ground 2	Ground 2	LAM Ground

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 5-4 Lightning (C12) connector configurations

Pin	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
1	Not Connected	Not Connected	Not Connected	LAM DW
2	USB D+	USB D+	Device Serial TX	USB D+
3	Ground 1	Ground 1	Ground 1	Ground 1
4	USB D-	USB D-	Accessory Serial TX	USB D-
5	Accessory Power	Accessory Power	Accessory Power	LAM Power
6	Not Connected	Not Connected	Not Connected	LAM D-
7	Ground 2	Ground 2	Ground 2	Ground 2
8	Not Connected	Not Connected	Not Connected	LAM D+
9	Device Power	Device Power	Device Power	LAM Ground

Table 5-5 Lightning (C48) connector configurations

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial
1	Ground	Ground	Ground
2	USB D-	USB D-	Accessory Serial TX
3	USB D+	USB D+	Device Serial TX
4	Device Power	Device Power	Device Power

Table 5-6 Lightning (C68) connector configurations

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
1	Ground 1	Ground 1	Ground 1	Ground 1
2	USB D-	USB D-	Accessory Serial TX	USB D-

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Pad	Configuration A: USB Host Mode	Configuration B: USB Device Mode	Configuration C: Serial	Configuration E: Lightning Audio Module
3	USB D+	USB D+	Device Serial TX	USB D+
4	Device Power	Device Power	Device Power	Device Power
5	Accessory Power	Accessory Power	Accessory Power	LAM Power
6	Not Connected	Not Connected	Not Connected	LAM DW
7	Not Connected	Not Connected	Not Connected	LAM D-
8	Not Connected	Not Connected	Not Connected	LAM D+
9	Ground 2	Ground 2	Ground 2	LAM Ground

5.4 Connector Mechanical Requirements

All accessories that integrate the Lightning connector into their design must comply with the mechanical requirements in this section. Different requirements apply, depending on the nature of the integration (e.g. cable, dongle, dock) and the Apple devices with which the accessory is compatible. If an accessory integrates more than one Lightning connector in its design, it must comply with the relevant requirements for each integration. Similarly, if an accessory declares compatibility with more than one Apple device, it must comply with the union of all relevant requirements for each device.

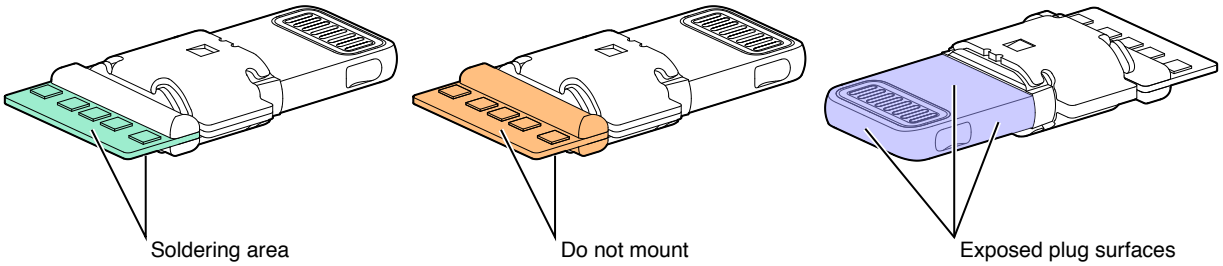
5.4.1 All Accessories

The critical mechanical areas for mounting and connecting a Lightning connector are shown in Figure 5-16 (page 146), Accessory designs must respect these areas when meeting the following requirements:

- The Lightning connector must not be subjected to any process, such as a reflow process, that heats it outside of the soldering area to a temperature above 80 °C for more than 30 minutes. Recommended processes include soldering, hot-bar, and over-molding.
- The accessory should not mount or hold on to the Lightning connector's PCB.
- Any process performed on the Lightning connector must not cause any buildup of material on its exposed plug surfaces.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-16 Connector critical areas

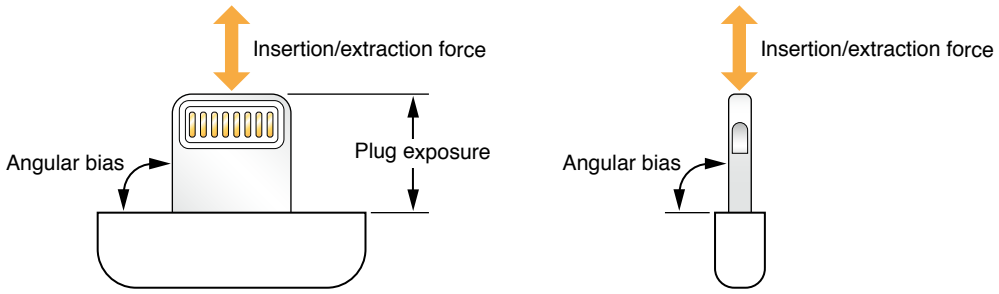


Additionally, all accessory integrations of a Lightning (C11) connector must make physical contact with the connector in the area labeled 'MIN MOUNTING AREA' as shown in Figure 5-8 (page 135).

Figure 5-17 (page 146) illustrates the following requirements related to plug engagement:

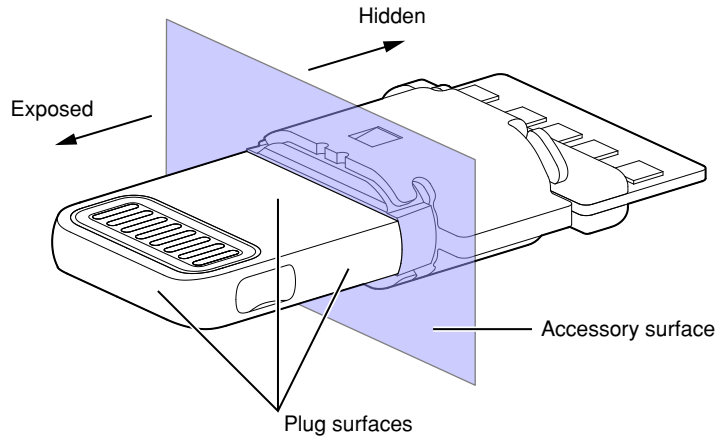
- At least 6.47 mm of the connector plug must be exposed. 6.65 mm +0.10 mm/-0.18 mm is recommended.
- When a force of 30 N is applied to the plug in the direction of device insertion, the exposed connector plug length must not decrease below 6.47 mm. The plug is designed to exert a force of 15 N during device insertion.
- The connector plug must not break away from the accessory when a force of 30 N or less is applied on the plug in the direction of device extraction.
- The connector must not be mounted in such a way that the connector plug is biased more than 2 degrees sideways.

Figure 5-17 Connector plug engagement



Note: All Lightning connector faces besides the cosmetic plug faces should be hidden within the accessory such that the faces are not visible as shown in Figure 5-18 (page 147). Gaps between the accessory surface and the Lightning connector must not exceed 0.25 mm.

Figure 5-18 Connector plug exposure



Accessories that use a Lightning (C48) or Lightning (C68) connector must shield the connector. See Shielding (page 152) and Shielding (page 163).

5.4.2 Cable Accessories

A cable is an accessory that integrates a Lightning connector in such a way that the rigid enclosure holding the connector contains only the components necessary to integrate the connector. All other components are held in a separate enclosure that is connected by wire or flex.

An enclosure refers to the rigid portion of an accessory that encloses the components of the accessory. If a cable incorporates a strain relief, the strain relief is not counted in the enclosure. An enclosure must not deform under normal loads.

Figure 5-19 Connector cable enclosure

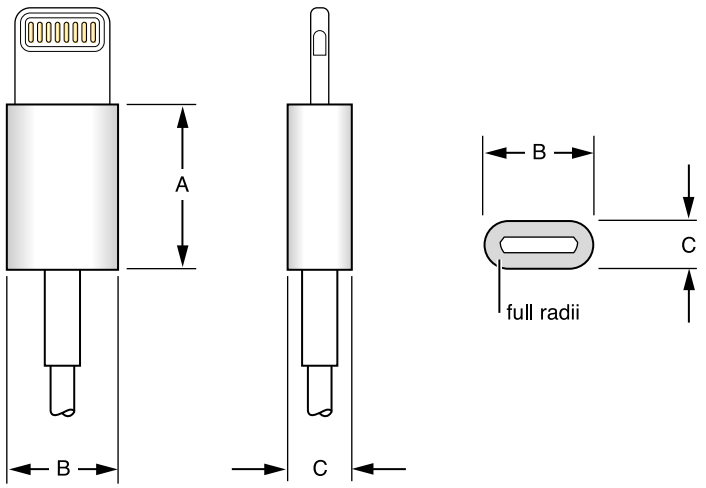
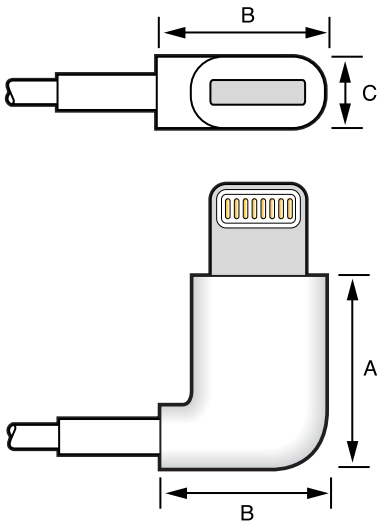


Figure 5-20 Connector cable enclosure with right angle



The Lightning connector enclosure dimensions must not exceed the following dimensions with full radii rounded edges as shown in Figure 5-19 (page 147) and Figure 5-20 (page 148):

- (A) 20 mm
- (B) 12.00 mm
- (C) 6.25 mm

Connector enclosure must contain rounded edges.

Although strain relief designs may vary, the cable enclosure must adhere to the specified requirements.

> **Note:** Cables that do not meet the enclosure dimensions will be treated as a dongle accessory. See Dongle Accessories (page 167) for more information.

5.4.2.1 Cables with USB Connectors

All accessory cables that incorporate both a Lightning connector and a USB-A plug (Standard, Mini, or Micro) must use a USB Device Mode Lightning connector configuration.

See Figure 5-21 (page 149) and Figure 5-22 (page 149) for all allowed USB to Lightning cable integrations.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-21 Allowable USB-A Lightning Cable Integrations

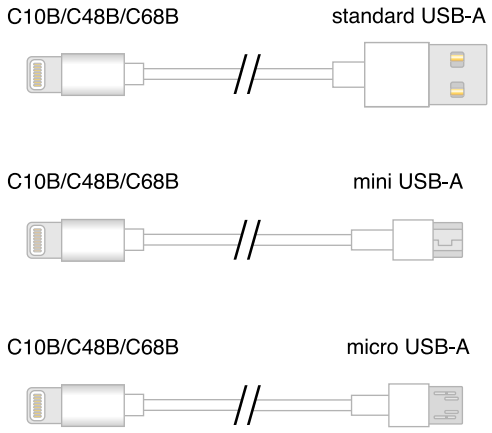
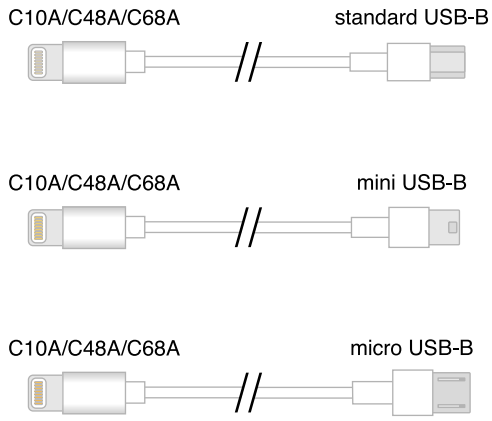


Figure 5-22 Allowable USB-B Lightning Cable Integrations



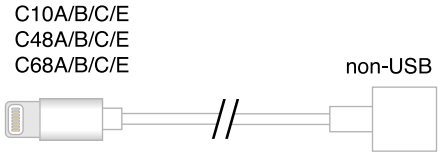
Accessory cables that incorporate a Lightning connector must not terminate in a USB receptacle.

5.4.2.2 Cables with Non-USB Connectors

Accessory cables that incorporate a Lightning connector must not terminate in a coaxial power connector. Examples of such connectors can be found in the IEC 60130-10, EIAJ, or DIN 45323 standards specifications.

Accessory cables that incorporate a Lightning connector must not terminate in a non-USB receptacle.

Figure 5-23 Example Non-USB Lightning Cable Integration



5.4.2.3 Cable Encapsulation

Additionally, exposed copper of cable/flex and exposed PCB copper should be encapsulated after soldering with electrically nonconductive and liquid sealing compound, for moisture protection. Some suggested compounds that can be used for this purpose are:

- Loctite 3703 UV Encapsulant
- Henkel UV9060 UV Encapsulant
- Cemedine SuperX 8008 Neo Encapsulant (RTV)
- Dymax 29990 UV Encapsulant

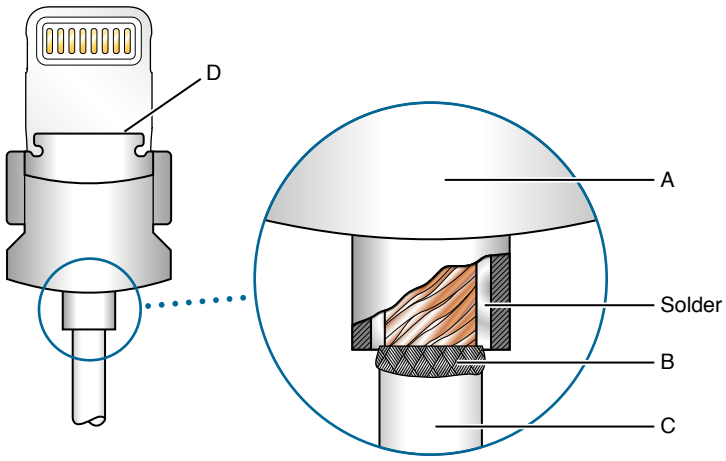
A jet dispense process is recommended for applying encapsulation. Some recommended vendors of encapsulation equipment are:

- Nordson EFD/ASYMTEK (http://www.nordson.com)
- Musashi Engineering (http://www.musashi-engineering.co.jp)

5.4.2.4 Cable EMC Considerations

Figure 5-24 (page 150) illustrates a typical cable accessory integration of the Lightning (C10) connector. Accessory developers are responsible for all components other than Lightning (C10), including the crimp and cable terminations.

Figure 5-24 Cable EMC Diagram



The following recommendations related to EMC apply to all cable accessories, using Figure 5-24 (page 150) as a visual reference:

- The cable (C) should have both braid and foil shield. The braid should have a minimum of 95% coverage.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Shield braid (B) should make 360 degree termination to the plug housing (A). Copper tape is recommended for this purpose. The termination should include mechanically secured crimp and soldering. 'Pigtail'-type connections should not be used.
- Connection from plug (D) to plug housing should make a 360 degree termination.
- Connector-to-connector resistance should be <100 mΩ maximum end to end.

5.4.2.5 Lightning (C48)/Lightning (C68) Cables

All cables that terminate in a Lightning (C48) or Lightning (C68) connector must additionally comply with the requirements in this section.

All cables that incorporate a Lightning (C48) or Lightning (C68) connector must pass the test procedures specified in Cable Accessories (page 173) and Lightning (C48)/Lightning (C68) Accessories (page 170).

5.4.2.5.1 Cable Termination

Lightning (C48)/Lightning (C68) cable termination must be done using one of the following methods:

- Hand soldering
- Hot bar soldering
- Laser soldering

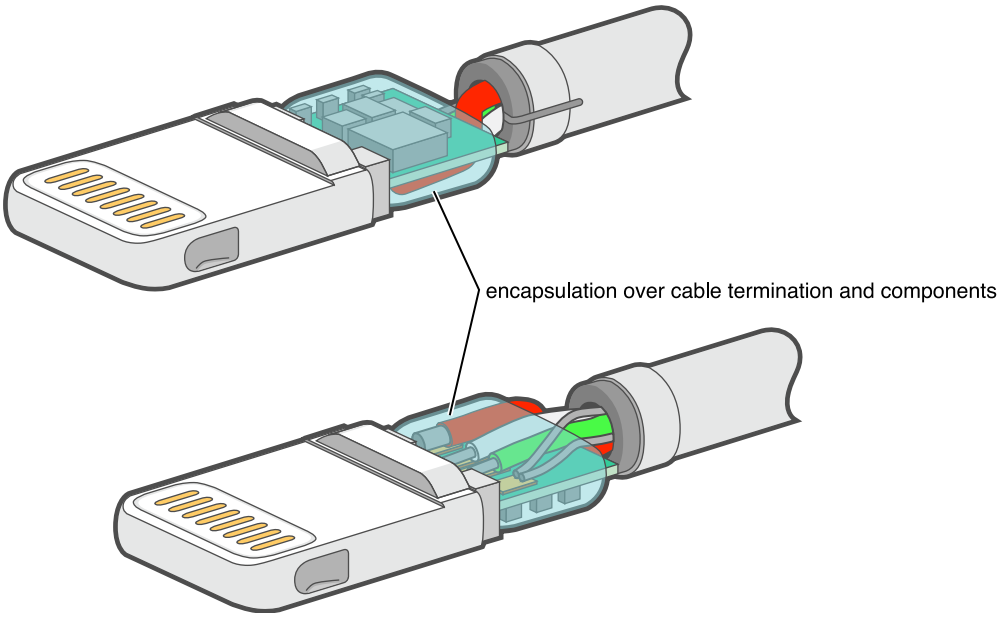
Ensure that none of the components of the connector are damaged during termination.

5.4.2.5.2 Encapsulation

Both sides of the Lightning (C48) or Lightning (C68) connector must be encapsulated (see Cable Encapsulation (page 150)) to protect it from liquid ingress and damage. UV cure encapsulation is recommended. Avoid processes that subject the connector to heat over 85 °C.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

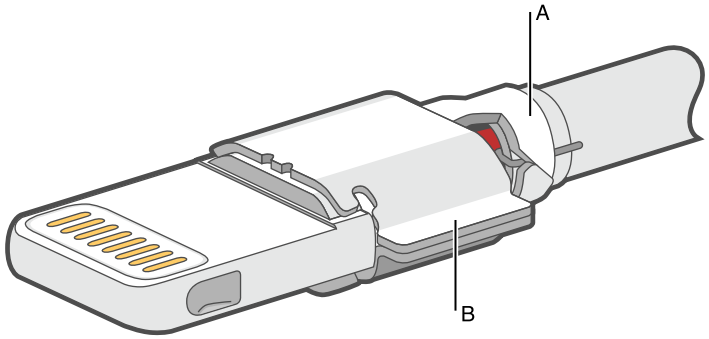
Figure 5-25 Lightning (C48) Cable Encapsulation Diagram



5.4.2.5.3 Shielding

All cables incorporating a Lightning (C48) or Lightning (C68) connector must incorporate a cable shields as shown in Figure 5-26 (page 152). Use of 0.40 mm thick SUS 304 1/2H material is recommended.

Figure 5-26 Lightning (C48) Cable Shielding Diagram



The shield must extend to crimp to the cable shield braid (A) as illustrated in Figure 5-26 (page 152). See Cable EMC Considerations (page 150) for additional information.

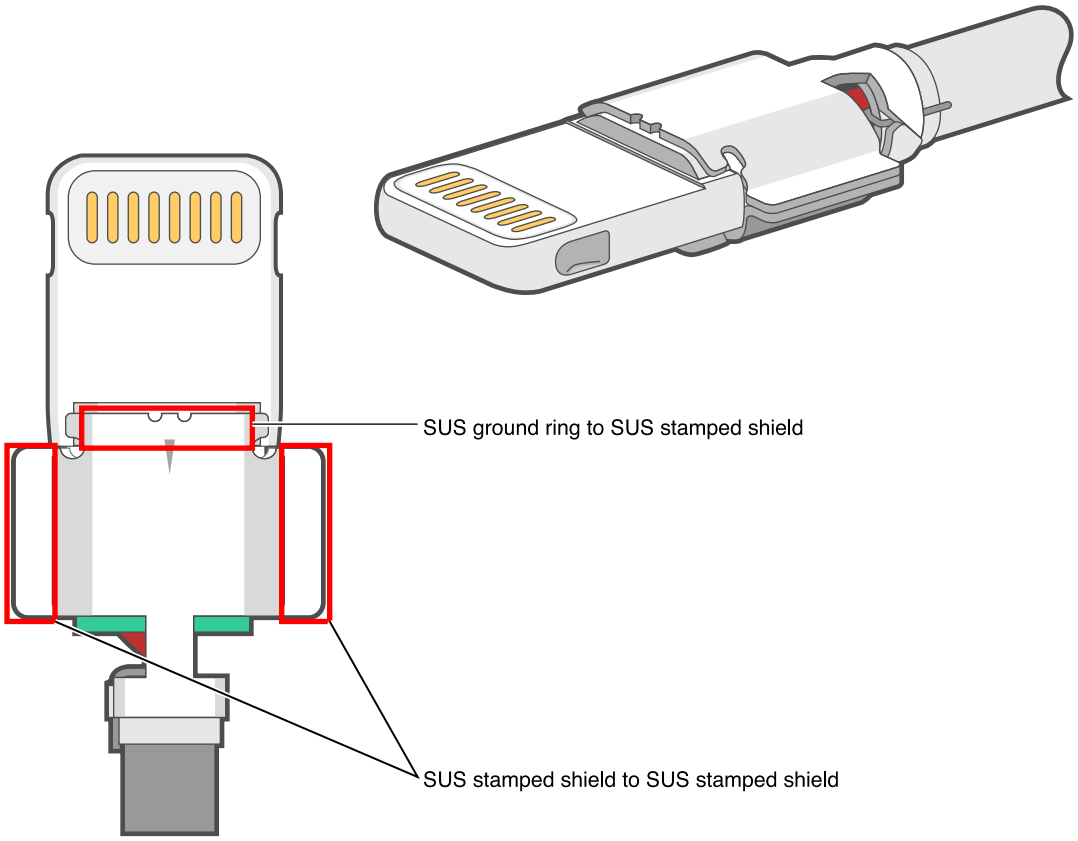
Additionally, flanges (B) are recommended for increased plug strength as illustrated in Figure 5-26 (page 152).

Shields must be free of burrs. Electropolishing is recommended as burrs can cut into wires and cause failures either in the factory or in the field.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The SUS stamped shield must be laser welded to the Lightning (C48) or Lightning (C68) ground ring as shown in Figure 5-27 (page 153). Recommended flange welding areas are also indicated. See Laser Welding (page 158).

Figure 5-27 Lightning (C48) Cable Shield Laser Welding Diagram



Apple recommends the following two-part cable shield design for Lightning (C48).

Figure 5-28 Lightning (C48) Recommended Cable Shield, Crimp

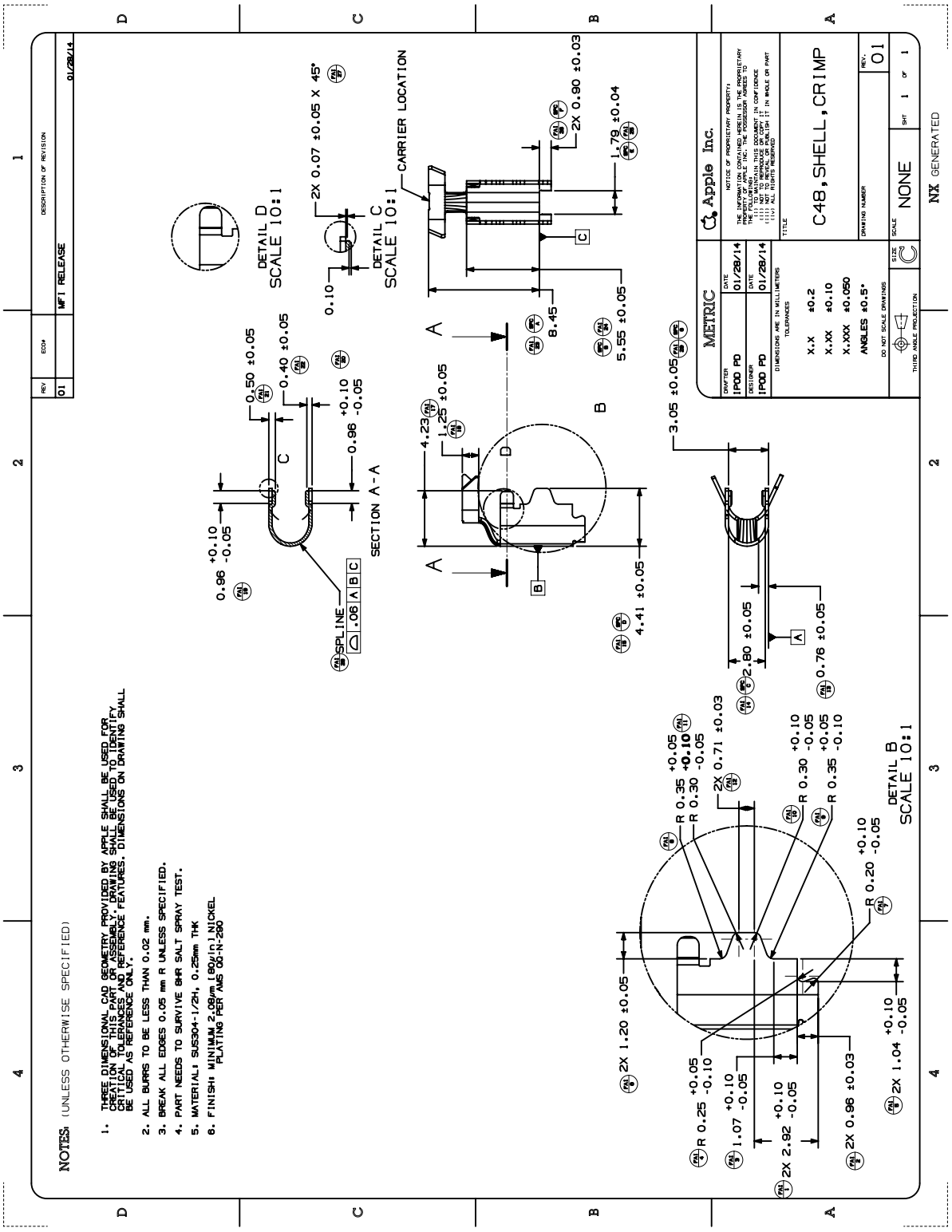
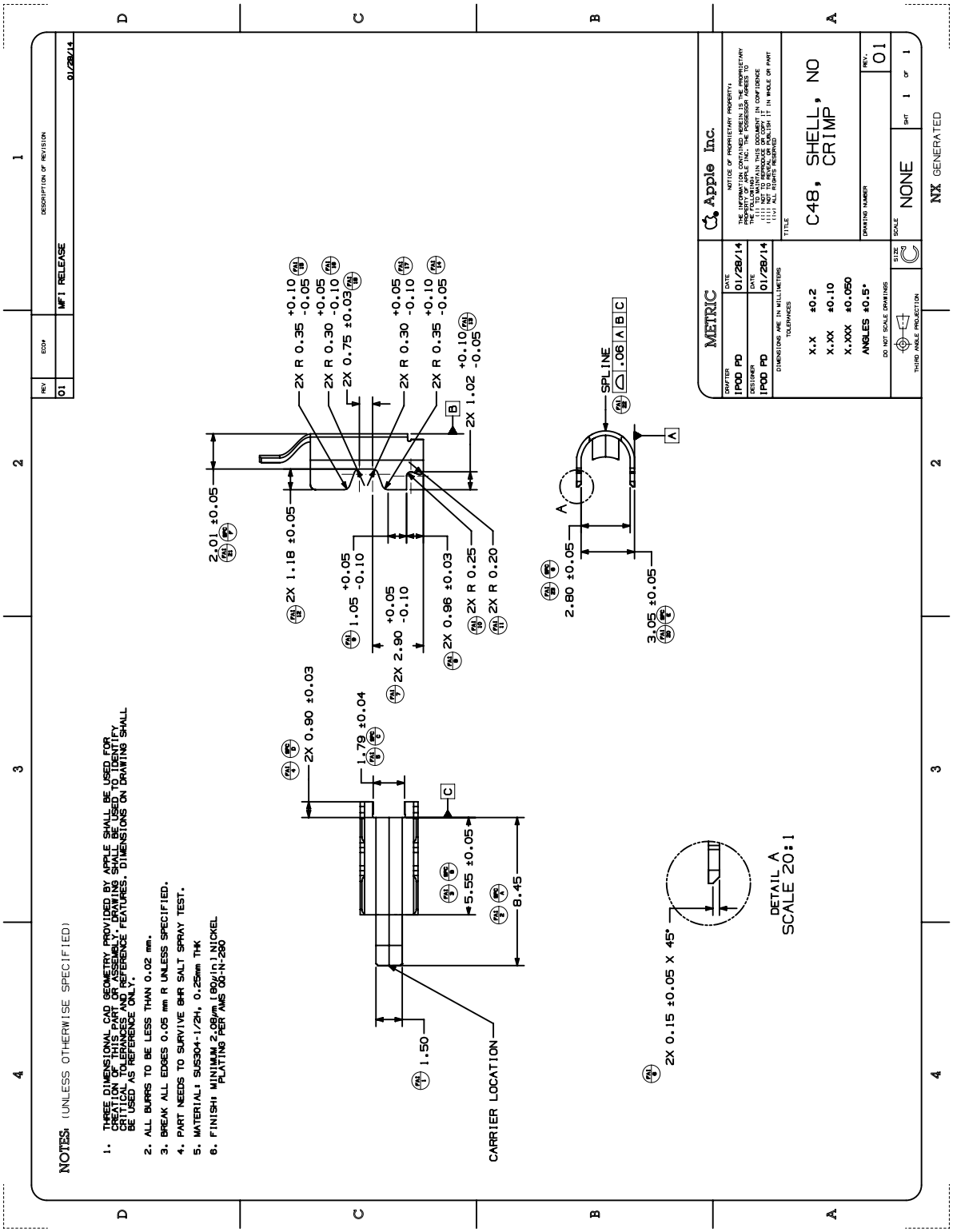


Figure 5-29 Lightning (C48) Recommended Cable Shield, No Crimp



Apple recommends the following two-part cable shield design for Lightning (C68).

Figure 5-30 Lightning (C68) Recommended Cable Shield, Crimp

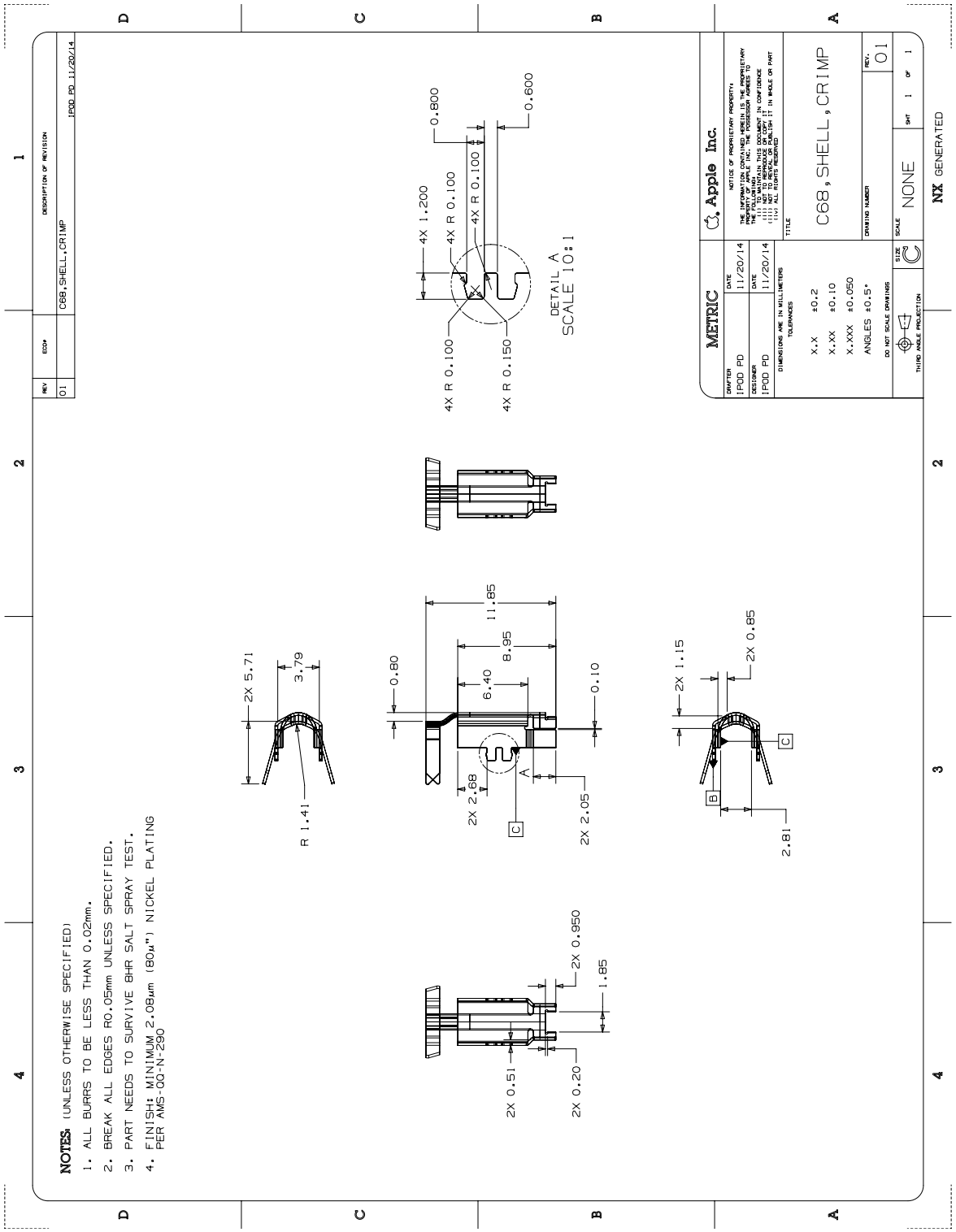
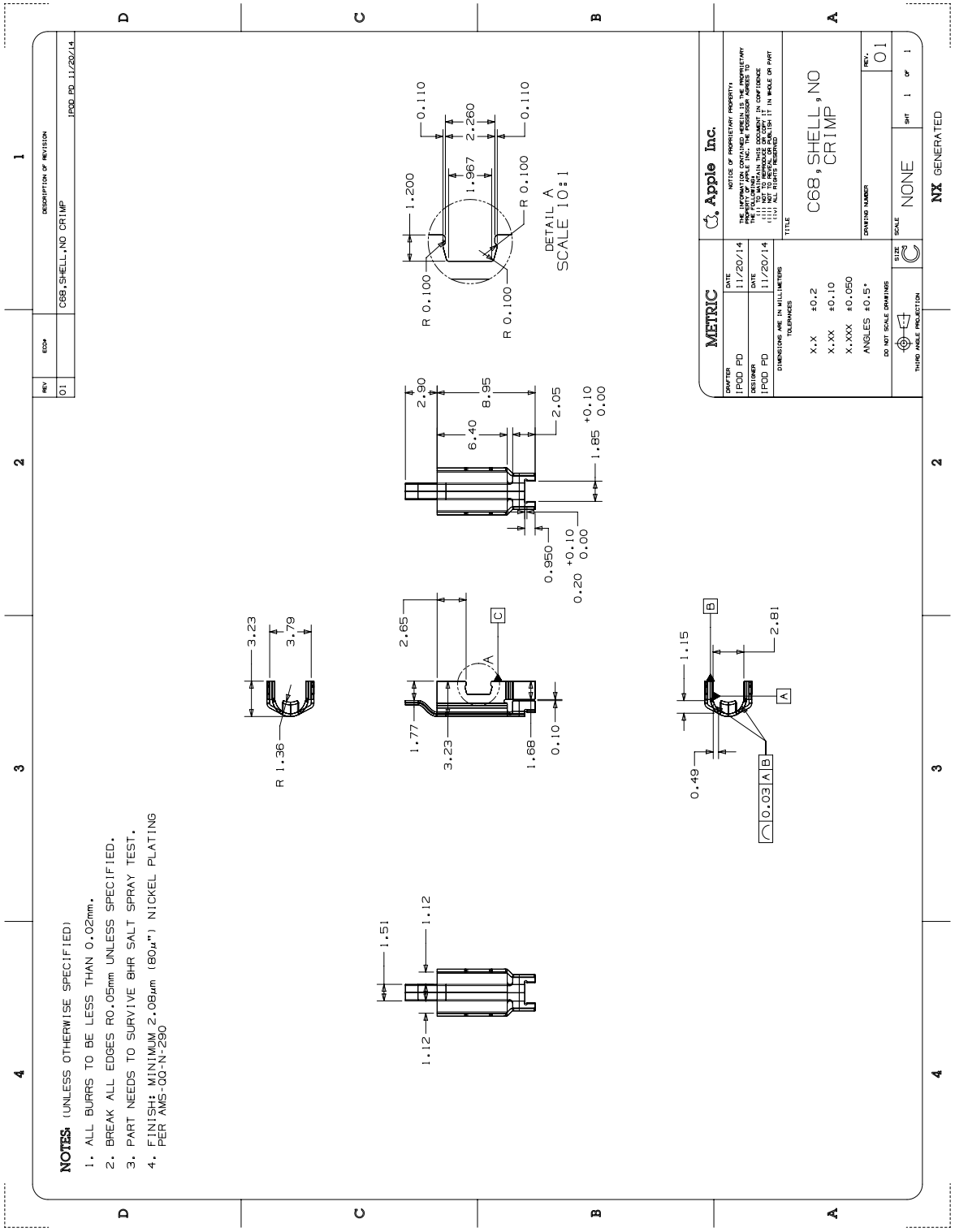


Figure 5-31 Lightning (C68) Recommended Cable Shield, No Crimp



5.4.2.5.4 Laser Welding

A SUS stamped shield must be laser welded to the Lightning (C48) or Lightning (C68) ground ring. Apple suggests the following laser welding equipment vendors:

- Han's Laser (http://www.hanslaser.net)
- Lype Laser (http://www.lype-laser.com)

5.4.3 Dock Accessories

A dock is an accessory that is designed to hold the Apple device when it is connected to the Lightning connector. In a dock, the Apple device can be moved relative to the Lightning connector (unlike in a form-fitting accessory, see Form-Fitting Accessories (page 167)).

All dock accessories must contain integrated backstops. See iPhone and iPod touch Dock Requirements (page 158) and iPad Dock Requirements (page 160) for more information.

All dock accessories must use either:

- Lightning (C48) or Lightning (C68) with a shield (see Shielding (page 163))
- Lightning (C11)

One of two approaches must be used to secure the Lightning connector to the dock accessory:

- 4 M1.6 screws are used to secure the Lightning connector to the dock accessory.
- 2 M1.6 screws are used in the bottom holes of the Lightning connector and 2 locating posts are fed through the top holes. Apple recommends that the size and tolerance of the locating posts is 1.68 ±0.05 mm.

The tip of the Lightning plug is made with a conductive material. Apple recommends that a continuity tester be designed and used during the accessory manufacturing process to assure that the minimum exposed plug length meets the requirement of 6.47 mm.

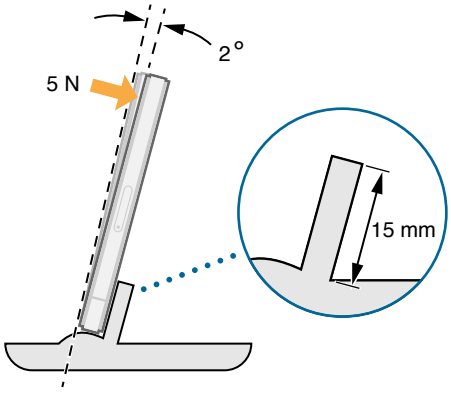
5.4.3.1 iPhone and iPod touch Dock Requirements

Any dock accessory that is intended to be used with an iPhone or iPod touch, or which can reasonably be connected to an iPhone or iPod touch, must comply with the requirements in this section in addition to the general requirements.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

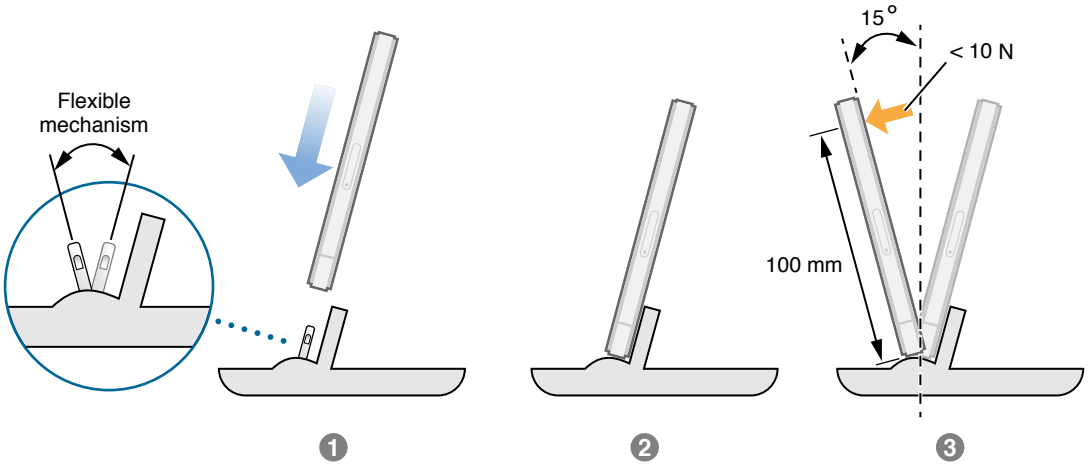
The accessory must have a backstop that contacts the iPhone or iPod touch at a point at least 15 mm vertically from the bottom of the device when a force of 5 N is applied perpendicular to the screen of the device at a point 100 mm vertically from the bottom of the device. The iPhone or iPod touch will rotate approximately 2 degrees from its rest position relative to the backstop when this force is applied. A typical backstop is shown in Figure 5-32 (page 159).

Figure 5-32 iPhone/iPod touch/iPod nano backstop



The accessory must have a flexible mechanism that allows a user to rotate the iPhone or iPod nano from the rest position to at least 15 degrees from vertical. This mechanism must prevent the user from applying 10 N perpendicular to the iPhone or iPod nano at a distance 100 mm from the bottom of the device before the device reaches the compliant limit.

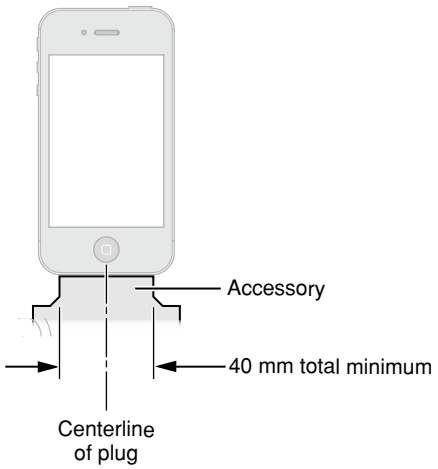
Figure 5-33 iPhone/iPod touch flexible mechanism



The accessory should contact the bottom of the device at two points, each of which is at least 20 mm away from the side of the connector, when the device is in the rest position.

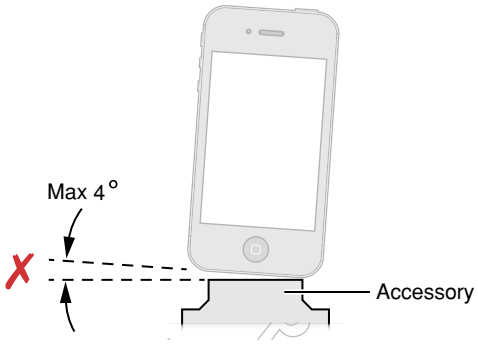
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-34 iPhone/iPod touch contact



The Lightning connector is designed to disconnect if biased more than 4 degrees sideways.

Figure 5-35 iPhone/iPod touch sideways angle



The accessory should not block any iPhone or iPod touch speakers or microphones. See Cases (page 487) for a pointer to dimensional drawings of iPhone and iPod touch that call out locations of speakers and microphone.

5.4.3.2 iPad Dock Requirements

Any dock accessory that is intended to be used with an iPad, or which can reasonably be connected to an iPad, must comply with the requirements in this section in addition to the general requirements.

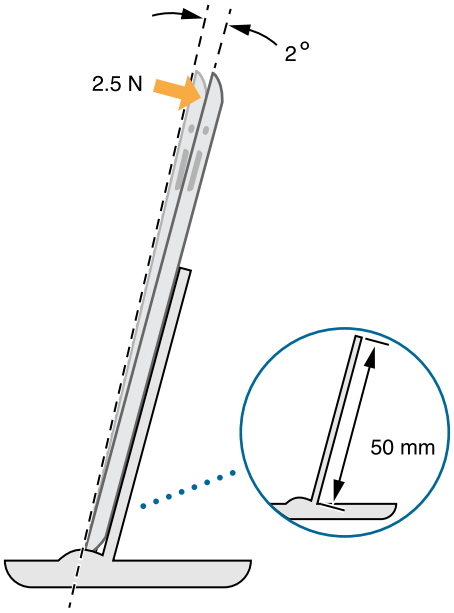
The accessory must have a backstop that contacts the iPad at a point at least 50 mm vertically from the bottom of the device when a force of 2.5 N is applied perpendicular to the screen of the device at a point 200 mm vertically from the bottom of the device.

The accessory should contact the bottom of the device at two points, each of which is at least 50 mm away from the side of the connector, when the device is in the rest position.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The accessory and its backstop should not block any speakers or microphones.

Figure 5-36 iPad backstop

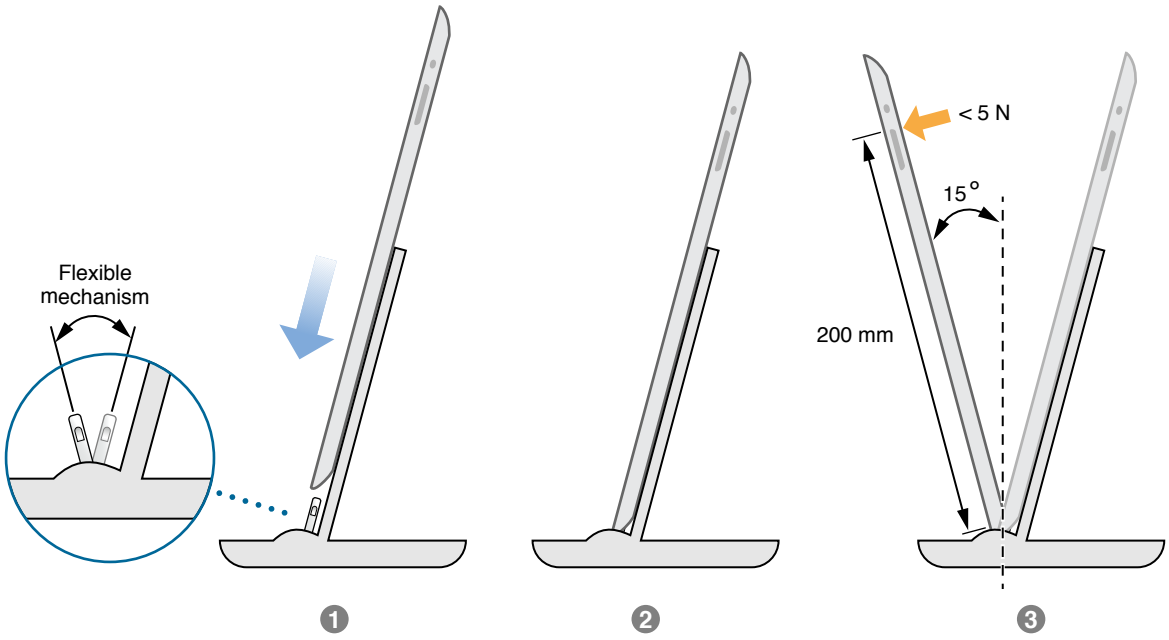


The accessory must have a flexible mechanism that allows a user to rotate the iPad from its rest position to at least 15 degrees from vertical. This mechanism must prevent the user from applying more than 5 N force perpendicular to the iPad at a distance of 200 mm from the bottom of the device before the device reaches the compliant limit.

This flexibility requirement is illustrated in Figure 5-37 (page 162). Drawings 1 and 2 show the iPad being engaged with the connector plug. Drawing 3 shows the device responding to a force of 5 N or less, applied 200 mm above the bottom of the device, by rotating up to 15 degrees from its vertical position.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-37 iPad flexible mechanism



5.4.3.3 iPod nano Dock Requirements

Any dock accessory that is intended to be used with an iPod nano, or which can reasonably be connected to an iPod nano, must comply with the requirements in this section in addition to the general requirements.

The accessory must have a backstop that contacts the iPod nano at a point at least 15 mm vertically from the bottom of the device when a force of 10 N is applied perpendicular to the screen of the device at a point 50 mm vertically from the bottom of the device.

The accessory should contact the bottom of the device at two points, each of which is at least 20 mm away from the side of the connector, when the device is in the rest position.

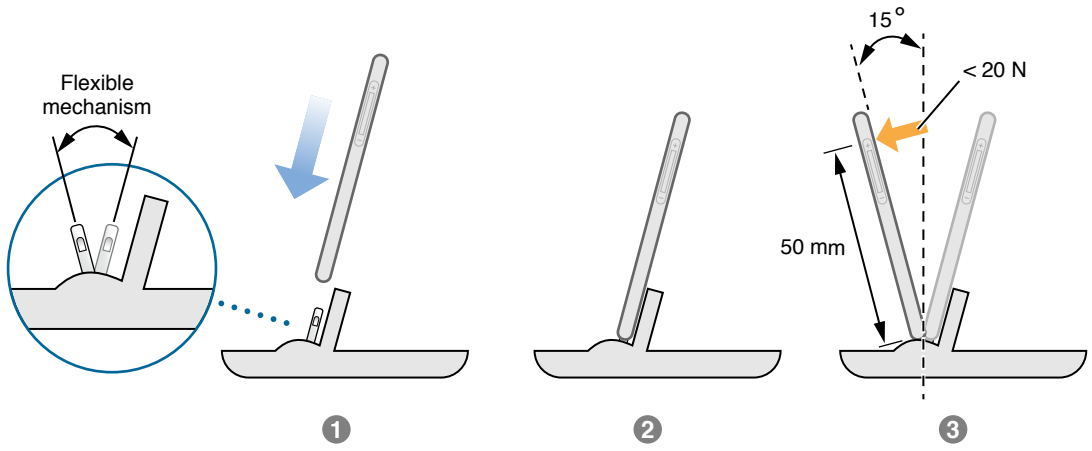
The accessory and its backstop should not block any speakers or microphones.

The accessory must have a flexible mechanism that allows a user to rotate the iPod nano from its rest position to at least 15 degrees from vertical by only applying a load to the device. This mechanism must prevent the user from applying more than 20 N force perpendicular to the iPhone or iPod touch at a distance of 50 mm from the bottom of the device before the device reaches the compliant limit.

This flexibility requirement is illustrated in Figure 5-38 (page 163). Drawings 1 and 2 show the iPod nano being engaged with the dock connector plug. Drawing 3 shows the device responding to a force of 20 N or less, applied 50 mm above the bottom of the device, by rotating up to 15 degrees from its vertical position.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-38 iPod nano flexible mechanism



5.4.3.4 Lightning (C48)/Lightning (C68) Docks

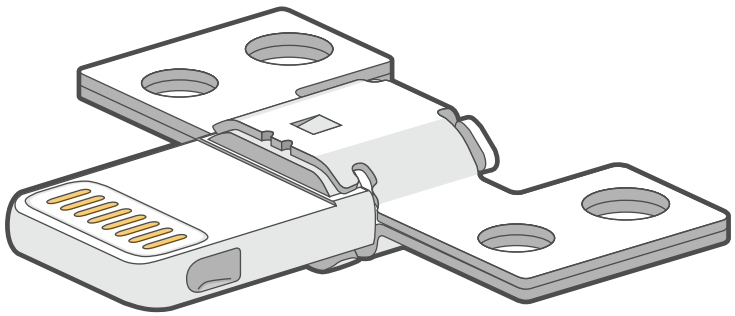
All docks incorporating a Lightning (C48) or Lightning (C68) connector must additionally comply with the requirements in this section.

All docks that incorporate a Lightning (C48) or Lightning (C68) connector must pass the test procedures specified in Dock Accessories (page 179) and Lightning (C48)/Lightning (C68) Accessories (page 170).

5.4.3.4.1 Shielding

All docks incorporating a Lightning (C48) or Lightning (C68) connector must incorporate a SUS stamped shield as shown in Figure 5-39 (page 163). Use of 0.40 mm thick SUS 304 1/2H material is recommended.

Figure 5-39 Lightning (C48) Dock Shielding Diagram

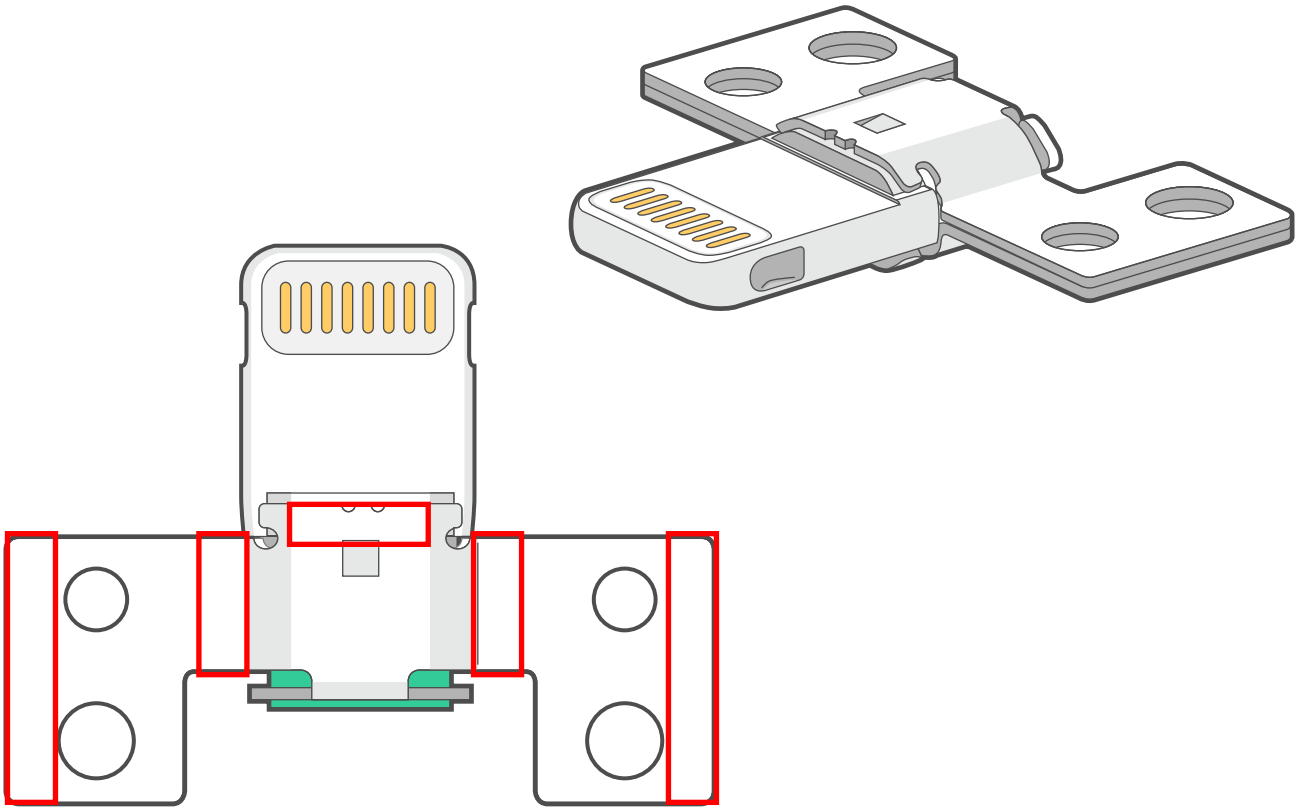


Shields must be free of burrs. Electropolishing is recommended as burrs can cut into wires and cause failures either in the factory or in the field.

The SUS stamped shield must be laser welded to the Lightning (C48) or Lightning (C68) ground ring as shown in Figure 5-40 (page 164). Recommended flange welding areas are also indicated. See Laser Welding (page 158).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-40 Lightning (C48) Dock Shield Laser Welding Diagram



Laser welds between shields and Lightning (C48) or Lightning (C68) must withstand 2000 Nmm bending force. See Dock Shield Bend Test (C48, C68) (page 179).

Apple recommends the following two-part dock shield design for Lightning (C48).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-41 Lightning (C48) Recommended Dock Shield Top

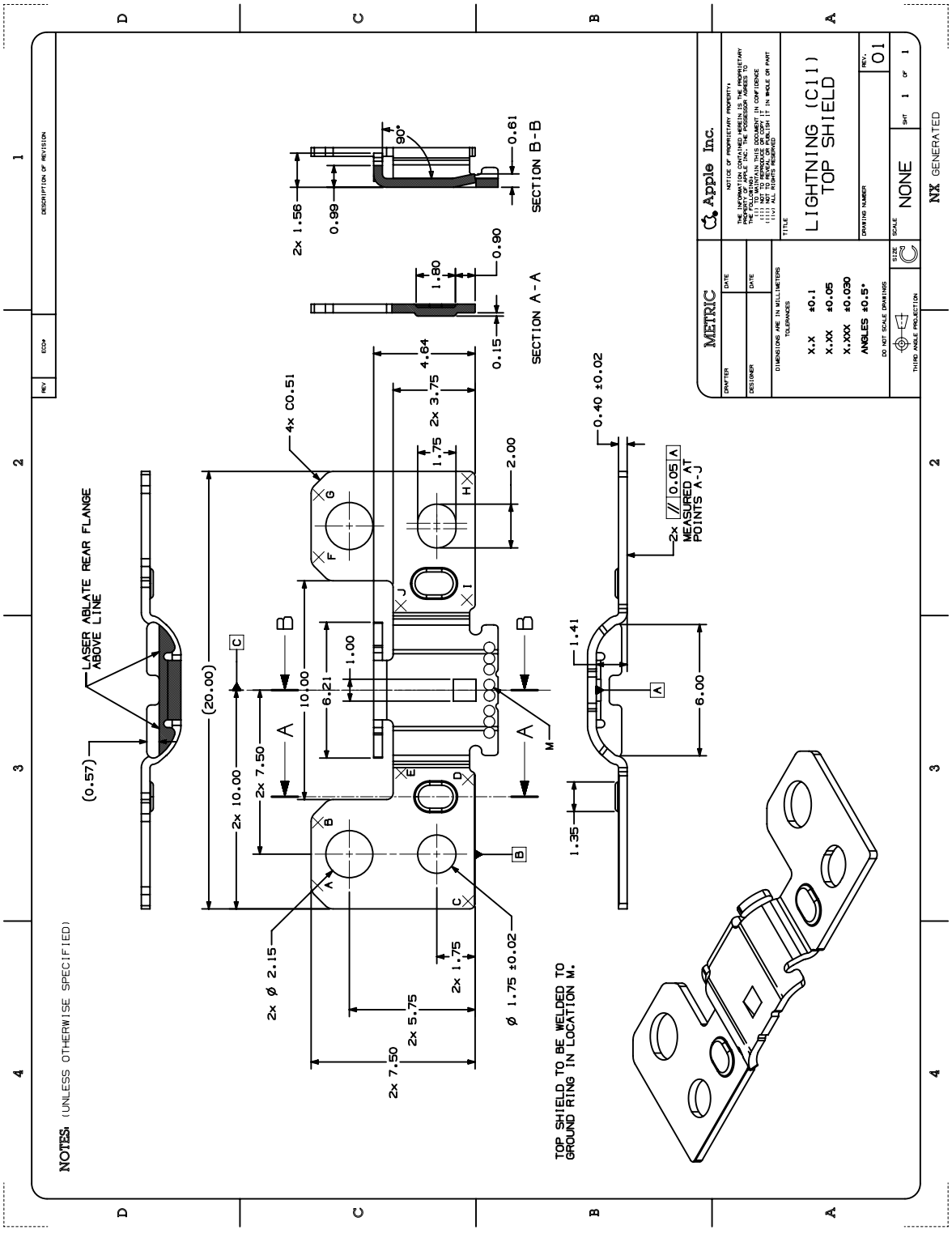
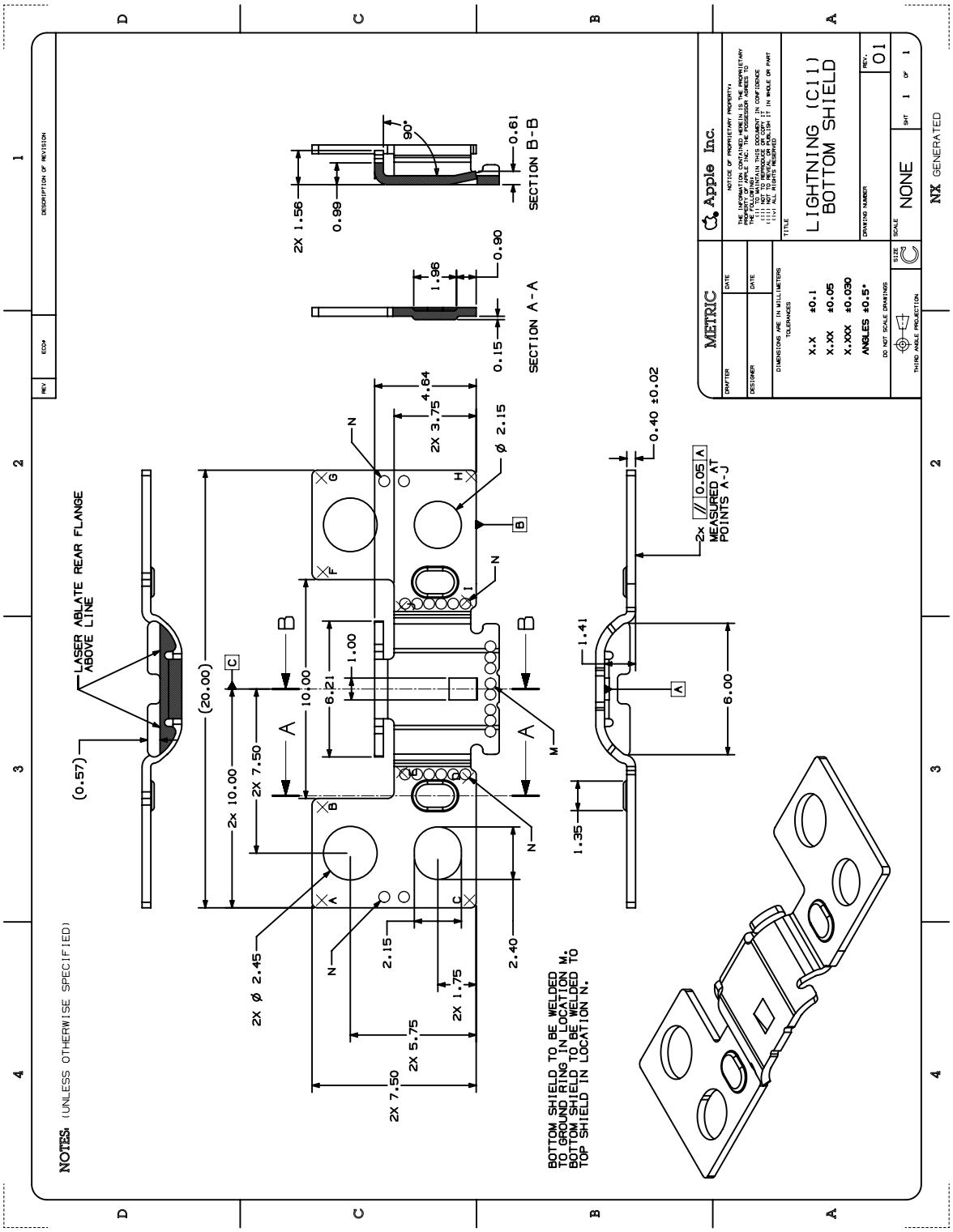


Figure 5-42 Lightning (C48) Recommended Dock Shield Bottom



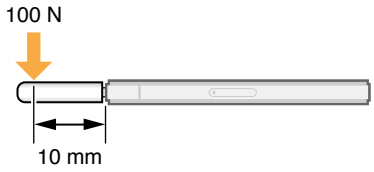
5.4.4 Dongle Accessories

A dongle accessory is an accessory that integrates the Lightning connector in an enclosure along with other components. A dongle accessory can be moved relative to the Apple device (unlike in a form-fitting accessory, see Form-Fitting Accessories (page 167)) and is not designed to hold up the Apple device (unlike in a dock accessory, see Dock Accessories (page 158)).

The following requirements apply to all dongle accessories:

- All dongle accessories must use either:
  - Lightning (C48) or Lightning (C68) with a shield (see Shielding (page 163))
  - Lightning (C11)
  - Lightning (C12)
- Dongle accessories that function exactly like the Apple Lightning to Micro-USB Adapter must use Lightning (C10B) or Lightning (C48B) with a shield. (See Shielding (page 152) and Shielding (page 163) for some options.)
- All dongle accessories must break or bend before the user can apply a 100 N force 10 mm away from the bottom of the device as illustrated in Figure 5-43 (page 167).

Figure 5-43 Dongle Maximum Allowable Force



5.4.5 Form-Fitting Accessories

A form-fitting accessory is an accessory that integrates the Lightning connector in such a way that it allows no relative movement between the Lightning connector and the Apple device when connected. Battery pack cases and game controllers which enclose the Apple device are examples of form-fitting accessories. Form-fitting accessories that substantally enclose the Apple device must comply with Cases (page 487).

> **Note:** Form-fitting accessories must avoid degrading the wireless performance of Apple devices. See RF Transmission and Reception (page 65)

Form-fitting accessories may mount the Lightning connector rigidly without any compliance. However, the rigid mounting should not contact the connector's PCB in any way, as shown in Figure 5-16 (page 146).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

If a Lightning (C48) is used, it must incorporate a shield. The dock shield is recommended, see Shielding (page 163).

5.5 Connector Electrical Requirements

Accessories do not need to supply power to the Lightning connector. Instead, the connector draws a small amount of power from the Apple device upon insertion. All power supplied by the accessory to the Device Power pad/pin is routed to the Apple device.

5.5.1 Connector Ground Pad/Pin Connection Requirements

The following requirements apply to connections made to the ground pads/pins:

- Ground/Ground 1 must be permanently connected to the accessory's system ground.
- For Lightning (C10), Lightning (C11), Lightning (C48), and Lightning (C68), Ground 2 must be connected to the accessory's system ground if the accessory draws power from the Apple device, but may be left unconnected (NC) otherwise.
- For Lightning (C12), Ground 2 must be connected to the accessory's system ground.
- USB Host or Device Mode accessories must connect a drain wire to Ground/Ground 1.
- The Lightning connector shield must not be used as the only ground path.

5.5.2 Connector Shielding Requirements

All accessory components that carry power or data signals to and from the Lightning connector pads/pins must be shielded by grounded metal in accordance with EMI, EMC, and RF regulations.

5.5.3 Connector ESD Protection Requirements

ESD protection is present on all pads/pins in the Lightning connector.

5.5.4 Connector Signal Integrity Requirements

The following signal integrity requirements apply to all accessories that transmit or receive USB data.

- The accessory must comply with all differential insertion loss requirements as specified in Section 7.1.17 in the USB 2.0 specification.
- The accessory must comply with all impedance requirements as specified in Section 7.1.1.3 in the USB 2.0 specification.

Jay Xiao Weifang Goertek Electronics Co.,Ltd.

5.5.5 Lightning (C48)/Lightning (C68) Accessory Testing Requirements

Every accessory that integrates a Lightning (C48) or Lightning (C68) connector must pass the following tests on the factory line:

- Over voltage protection (see Over Voltage Protection Test (page 170))
- Current limit (see Current Limit Test (page 171))
- Quiescent current consumption (see Quiescent Current Consumption Test (page 172))

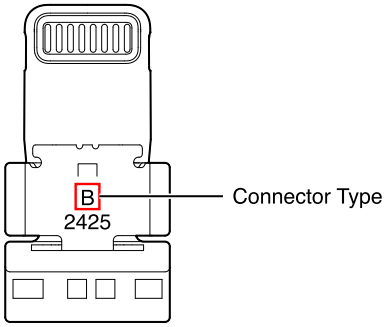
5.6 Test Procedures

5.6.1 All Accessories

5.6.1.1 Connector Version and Configuration

1. Via visual inspection, verify that the accessory uses Lightning (C10), Lightning (C11), Lightning (C12), Lightning (C48), or Lightning (C68) in accordance with the requirements in Connector Versions (page 129).
2. Via visual inspection, verify that the accessory uses the correct Lightning configuration (A, B, C, E, etc.)
   - Lightning (C10), Lightning (C11) and Lightning (C12) connectors have the configuration laser etched on the shield as shown in Figure 5-44 (page 169).
   - Lightning (C48) and Lightning (C68) connector trays will have a unique color stripe down the center.
     - Lightning (C48A)/Lightning (C68A) trays have a green stripe.
     - Lightning (C48B)/Lightning (C68B) trays have a yellow stripe.
     - Lightning (C48C)/Lightning (C68C) trays have a blue stripe.
     - Lightning (C68E) trays have an orange stripe.

Figure 5-44 Identifying Lightning (C10), Lightning (C11), or Lightning (C12) connector configuration



5.6.1.2 Static Plug Exposure

Using a caliper, verify that at least 6.47 mm of the connector plug is exposed throughout the contact area of the Apple device as shown in Figure 5-17 (page 146).

5.6.1.3 Plug Exposure Under Insertion Force

1. Place the accessory in a clamp or other mechanism to securely hold it in place.
2. Using a calibrated force gauge, apply 30 N of pressure to the connector in the direction of device insertion via the measuring tip of the force gauge.
3. Using a caliper, verify that at least 6.47 mm of the connector plug is exposed throughout the contact area of the Apple device as shown in Figure 5-17 (page 146).

5.6.1.4 Plug Mounting Under Extraction Force

1. Place the accessory in a clamp or other mechanism to securely hold it in place.
2. Using a calibrated force gauge, apply 30 N of pressure to the connector in the direction of device extraction by clamping on to the plug and pulling on it.
3. Verify that the connector does not pull or break away from the accessory.

5.6.1.5 Plug Bias

Using a protractor or similar measuring device, measure the angular bias and verify that no more than 2 degrees of sideways bias is present as shown in Figure 5-17 (page 146).

5.6.1.6 Hidden Connector Faces

Via visual inspection, verify that all Apple Lightning Connector faces besides the cosmetic plug faces are hidden within the accessory as shown in Figure 5-18 (page 147).

5.6.2 Lightning (C48)/Lightning (C68) Accessories

5.6.2.1 Over Voltage Protection Test

Note: This procedure is required for all Lightning (C48) or Lightning (C68) integrations on the factory line. See Lightning (C48)/Lightning (C68) Accessory Testing Requirements (page 169).

Verify that the Lightning (C48) or Lightning (C68) connector is functioning correctly after integration by performing the following test procedure:

1. For cable integrations that terminate in a USB-A plug:
   a. Connect the USB side of the cable to the MFi USB Breakout.
   b. Connect the MFi USB Breakout to the power supply (pin 9 to power and pin 3 to ground).
2. For all other integrations, connect the accessory's power and ground terminating points to the power supply.
3. Connect the Lightning (C48) or Lightning (C68) side to the MFi Lightning Breakout.
4. Attach digital multimeter across PWR and GND on MFi Lightning Breakout to measure voltage.
5. Apply 5V on the power supply.
6. Ramp the power supply from 5V to 9V
7. Verify that the output voltage on PWR tracks VBUS until 6.0V ≤ VBUS ≤ 6.6V. At this point PWR must drop to 0V and remain until VBUS has dropped back below this threshold.

5.6.2.2 Current Limit Test

Note: This procedure is required for all Lightning (C48) or Lightning (C68) integrations on the factory line. See Lightning (C48)/Lightning (C68) Accessory Testing Requirements (page 169).

Verify that the Lightning (C48) or Lightning (C68) connector is functioning correctly after integration by performing the following test procedure:

1. For cable integrations that terminate in a USB-A plug:
   a. Connect the USB side of the cable to the MFi USB Breakout.
   b. Connect the MFi USB Breakout to the power supply (pin 9 to power and pin 3 to ground).
2. For all other integrations, connect the accessory's power and ground terminating points to the power supply.
3. Set current limit on power supply to at least 20 mA.
4. Connect the Lightning (C48) or Lightning (C68) side to the MFi Lightning Breakout.
5. Attach digital multimeter inline with PWR to measure current.
6. Apply 5V on the power supply.
7. Attach 3.5V SMU load (see Note below) across PWR and GND on MFi Lightning Receptacle, measure current through load. Current must be < 8 mA.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8. Attach a short across PWR and GND on MFi Lightning Receptacle, measure current through load. Current must not exceed 15 mA.

Note: The following four-quadrant SMU or equivalent set of power loads are required:

- 500Ω , > 100 mW
- "short" load, > 20 mA

5.6.2.3 Quiescent Current Consumption Test

Note: This procedure is required for all Lightning (C48) or Lightning (C68) integrations on the factory line. See Lightning (C48)/Lightning (C68) Accessory Testing Requirements (page 169).

Verify that the Lightning (C48) or Lightning (C68) connector is functioning correctly after integration by performing the following test procedure:

1. For cable integrations that terminate in a USB-A plug:
   a. Connect the USB side of the cable to the MFi USB Breakout.
   b. Connect the MFi USB Breakout to the power supply (pin 9 to power and pin 3 to ground).
2. For all other integrations, connect the accessory's power and ground terminating points to the power supply.
3. Attach digital multimeter inline with VBUS to measure current.
4. Apply -1 V on the power supply and measure current consumption. Current must not exceed 17 mA.
5. Apply 5 V on the power supply and measure current consumption. Current must not exceed 1 mA.

5.6.3 Mixed 30-pin/Lightning Connector Accessories

5.6.3.1 iAP1/iAP2 Messages

1. Run ATS in "ATS Capture" mode on each connector.
2. Run through all possible use cases.
3. Verify there are no errors and warnings in the summary panel and ATS trace.
4. Verify that both accessories are using the same data transport.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

5.6.3.2 Power Only

1. Verify that both connectors can provide the required current when all ports are under a full load, see Lightning Connectors on Power-Providing Accessories (page 664).
2. Verify connectors charge each Apple device simultaneously when connected.
3. Verify that iAP1/iAP2 messages are used to inform the device of available power.

5.6.3.3 Power and Data

1. Connect an Apple device on one connector, verify that a data transport has been established.
2. Connect a second Apple device to the secondary connector.
3. Verify that a data transport has been terminated with the first attached Apple device.
4. Verify that a data transport has been established with the last attached Apple device.

5.6.4 Cable Accessories

Using a caliper, verify that the connector enclosure dimensions are smaller than the limits specified in Figure 5-19 (page 147).

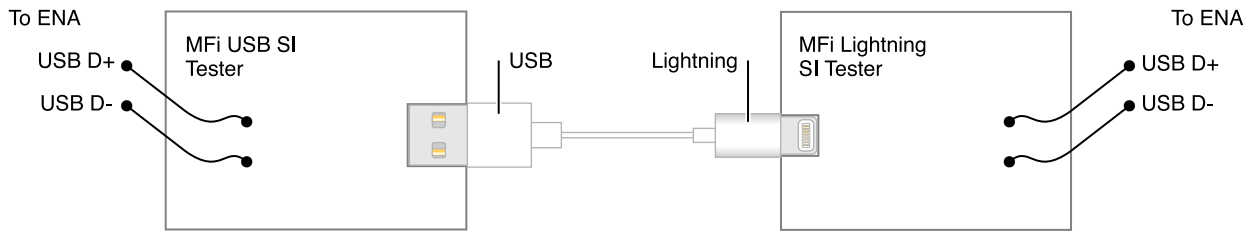
5.6.4.1 Signal Integrity Tests (C10A, C10B, C48A, C48B, C68A, C68B)

All cables that incorporate a Lightning (C10A), Lightning (C10B), Lightning (C48A), Lightning (C48B), Lightning (C68A), or Lightning (C68B) connector on one end and a USB plug on the other must be tested for differential insertion loss and differential impedance.

Equipment needed for differential insertion loss and differential impedance testing:

- Agilent E5071C-4K5 ENA with Enhanced TDR
- Agilent N4433A-010 RF ECal module, 300 kHz to 20 GHz, 3.5 mm, 4-port
- MFi USB SI Tester (MFI677-2103)
- MFi Lightning SI Tester (MFI677-1642)

Figure 5-45 Signal Integrity Test Setup for USB cables



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

On the Agilent ENA, enter TDR mode and run the Setup Wizard and select "Full Calibration". Make sure "Differential Two-Port" is selected when prompted. Run through the steps required to calibrate and test the accessory.

In the "Fixture Compensation" step, make the following connections:

1. MFi USB SI Tester "PIN3" to port 1 of Agilent ENA
2. MFi USB SI Tester "PIN2" to port 2 of Agilent ENA
3. MFi Lightning SI Tester "PIN2" to port 3 of Agilent ENA
4. MFi Lightning SI Tester "PIN3" to port 4 of Agilent ENA

Set limit lines for Tdd11 (differential impedance graph) and Sdd12/Sdd21 (differential insertion loss graph). Results should look similar to Figure 5-46 (page 174) and Figure 5-47 (page 175) as shown below.

Figure 5-46 Differential impedance graph with limit lines

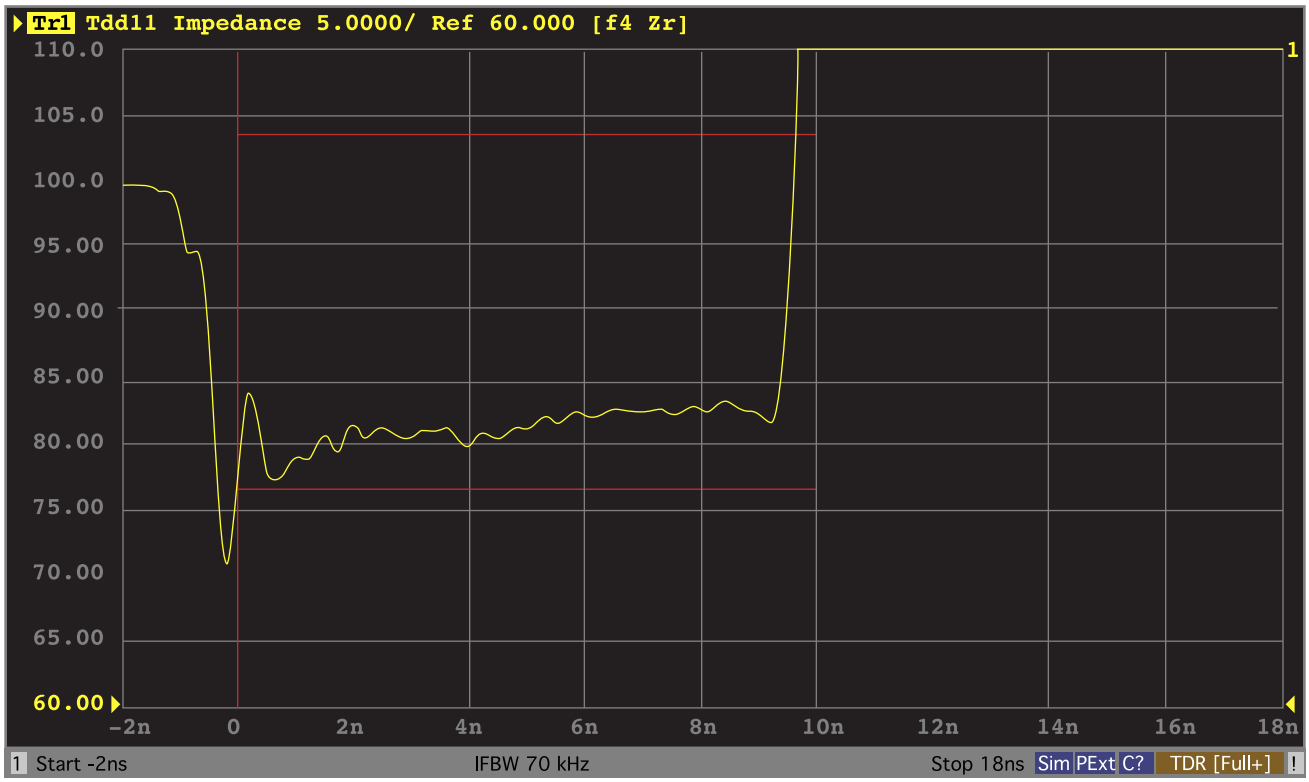
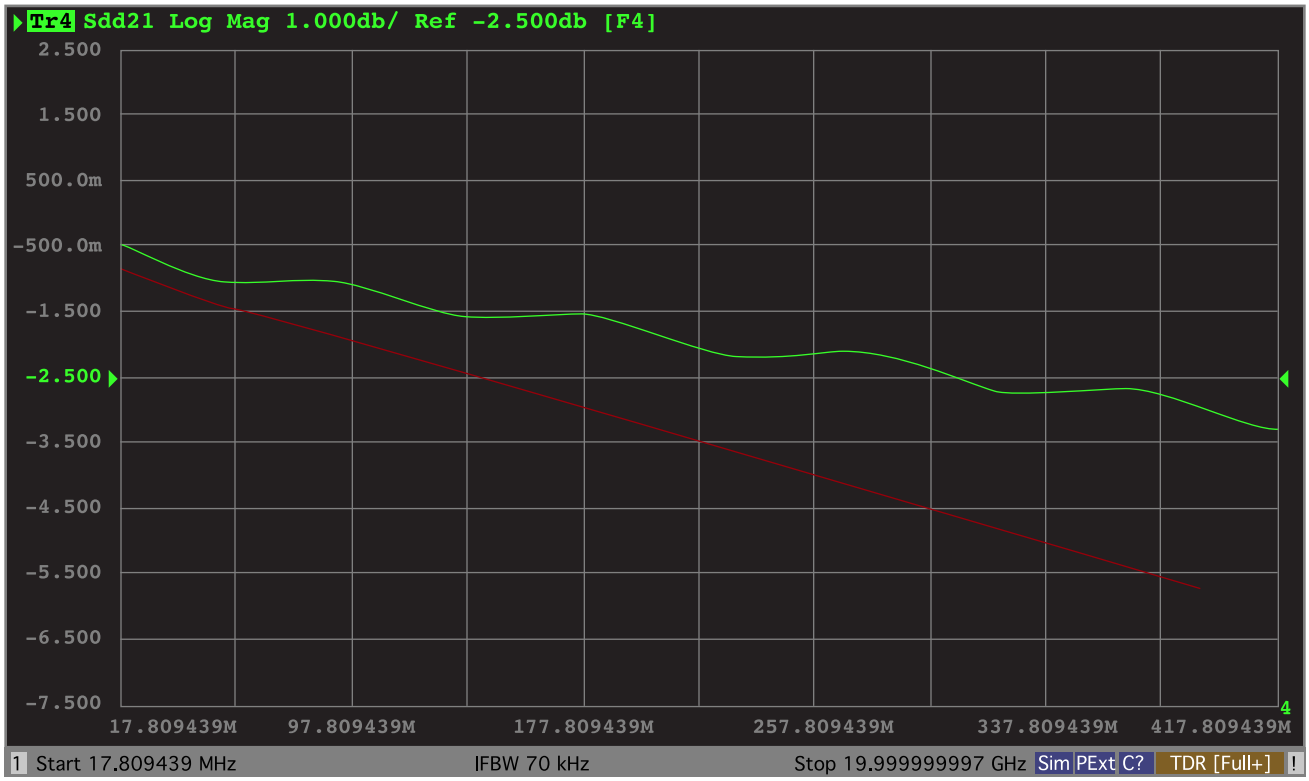


Figure 5-47 Differential insertion loss graph with limit lines



> **Note:** Differential insertion loss and differential impedance requirements can be found in chapter 7 of the *Universal Serial Bus Specification*, revision 2.0.

5.6.4.2 DC Resistance Tests (C10, C48, C68)

For cables that incorporate a Lightning (C10), Lightning (C48), or Lightning (C68) connector, refer to Figure 5-48 (page 176), Figure 5-49 (page 176), and Figure 5-50 (page 176) for DC resistance test setup.

Equipment needed for DC resistance testing:

- Digital multi-meter
- Power supply
- 2.5A load
- MFi Lightning Tester (MFI677-00167)
- MFi USB Breakout (MFI677-2104)

Figure 5-48 DCR Ground in Parallel with Shield Conductor Test Setup ($R_{GND||SHIELD}$)

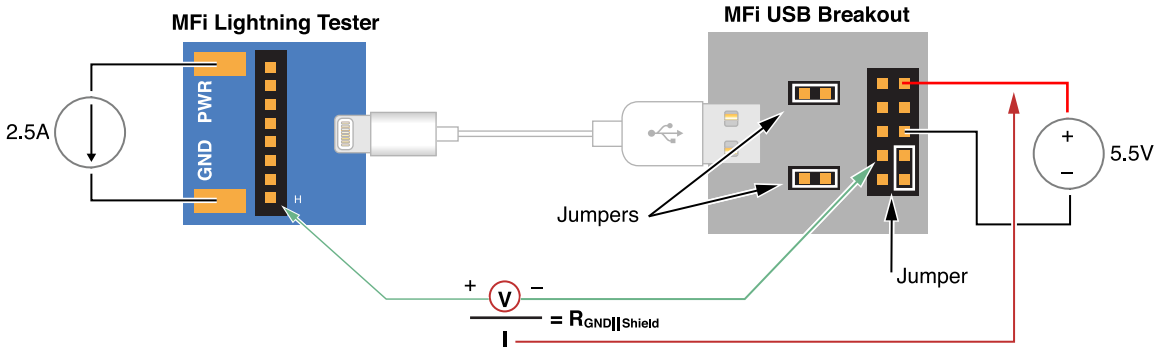


Figure 5-49 DCR VBUS Conductor Test Setup (R_{VBUS})

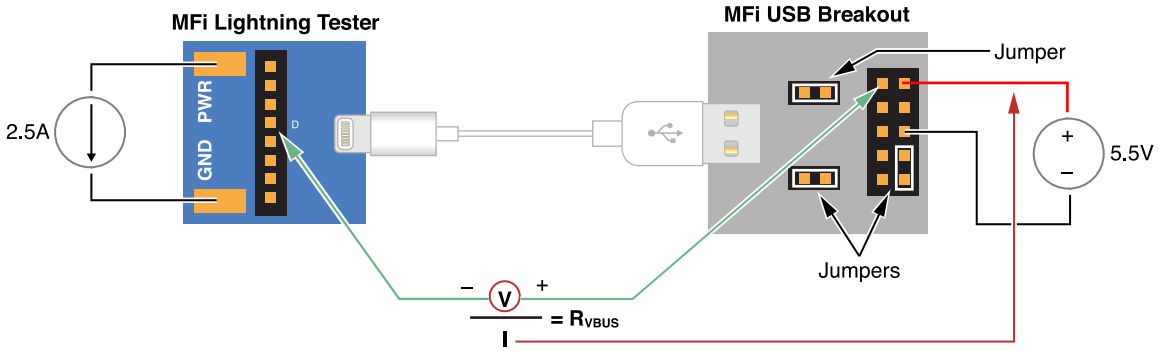
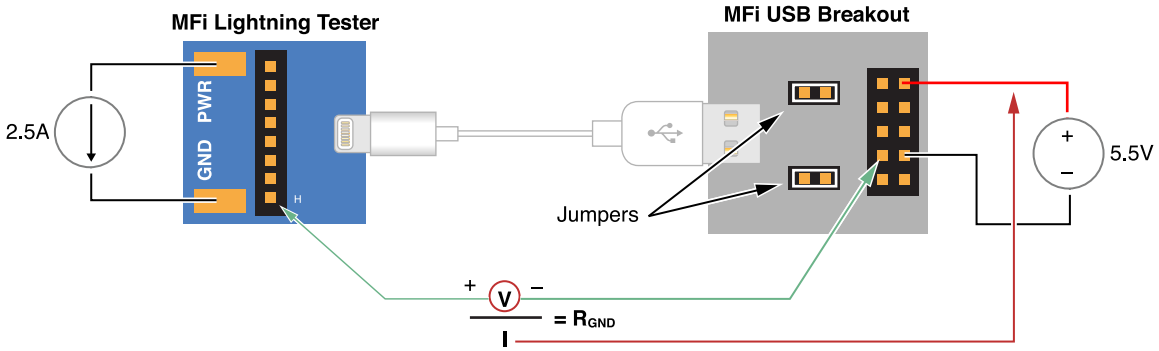


Figure 5-50 DCR Ground Conductor Test Setup (R_{GND})



1. For cable integrations that terminate in a USB-A plug:
   a. Connect the USB side of the cable to the MFi USB Breakout.
   b. Connect the MFi USB Breakout to the power supply (pin 9 to power and pin 3 to ground).
2. For cable integrations that do not terminate in a USB-A plug:
   a. Connect the cable accessory's power and ground terminating ends of the PCB to the power supply.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Connect the Lightning side to the MFi Lightning Tester.
4. Attach a 2.5A load across PWR and GND on MFi Lightning Tester.
5. Apply 5.5V on the power supply.
6. For ground conductor in parallel with shield resistance ($R_{GND||SHIELD}$):
    a. Measure the voltage difference between pin H of the MFi Lightning Tester and pin 4 of the MFi USB Breakout.
    b. Measure the current flowing through pin 9 of the MFi USB Breakout and the power supply.
    c. Calculate $R_{GND||SHIELD}$. $R_{GND||SHIELD}$ should not exceed what is specified in Table 45-2 (page 665). See Figure 5-48 (page 176) for more information.
7. For VBUS conductor resistance (R_{VBUS}):
    a. Measure the voltage difference between pin D of the MFi Lightning Tester and pin 10 of the MFi Lightning Breakout.
    b. Measure the current flowing through pin 9 of the MFi USB Breakout and the power supply.
    c. Calculate R_{VBUS}. R_{VBUS} should not exceed what is specified in Table 45-2 (page 665). See Figure 5-49 (page 176) for more information.
8. For ground conductor resistance (R_{GND}):
    a. Measure the voltage difference between H of the MFi Lightning Tester and pin 4 of the MFiUSB Breakout.
    b. Measure the current flowing through pin 9 of the MFi USB Breakout and the power supply.
    c. Calculate R_{GND}. R_{GND} should not exceed what is specified in Table 45-2 (page 665). See Figure 5-50 (page 176) for more information.

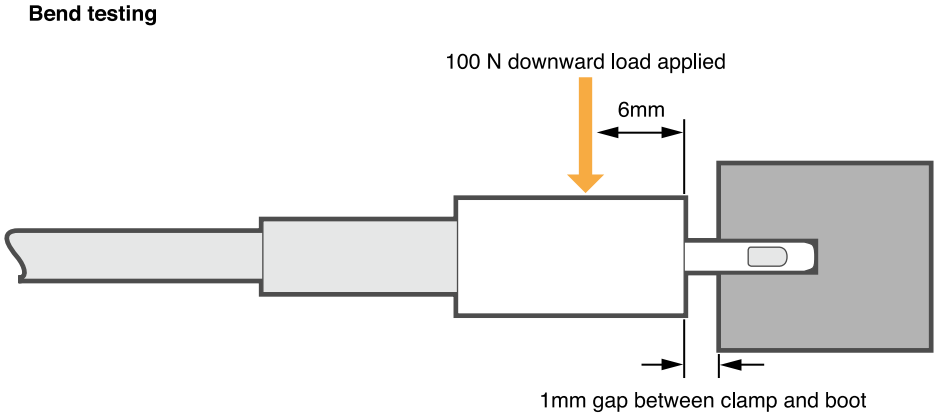
5.6.4.3 Cable Shield Bend Test (C48, C68)

All cables that incorporate a Lightning (C48) or Lightning (C68) connector must past this test.

Verify that the cable shielding is still intact after performing a bend test as shown in Figure 5-51 (page 178).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-51 Bend Test Setup for Lightning Cable Shield



5.6.4.4 180° Cable Bend Test

Verify that the cable shielding is still intact after performing a 180° cable bend test:

1. F_{Pull} = 0.453 kgf, d = 225 mm. See Figure 5-52 (page 178)
2. Cable tester should rotate at 42 RPM. Apple suggests a cable tester from http://www.yhtec.com.cn, part number: YH-8801K.
3. Fix connector so strain relief is centered on a rotating surface. Strain relief and cable should be free to bend. See Figure 5-53 (page 179).
4. Rotate the tester counter clockwise by 180°. See Figure 5-54 (page 179).
5. Rotate the tester clockwise by 360°. See Figure 5-55 (page 179).
6. Return the tester to its original position. See Figure 5-53 (page 179).
7. Cable should go through at least 50 cycles and still remain functional.

Figure 5-52 Bend Test Setup for Lightning Cables

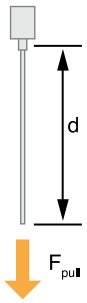


Figure 5-53 Tester Setup for Lightning Cables

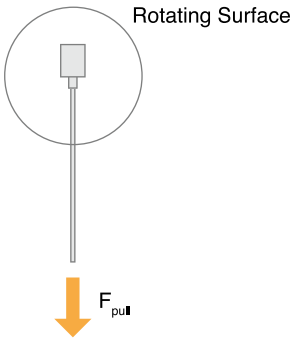


Figure 5-54 180° Cable Bend Test Setup for Lightning Cables

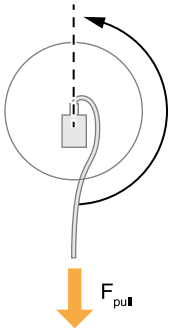
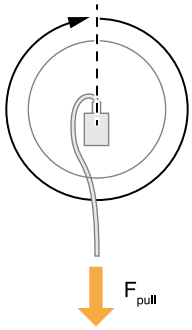


Figure 5-55 360° Cable Bend Test Setup for Lightning Cables



5.6.5 Dock Accessories

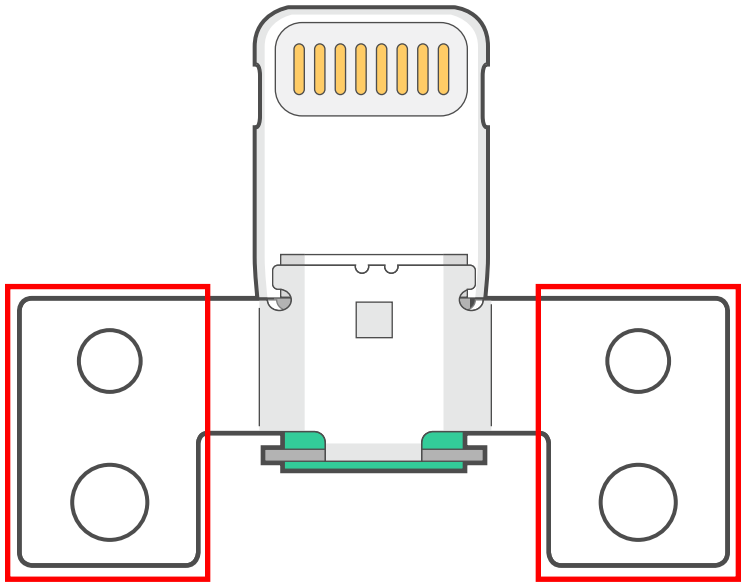
5.6.5.1 Dock Shield Bend Test (C48, C68)

All docks that incorporate a Lightning (C48) or Lightning (C68) connector must past this test.

Clamp on the shields as shown in Figure 5-56 (page 180).

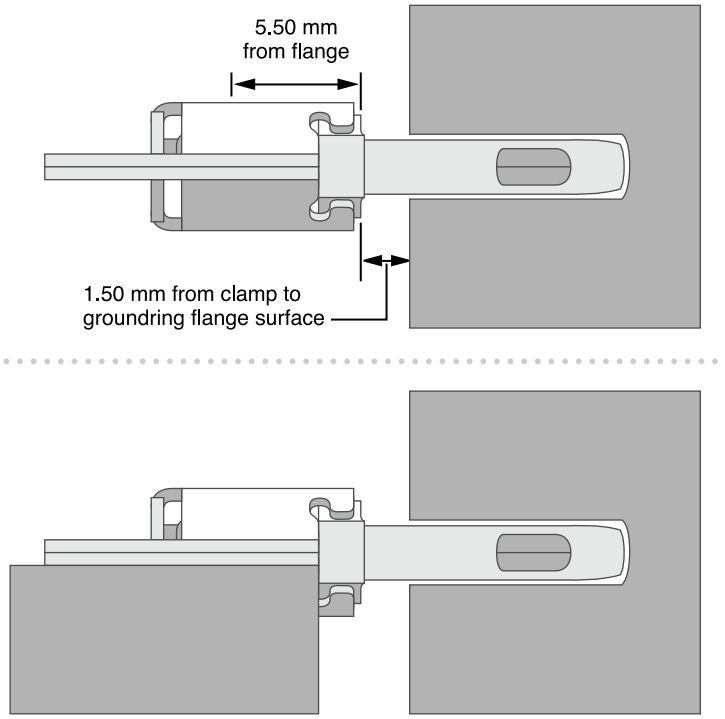
Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-56 Bend Test Clamp Region for Lightning Dock Shield



Ensure a 1.5 mm gap between plug clamp and groundring flange as shown in Figure 5-57 (page 180) and apply a force of at least 286 N at a distance of 5.50 mm from flange.

Figure 5-57 Bend Test Setup for Lightning Dock Shield



Verify that laser welds are intact.

5.6.5.2 iPhone and iPod touch Docks

5.6.5.2.1 Backstop

1. Connect the iPhone or iPod touch to the dock.
2. Using a calibrated force gauge, apply 5 N of force perpendicular to the screen at a point 100 mm vertically from the bottom of the device as shown in Figure 5-33 (page 159).
3. Verify that a backstop contacts the device at a point at least 15 mm vertically from the bottom of the device when the force is applied, as shown in figure Figure 5-32 (page 159).

5.6.5.2.2 Flexible Mechanism

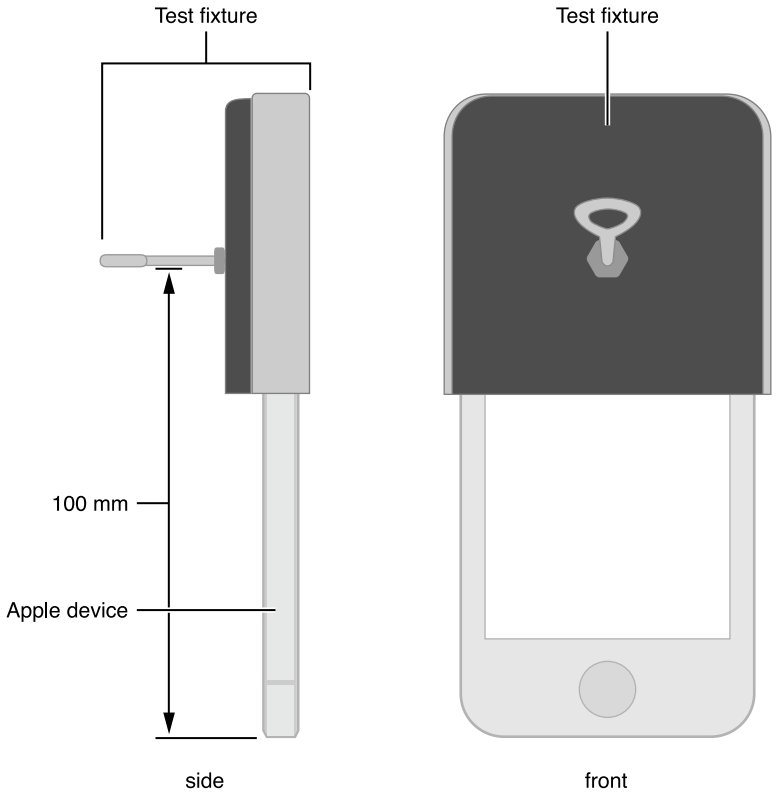
The flexible mechanism requirement is shown in Figure 5-33 (page 159). Drawings 1 and 2 show the device being engaged with the dock connector plug. Drawing 3 shows the device responding to a force of 10 N or less, applied 100 mm above the bottom of the device, by rotating up to 15 degrees from vertical.

1. Connect the iPhone or iPod touch to the dock and allow both to settle into a rest position.
2. Using a calibrated force gauge, apply force perpendicular to the device at a point 100 mm from the bottom of the device. Verify that a flexible mechanism prevents the user from applying more than 10 N of force to the device.
3. Using a protractor, verify that the device can be rotated 15 degrees from vertical.

Some docks have a backstop or other surface that limits access to the back of the device. A mechanism fitting over the front of the device similar to the one shown in Figure 5-58 (page 182) should be constructed to test them.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-58 Fitting mechanism to enable testing of certain docks that limit device access



5.6.5.3 iPad Docks

5.6.5.3.1 Backstop

1. Connect the iPad to the dock.
2. Using a calibrated force gauge, apply 2.5 N of force perpendicular to the screen at a point 200 mm vertically from the bottom of the device as shown in Figure 5-37 (page 162).
3. Verify that a backstop contacts the device at a point at least 50 mm vertically from the bottom of the device when the force is applied.

5.6.5.3.2 Flexible Mechanism

The flexible mechanism requirement is shown in Figure 5-37 (page 162). Drawings 1 and 2 show the device being engaged with the dock connector plug. Drawing 3 shows the device responding to a force of 5 N or less, applied 200 mm above the bottom of the device, by rotating up to 15 degrees from vertical.

1. Connect the iPad to the dock and allow both to settle into a rest position.

2. Using a calibrated force gauge, apply force perpendicular to the device at a point 200 mm from the bottom of the device. Verify that a flexible mechanism prevents the user from applying more than 5 N of force to the device.
3. Using a protractor, verify that the device can be rotated 15 degrees from vertical.

Some docks have a backstop or other surface that limits access to the back of the device. A mechanism fitting over the front of the device similar to the one shown in Figure 5-58 (page 182) should be constructed to test them.

5.6.5.4 iPod nano Docks

5.6.5.4.1 Backstop

1. Connect the iPod nano to the dock.
2. Using a calibrated force gauge, apply 20 N of force perpendicular to the screen at a point 50 mm vertically from the bottom of the device.
3. Verify that a backstop contacts the device at a point at least 15 mm vertically from the bottom of the device when the force is applied, as shown in figure Figure 5-32 (page 159).

5.6.5.4.2 Flexible Mechanism

The flexible mechanism requirement is shown in Figure 5-38 (page 163). Drawings 1 and 2 show the device being engaged with the dock connector plug. Drawing 3 shows the device responding to a force of 20 N or less, applied 50 mm above the bottom of the device, by rotating up to 15 degrees from vertical.

1. Connect the iPod nano to the dock and allow both to settle into a rest position.
2. Using a calibrated force gauge, apply force perpendicular to the device at a point 50 mm from the bottom of the device. Verify that a flexible mechanism prevents the user from applying more than 20 N of force to the device.
3. Using a protractor, verify that the device can be rotated 15 degrees from vertical.

Some docks have a backstop or other surface that limits access to the back of the device. A mechanism fitting over the front of the device similar to the one shown in Figure 5-58 (page 182) should be constructed to test them.

Jay Zhang jay.zhang@goertek.com GOERTEK ELECTRONICS CO.,LTD. WEIFANG

5.6.6 Dongle Accessories

5.6.6.1 Shield Bend Test (C48, C68)

All dongles that incorporate a Lightning (C48) or Lightning (C68) connector must past one of the shield bend tests:

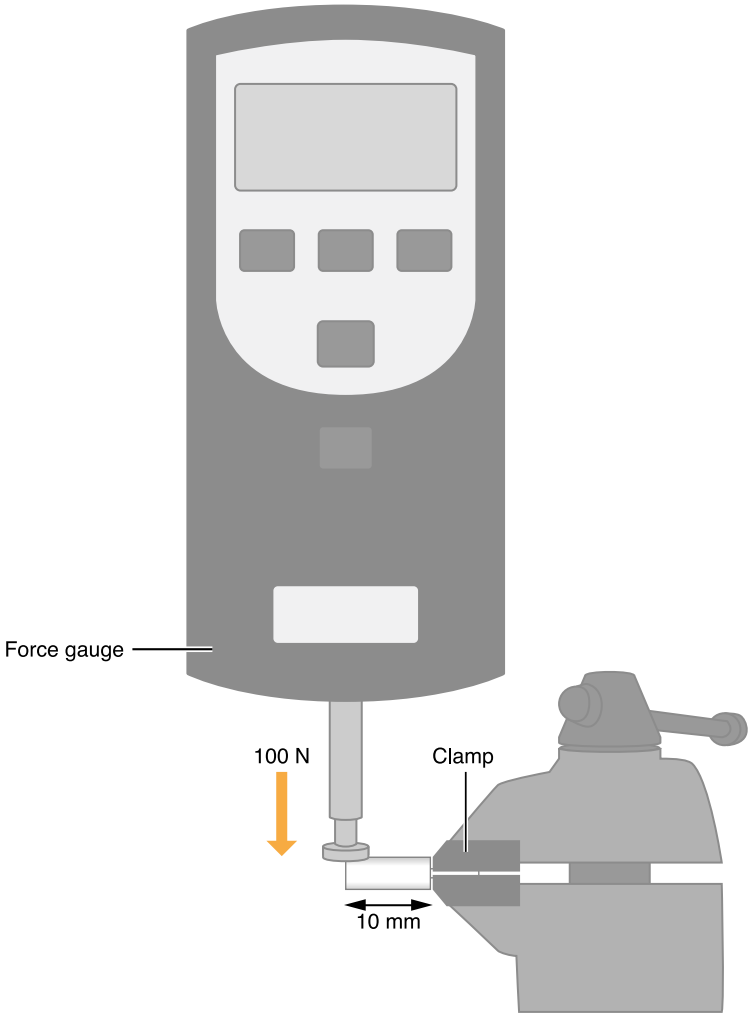
- Dock Shield Bend Test (C48, C68) (page 179)
- Cable Shield Bend Test (C48, C68) (page 177)

5.6.6.2 Break / Bend Test

1. Place the dongle's exposed plug surface into a clamp or other secure holding mechanism.
2. Position the measuring tip of a force gauge 10 mm away from the bottom of the dongle and apply 100 N of force as shown in Figure 5-59 (page 185).
3. Verify that the dongle breaks or bends before 100 N of force is applied.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 5-59 Dongle testing using a force gauge



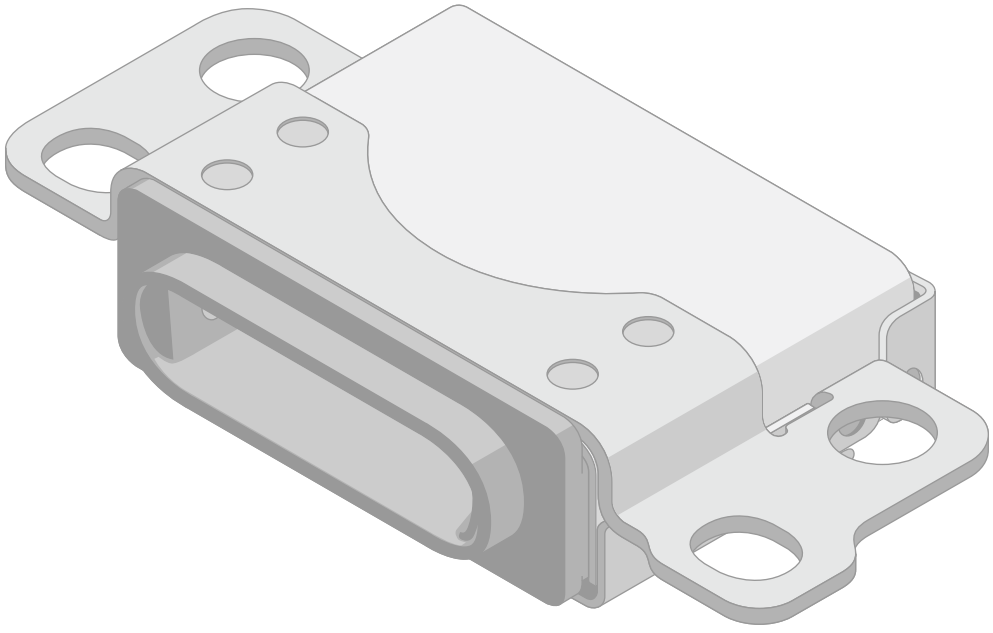
6. Apple Lightning Receptacle

Accessories may incorporate a Lightning Receptacle to

- Draw power from Apple branded or MFi certified USB power adapters.
- Draw power from USB Dedicated Charging Ports and USB hosts, such as a Mac.
- Enable USB connection to a Mac.
- Enable passthrough USB charge/sync of an Apple device while the accessory is attached to the Apple device's Lightning connector.

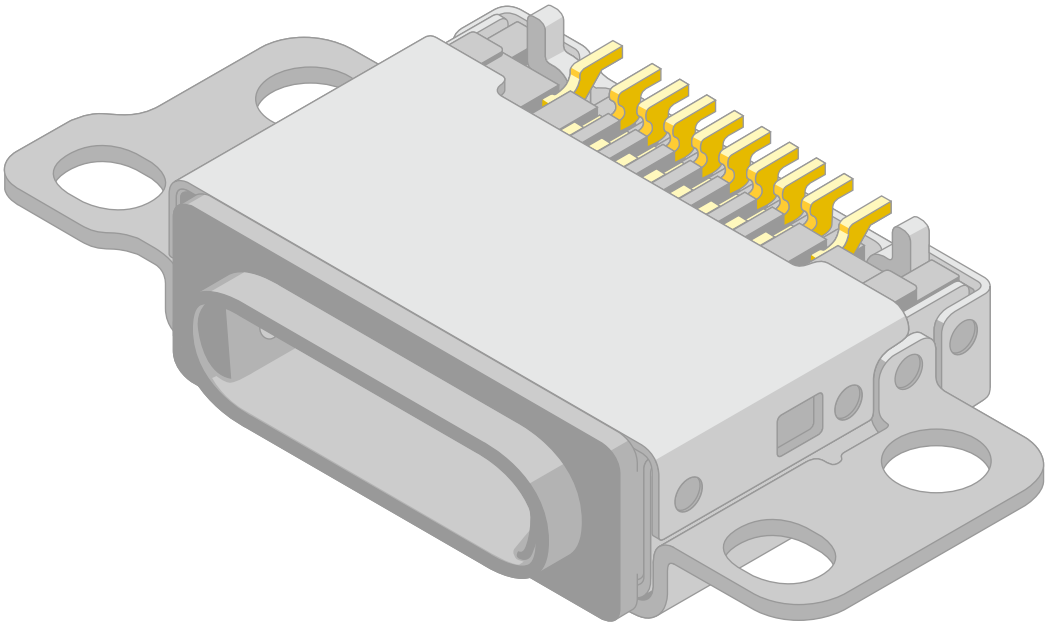
The Lightning Receptacle does not support any accessory interface features other than the ones listed above.

Figure 6-1 Lightning Receptacle Top View



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 6-2 Lightning Receptacle Bottom View



Examples of accessories that may benefit from including Lightning Receptacle include docks, battery pack cases, game controllers, Lightning headsets, and Bluetooth speakers.

Wireless accessories may use a Lightning Receptacle to charge their internal batteries at higher rates than those permitted by the USB specification using any USB to Lightning cable and Apple branded or MFi certified USB power adapters. This can be useful for game controllers (see HID Game Controller (page 591)), headsets (see Headsets (page 549)), and speakers.

6.1 Lightning Receptacle Requirements

Accessories that incorporate a Lightning Receptacle must also support/integrate one or more of the following features:

- iAP2
- Lightning Audio Module (see Apple Lightning Audio Module (page 93))
- AirPlay
- Wi-Fi Accessory Configuration

For example, accessories that only provide power to the Apple device do not qualify to use the Lightning Receptacle.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

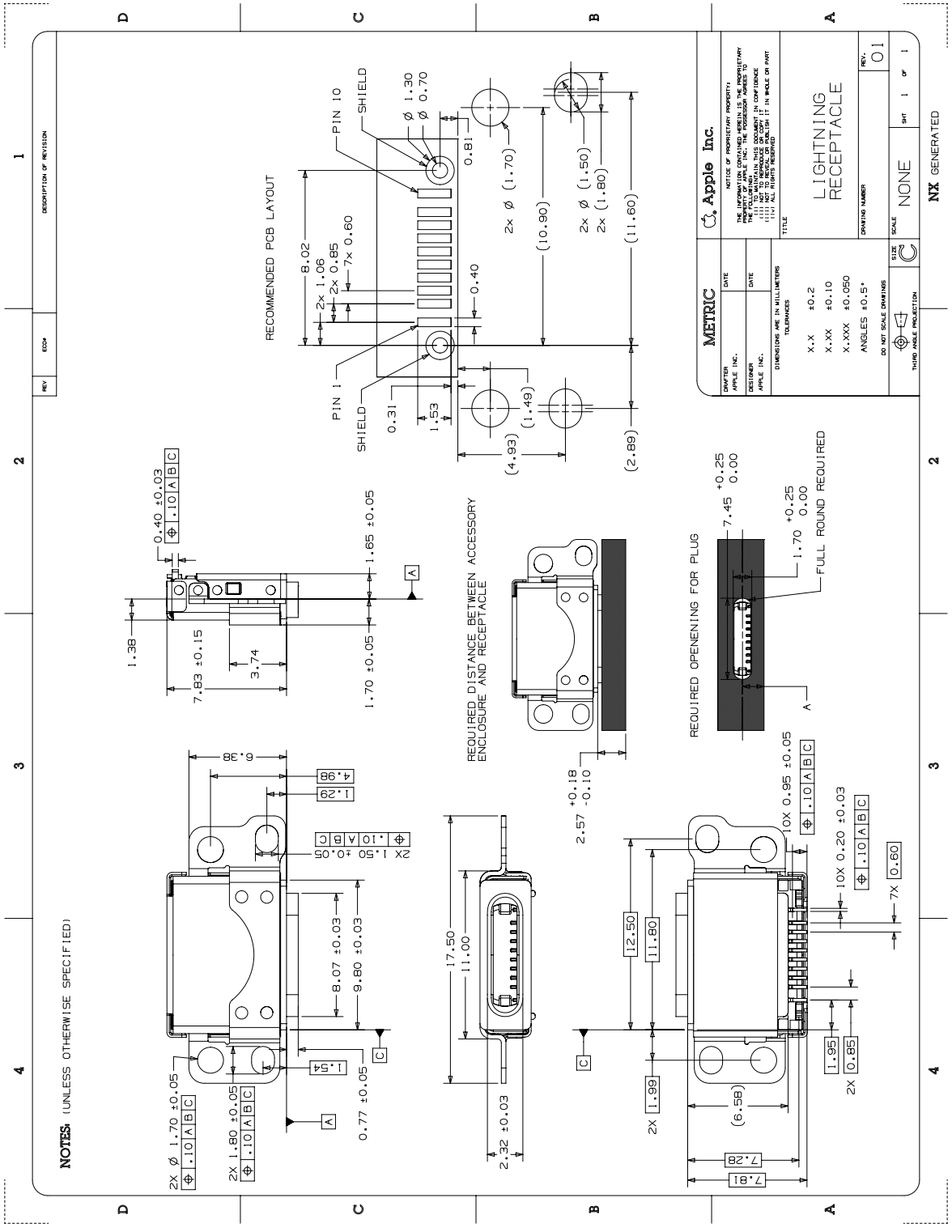
6.2 Lightning Receptacle Mechanical Requirements

Accessories must not incorporate more than one Lightning Receptacle.

The Lightning Receptacle is designed to withstand a standard reflow profile. Apple recommends a peak reflow temperature below 250 °C for 20 seconds maximum to prevent damage to the receptacle.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 6-3 Lightning Receptacle Dimensions



6.2.1 Mounting

A stainless steel trim ring is recommended around the opening of the receptacle for maximum durability and robustness.

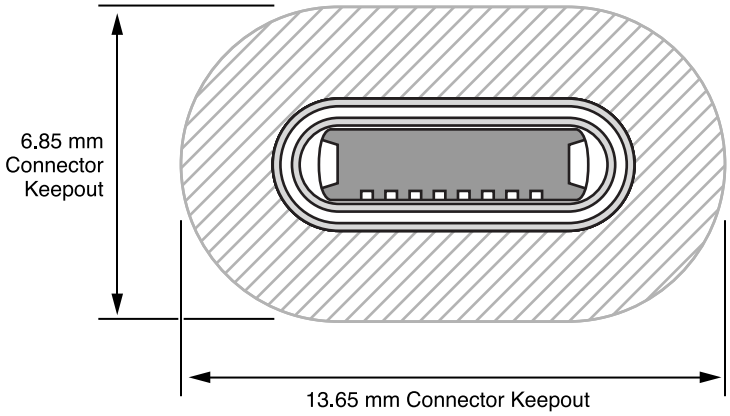
The receptacle must be recessed from the surface of the accessory as specified in Figure 6-3 (page 189). The depth of the receptacle to the surface of the opening must be at least 6.47 mm and at most 6.65 mm.

The opening for the Lightning connector must be fully rounded and meet the dimensions in Figure 6-3 (page 189).

The receptacle must be mounted in all 4 mounting holes with screws. Holes are designed to fit M1.2 screws.

6.2.2 Receptacle Keepout Area

Figure 6-4 Lightning Receptacle Keepout

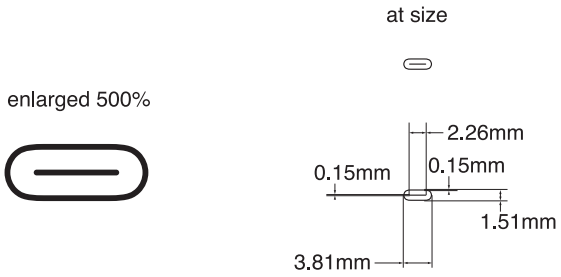


The accessory must keep a 13.65 mm by 6.85 mm area with full radii rounded edges clear around the receptacle for a distance of 14.0 mm as shown in Figure 6-4 (page 190).

6.2.3 Receptacle Label

The accessory may label the receptacle with the following icon (all dimensions in mm):

Figure 6-5 Lightning Receptacle Icon



The accessory must use the Lightning Receptacle icon art asset posted on the MFi Portal.

The accessory must not use any other icon or text to label the receptacle.

The icon should be aligned with the center of the receptacle and should be located either directly below or to the right of the receptacle.

The icon color should be black or white, whichever provides most contrast against the surface.

The icon must be legible after 1 million cycles of use.

6.2.4 Receptacle Encapsulation

Exposed contacts, solder joints and exposed PCB copper should be encapsulated after soldering with electrically nonconductive and liquid sealing compound, for moisture protection. Some suggested compounds that can be used for this purpose are:

- Loctite 3703 UV Encapsulant
- Henkel UV9060 UV Encapsulant
- Cemedine SuperX 8008 Neo Encapsulant (RTV)
- Dymax 29990 UV Encapsulant

A jet dispense process is recommended for applying encapsulation. Some recommended vendors of encapsulation equipment are:

- Nordson EFD/ASYMTEK (http://www.nordson.com)
- Musashi Engineering (http://www.musashi-engineering.co.jp)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

6.3 Lightning Receptacle Electrical Requirements

Accessories that integrate a Lightning Receptacle must include the Lightning Receptacle Controller. See Apple Lightning Receptacle Controller (page 208).

Accessories that integrate a Lightning Receptacle must use one or more of the following data transports to connect to the Apple device:

- Serial
- Lightning Audio Module Serial
- Wi-Fi
- Bluetooth

Accessories that integrate a Lightning Receptacle and also connect to an Apple device via a Lightning connector may support passthrough USB charge/sync for the Apple device. To support passthrough charge/sync, such accessories must:

- Integrate one of the following Lightning connectors:
  - Lightning (C10B), Lightning (C48B), Lightning (C11B), Lightning (C12B), or Lightning (C68B)
  - Lightning (C10E), Lightning (C11E), Lightning (C12E), or Lightning (C68E)
- Connect USB D+, USB D-, and Ground from the Lightning connector directly to the Lightning Receptacle Controller.
- Connect Device Power from the Lightning connector to the gated Receptacle Power, see Reference Circuit (page 213).
- Not use the USB transport for any communication with the Apple device while gated Receptacle Power is active, see Reference Circuit (page 213).
- Not draw power from the Lightning Receptacle unless they also integrate a Lightning Audio Module and use it to reserve a portion of pass-through power for the accessory's own use (see accPassthroughPowerSourceInformation (0x24) (page 122)).
- Pass the Lightning receptacle USB Integrity test procedures (see Passthrough USB Signal Integrity (page 198)).

Accessories that draw power from the Lightning Receptacle must:

- Correctly identify all Apple branded or MFi certified USB power adapters that connect resistor networks to USB D+ and USB D- (see Identifying Power Supplies Using Resistor Networks (page 193)).
- Correctly identify all USB Dedicated Charging Ports (DCP) as defined in the USB Battery Charging 1.2 specification.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Enumerate as a USB device when connected to a USB host, such as a Mac, and:
  - Not draw more than 100 mA of current until they have been successfully enumerated.
  - Request no more than 500 mA of charging current in their USB device descriptor.
- Not draw more power than the USB power source claims it is capable of providing via one of the above methods.

Accessories that draw power from a Lightning Receptacle to charge an internal battery must be capable of drawing at least 1.0 A from an Apple branded or MFi certified USB power source. 2.4 A is recommended.

6.3.1 Identifying Power Supplies Using Resistor Networks

Apple branded and MFi certified USB power adapters use resistor networks to identify their current capability.

Accessories that draw power from the Lightning Receptacle must be able to identify all of the resistor values in Table 6-1 (page 193) and must not attempt to draw more than the current specified. Accessories must take into account the variation of USB VBUS voltage and resistor tolerance (see Connectors That Do Not Implement iAP2 (page 665)).

Table 6-1 USB D+/D- resistor values for power-providing accessory connectors that do not implement iAP2

Current	R1	R2	R3	R4
500 mA	75.0 kΩ	49.9 kΩ	75.0 kΩ	49.9 kΩ
1000 mA	75.0 kΩ	49.9 kΩ	43.2 kΩ	49.9 kΩ
1000 mA	75.0 kΩ	49.9 kΩ	24.9 kΩ	49.9 kΩ
2100 mA	43.2 kΩ	49.9 kΩ	75.0 kΩ	49.9 kΩ
2400 mA	43.2 kΩ	49.9 kΩ	43.2 kΩ	49.9 kΩ
1000 mA	43.2 kΩ	49.9 kΩ	24.9 kΩ	49.9 kΩ
1000 mA	24.9 kΩ	49.9 kΩ	75.0 kΩ	49.9 kΩ
1000 mA	24.9 kΩ	49.9 kΩ	43.2 kΩ	49.9 kΩ
1000 mA	24.9 kΩ	49.9 kΩ	24.9 kΩ	49.9 kΩ

The following procedure is recommended for identifying the current limit based on the resistor values:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1. Read the VBUS voltage using an ADC. If value is less than 4.5 V, return no resistors detected.
2. For each D+ and D-, pull-down the line and read the voltage using an ADC. If either value is less than 1 V, return no resistors detected.
3. For each D+ and D-, disable the pull-down and allow the voltage to return to normal.
4. Read the D+ voltage using an ADC.
5. Read the D- voltage using an ADC.
6. For each value:
   - If value is > 2.995 V (based on 1 MΩ load impedance), then assume 24.9 kΩ .
   - If value is between 2.320 V and 2.995 V (based on 1 MΩ load impedance), then assume 43.2 kΩ .
   - If value is < 2.320 V (based on 1 MΩ load impedance), then assume 75.0 kΩ .
7. Use a table to look up max current, see Table 6-1 (page 193).

If resistor values could not be identified, the accessory should proceed to identify the power available based on the USB Battery Charging 1.2 specification.

6.3.2 Pins and Assignments

Figure 6-6 Lightning Receptacle Bottom View with Pin Assignments

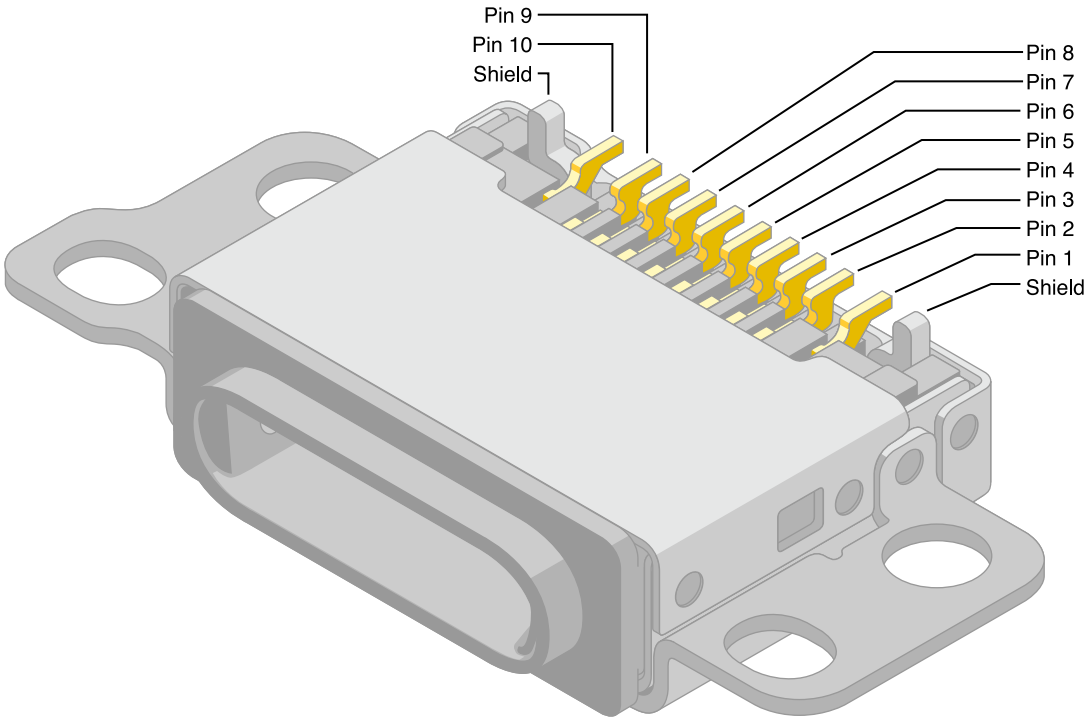


Table 6-2 Lightning Receptacle pins

Pin	Name	Assignment
1	RCP_1	see Reference Circuit (page 213)
2	RCP_2	Ground, see Reference Circuit (page 213)
3	RCP_3	see Reference Circuit (page 213)
4	RCP_4	see Reference Circuit (page 213)
5	RCP_5	see Reference Circuit (page 213)
6	RCP_6	Receptacle Power, see Reference Circuit (page 213)
7	RCP_7	see Reference Circuit (page 213)
8	RCP_8	see Reference Circuit (page 213)
9	RCP_9	see Reference Circuit (page 213)
10	RCP_10	Must be connected to RCP_1.

The receptacle shield must be grounded.

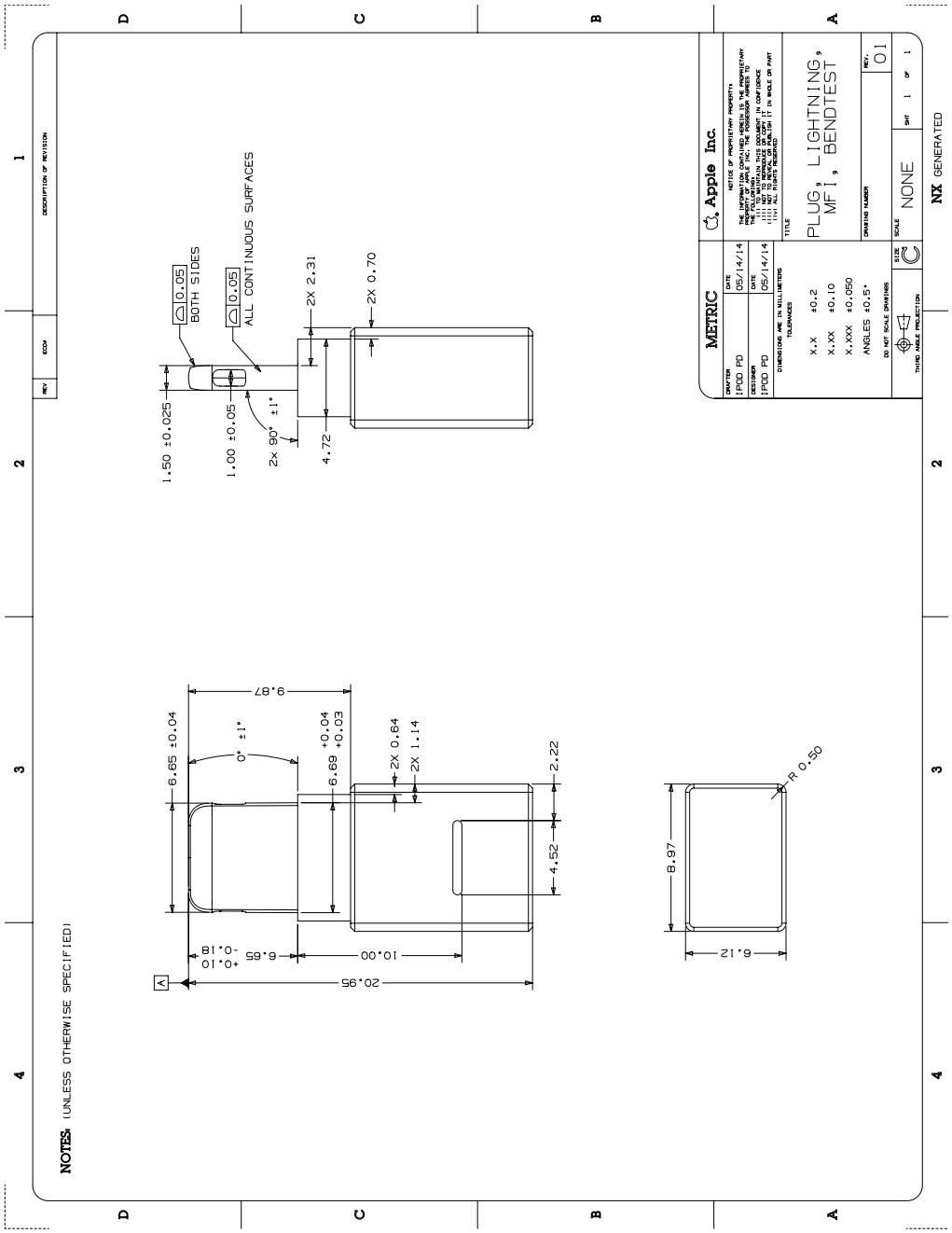
Signals from the following pads must be routed as differential pairs with 90 Ω differential impedance:

- RCP_3 and RCP_4
- RCP_7 and RCP_8

6.4 Test Procedures

These test procedures require the use of a test plug as specified in Figure 6-7 (page 196).

Figure 6-7 Lightning Receptacle Mechanical Test Plug



6.4.1 Static Receptacle Opening

1. Verify that an Apple branded Lightning cable inserts fully into the receptacle.
2. Using a caliper,
   a. Verify that the width of the receptacle opening is at least 7.45 mm and at most 7.70 mm.
   b. Verify that the height of the receptacle opening is at least 1.70 mm and at most 1.95 mm.

c. Verify that the depth of the receptacle to the surface of the opening is at least 6.47 mm and at most 6.65 mm.

d. Verify that the an area at the surface is clear for at least 13.65 mm by 6.85 mm with full radii rounded edges for a distance of 14.0 mm normal to the surface.

3. Verify that the receptacle opening is fully rounded.

4. If labeled, verify that the only label for the receptacle is the icon in Figure 6-5 (page 191).

6.4.2 Receptacle Mounting Under Insertion Force

1. Place the accessory in a clamp or other mechanism to securely hold it in place.

2. Using a caliper, measure the depth of the receptacle to the surface of the opening.

3. Fully insert the mechanical test plug into the receptacle.

4. Using a calibrated force gauge, apply 30 N of pressure to the mechanical test plug in the direction of plug insertion via the measuring tip of the force gauge.

5. Verify that the receptacle does not break away from the surface of the accessory.

6. Verify that the depth of the receptacle to the surface of the opening has not changed.

6.4.3 Receptacle Mounting Under Extraction Force

1. Place the accessory in a clamp or other mechanism to securely hold it in place.

2. Fully insert the mechanical test plug into the receptacle.

3. Using a calibrated force gauge, apply 30 N of pressure to the connector in the direction of plug extraction by clamping on to the plug and pulling on it.

4. Verify that the receptacle does not pull or break away from the accessory.

6.4.4 Bend Test

1. Place the accessory in a clamp or other mechanism to securely hold it in place.

2. Fully insert the mechanical test plug into the receptacle.

3. Using a calibrated force gauge, apply 100 N of pressure to the side of the mechanical test plug with the edge of the force gauge at a distance of 10 mm from the accessory surface.

4. Rotate the accessory ±90 degrees, two times each, and apply the same force from each position.

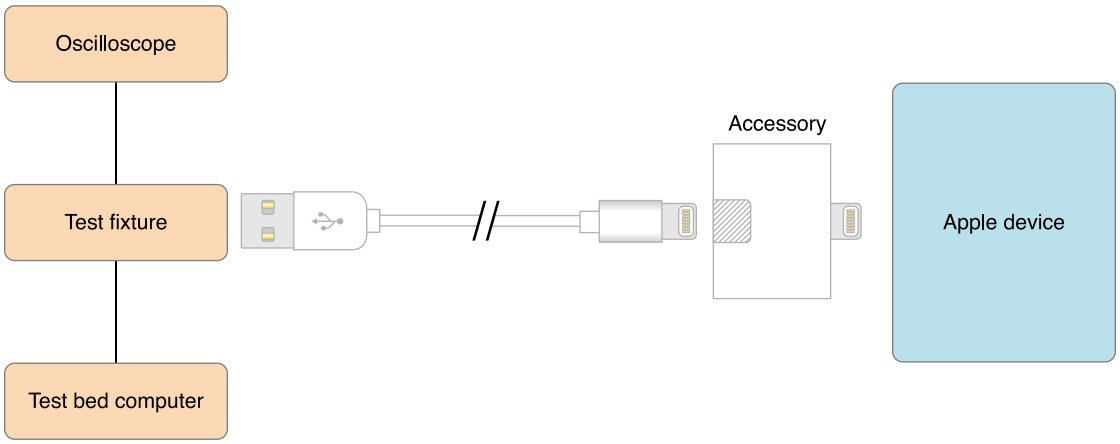
5. Perform a continuity test on all receptacle contacts and verify that continuity is maintained and there are no open circuits.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

6.4.5 Passthrough USB Signal Integrity

Accessories that integrate a Lightning Receptacle and support passthrough USB charge/sync to an Apple device via an integrated Lightning connector must pass the High-Speed and Full-Speed USB signal integrity test procedures.

Figure 6-8 Passthrough USB Signal Integrity Test Environment



6.4.5.1 High-Speed USB Signal Integrity

Validating High-Speed USB signal integrity of accessories that integrate a Lightning Receptacle and support passthrough USB charge/sync to an Apple device via an integrated Lightning connector requires:

- Apple branded USB to Lightning 2 meter cable
- iPhone 6s
- All equipment specified by the USB-IF to execute a Device High-Speed Electrical Test Procedure (DHSET), including but not limited to:
  - Approved oscilloscope
  - Device High-Speed Signal Quality Test Fixture
  - High-Speed Electrical Test Bed computer

Using the equipment specified above:

1. Connect the Test Port of the Device High-Speed Signal Quality Test Fixture to the accessory's Lightning Receptacle using the Apple USB to Lightning cable
2. Attach the iPhone 6s to the accessory's Lightning plug
3. Measure the Device High-Speed Signal Quality (eye diagram) of the iPhone 6s
4. Verify that the eye diagram of the iPhone 6s meets the requirements

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The eye diagram must meet the requirements in Figure 6-9 (page 199) and Table 6-3 (page 199).

Figure 6-9 High-Speed Eye Diagram Template at the SMA test fixture

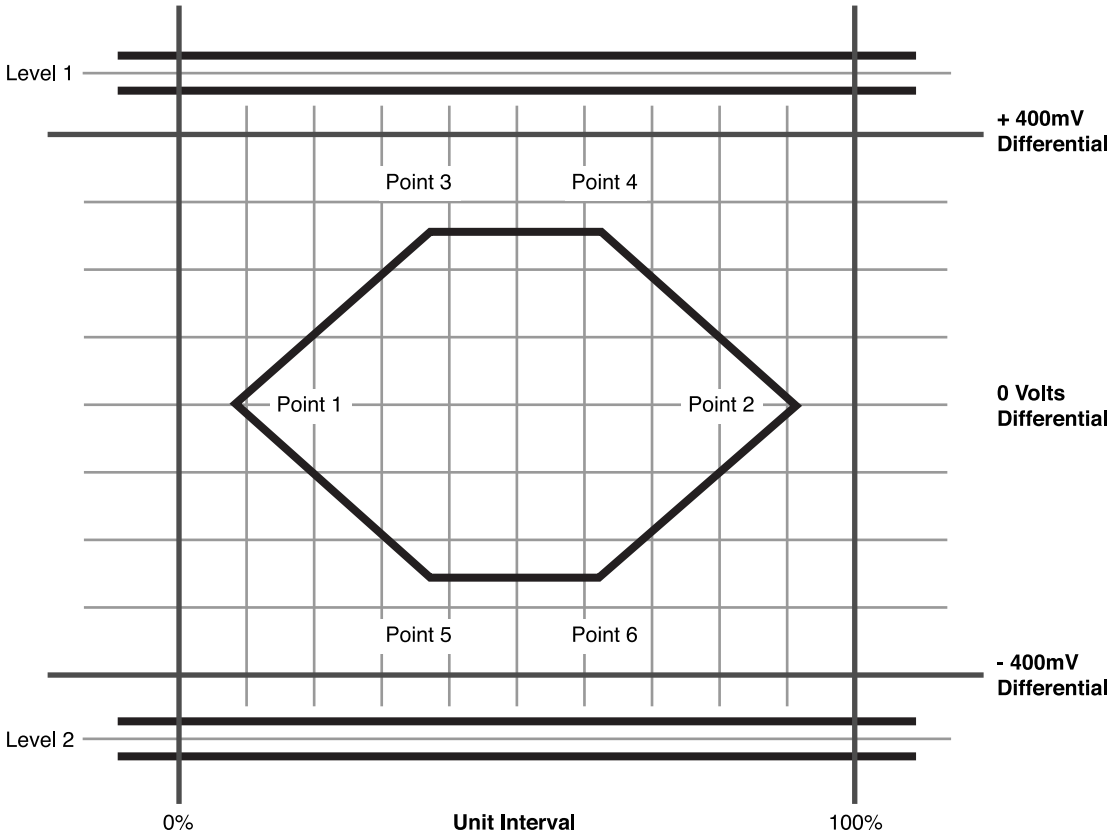


Table 6-3 High-Speed Eye Diagram Template Specifications

	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition -475 in all others	N/A
Point 1	0 V	7.5% UI
Point 2	0 V	92.5% UI
Point 3	260 mV	37.5% UI
Point 4	260 mV	62.5% UI
Point 5	-260 mV	37.5% UI
Point 6	-260 mV	62.5% UI

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

6.4.5.2 Full-Speed USB Signal Integrity

Validating Full-Speed USB signal integrity of accessories that integrate a Lightning Receptacle and support passthrough USB charge/sync to an Apple device via an integrated Lightning connector requires:

- Apple branded USB to Lightning 2 meter cable
- iPhone 6s
- All equipment specified by the USB-IF to execute a Device Full-Speed Electrical Test Procedure, including but not limited to:
  - Approved oscilloscope
  - Device Full-Speed Signal Quality Test Fixture
  - Full-Speed Electrical Test Bed computer

Using the equipment specified above:

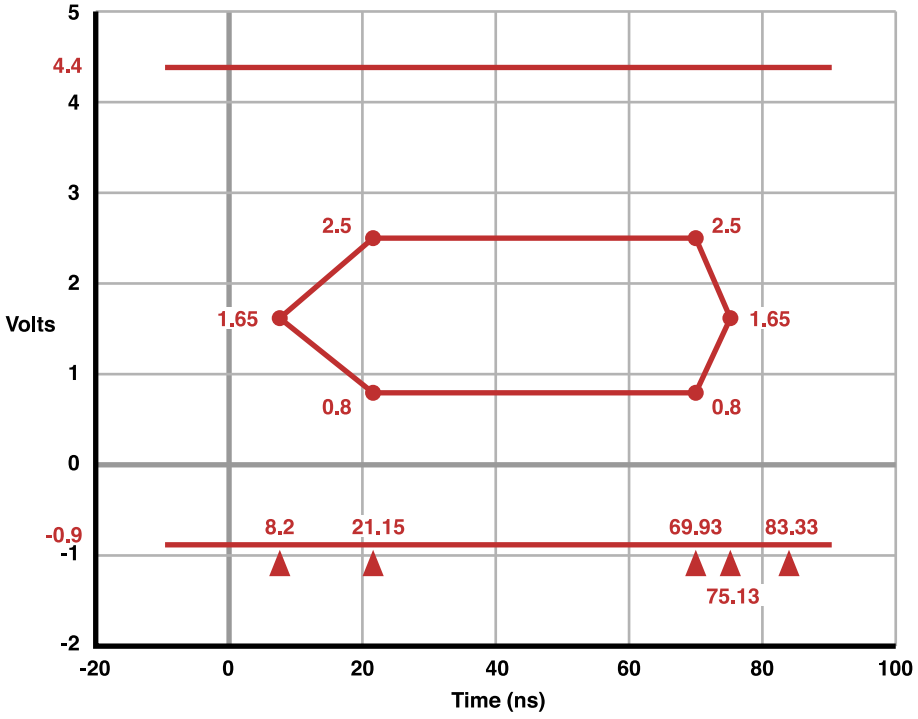
1. Connect the Test Port of the Device Full-Speed Signal Quality Test Fixture to the accessory's Lightning Receptacle using the Apple USB to Lightning cable
2. Attach the iPhone 6s to the accessory's Lightning plug
3. Measure the Device Full-Speed Signal Quality (eye diagram) of the iPhone 6s
4. Verify that the eye diagram of the iPhone 6s meets the requirements

For more information on the Full-Speed Electrical Test Procedure, refer to section B.6.3 Test Steps of USB-IF Full and Low Speed Electrical and Interoperability Compliance Test Procedure.

The eye diagram must meet the requirements in Figure 6-10 (page 201).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 6-10 Full-Speed Eye Diagram Template at the SMA test fixture



6.4.6 T37 USB Power Source Identification Tester for C37

The Lightning Receptacle Tester (MFI939-01036) is a USB power source that uses a resistor network to identify its current capability. See Identifying Power Supplies Using Resistor Networks (page 193).

The Lightning Receptacle Tester is designed to provide VBUS and D+/D- voltages with ±1.0% and ±2.2% of specification to take into account variations in tolerances and to provide additional margin for testing. See Table 6-4 (page 201) and Table 6-5 (page 201).

Table 6-4 Lightning Receptacle Tester VBUS Output Voltages

	Min (-1.0%)	Nominal	Max (+1.0%)
VBUS	4.85	4.9 to 5.25	5.30

Table 6-5 Lightning Receptacle Tester D+/- Output Voltages

	Min (-2.2%)	Nominal	Max (+2.2%)
D+/- 24.9 kΩ pull-up	3.16	3.33	3.61
D+/- 43.2 kΩ pull-up	2.54	2.68	2.90

Copyright © 2015 Apple Inc. All Rights Reserved.

	Min (-2.2%)	Nominal	Max (+2.2%)
D+/- 75.0 kΩ pull-up	1.89	2.00	2.16

The Lightning Receptacle Tester provides controls to vary the following parameters:

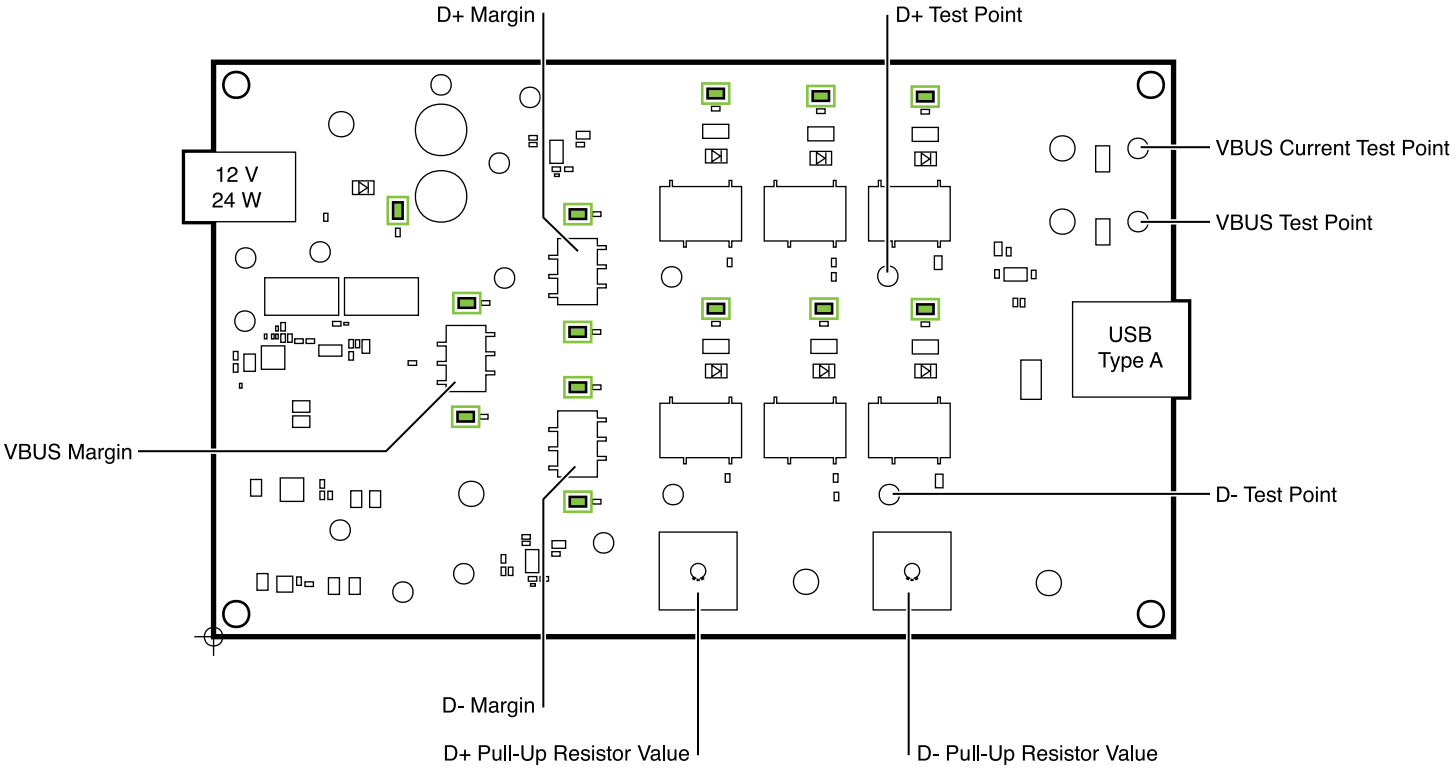
- D+ pull-up resistor value (24.9 kΩ , 43.2 kΩ , or 75.0 kΩ)
- D- pull-up resistor value (24.9 kΩ , 43.2 kΩ , or 75.0 kΩ)
- VBUS voltage margin (min/max)
- D+ pull-up resistor margin (min/max)
- D- pull-up resistor margin (min/max)

The Lightning Receptacle Tester provides:

- USB-A receptacle
- VBUS current test point (1V ≈ 1A)
- VBUS voltage test point
- D+ voltage test point
- D- voltage test point

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 6-11 T37 USB Power Source Identification Tester



To use the Lightning Receptacle Tester:

1. Connect a 12 V, 24 W power source to the barrel jack.
2. Set the desired D+/D- pull-up resistor values.
3. Set the desired VBUS voltage margin (min or max).
4. Set the desired D+/- resistor value margins (min or max).
5. Connect the accessory using the USB receptacle and an Apple branded Lightning cable.
6. Verify that the accessory is not drawing more than the maximum current as specified in Table 6-1 (page 193).

6.4.7 Power Supply Resistor Network Detection Test

The following equipment is required for validating resistor network detection:

- Lightning Receptacle Tester (see T37 USB Power Source Identification Tester for C37 (page 201))
- Digital multimeter

To perform the power supply resistor network detection test:

Copyright © 2015 Apple Inc. All Rights Reserved.

1. Connect a 12 V, 24 W power source to the barrel jack.
2. For every set of D+/D- pull-up resistor values in Table 6-1 (page 193):
   a. Set the D+/D- pull-up resistor values.
   b. For all 6 combinations of VBUS min/max, D+ min/max, and D- min/max:
      a. Set the VBUS voltage margin switch.
      b. Set the D+ and D- resistor value margin switches.
      c. Connect the accessory using the USB receptacle and an Apple branded Lightning cable.
      d. Verify that the accessory is not drawing more than the maximum current.
      e. Disconnect the accessory.

Table 6-6 (page 204) may be used to record the current for each switch combination.

Table 6-6 Test Results Matrix

Max Current	R1	R3	VBUS Margin	D+ Margin	D- Margin	Measured Current
500 mA	75.0 kΩ	75.0 kΩ	min	min	min	
			min	min	max	
			min	max	min	
			min	max	max	
			max	min	min	
			max	min	max	
			max	max	min	
			max	max	max	
1000 mA	75.0 kΩ	43.2 kΩ	min	min	min	
			min	min	max	
			min	max	min	
			min	max	max	
			max	min	min	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Max Current	R1	R3	VBUS Margin	D+ Margin	D- Margin	Measured Current
			max	min	max	
			max	max	min	
			max	max	max	
1000 mA	75.0 kΩ	24.9 kΩ	min	min	min	
			min	min	max	
			min	max	min	
			min	max	max	
			max	min	min	
			max	min	max	
			max	max	min	
			max	max	max	
2100 mA	43.2 kΩ	75.0 kΩ	min	min	min	
			min	min	max	
			min	max	min	
			min	max	max	
			max	min	min	
			max	min	max	
			max	max	min	
			max	max	max	
2400 mA	43.2 kΩ	43.2 kΩ	min	min	min	
			min	min	max	
			min	max	min	
			min	max	max	
			max	min	min	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Max Current	R1	R3	VBUS Margin	D+ Margin	D- Margin	Measured Current
			max	min	max	
			max	max	min	
			max	max	max	
1000 mA	43.2 kΩ	24.9 kΩ	min	min	min	
			min	min	max	
			min	max	min	
			min	max	max	
			max	min	min	
			max	min	max	
			max	max	min	
			max	max	max	
1000 mA	24.9 kΩ	75.0 kΩ	min	min	min	
			min	min	max	
			min	max	min	
			min	max	max	
			max	min	min	
			max	min	max	
			max	max	min	
			max	max	max	
1000 mA	24.9 kΩ	43.2 kΩ	min	min	min	
			min	min	max	
			min	max	min	
			min	max	max	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Max Current	R1	R3	VBUS Margin	D+ Margin	D- Margin	Measured Current
			max	min	min	
			max	min	max	
			max	max	min	
			max	max	max	
1000 mA	24.9 kΩ	24.9 kΩ	min	min	min	
			min	min	max	
			min	max	min	
			min	max	max	
			max	min	min	
			max	min	max	
			max	max	min	
			max	max	max	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

7. Apple Lightning Receptacle Controller

Accessories may incorporate a Lightning Receptacle Controller to establish electrical connections with Apple branded or MFi certified power supplies, USB Dedicated Charging Ports, and USB hosts, such as a Mac.

The Apple Lightning Receptacle Controller chip must be used in conjunction with the Lightning Receptacle, see Apple Lightning Receptacle (page 186).

7.1 Mechanical

The Lightning Receptacle Controller has the following mechanical properties:

- Package: WLCSP 36 balls, 2.07 mm x 2.07 mm x 0.435 mm height, 0.35 mm pitch
- Operating temperature: -20 °C to 85 °C

PCB design rules:

- For this 0.35 mm BGA, a UBM size of 0.2 mm with a max solder ball size of 0.21 mm is recommended.
- On the PCB side, a copper defined pad of size 0.2 mm is recommended.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

7.1.1 Physical Configuration

Figure 7-1 Lightning Receptacle Controller package

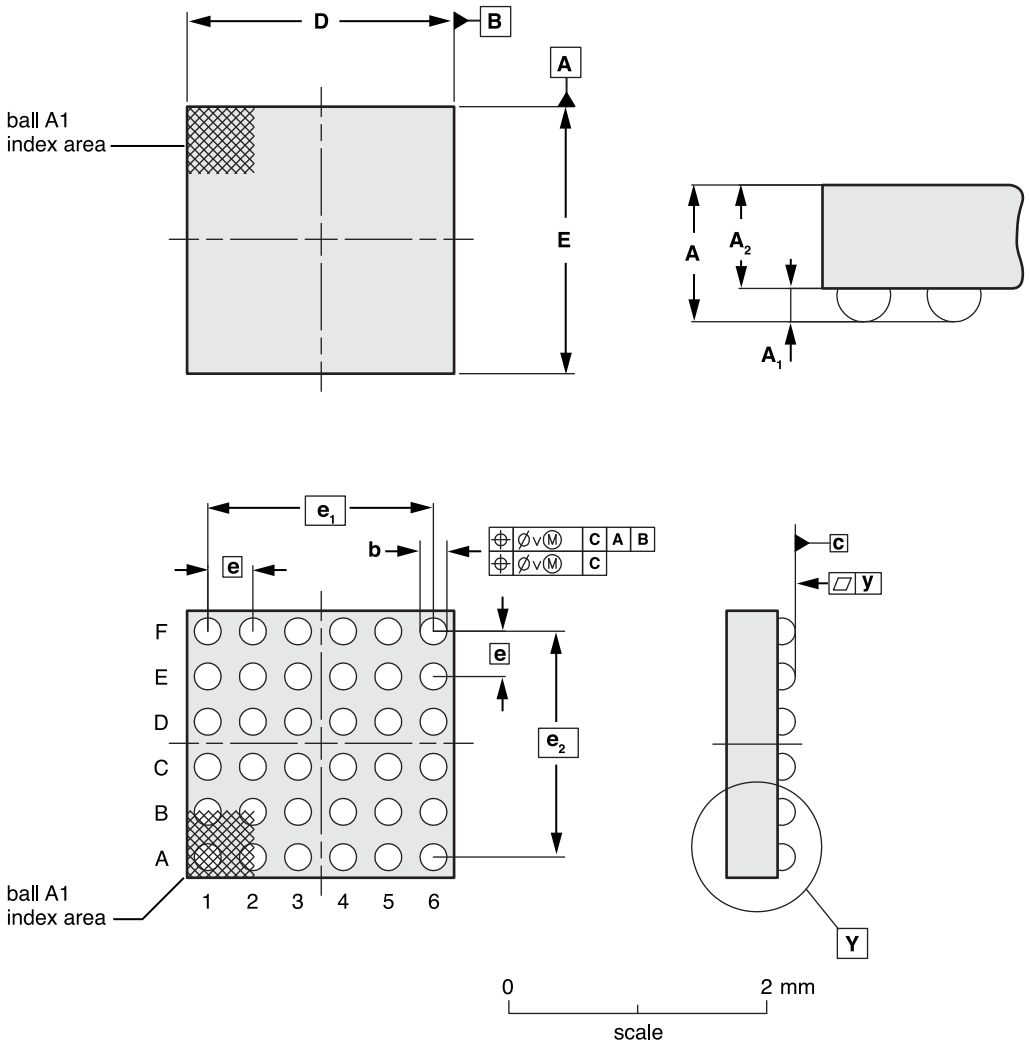


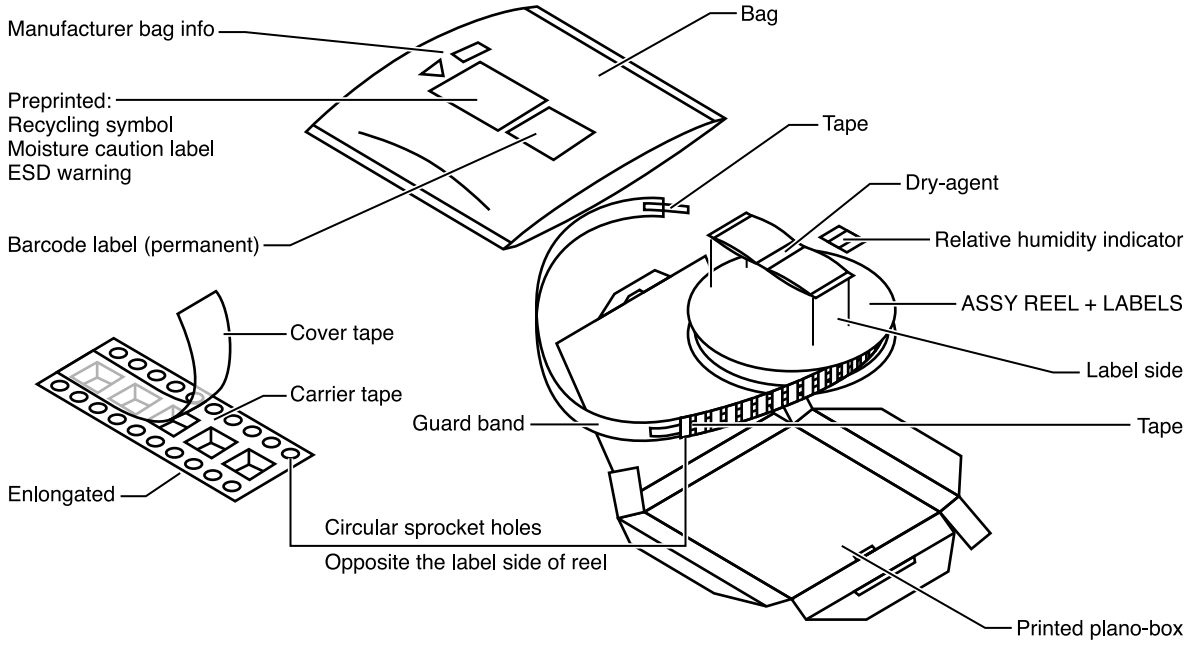
Table 7-1 Lightning Receptacle Controller package measurements

Dimension	Minimum	Nominal	Maximum
A	0.405 mm	0.435 mm	0.465 mm
A_1	0.115 mm	0.130 mm	0.145 mm
A_2	0.280 mm	0.305 mm	0.330 mm
b	0.178 mm	0.208 mm	0.238 mm
D	2.04 mm	2.07 mm	2.10 mm

 Copyright © 2015 Apple Inc. All Rights Reserved.

Dimension	Minimum	Nominal	Maximum
E	2.04 mm	2.07 mm	2.10 mm
e		0.35 mm	
e_1		1.75 mm	
e_2		1.75 mm	
v		0.05 mm	
w		0.015 mm	
y		0.03 mm	

Figure 7-2 Lightning Receptacle Controller packing tape and reel



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 7-3 Lightning Receptacle Controller packing tape

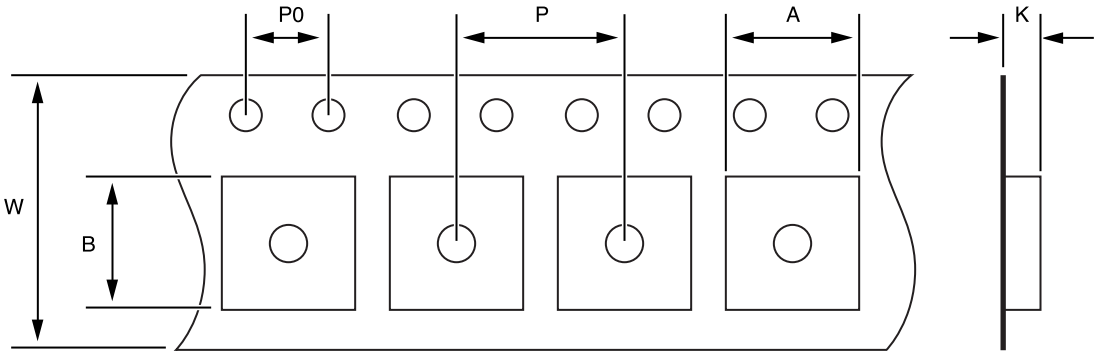


Table 7-2 Lightning Receptacle Controller packing tape dimensions

Label	Dimension
P0	4 mm
P	4 mm
A	2.24 mm
B	2.24 mm
K	0.75 mm
W	8 mm

7.1.2 Recommended Handling

Apple recommends the following when handling the Lightning Receptacle Controller:

- Use Pb-free solder.
- Use no-clean flux.
- Mitigate handling issues by assembling onto boards directly from tape and reel.
- Mitigate handling issues by using Teflon tipped tweezers (only if necessary) or bamboo tweezers. For example, Excelta part no. 159-RT.
- Use a vacuum wand if manual handling is required. For example:
  - Techni-tool vacuum system part no. 758SO100 (http://www.tehcni-tool.com)
  - Virtual Industry vacuum tip part no. VSPT4030-SD and VSPT2030-SD (http://www.virtual-ii.com

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

7.1.3 Recommended Operating Conditions

The Lightning Receptacle Controller is designed to work in ambient temperatures between -20 and 85 °C and stored in temperatures between -65 and 150 °C.

7.2 Electrical

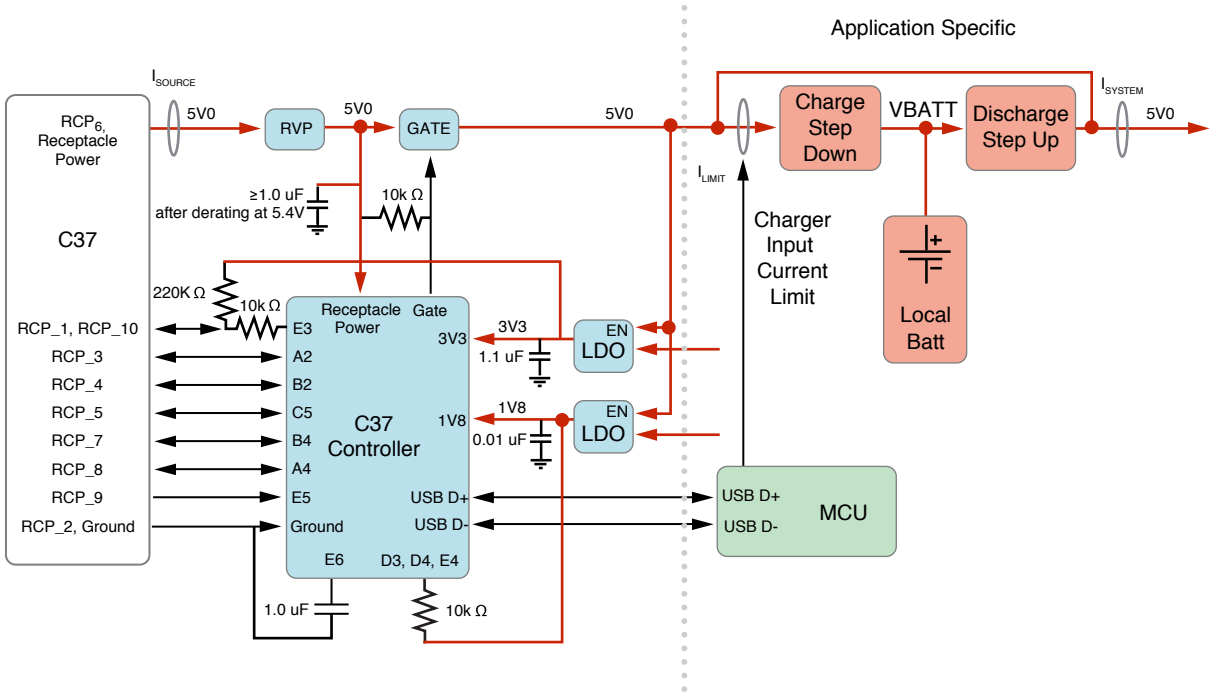
The Lightning Receptacle Controller has the following electrical properties:

- Power supplies:
  - V_{DD}(1V8)
  - V_{DD}(3V3) (or V_{DD}(3V0))
- ESD 1500 V HBM, 500 V CDM

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

7.2.1 Reference Circuit

Figure 7-4 Lightning Receptacle Controller reference circuit



1. MCU identifies max source current, I_{SOURCE}, via USB D+/D-
2. Design ensures $I_{LIMIT} \leq I_{SOURCE} - I_{SYSTEM}$

Note: This reference circuit does not support passthrough USB charge/sync. See Lightning Receptacle Electrical Requirements (page 192) for additional requirements specific to passthrough USB charge/sync.

The RVP (reverse voltage protection) and GATE (power gate) transistors can both be the same model since the voltage, current and resistance requirements are the same. Vds and Vgs should withstand at least ±6 V and should be rated for continuous operation at Vds or Vgs as high as ±5.4 V. The drain current should be rated to at least 2.4 A. An Rdson of no more than 10 mΩ is recommended for each transistor at Vgs = 4.25 V, Id = 2.4 A in order to avoid significant impact to the overall roundtrip resistance. This accounts for the lowest VBUS voltage, highest roundtrip cable resistance, and maximum current.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The capacitor connected to Receptacle Power at F6 must be at least 1.0 uF after derating at 5.4 V. A 0402 25 V X5R 1 uF capacitor is recommended.

The two LDOs must only be enabled when Receptacle Power is 5 V. The LDO power source may come from Receptacle Power or System Power.

Signals from D3, D4, and E4 must be combined and pulled-up to V_{DD}(1V8) using a 10 kΩ resistor.

All un-assigned or Reserved signals must be soldered but left Not Connected (NC).

Table 7-3 Lightning Receptacle Controller pad assignments

Pad	Assignment
A1	Reserved
A2	RCP_3, see Pins and Assignments (page 194)
A3	USB D+
A4	RCP_8, see Pins and Assignments (page 194)
A5	Reserved
A6	Ground
B1	Reserved
B2	RCP_4, see Pins and Assignments (page 194)
B3	USB D-
B4	RCP_7, see Pins and Assignments (page 194)
B5	Reserved
B6	Reserved
C1	Ground
C2	Reserved
C3	Reserved
C4	Reserved
C5	RCP_5, see Pins and Assignments (page 194)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Pad	Assignment
C6	Reserved
D1	Reserved
D2	Reserved
D3	10 kΩ pull-up to V_{DD}(1V8)
D4	10 kΩ pull-up to V_{DD}(1V8)
D5	Reserved
D6	Power Gate Enable (Low) with 10 kΩ pull-up to Receptacle Power
E1	Reserved
E2	Reserved
E3	RCP_1 and RCP_10 with 220 kΩ pull-up to V_{DD}(3V3) and 10 kΩ in series, see Pins and Assignments (page 194)
E4	10 kΩ pull-up to V_{DD}(1V8)
E5	RCP_9, see Pins and Assignments (page 194)
E6	1.0 uF to Ground
F1	Reserved
F2	Reserved
F3	V_{DD}(1V8)
F4	V_{DD}(3V3)
F5	Ground
F6	Receptacle Power with reverse voltage protection

7.2.2 Input/Output Electrical Characteristics

The following electrical characteristics apply to the Lightning Receptacle Controller.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 7-4 Input/Output Electrical Limiting Values

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD}(3V0)	supply voltage 3.0V		-0.5	+4.6	V
V_{DD}(1V8)	supply voltage 1.8V		-0.5	+4.6	V
V_I	input Voltage	RCP_1, RCP_3, RCP_4, RCP_5, RCP_7, RCP_8, RCP_9, RCP_10	-0.5	+7	V
		RCP_6	-0.5	+20	V

Table 7-5 Input/Output Electrical Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}(3V0)	supply voltage 3.0V		2.85	3.0	3.465	V
V_{DD}(1V8)	supply voltage 1.8V		1.7	-	3.465	V
V_I	input voltage	V_{DD}(3V0) CMOS inputs	-0.3	-	+3.6	V
		V_{DD}(1V8) CMOS inputs	-0.3	-	+3.6	V
		RCP_1, RCP_3, RCP_4, RCP_5, RCP_6, RCP_7, RCP_8, RCP_9, RCP_10	-0.3	-	+6.0	V

Table 7-6 Receptacle power and LDO AC/DC characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{I(RCP\_6)(max)}$	maximum input voltage on pin RCP_6	faulty conditions	-	-	+20	V
$V_{I(RCP\_6)}$	input voltage on pin RCP_6	normal operating condition	-0.3	-	+6	V
$I_{sink(RCP\_6)}$	sink current on pin RCP_6	when cable/source impedance is being checked by host system	15	-	27	mA

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

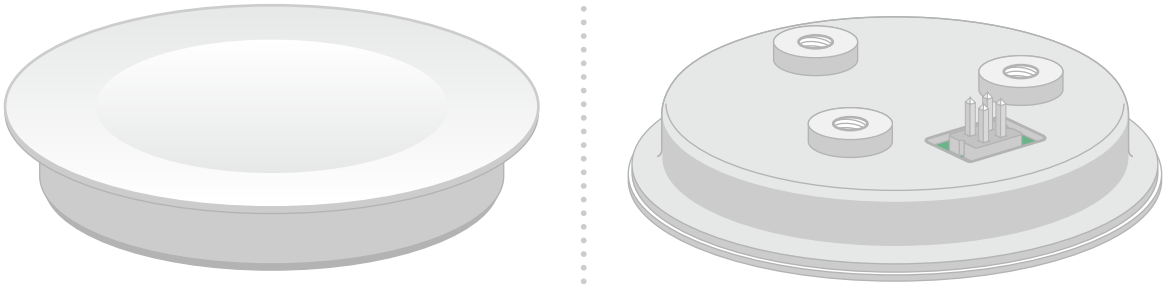
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OUT_LDOacc}	accuracy of voltage generated by LDO	LDO load current of 8mA	-150	-	+150	mV

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8. Apple Magnetic Charging Module

The Magnetic Charging Module enables dock accessories to recharge the Apple Watch when it is not being worn by the user.

Figure 8-1 Magnetic Charging Module



8.1 Magnetic Charging Module Requirements

All Magnetic Charging Module accessories must be docks that:

- Hold the Apple Watch when it is magnetically connected to the Magnetic Charging Module.
- Permit the Apple Watch to move relative to the Magnetic Charging Module.

Magnetic Charging Module must not be integrated into cables or wearable accessories such as wrist straps or watch bands.

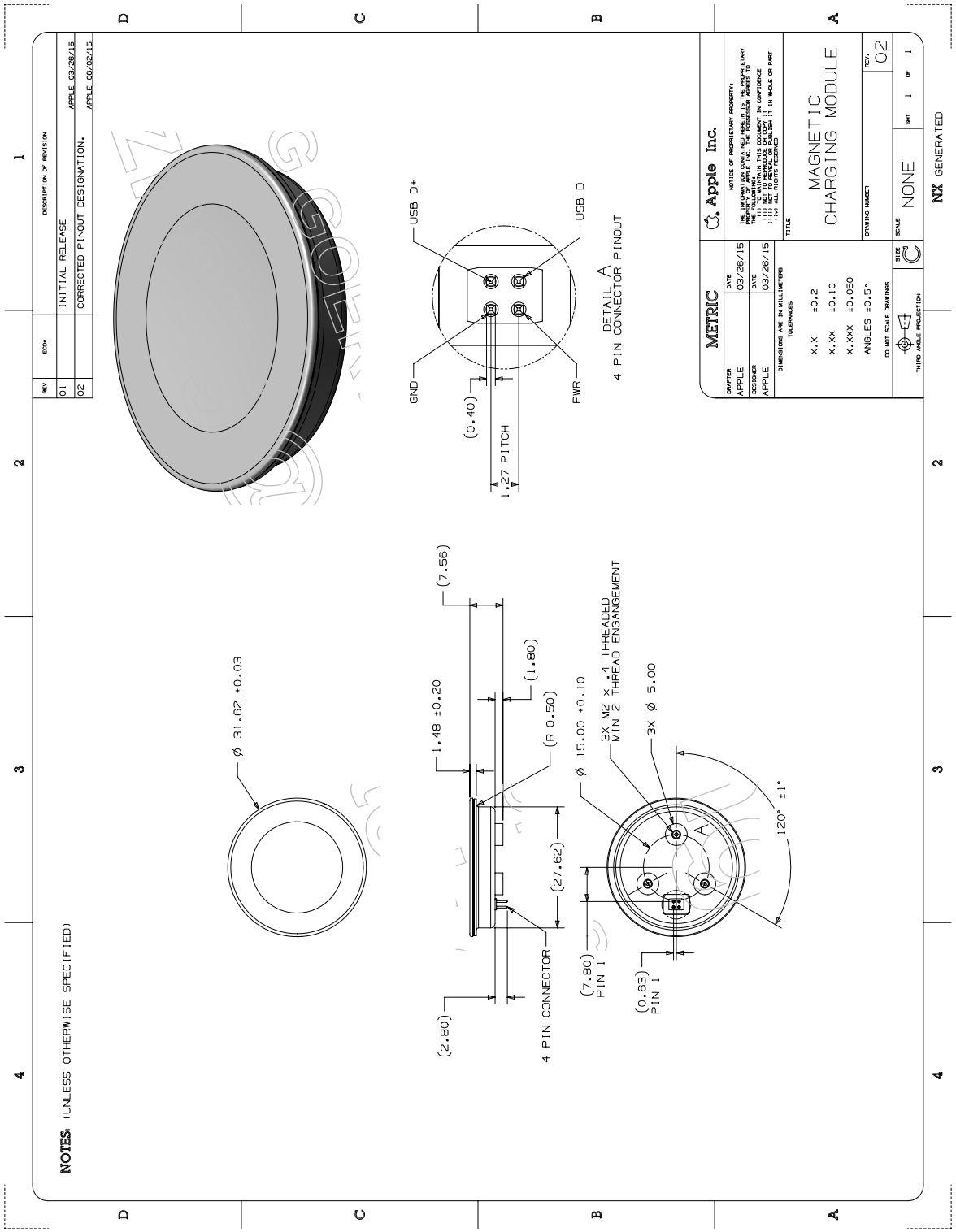
Apple Magnetic Charging Module docks may also integrate a Lightning connector so long as the integration complies with Lightning dock requirements (see Dock Accessories (page 158)).

Accessories that incorporate a Magnetic Charging Module must provide power to Apple Watch. See Magnetic Charging Module Electrical Requirements (page 221).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8.2 Magnetic Charging Module Mechanical Requirements

Figure 8-2 Magnetic Charging Module Dimensions



Accessories must not interfere with the body of an Apple Watch.

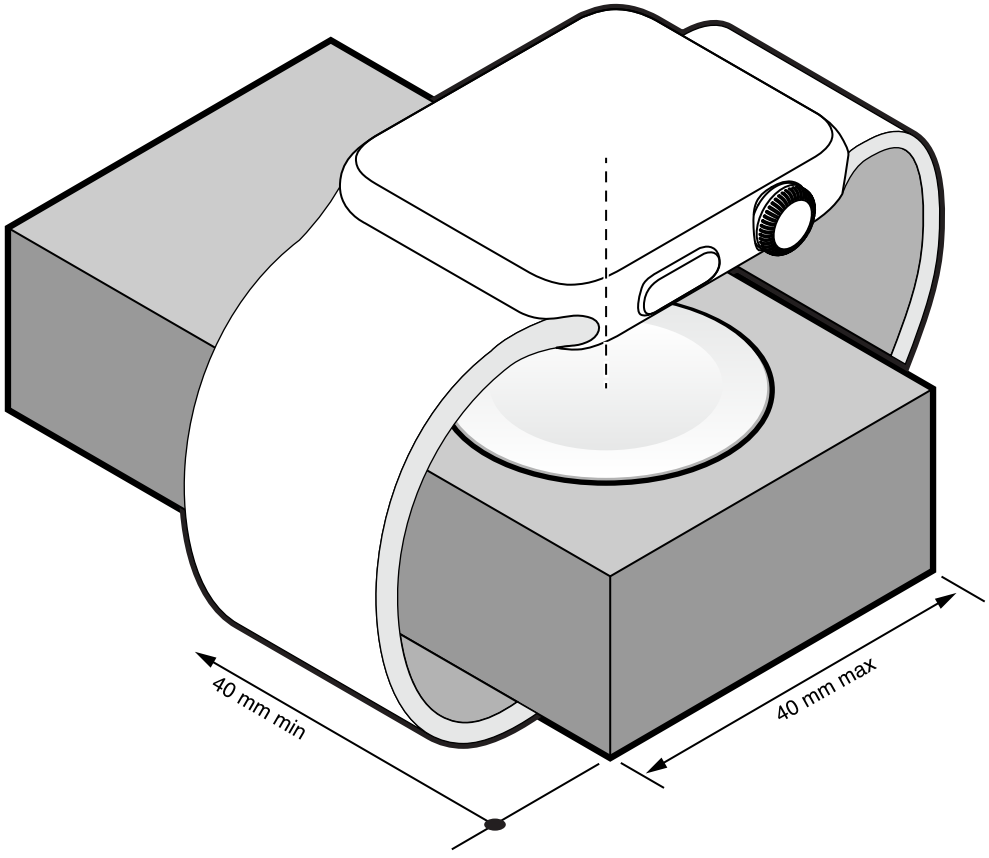
Accessories should be compatible with all Apple bands including the Milanese Loop and the Link Bracelet. Additionally, the accessory should not:

- Require the watch band to be removed or detached.
- Make contact with the watch band in a manner that may scratch or damage the band.

To avoid interference with Apple Watch bands, accessories should:

- Not exceed 20 mm in radius around the center of the Magnetic Charging Module surface if the Apple Watch can be attached in any orientation.
- Not exceed a length of 40 mm across the surface of the Magnetic Charging Module for a width of 40 mm along the intended orientation of the Apple Watch if Apple Watch is intended to be attached in a specific orientation. See Figure 8-3 (page 220).

Figure 8-3 Magnetic Charging Module Charging Arm Clearance



The surface of the Magnetic Charging Module must be 0.40 mm ±0.25 mm proud of the surface of the accessory.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The Magnetic Charging Module must be mounted such that any combination of Apple Watch and band must not disengage and fall off due to gravity.

The Magnetic Charging Module must not be mounted more than 45 degrees from horizontal. Horizontal mounting is suggested.

The Magnetic Charging Module must be mounted using at least one of the following:

- All 3 mounting holes with screws. Holes are designed to fit M2 x 0.4 threaded screws. A minimum 2 thread engagement is required.
- An adhesive to bond the lip of the cap to the surface of the accessory. Apple recommends pressure sensitive adhesive (PSA) in a ring with an inner diameter of 29.12 mm, a width of 1.05 mm to 1.25 mm, and a thickness of 0.15 mm.

The Magnetic Charging Module connector must not be mounted in proximity to magnetic steel.

All RF/metal keep out zones for the Apple Watch must be respected, see Apple Watch 38 mm (page 883) and Apple Watch 42 mm (page 884).

The Magnetic Charging Module's maximum storage temperature is 60 °C.

8.3 Magnetic Charging Module Electrical Requirements

All accessories that integrate a Magnetic Charging Module must comply with the electrical requirements in this section.

8.3.1 Power

Accessories that integrate a Magnetic Charging Module must provide power from either:

- An external USB power source.
- An internal power supply.

If the accessory integrates an internal power supply, it:

- May accept external power from an integrated USB-B receptacle (Standard, Mini, Micro) or a non-USB receptacle so long as the accessory actively conditions and regulates the external power to comply with Magnetic Charging Module power requirements.
- May consume some of the external power for its own purposes so long as all Magnetic Charging Module power requirements are met.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

If the accessory relies on an external USB power source to provide power to the Magnetic Charging Module and does not contain its own internal power supply, it:

- Must connect to the external USB power source via a captive cable that terminates in a USB-A plug.
- Must not consume any power from the external USB power source.
- Must not monitor or modify the USB D+/D- signals from the external USB power source.

8.3.1.1 Internal Power Supplies

All internal power supplies in Magnetic Charging Module accessories must:

- Regulate input voltage at the VBUS pin of the Magnetic Charging Module to 4.75 V - 5.4 V under any load from 0 A to 1 A.
- Hold VBUS ripple below 20 mVpp under any load from 0 A to 1 A.
- Implement over-current protection with a 1.5 A threshold.
- Connect USB D+/D- to a resistor network as shown in Figure 45-2 (page 665), using resistor values for a 1000 mA power source as defined in Table 45-3 (page 666).

8.3.1.2 External USB Power Sources

All accessories that rely on an external USB power source to provide power to the Magnetic Charging Module must:

- Have maximum 500 mΩ round trip DCR (USB VBUS to GROUND) between the Magnetic Charging Module connector and the accessory's USB-A plug.
- Pass the USB-IF full-speed signal quality test. See Full-Speed USB (page 703).
- Meet the USB-IF inrush current specification of 51.5 µC.
- Meet the USB-IF suspend current specification of 2.5 mA in the following test scenarios:
  - Suspend Current in USBIFCV on PC
  - Suspend Current in HSET on PC
  - Suspend Current With PC

For USB-IF test procedures, see the USB-IF *Full and Low Speed Electrical and Interoperability Compliance Test Procedure* and *Gold Suite Test Procedure*.

Overcurrent protection (OCP) is optional. If the accessory implements OCP, the round trip DCR measurement must include the on-resistance of the OCP switch.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8.3.1.3 Apple Watch Charging Efficiency

Accessories that integrate a Magnetic Charging Module must not impair Apple Watch's ability to efficiently charge from provided power. Strict compliance with all the Magnetic Charging Module integration requirements is key.

8.3.2 Pins and Assignments

The module pin assignments are shown in Figure 8-2 (page 219).

Table 8-1 Magnetic Charging Module Pins

Pin	Name	Assignment
1	PWR	USB VBUS
2	GND	Ground
3	USB D+	USB D+
4	USB D-	USB D-

The module housing must be grounded. Any metallic accessory housing must share the same ground as the module housing.

Signals from the USB D+ and D- pins must be routed as a differential pair.

One of the following approaches must be used to connect to the module pins:

- Use a 4-pin 0.050" micro pitch connector mounted to a PCB. The PCB must be attached to the module using the 3 mounting holes.
- Use a 4-pin 0.050" micro pitch connector attached to a cable.
- Solder pins directly to a PCB using through-hole mounting. The pins must not be reflow soldered, but must be manually soldered with heat contained to the pin region.
- Solder wires directly to the pins.

Micro pitch connectors are available from Samtec (http://www.samtec.com) and others.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8.3.3 EMC

Apple recommends shielding the magnetic field from the charging coil and maintaining a low impedance shield termination for the cable to comply with regulatory EMC requirements for the completed product. The implementation, final compliance testing, report preparation, and labeling are the responsibility of the company marketing the product.

Cable termination is critical for reduced emissions. Cable termination and connectors should be kept away from the Magnetic Charging Module and should be routed away from the face of the module.

If an adapter board is used between the Magnetic Charging Module and a cable, the cable should have 360 degree termination to the board. The design of the adapter board should limit exposed traces on external layers.

If emissions are present, adding clamp-on ferrites/absorbers to the cable can help reduce the emissions. The selected ferrite/absorber material should be rated for the failing frequencies.

The Apple Watch models with wristband combinations (listed at http://www.apple.com/watch/gallery/) should be tested with the Magnetic Charging Module for EMC/EMI.

Depending on the use cases, testing should be performed with one or all of the following power supplies:

- 15" MacBook Pro
- Apple USB Power Adapter (US) Model A1385
- Apple USB Power Adapter (Int.) Model A1400
- Apple USB Power Adapter (UK) Model A1552
- Apple USB Power Adapter (China) Model A1443
- Apple USB Power Adapter (Australia) Model A1444
- Apple USB Power Adapter (Brazil) Model A1486
- Apple USB Power Adapter (Korea) Model A1487
- Apple USB Power Adapter (Argentina) Model A1501

If different power sources are used than listed above, emission testing should be performed with those power sources ON.

In addition to the cases above, the charging device should be tested in idle mode for emissions.

Emissions tests should be conducted in accordance with standards referenced in the following:

- FCC CFR 47, Part 15
- ICES-003, Issue 5, CAN/CSA-CEI/IEC CISPR 22-10

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- CISPR 22:2008
- EN55022:2010
- AS/NZS CISPR 22:2009, TCVN 7189:2009
- VCCI V-3/2013.04
- GB 9254-2008, GB 17625.1-2012, GB 17625.2-2007, CNS 13438-2006
- CISPR 24: 2010
- EN 55024: 2010

Once the highest emissions combination is identified, complete testing should be performed on that configuration.

In some regulatory domains EMC certification is required before placing the accessory on the market.

8.4 Factory Configuration

The Magnetic Charging Module exposes a USB Virtual COM Port (VCP) on the 4-pin header to use for factory configuration. Connect the 4 pins to a cable terminating in a USB-A plug. See USB Virtual COM Port (page 230).

The Magnetic Charging Module must be configured at the time of accessory manufacturing to set the following parameters:

- Vendor ID (VID)
- Product ID (PID)
- Vendor Name
- Product Name
- Model Number
- Serial Number

Every Magnetic Charging Module:

- Must not be configured with blank or generic string values.
- Must be configured with a unique Serial Number (i.e. serialized).
- Must set the Vendor Name, Product Name, and Model Number to human-readable strings that match names appearing on the accessory or its packaging.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Must set the Vendor ID (VID) as assigned by the USB-IF and a unique Product ID (PID) assigned by the accessory developer. The VID must correspond to the brand name that appears on the accessory or its packaging. See USB Host Mode (page 712).

The Magnetic Charging Module factory configuration interface provides commands to read and set each configuration parameter. Commands take two forms:

- "[*commandName*]" to read the current value of a parameter.
- "[*commandName*–*parameterValue*]" to set the value of a parameter.

Command responses take the form: "<*commandName*–*value*>".

The following commands are supported:

- Set Vendor Name (page 226)
- Set Product Name (page 226)
- Set Model Number (page 227)
- Set Serial Number (page 227)
- Set USB Vendor ID (page 227)
- Set USB Product ID (page 228)
- Lock Configuration (page 228)

8.4.1 Set Vendor Name

Description: Sets the Vendor Name string.

Format: `[SetVN–vendorName]`

Parameters:

- *vendorName*: A string representation of the vendor name from the manufacturer (up to 64 UTF8 characters).

Example: `[SetVN–Example Product Developer]`

Response: `<SetVN–Example Product Developer>`

8.4.2 Set Product Name

Description: Sets the Product Name string.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Format: `[SetPN–productName]`

Parameters:

- *productName*: A string representation of the product name from the manufacturer (up to 64 UTF8 characters).

Example: `[SetPN–Example Product]`

Response: `<SetPN–Example Product>`

8.4.3 Set Model Number

Description: Sets the Model Number string.

Format: `[SetMN–modelNumber]`

Parameters:

- *modelNumber*: A string representation of the model number from the manufacturer (up to 32 UTF8 characters).

Example: `[SetMN–ABC 123]`

Response: `<SetMN–ABC 123>`

8.4.4 Set Serial Number

Description: Sets the Serial Number string.

Format: `[SetSN–serialNumber]`

Parameters:

- *serialNumber*: A string representation of the serial number from the manufacturer (up to 32 UTF8 characters).

Example: `[SetSN–ABCDEF123456]`

Response: `<SetSN–ABCDEF123456>`

8.4.5 Set USB Vendor ID

Description: Sets the USB Vendor ID hex value.

Format: `[SetUV–vendorID]`

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Parameters:

- *vendorID*: A string representation of the hex value of the vendor ID from the manufacturer, without the `0x` prefix.

Example: `[SetUV-0000]`

Response: `<SetUV-0000>`

8.4.6 Set USB Product ID

Description: Sets the USB Product ID hex value.

Format: `[SetUP-`*`productID`*`]`

Parameters:

- *productID*: A string representation of the hex value of the product ID from the manufacturer, without the `0x` prefix.

Example: `[SetUP-0000]`

Response: `<SetUP-0000>`

8.4.7 Lock Configuration

Description: Locks the configuration for production use.

Format: `[LkCfg-`*`mode`*`]`

Parameters:

- *mode*:
  - 1 = Configure the next boot into production mode, then return to VCP mode. This setting may be used for testing.
  - 2 = Permanently boot in production mode. This setting must be used for all production units.
  - All other values are reserved.

Example: `[LkCfg-2]`

Response: `<LkCfg-2>`

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8.4.8 Example Sequence

The following is an example factory configuration sequence which tests the settings in production mode before permanently locking the Magnetic Charging Module:

1. Apply power to Magnetic Charging Module and connect to USB VCP interface.
2. Issue all `SetXX` commands to program module.
3. Issue `LkCfg-1` (next boot will come up in production mode).
4. Cycle power, Magnetic Charging Module boots in production mode.
5. Verify configuration (see Verifying Configuration (page 229)).
6. Cycle power, Magnetic Charging Module boots into VCP mode.
7. Issue `LkCfg-2` (permanently lock firmware settings).
8. Cycle power, Magnetic Charging Module permanently boots in production mode.
9. Verify configuration (see Verifying Configuration (page 229)).

The following is an example factory configuration sequence which permanently locks the Magnetic Charging Module immediately after configuration:

1. Apply power to Magnetic Charging Module and connect to USB VCP interface.
2. Issues all `SetXX` commands to program module.
3. Issue `LkCfg-2` (permanently lock firmware settings).
4. Cycle power, Magnetic Charging Module permanently boots in production mode.
5. Verify configuration (see Verifying Configuration (page 229)).

8.4.9 Verifying Configuration

Verify that the Magnetic Charging Module has been correctly configured with the following procedure or its equivalent for another computer/operating system combination:

1. Connect the Magnetic Charging Module to the USB receptacle on a Mac running OS X.
2. Run the 'System Information' app
3. Select 'USB' under the 'Hardware' category on the left pane
4. Confirm that the Magnetic Charging Module is appearing to the Mac as a USB device using the configured settings (Vendor ID, Product ID, Product Name, Serial Number, etc.).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8.5 USB Virtual COM Port

The USB Virtual COM Port (VCP) enables configuration of the Magnetic Charging Module.

When using a Mac running OS X, the Magnetic Charging Module is recognized as a USB serial device. The accessory will appear as `/dev/tty.usbmodem–<identifier>`. The interface may be accessed using various serial interface programs (e.g. executing `screen /dev/tty.usbmodem–<identifier>` given the appropriate identifier in a terminal).

When using a computer running Windows, the STMicro VCP driver is required. The STM32 Virtual COM Port Driver (part number STSW-STM32102) is available at http://www.st.com/. Once installed, edit the `stmcdc.inf` file to add Apple VID 0x05AC and Magnetic Charging Module PID 0x1709 to the DeviceList sections as follows:

```
[DeviceList.NT]


[DeviceList.NTamd64]
```

If the VCP is not identified properly upon connection, update the driver using the Windows Device Manager. If the accessory is listed as a Composite device, select it and update the driver (choose "Have Disk", and browse to the folder containing the updated `stmcdc.inf` file).

8.6 Test Procedures

Test procedures for accessories integrating the Magnetic Charging Module are contained in this section. Tests should be performed with an Apple Watch 42 mm Stainless Steel Case with Link Bracelet.

8.6.1 Mechanical

This section contains mechanical test procedures for accessories integrating the Magnetic Charging Module.

8.6.1.1 Product Design

1. Verify (via visual inspection) that the accessory:
   a. Does not interfere with the body of the Apple Watch when connected for charging.
   b. Does not require the watch band to be removed or detached.
   c. Does not make contact with the watch band in a manner that may scratch or damage the band.
2. Verify (via visual inspection and with ferromagnetic metal) that the accessory does not have magnetic steel in proximity to the connector.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Using a caliper, verify that the Magnetic Charging Module is 0.40 mm ±0.25 mm proud of the surface of the accessory.
4. Using a protractor, verify that the Magnetic Charging Module is not mounted at an angle greater than 45 degrees from the horizontal.

8.6.1.2 Drop Test

1. Drop the accessory onto plywood from a height of 32 inches.
2. Verify (via visual inspection) that the Magnetic Charging Module has not been dislodged from the accessory housing.
3. Verify that the accessory still charges the Apple Watch.

8.6.2 Electrical

This section contains electrical test procedures for accessories integrating the Magnetic Charging Module.

8.6.2.1 Internal Power Supplies

This section contains electrical test procedures for accessories containing internal power supplies.

8.6.2.1.1 Equipment

The following equipment is required for performing the electrical test procedures:

- Electronic load rated at 10 W or higher, capable of constant current (CC) mode.
- Oscilloscope with 100 MHz or higher bandwidth.

8.6.2.1.2 Test Setup

1. Disconnect the Magnetic Charging Module from the accessory's 4-pin connector.
2. Connect the electronic load and the oscilloscope channel to VBUS and GND pins on the connector.
3. Connect the accessory to the power source.

8.6.2.1.3 Supply Voltage (DC)

1. Configure the oscilloscope as follows:
    - Horizontal scale: 1 ms/div
    - Vertical scale: 1 V/div
    - Channel coupling: DC

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. With electronic load set to CC mode, record the average VBUS voltage on the scope for each load from 0 A to 1 A in 200 mA steps.
3. Verify that all recorded values are within 4.75 V and 5.40 V.

8.6.2.1.4 Supply Voltage Ripple

1. Configure the oscilloscope as follows:
   - Horizontal scale: 1 ms/div
   - Vertical scale: 10 mV/div
   - Channel coupling: AC
2. With electronic load set to CC mode, record the peak-peak VBUS voltage on the scope using cursors for each load from 0 A to 1 A in 200 mA steps. Capture at least 10 cycles and measure the worst-case peak-peak voltage, adjusting the horizontal scale if necessary based on the frequency of the ripple.
3. Verify that all recorded values are below 20 mV.

8.6.2.1.5 Overcurrent Protection (OCP)

1. Configure the oscilloscope as follows:
   - Horizontal scale: 1 ms/div
   - Vertical scale: 1 V/div
   - Channel coupling: DC
2. With electronic load set to CC mode, observe the average VBUS voltage on the scope as the load is varied from 1.4 A to 1.6 A in 10 mA increments.
3. Verify that the VBUS voltage reading on the scope falls within 4.75 V and 5.4 V with the electronic load setting of 1.4 A and the output current draw falls to less than 20 mA with the electronic load setting of 1.5 A ±2% (between 1.47 A and 1.53 A).

8.6.2.2 External USB Power Sources

This section contains electrical test procedures for accessories using external USB power sources.

8.6.2.2.1 Equipment

The following equipment is required:

- DMM with 4-wire Kelvin sense capability

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8.6.2.2.2 Test Setup

1. Disconnect the Magnetic Charging Module from the accessory's 4-pin connector.

8.6.2.2.3 Round-Trip DC Resistance (DCR)

1. Connect the "Force" and "Sense" leads of the DMM to the VBUS and GND pins on the accessory's 4-pin connector.
2. Short the VBUS and GND signals at the USB-A plug using a breakout board.
3. Configure the DMM to perform a 4-wire Kelvin sense resistance measurement.

8.6.2.2.4 Round-Trip DC Resistance (DCR) with Overcurrent Protection (OCP)

When OCP circuitry is included, the total resistance (on-resistance of the OCP switch and the DCR of the cable) should be measured.

1. Connect the power supply's positive and negative terminals to the USB-A plug's VBUS and GND pins respectively using a USB breakout board.
2. Connect the electronic load to the VBUS and GND pins on the accessory's 4-pin connector.
3. Connect the positive lead of the DMM to the USB-A plug's VBUS pin on the breakout board.
4. Connect the negative lead of the DMM to the 4-pin connector's VBUS pin.
5. Set the power supply to supply 5 V and turn it on.
6. Set the electronic load to draw 100 mA.
7. Record the voltage, V_{VBUS} in mV, from the DMM.
8. Connect the positive lead of the DMM to the 4-pin connector's GND pin.
9. Connect the negative lead of the DMM to the USB-A plug's GND pin on the breakout board.
10. Record the voltage, V_{GND} in mV, from the DMM.
11. Calculate the total resistance: DCR = (V_{VBUS} + V_{GND}) / 100

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

9. Apple 30-pin Connector

These connectors must not be the sole or main point of physical support for an iPhone, iPad, or iPod. Accessories must use the universal dock & dock adapter approach to provide support, or, alternatively, must implement another means (other than the connector) to provide physical support to an iPhone, iPad, or iPod.

See *MFi Accessory Hardware Specification R9* for full details of the 30-pin connector.

9.1 Connector Versions

9.1.1 Mid-Plane Plug

This connector should be used with products that require a metal shell around the connector (for shielding) or with products that do not require latching pins (friction or detent). This plug is designed so that the pins on the connector will straddle both sides of a PCB and is not designed for surface mount solutions. It is commonly used with products that have electronics right in the plug that plugs into the iPod.

The mid-plane plug is designed for use with the Mid-Plane Plug (Top Shell) (page 235) & Mid-Plane Plug (Bottom Shell) (page 235), which provide a passive/friction latching mechanism.

*Note: Only 16 pins are present in this connector. See the connector drawing for details. Licensees are **strongly** encouraged to purchase samples of this connector to confirm that it meets their requirements.*

See Figure 9-1 (page 238) for a dimensional drawing.

9.1.2 Mid-Plane Plug with Friction Latches

This connector is designed so that the pins on the connector will straddle both sides of a PCB. It is commonly used with products that have electronics within the connector's housing.

This connector should be used with products where a metal shell around the connector is not desired or supported, but does require friction latching pins.

See Figure 9-2 (page 239) for a dimensional drawing.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

9.1.3 Mid-Plane Plug with Friction Latches, Power Only

This connector is designed so that the pins on the connector will straddle both sides of a PCB.

This connector should be used with products where a metal shell around the connector is not desired or supported, but does require friction latching pins.

Note: Only 10 pins are present in this connector. See the connector drawing for details.

See Figure 9-3 (page 240) for a dimensional drawing.

9.1.4 Mid-Plane Plug (Top Shell)

This component is used in combination with Mid-Plane Plug (Bottom Shell) (page 235). It includes latch 'fingers' that hold the connector into the iPod. Users are not required to press detent buttons on the side of the connector to disconnect the accessory.

This is not a plastic over-mold or shell, and helps shield against EMI.

See Figure 9-4 (page 241) for a dimensional drawing.

9.1.5 Mid-Plane Plug (Bottom Shell)

Used in combination with Mid-Plane Plug (Top Shell) (page 235).

See Figure 9-5 (page 242) for a dimensional drawing.

9.1.6 Cable-End Plug

This connector is designed to be soldered to the end of a cable. It should be used with cable assemblies, where the connector cannot be unplugged unless the user presses release buttons on the side of the connector housing. This connector has solder tail pins, and is not suitable for mounting onto a PCB.

The cable-end plug is designed for use with Cable-End Plug (Top Shell) (page 236) & Cable-End Plug (Bottom Shell) (page 236), which provide an active latch mechanism.

See Figure 9-6 (page 243) for a dimensional drawing.

9.1.7 Cable-End Plug, Power Only

This version of the cable-end connector is modified for use in 'Power & Sync Only' products.

This connector is designed to be soldered to the end of a cable. It should be used with cable assemblies, where the connector cannot be unplugged unless the user presses release buttons on the side of the connector housing. This connector has solder tail pins, and is not suitable for mounting onto a PCB.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Designed for use with Cable-End Plug (Top Shell) (page 236) & Cable-End Plug (Bottom Shell) (page 236), which provide an active latch mechanism.

Note: Only 10 pins are present in this connector. See the connector drawing for details.

See Figure 9-7 (page 244) for a dimensional drawing.

9.1.8 Cable-End Plug (Top Shell)

This component is used in combination with Cable-End Plug (Bottom Shell) (page 236)

It includes latch 'fingers' that hold the connector into the iPod. Users are required to press detent buttons on the side of the connector to disconnect the accessory.

This is not a plastic over-mold or shell, and helps shield against EMI.

See Figure 9-8 (page 245) for a dimensional drawing.

9.1.9 Cable-End Plug (Bottom Shell)

This component is used in combination with Cable-End Plug (Top Shell) (page 236)

See Figure 9-9 (page 246) for a dimensional drawing.

9.1.10 Vertical Dock Plug

The vertical dock is used in products where an Apple device will sit vertically (perpendicular to the PCB), such as battery packs.

See Figure 9-10 (page 248) for a dimensional drawing.

9.1.11 Angled Dock Plug

The angled dock is tilted 15 degrees off vertical, with a support tab. It is designed for use with plastic sleeve and adapters used with the universal dock.

See Figure 9-11 (page 249) for a dimensional drawing.

9.1.12 Receptacle

The receptacle connector is only available in sample quantities, and is only for use in manufacturing and R&D test fixtures.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: This connector is not available for use in production and/or shipping products.

See Figure 9-12 (page 250) for a dimensional drawing.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

9.2 Connector Dimensional Drawings

Figure 9-1 Mid-Plane Plug

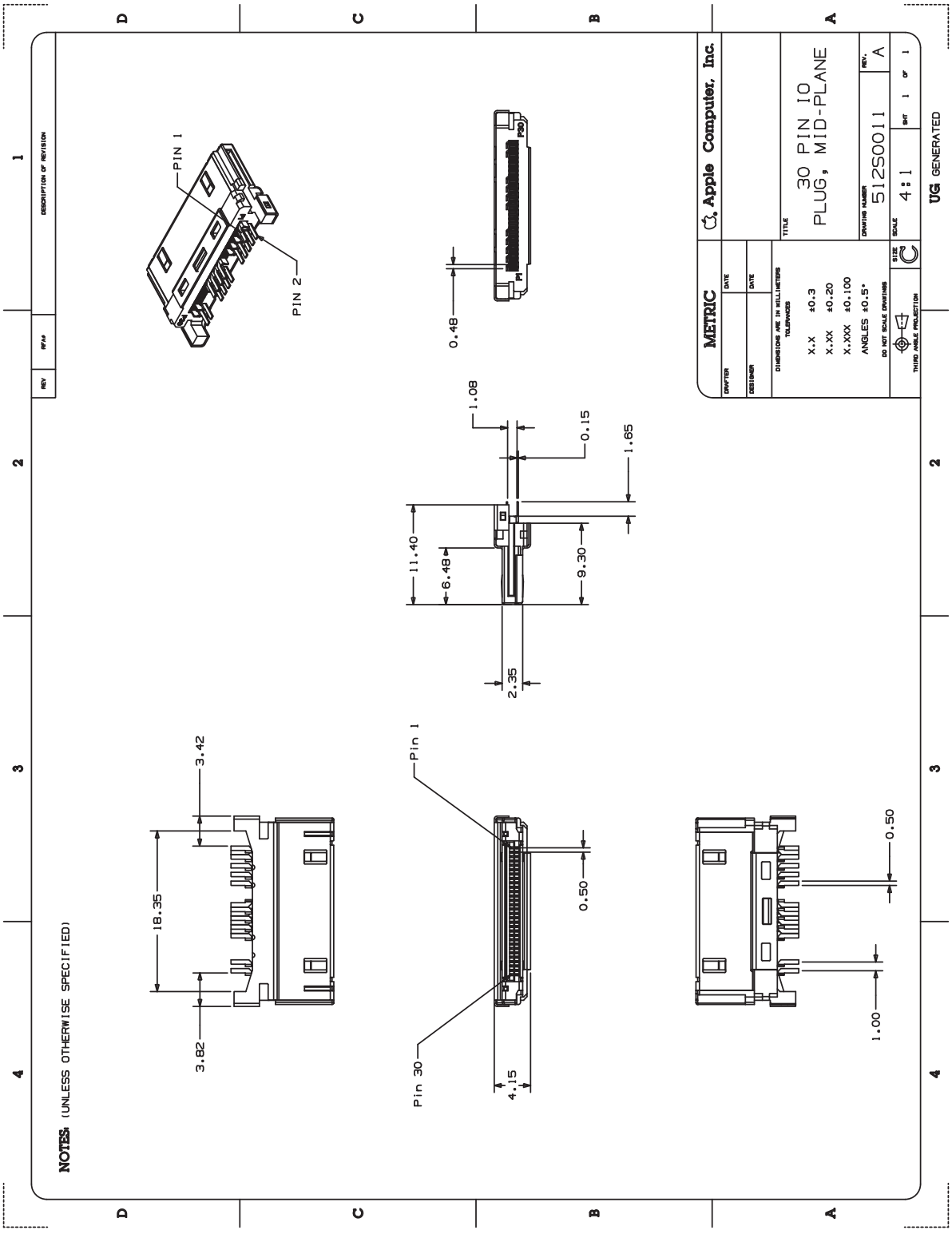


Figure 9-2 Mid-Plane Plug, with Friction Latches

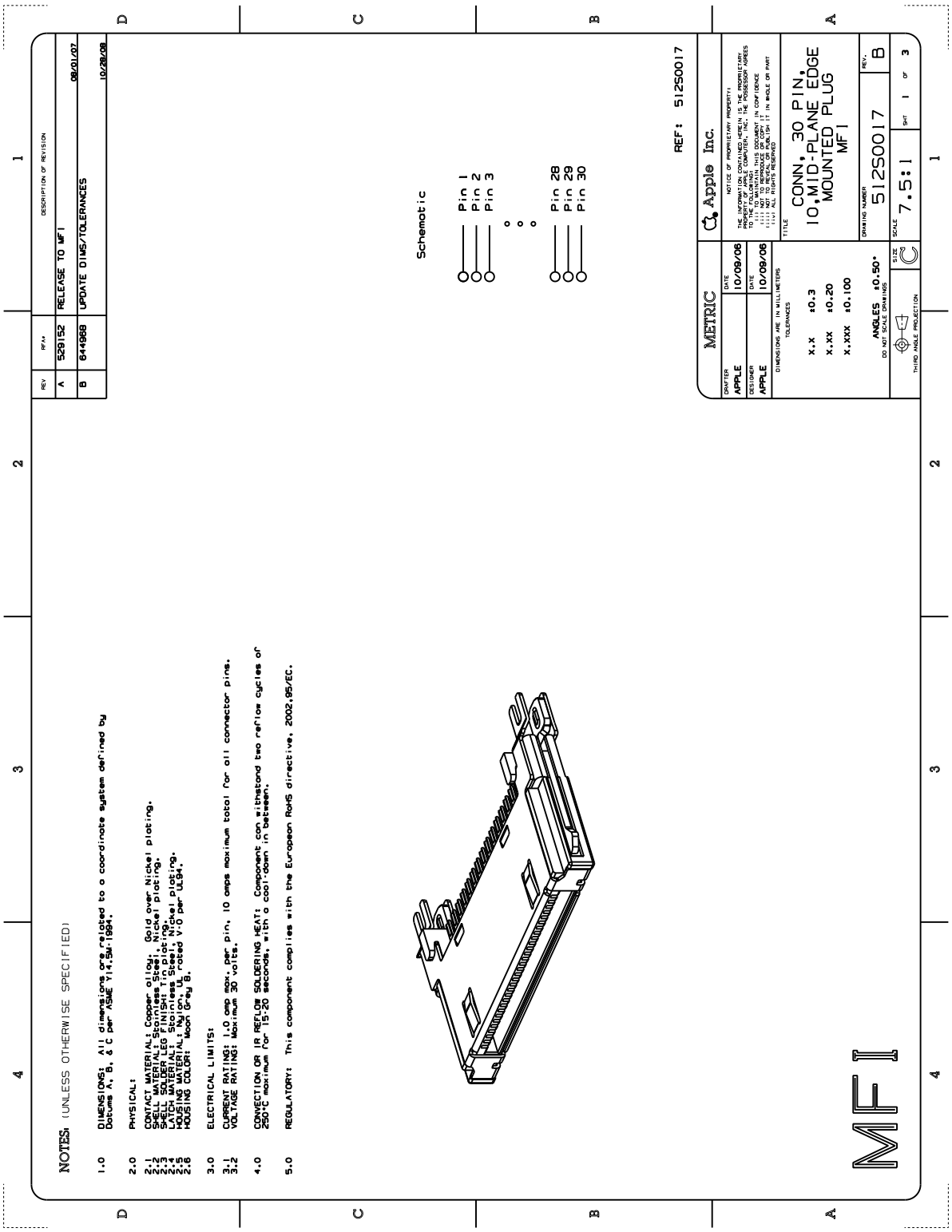


Figure 9-3 Mid-Plane Plug with Friction Latches, Power Only

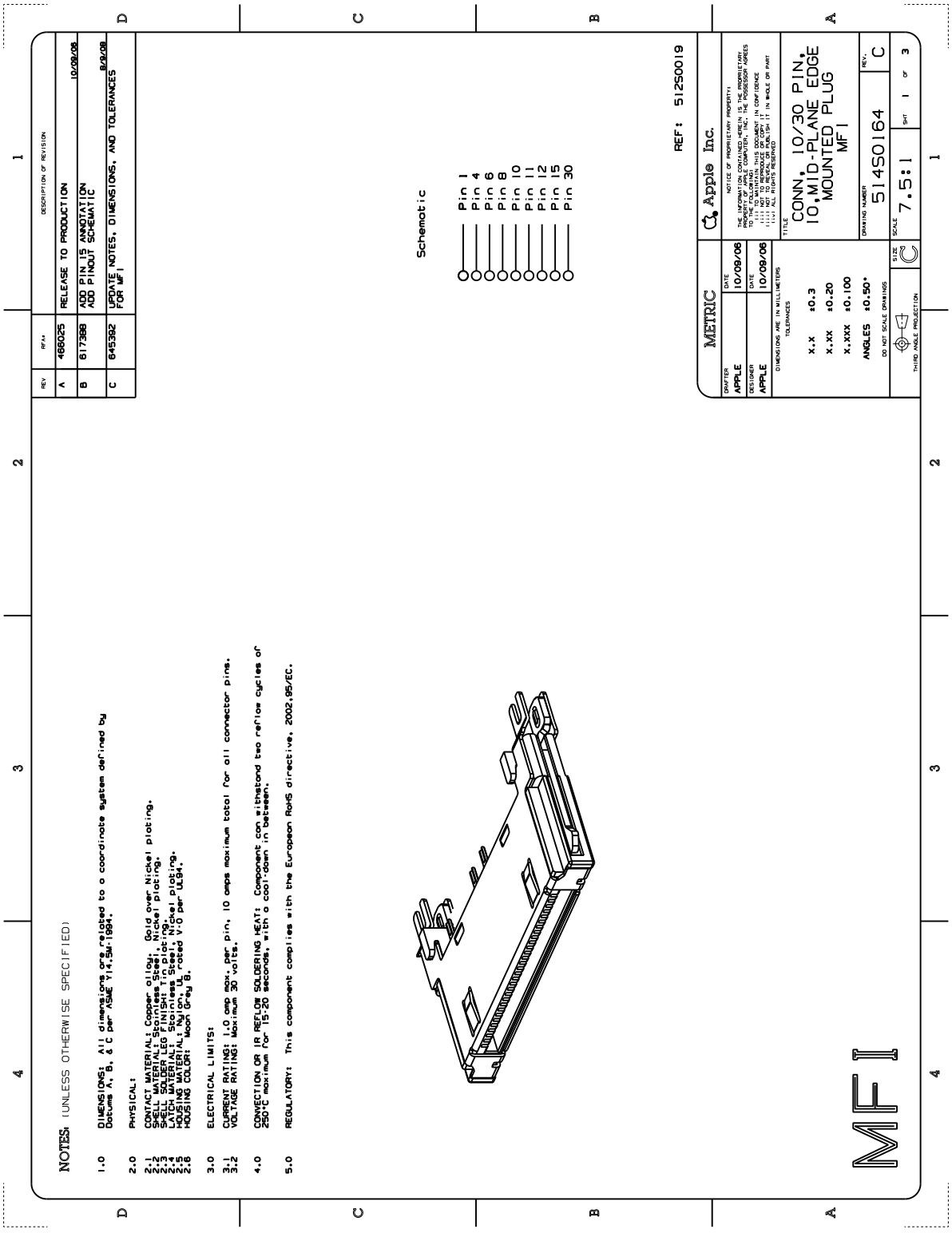


Figure 9-4 Top Shell for Mid-Plane Plugs

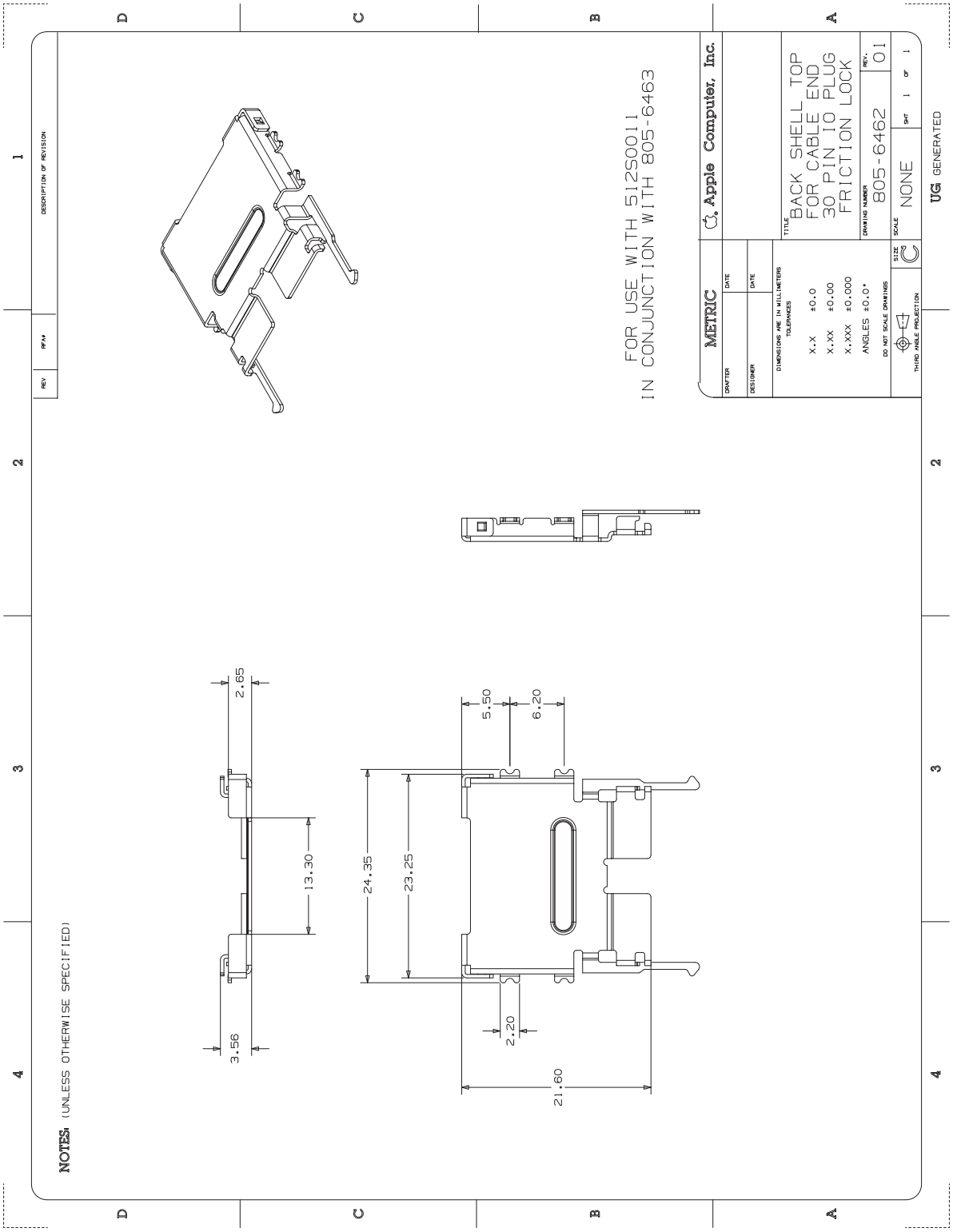


Figure 9-5 Bottom Shell for Mid-Plane Plugs

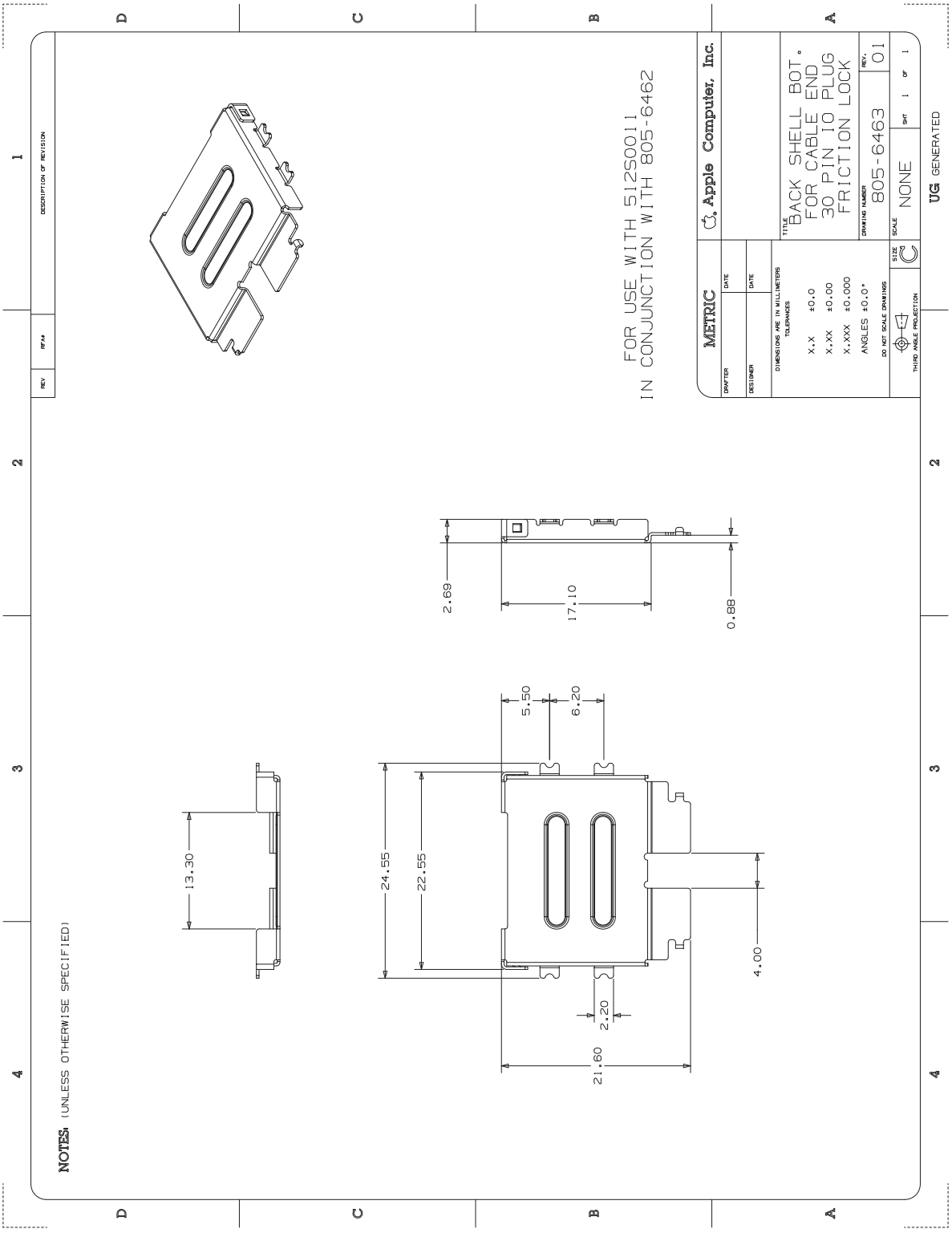


Figure 9-6 Cable-End Plug

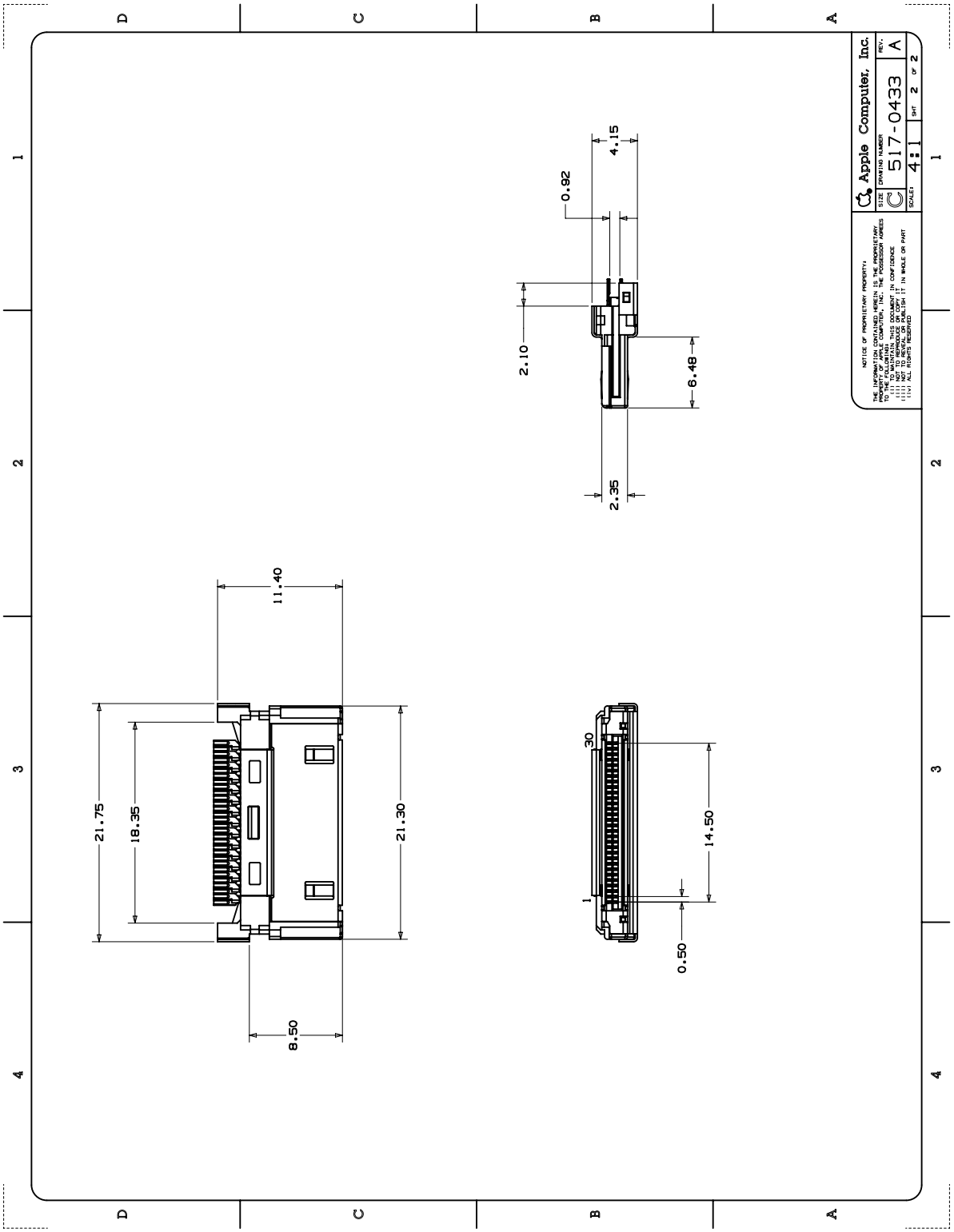


Figure 9-7 Cable-End Plug, Power Only

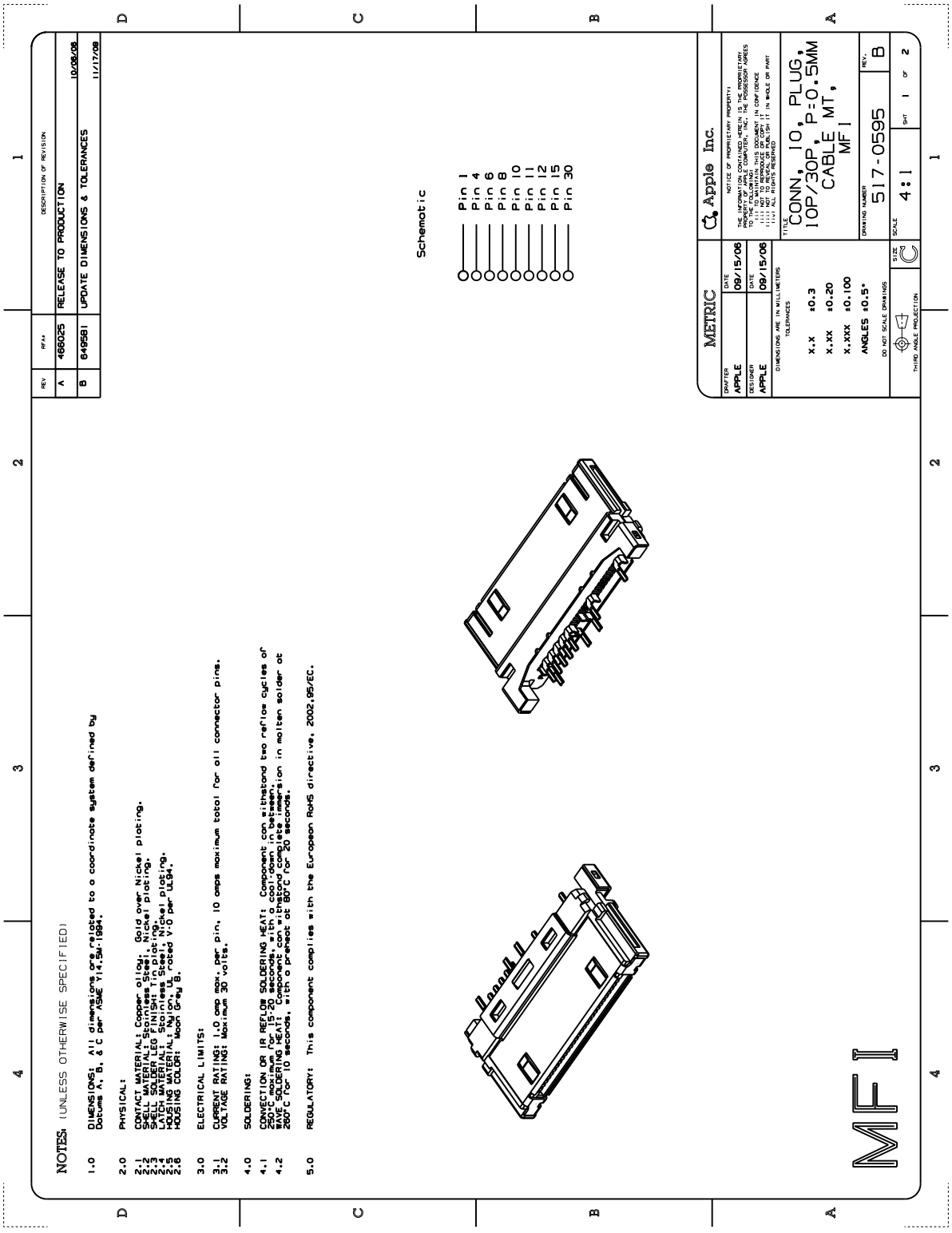


Figure 9-8 Top Shell for Cable End Plugs

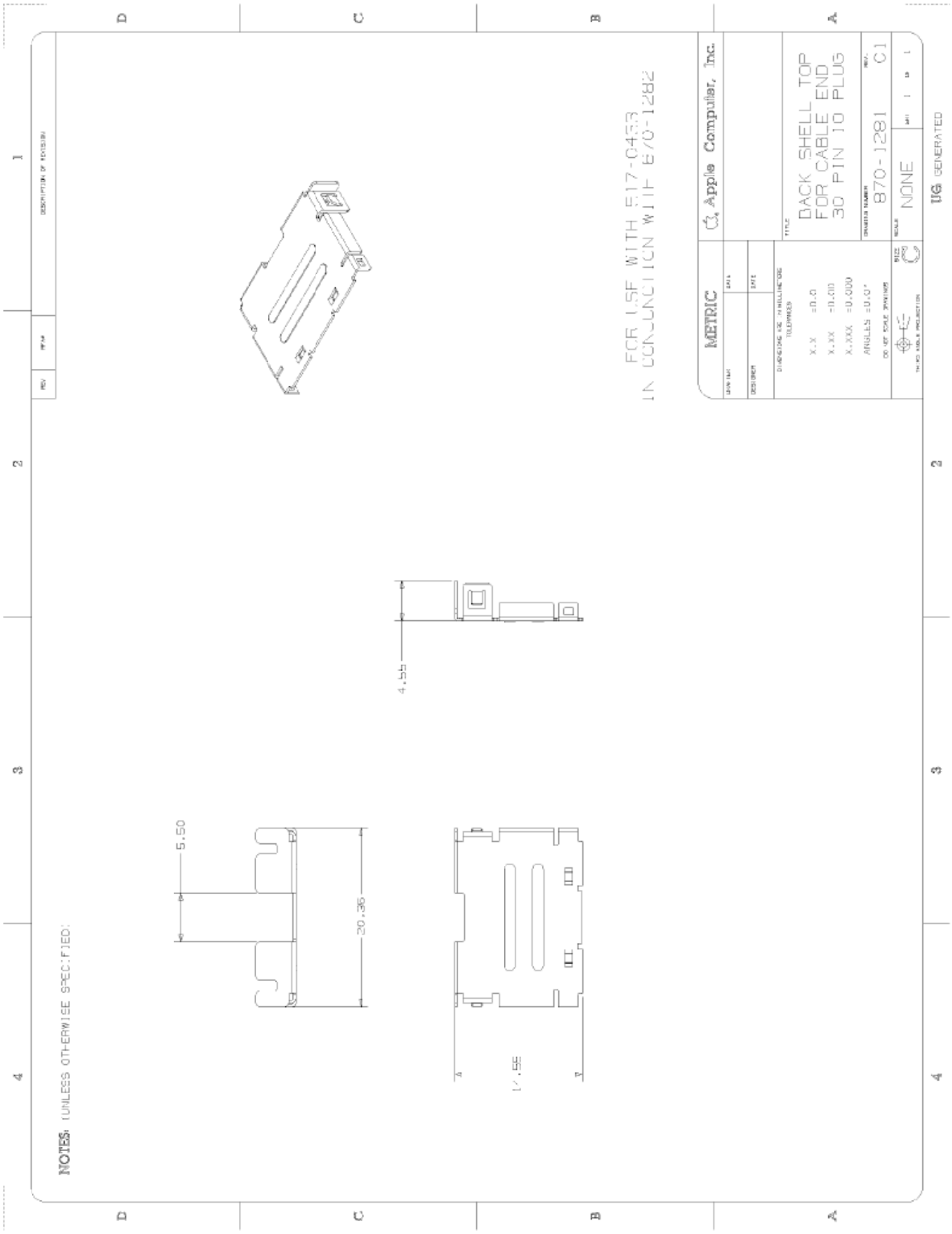


Figure 9-9 Bottom Shell for Cable End Plugs

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

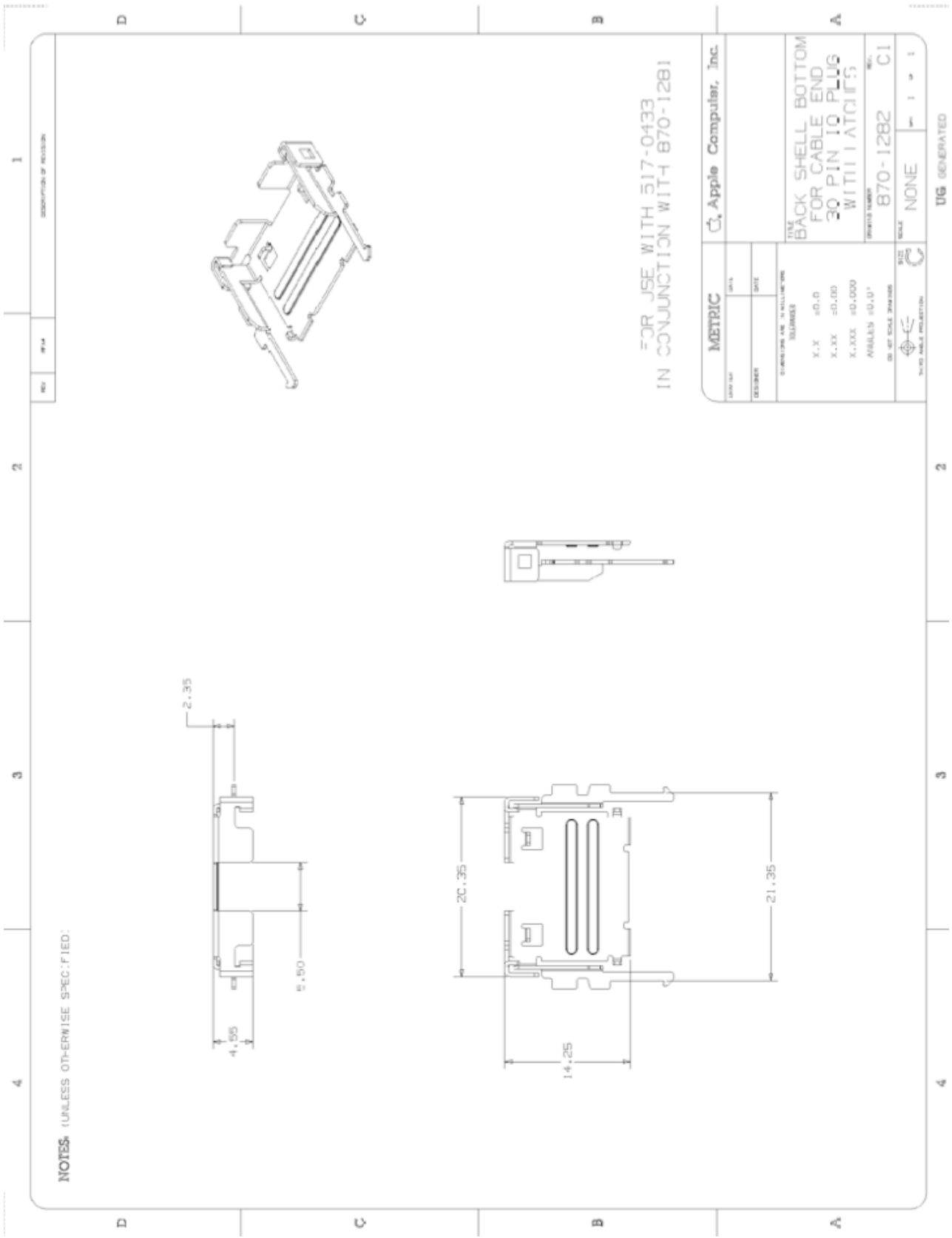

FOR USE WITH 517-0433
IN CONJUNCTION WITH 870-1281
Apple Computer, Inc.
METRIC
BACK SHELL BOTTOM
FOR CABLE END
30 PIN IO PLUG
WITH LATCHES
870-1282
C1
NONE
NOTES: (UNLESS OTHERWISE SPECIFIED)
2.35
5.50
4.55
20.35
21.35
14.25

Figure 9-10 Vertical Dock Plug

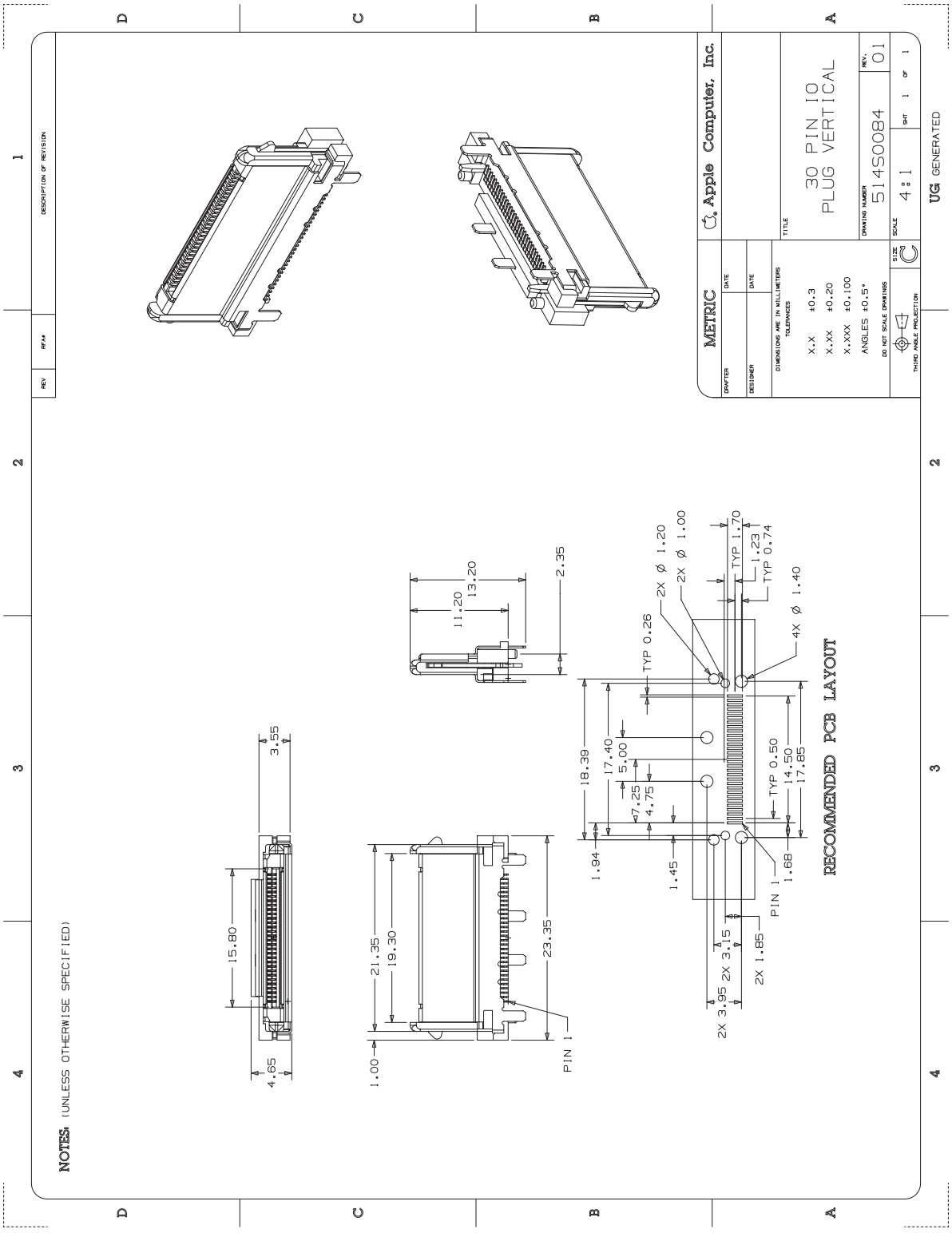


Figure 9-11 Angled Dock Plug

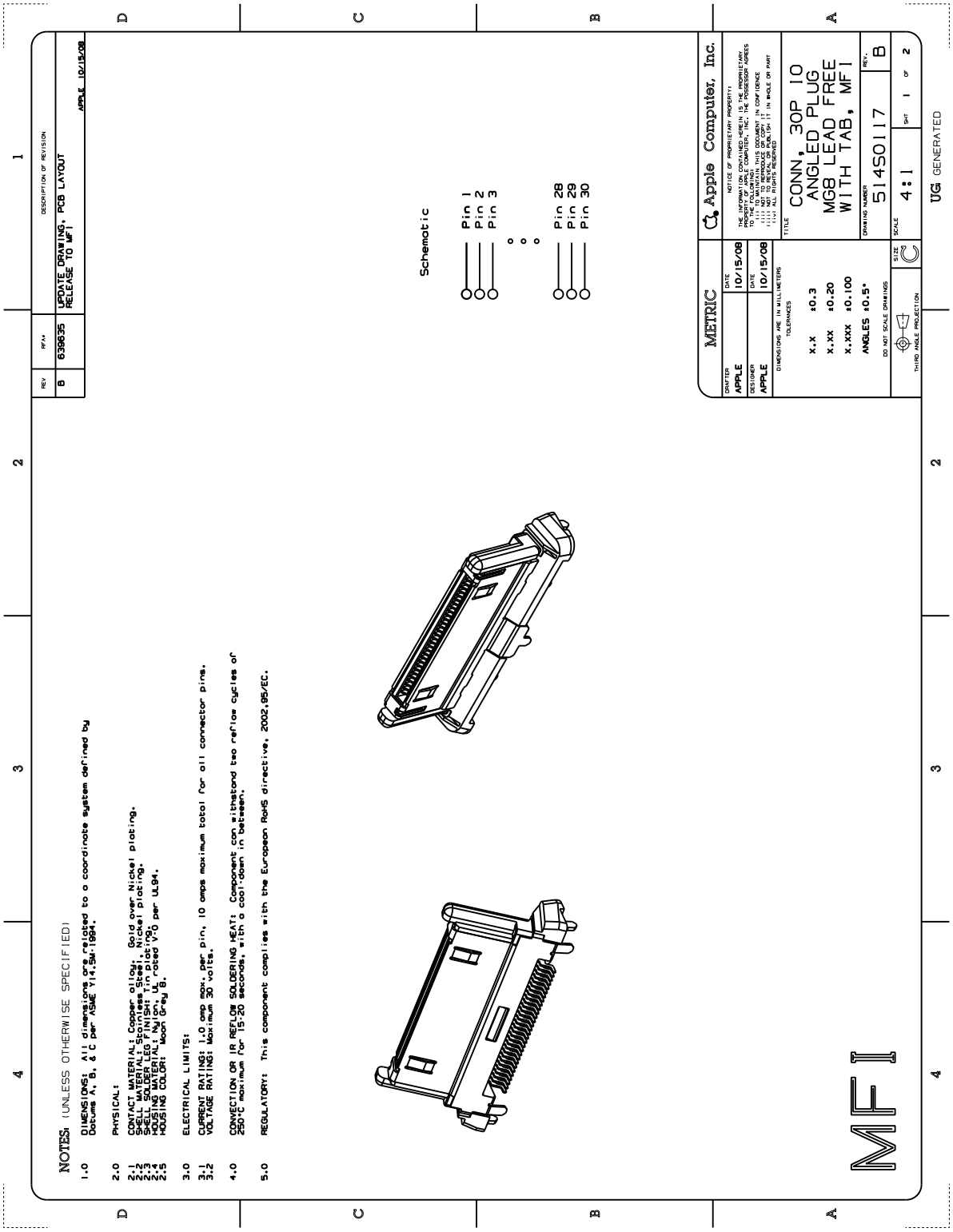
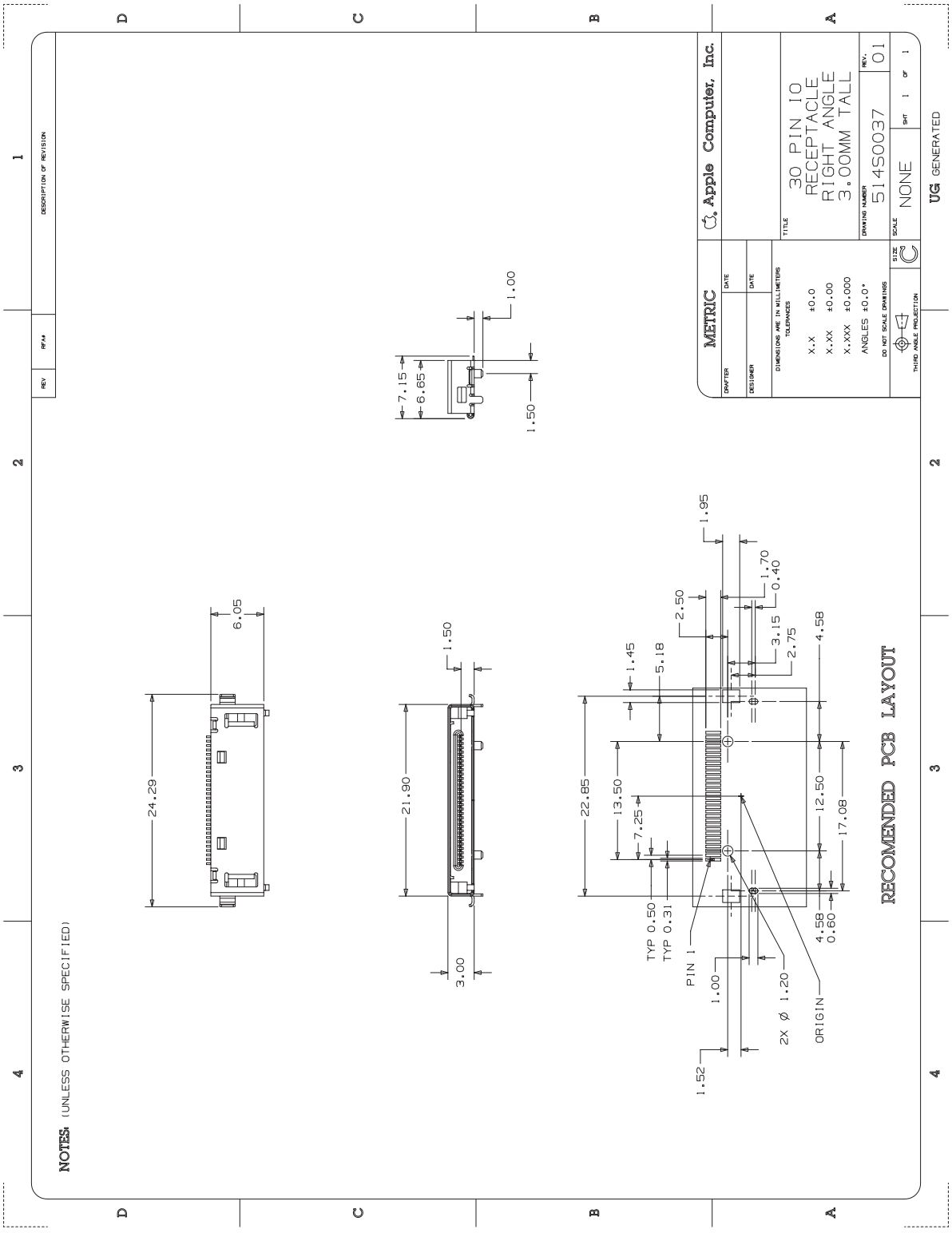


Figure 9-12 Receptacle



10. Apple Smart Connector

Note: This chapter is a Developer Preview and is not intended for use in the development of Proposed Products or Licensed Products under a MFi License. Although this content has been reviewed for accuracy, it is not final. Apple is supplying this content to help accessory developers plan for the adoption of the accessory interface features described herein. This information is subject to change, and accessories implemented according to this content must be tested with final operating system software and final documentation before going through self certification.

The Apple Smart Connector must be used in conjunction with the Apple Smart Connector Module (page 253).

10.1 Reversibility

The Apple Smart Connector is not reversible.

10.2 Smart Connector Pad Configuration

The Apple Smart Connector contains 3 pads:

- Data
- Power
- Ground

10.3 Mechanical Requirements

All accessories that integrate the Smart Connector into their design must comply with the mechanical requirements in this section.

Accessories must combine the following additional components with a Smart Connector:

- Two M1.2 screws
- 0.3 mm thick foam

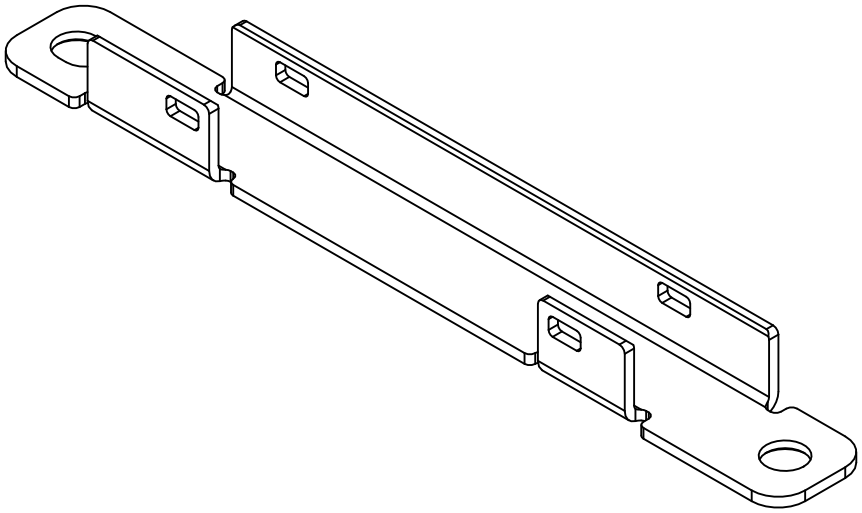
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- 0.286 mm ±0.05 mm thick stiffener

Accessory designs must respect these areas when meeting the following requirements:

- Mated assembly shall be subjected to mechanical shock per EIA-364-27B, test condition A, three discontinuities of 1 µs or longer duration.
- Any process performed on the Smart Connector must not cause any buildup of material on its exposed contact surface.
- Pin height must be 0.45 mm ±0.05 mm after being assembled onto the accessory.
- All 4 snaps between the connector and bracket must be engaged.
- Assembled Smart Connector must not short any pins to the bracket.
- Accessory's Smart Connector must make contact normal to the Apple device's Smart Connector.

Figure 10-1 Smart Connector Module Mounting Bracket



10.4 Electrical Requirements

Smart Connector accessories must meet ESD protection requirements Level 2 outlined in IEC 61000-4-2. Level 4 is recommended.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

11. Apple Smart Connector Module

Note: This chapter is a Developer Preview and is not intended for use in the development of Proposed Products or Licensed Products under a MFi License. Although this content has been reviewed for accuracy, it is not final. Apple is supplying this content to help accessory developers plan for the adoption of the accessory interface features described herein. This information is subject to change, and accessories implemented according to this content must be tested with final operating system software and final documentation before going through self certification.

Accessories may make use of the Apple Smart Connector Module to create the following types of accessories that attach to Apple devices with a Apple Smart Connector (page 251):

- Form-fitting
- Docks

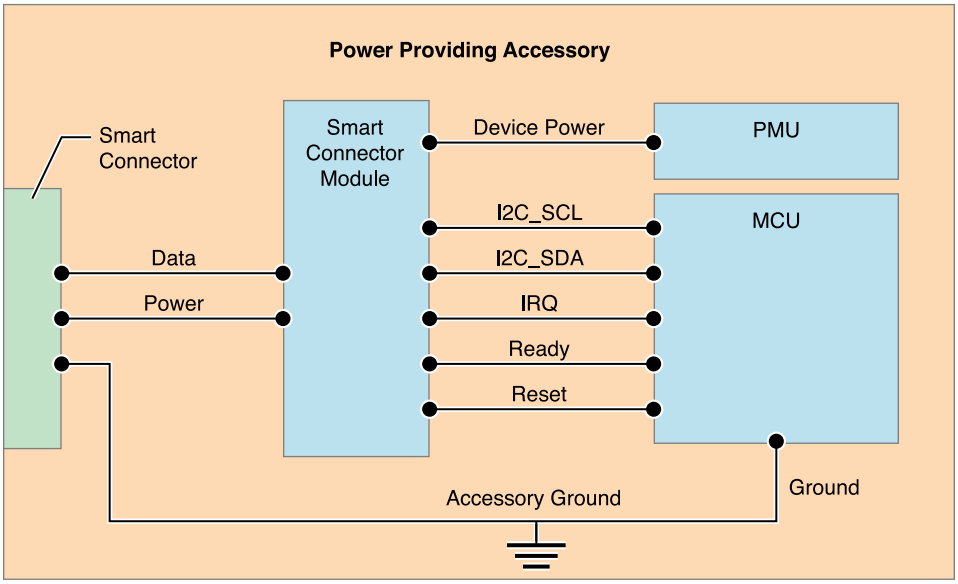
Cable and dongle accessories must not integrate a Smart Connector.

Accessories using the Smart Connector Module may support the following features:

- iAP2 Control Session
- Native HID
- Accessory Power
- Device Power

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 11-1 Smart Connector Module Accessory (Power Providing) Example



11.1 Overview

Key accessory-facing features of the Smart Connector Module are:

- I2C Interface
- Accessory Power
- Device Power

11.2 Firmware Updates

Smart Connector Module accessories must be capable of field-deployable firmware updates from Apple devices running iOS and/or Macintosh computers running OS X.

11.3 Authentication

Accessories that communicate with an Apple device via the Smart Connector Module do not need to integrate an Apple Authentication Coprocessor.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

11.4 Smart Connector

Accessories that integrate the Smart Connector Module must attach to the Apple device using a Apple Smart Connector (page 251).

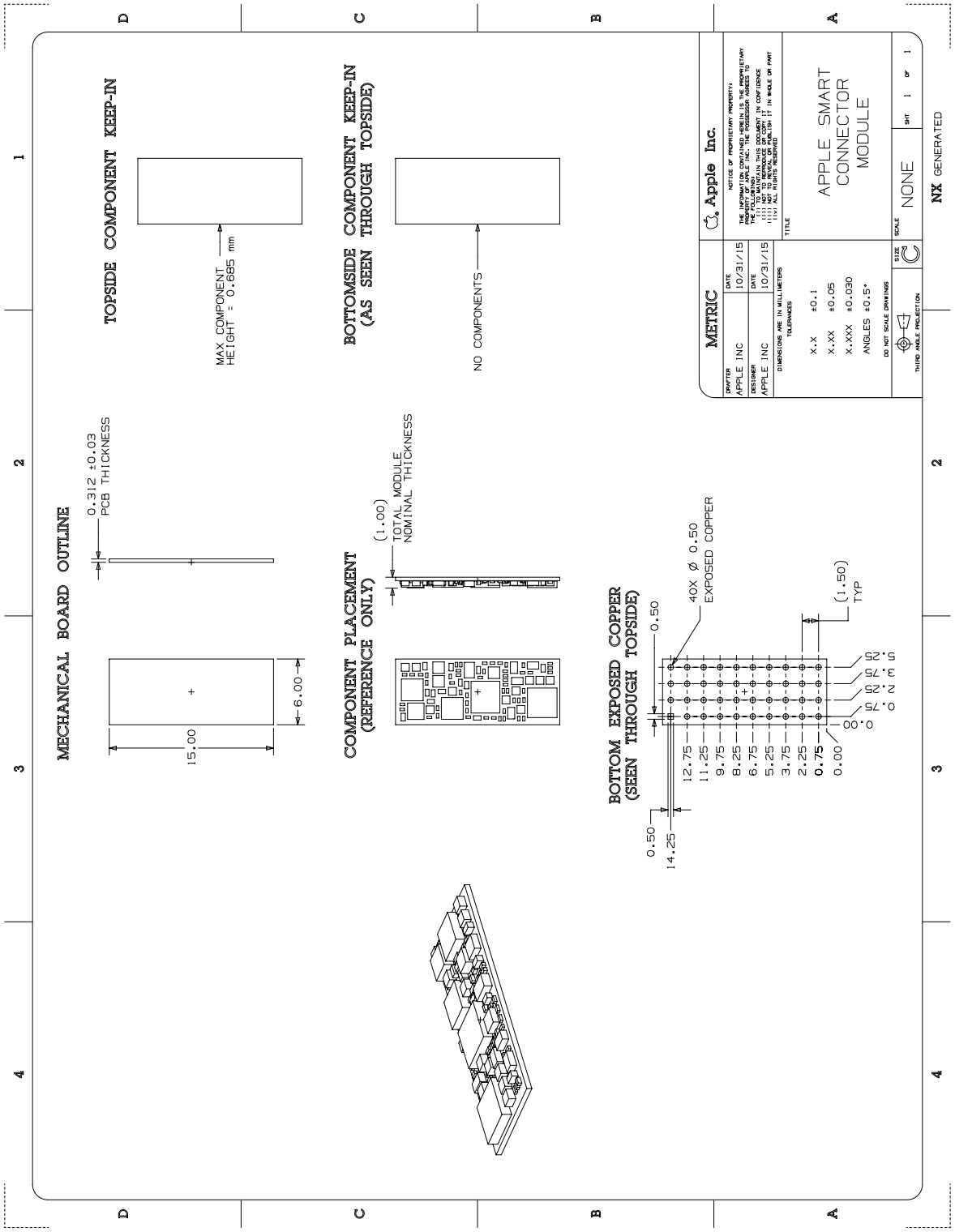
11.5 Mechanical

The Smart Connector Module has the following mechanical characteristics:

- 6 mm by 15 mm by 1 mm LGA package on tape and reel
- 0.5 mm offset
- 1.5 mm center to center on all pads

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

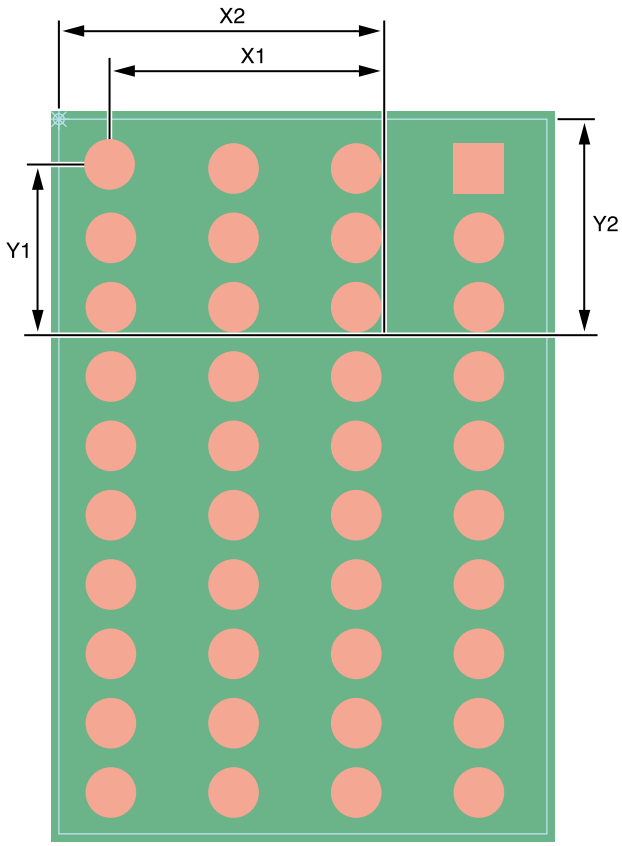
Figure 11-2 Smart Connector Module Mechanical Package



Pickup offsets for the Smart Connector Module (see Figure 11-3 (page 257)) are:

- (X1) 3.25 mm
- (Y1) 3.25 mm
- (X2) 4.00 mm
- (Y2) 4.00 mm

Figure 11-3 Smart Connector Module pickup offsets (bottom view)



11.6 Pad Layout and Assignments

The following diagrams detail the layout, names, and description of the Smart Connector Module pads. All Reserved pads must be soldered but left Not Connected (NC).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 11-4 Smart Connector Module pad layout (top view looking through module)

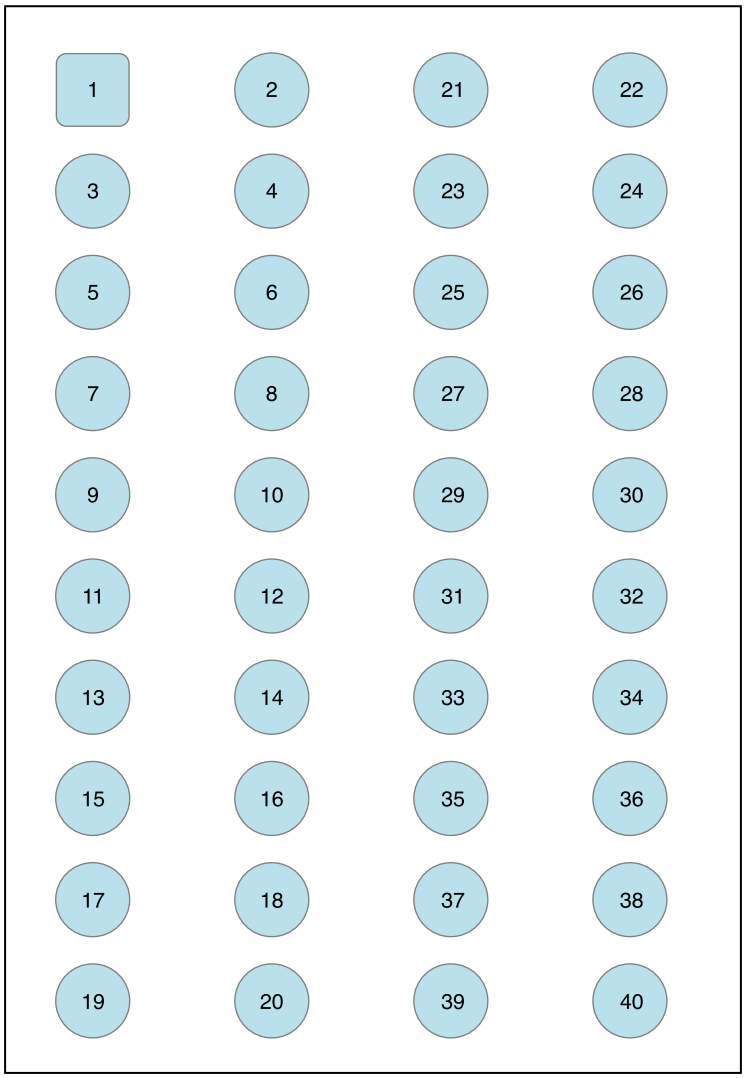


Table 11-1 Smart Connector Module pad assignments

Pad	Name	Assignment
1-6	Power	connects to Smart Connector Power pad
7	Data	connects to Smart Connector Data pad
8-10	Ground	connects to Smart Connector Ground pad
11-20	Reserved	
21-26	Accessory Power	
27-32	Device Power	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Pad	Name	Assignment
33	Reserved	
34	SDA	
35	Reserved	
36	SCL	
37	Reserved	
38	RESET	
39	READY	
40	IRQ	

11.7 Input/Output Electrical Characteristics

The following electrical characteristics apply to all Smart Connector Module I/O and I2C pads:

Table 11-2 Smart Connector Module Input/Output Characteristics

Characteristic	Minimum	Maximum
High-level input	1.26 V	1.8 V for I/O, 5.0 V for I2C
Low-level input	0 V	0.54 V
High-level output	1.35 V	1.8 V
Low-level output	0 V	0.45 V

Table 11-3 Smart Connector Module I/O and I2C pad voltage characteristics

Pin	Characteristic	Vmin	Vtypical	Vmax	Notes
SDA	Data	0 V		5.0 V	Requires a pull up resistor.
SCL	Clock	0 V		5.0 V	Requires a pull up resistor.
RESET	Output	0 V	1.8 V	1.8 V	Non-buffered GPIO.
IRQ	Input	0 V		5.0 V	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Pin	Characteristic	Vmin	Vtypical	Vmax	Notes
READY	Output	0 V	1.8 V	1.8 V	Non-buffered GPIO.

Table 11-4 Smart Connector Module I/O and I2C pad current characteristics

Pin	Characteristic	Max Source	Typical Source	Typical Sunk	Maximum Sunk
SDA	I2C	16 mA	8 mA	8 mA	22 mA
SCL	I2C	16 mA	8 mA	8 mA	22 mA
RESET	Output	16 mA	8 mA	8 mA	22 mA
IRQ	Input	16 mA	8 mA	8 mA	22 mA
READY	Output	16 mA	8 mA	8 mA	22 mA

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

12. Accessory Authentication

Accessory authentication provides a mechanism for Apple devices to trust the identities and feature sets of accessories that are compliant with this specification.

It is also possible for an accessory to require the Apple device to authenticate itself to the accessory. For more details, see Device Authentication (page 523).

12.1 Accessory Authentication Requirements

The accessory hardware must incorporate an Apple Authentication Coprocessor, version 2.0B or 2.0C. Version 2.0C is strongly recommended and specified in Apple Authentication Coprocessor 2.0C (page 66). Earlier versions of the authentication coprocessor are not allowed.

An accessory may use one authentication processor to authenticate itself to more than one Apple device. Conversely, multiple accessories must not share one authentication coprocessor or authenticate on behalf of each other to the same Apple device.

12.1.1 Accessory Authentication over iAP2 Requirements

The accessory must have successfully established an iAP2 link over an iAP2 transport.

All accessories that support the Accessory Authentication feature via iAP2 must send or receive the following iAP2 control session message(s):

- RequestAuthenticationCertificate (page 805)
- AuthenticationCertificate (page 805)
- RequestAuthenticationChallengeResponse (page 805)
- AuthenticationResponse (page 806)
- AuthenticationFailed (page 806)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

12.2 Accessory Authentication Usage

12.2.1 Accessory Authentication over iAP2 Usage

Authentication over iAP2 is always initiated with the transmission of a RequestAuthenticationCertificate (page 805) message from the Apple device to the accessory.

If control session version 2 is in use and the RequestAuthenticationCertificate (page 805) message contains a RequestAuthenticationCertificateSerialNumber parameter, the accessory should retrieve the X.509 certificate serial number from its Apple authentication coprocessor and transmit it to the Apple device using an AccessoryAuthenticationSerialNumber (page 807) message.

If the serial number is not available or the RequestAuthenticationCertificateSerialNumber parameter is not present, the accessory must retrieve the X.509 certificate from its Apple authentication coprocessor and transmit it to the Apple device using an AuthenticationCertificate (page 805) message. The certificate must be sent within 1 second of receiving the RequestAuthenticationCertificate (page 805).

If the certificate is valid, the Apple device will respond with a RequestAuthenticationChallengeResponse (page 805) message. The accessory then must use the Apple authentication coprocessor to compute a response to the authentication challenge and send an AuthenticationResponse (page 806) message with the response. This response must be sent within 2 seconds of receiving the RequestAuthenticationChallengeResponse (page 805) message.

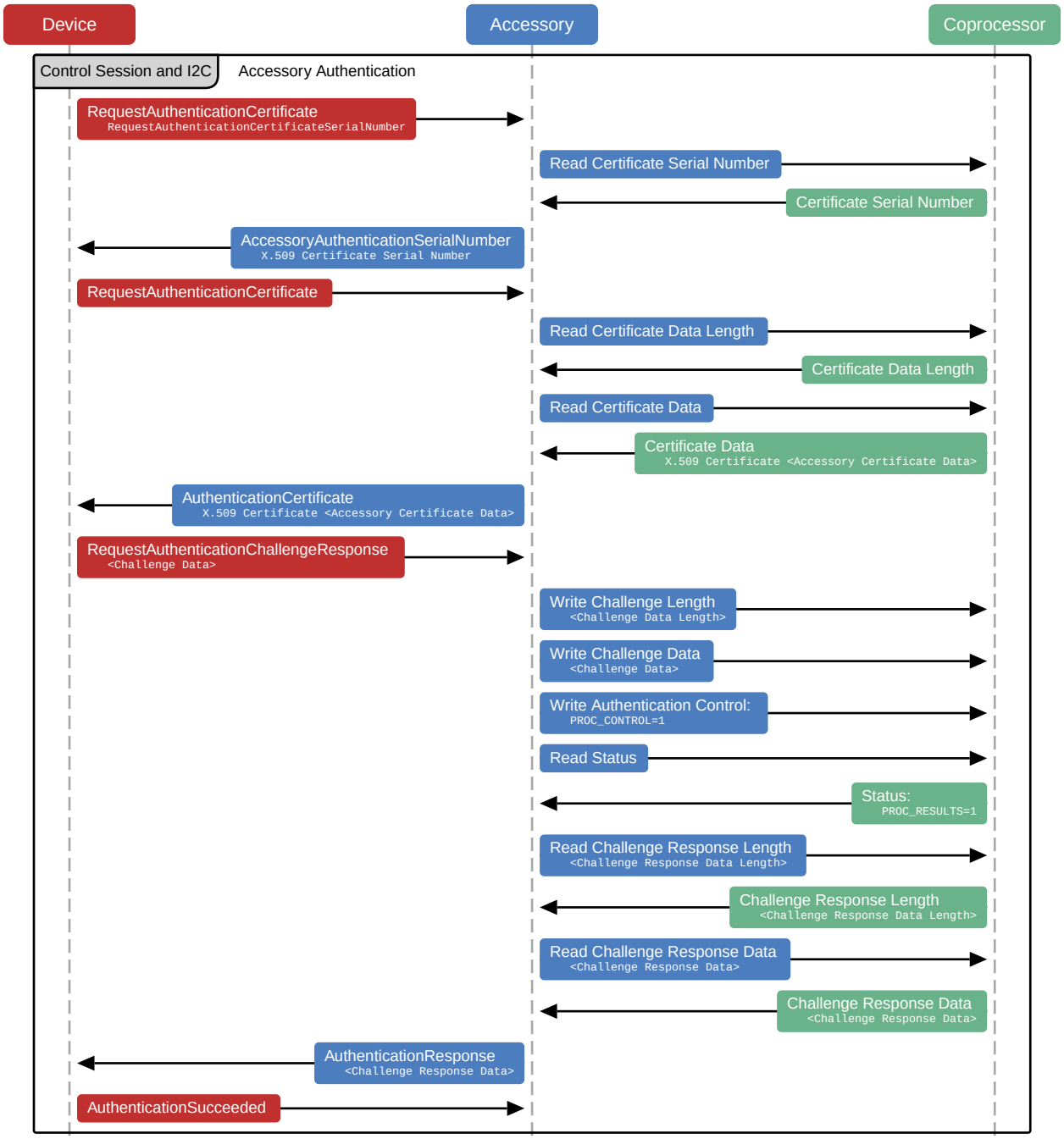
If the authentication response is correct, then the Apple device will send an AuthenticationSucceeded (page 806) message. The accessory must immediately proceed to Accessory Identification (page 265) unless it is using control session version 2 and has already identified (see Control Session (page 789)). With control session version 2, once the Apple device starts authentication, the accessory must complete the process within 15 seconds.

To demonstrate correct handling of the AuthenticationFailed (page 806) message during self-certification, accessories must send an empty or invalid certificate when the Apple Authentication Coprocessor is not present.

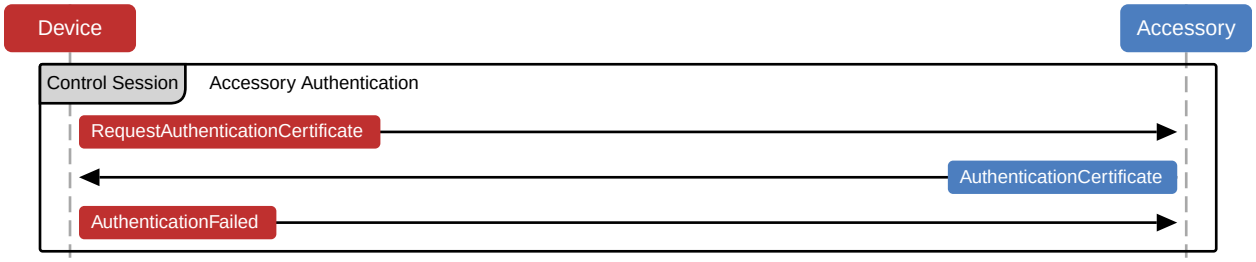
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

12.3 Accessory Authentication Examples

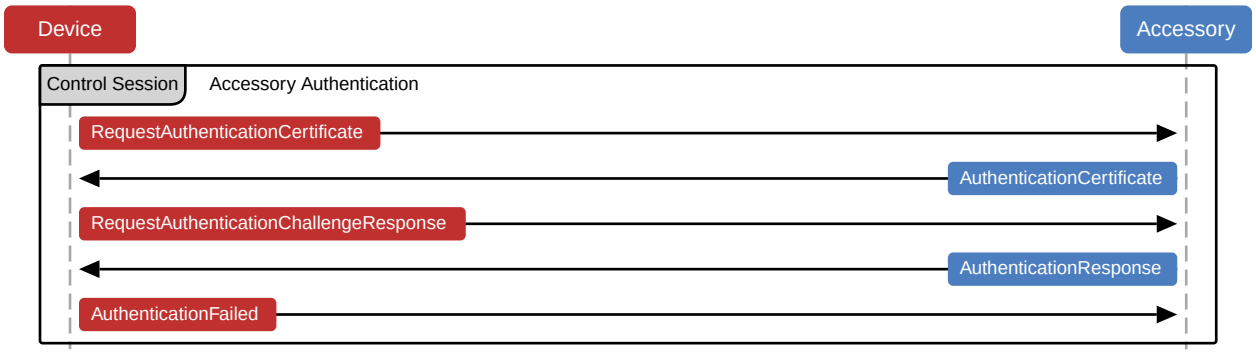
12.3.1 Typical Accessory Authentication over iAP2



12.3.2 Accessory Authentication over iAP2 Failure Due To Invalid Certificate



12.3.3 Accessory Authentication over iAP2 Failure Due To Invalid Response



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

13. Accessory Identification

All accessory interface features other than charging require that the identification feature be supported.

13.1 Accessory Identification Requirements

The accessory must have successfully completed Accessory Authentication (page 261) unless using control session version 2. See Control Session (page 789).

Control session version 2 allows the Apple device to choose the order of identification and authentication, so the accessory must be ready for either StartIdentification (page 807) or RequestAuthenticationCertificate (page 805) after successful link layer negotiation.

13.2 Accessory Identification Usage

All accessories that support the Accessory Identification feature via iAP2 must send or receive the following iAP2 control session message(s):

StartIdentification (page 807)

IdentificationInformation (page 807)

IdentificationAccepted (page 816)

IdentificationRejected (page 817)

CancelIdentification (page 819)

All accessories that support the Accessory Identification feature via iAP2 may also send or receive the following iAP2 control session message(s):

IdentificationInformationUpdate (page 819)

Identification is always initiated with the transmission of a StartIdentification (page 807) message from the Apple device to the accessory. The accessory must respond with a IdentificationInformation (page 807) message. The IdentificationInformation (page 807) message is evaluated by the Apple device to determine whether all of the requested iAP2 connection features are supported. If the parameters are valid and the Apple device can support all of the requested messages, it will send a IdentificationAccepted (page 816) message to the accessory.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Otherwise, a IdentificationRejected (page 817) message is sent detailing which invalid parameters and unsupported messages were present in the previous IdentificationInformation (page 807) message. Note that it is possible for the list of unsupported messages to be empty; that is an indication of incorrect accessory implementation, rather than unsupported features on a specific device.

Accessory Test System (ATS) output and Apple device console logs (see 'Getting Crash Logs and Console Output' at http://developer.apple.com/library/ios/#qa/qa1747/ will also provide additional supporting information which is useful for accessory development. Apple strongly recommends that accessory developers pay close attention to both ATS and the device logs.

If the identification information has been rejected, the accessory must send another IdentificationInformation (page 807) message with different parameters or abort the identification process using CancelIdentification (page 819). Cancellation of the identification process will cause the Apple device to inform the user that the accessory is not supported.

Failure of the accessory to send IdentificationInformation (page 807) within 1 second of either StartIdentification (page 807) or IdentificationRejected (page 817) is grounds for failing to pass self certification.

Once IdentificationAccepted (page 816) is received, the accessory must not send any more identification messages other than IdentificationInformationUpdate (page 819).

It is possible for an accessory to update its identification information after IdentificationAccepted (page 816) by sending an IdentificationInformationUpdate (page 819) message. For example, the user may change the accessory's active language. When that occurs, the accessory must send an IdentificationInformationUpdate (page 819) message containing both the new active language and accessory information strings in the new language.

If the identification process fails or the accessory cancels identification, the Apple device may choose to restart the identification process or terminate the iAP2 link.

13.2.1 Accessory Identification of Manufacturing Information

Accessories must provide values for the following IdentificationInformation (page 807) message parameters that match the accessory's marking and packaging:

- `Name`
- `ModelIdentifier`
- `Manufacturer`
- `SerialNumber`

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Additionally, the `FirmwareVersion` and `HardwareVersion` parameter values must uniquely reflect the current revision of the accessory's firmware and hardware respectively.

Accessories must not provide blank or generic string values (such as those provided by a reference design or component vendor) for any manufacturing information parameters.

The SerialNumber parameter should be unique for each accessory (i.e. serialized); Apple devices may use this to differentiate multiple accessories of the same type.

13.2.2 Accessory Identification of Sent/Received iAP2 Control Session Messages

The `MessagesSentByAccessory` and `MessagesReceivedFromDevice` parameters in IdentificationInformation (page 807) are critical to accessory identification. Accessories must exhaustively enumerate all iAP2 control session messages that they can send or receive. During the self certification process, the accessory must be used in a manner that demonstrates correct implementation of all declared messages.

There is only one exception to this requirement. Messages associated with the Authentication and Identification features do not need to be in the `MessagesSentByAccessory` or `MessagesReceivedFromDevice` lists, as all accessories that implement iAP2 must support those features.

13.2.3 Accessory Identification of Power Capabilities

All accessories must provide values for the `PowerProvidingCapability` and `MaximumCurrentDrawnFromDevice` parameters in their IdentificationInformation (page 807) messages.

If an accessory does not provide power to the Apple device, `PowerProvidingCapability` must be set to `None`.

If an accessory does not draw power from the Apple device, `MaximumCurrentDrawnFromDevice` must be set to `0`.

Otherwise, the accessory must set those parameters as required by Power (page 660).

13.2.4 Accessory Identification of iAP2 Transport Components

Every accessory that implements iAP2 must identify one and only one transport component that supports an iAP2 connection. That transport component must contain a `TransportSupportsiAP2Connection` parameter. Depending on the type of transport, additional identifying information, such as a Media Access Control (MAC) address, may be required. The iAP2 connection must be implemented using the identified transport component.

Some accessory interface features require a specific transport component to be implemented in order to function. This requirement is independent of whether or not an iAP2 connection is implemented on that transport. For example, an accessory must identify a `USBDeviceModeTransportComponent` for Digital Audio

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

over the USB Device Mode transport. Every identified transport component must have a unique identifier and human-readable name that reflects the intended purpose of the component, such as 'iAP2' or 'Digital Audio Speakers'.

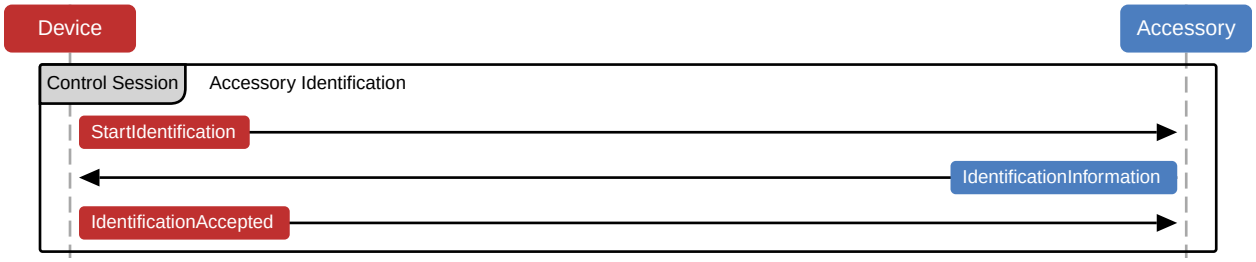
All transport components must have unique identifiers.

13.2.5 Additional Accessory Identification Parameters

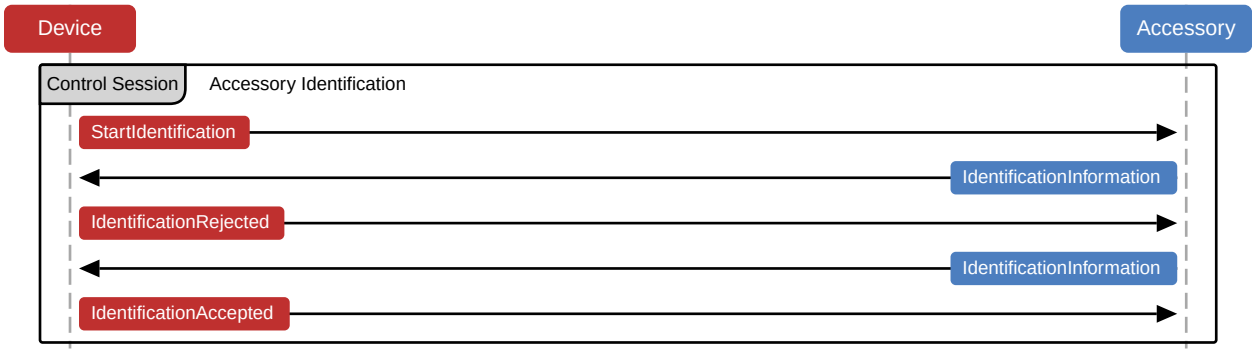
Several accessory interface features require the accessory to supply additional identification parameters. See the appropriate sections in the specification for details.

13.3 Accessory Identification Examples

13.3.1 Typical Accessory Identification

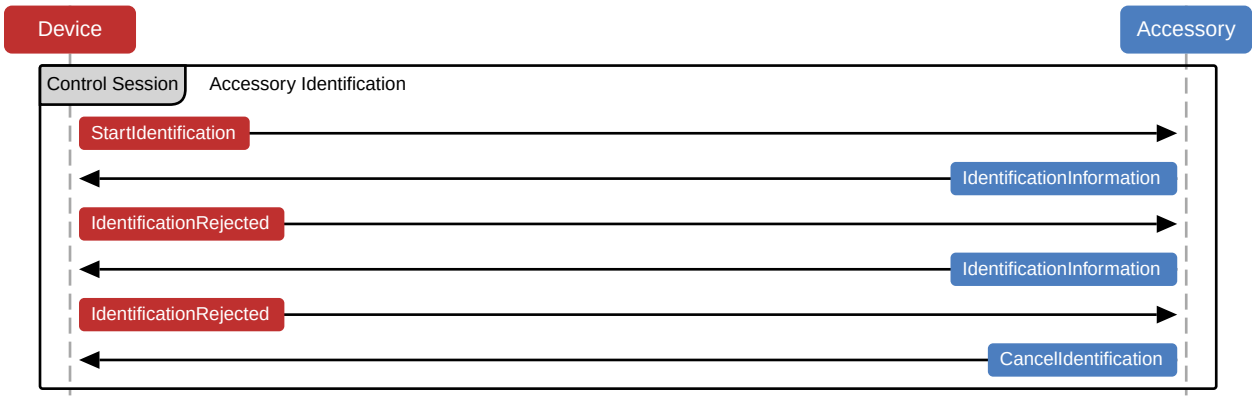


13.3.2 Successful Accessory Identification With Two Tries

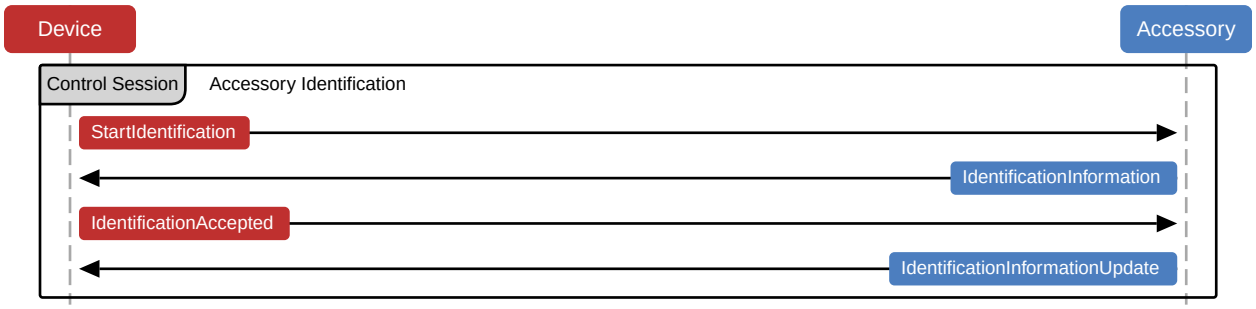


2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

13.3.3 Unsuccessful Accessory Identification After Two Tries



13.3.4 Accessory Identification With Information Update



13.4 Test Procedures

Equipment Required

- Latest revision of ATS software
- ATS hardware

1. Verify that the accessory identifies with all of the required accessory information fields in the ATS iAP Summary.
   a. Connect the accessory and the Apple device to ATS.
   b. View iAP Summary page.
   c. Confirm required fields (Name, brand, etc.) are correctly filled in (no "filler" entries, such as ACME, 1234, etc.)
2. Verify that the accessory supports and is connected to a non-IDPS supported device such as iPod Classic, that it properly falls back to Identify Device Lingoes when attached to the non-IDPS device.
   a. Connect the accessory and the non-IDPS Apple device to ATS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

b. View iAP summary page.

c. Verify Identification Method is IDL.

3. Verify accessory does not claim a lingo or message that is does not use.

a. Consult accessory Product Plan for claimed messages/lingoes.

b. Compare to Summary Page in ATS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14. AirPlay

14.1 Introduction

AirPlay lets a user wirelessly stream audio from their Apple device, iTunes or OS X computer to any AirPlay-enabled accessory connected to the network.

With an AirPlay-enabled accessory, users can:

- stream audio to any room in the house.
- choose which AirPlay accessories they want to play audio on.
- stream audio to a variety of AirPlay accessories.

AirPlay includes a number of Apple technologies to allow for a complete music streaming experience for the user.

14.2 User requirements for using AirPlay

14.2.1 Hardware

AirPlay requires the following:

- A device running iOS version 4.3.3 or greater, iTunes version 10.2.2 or greater, or OS 10.8.0 or greater to act as an audio source.
- An AirPlay accessory with built-in or external speakers.

14.2.2 Network

AirPlay requires a wired or Wi-Fi TCP/IP network connection between the source and accessory. The AirPlay-enabled accessory must reside on both the same local network and subnet as the AirPlay source device.

Wired networks require a minimum of 100Base-T connection.

Wireless networks require an access point compatible with any of:

- 802.11b/g

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- 802.11n
- 802.11ac

14.3 AirPlay Accessory Requirements

All AirPlay accessories must incorporate the following features and requirements.

14.3.1 Accessory Features

AirPlay accessories must include the following:

- A network connection.
- A built-in speaker or audio output capability.
- An accessory status indicator.
- A network status indicator.
- The ability for the user to perform a firmware update.

AirPlay accessories must not:

- Repeat or rebroadcast the AirPlay audio stream to any devices or components, other than to:
  - Speakers attached to the AirPlay-enabled accessory via internal or external wiring, PCB traces, etc.
  - Speakers wirelessly attached to the AirPlay-enabled accessory in a manner designed for single-zone use only (i.e. multi-zone support is not permitted).
- Background or render the audio in an inaudible way while the device is in an alternate mode.

AirPlay accessories are encouraged not to use proprietary Wi-Fi technologies that overlap with the international Wi-Fi spectrum.

14.3.2 Network Requirements

All AirPlay accessories must support a wired or Wi-Fi network connection and Bonjour, Apple's service discovery protocol. Devices must also support changing the Bonjour name to a user defined value. The default name of all devices must be unique out of the box.

AirPlay accessories that support Wi-Fi networks must also:

- support 802.11b/g, 802.11n, or 802.11ac.
- complete Wi-Fi certification (see Certification Requirements (page 273)).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.3.3 Implementation Requirements

All AirPlay accessories must meet all applicable requirements in this specification and must incorporate the Apple Authentication Coprocessor (see Apple Authentication Coprocessor 2.0C (page 66)).

14.3.4 Certification Requirements

AirPlay accessories must complete the following certification requirements:

- Wi-Fi Alliance "Wi-Fi certified" program.
- Wi-Fi Multimedia (WMM) certification.
- MFi program certification.

NOTE: Pre-existing MFi program certification on an accessory is not sufficient for AirPlay accessory certification.

14.4 AirPlay Setup Experience

Getting started

1. Install the latest version of iTunes on a Mac or PC.
2. Connect all necessary cables to the AirPlay accessory.
3. Connect the AirPlay accessory to power.
4. Connect the AirPlay accessory to the network (see Network configuration).

Network configuration Joining the network is the first touch point the consumer will have with the AirPlay Accessory. As such, it is important that thought is put into the best way to get the Accessory onto the network.

AirPlay Accessories must minimally support the use of a Bonjour HTTP record and a web based configuration page that is compatible with the Safari browser on Mac, PC, iPhone, iPod touch and iPad. Any additional methodologies to improve the user experience of joining the network are encouraged. Additional methodologies must support Mac OS and iOS, though other operating systems are allowed.

If the AirPlay Accessory includes a Wi-Fi interface, it must be configurable via that interface. Specifically, when configuring using a wireless methodology it is required that the AirPlay accessory operate as an Access Point in its unconfigured state instead of in Ad-Hoc mode for greater ease of use and Accessory compatibility. While in configuration mode, the Accessory must also expose Apple's Wi-Fi Accessory Configuration (WAC) feature to the user according to that feature's functional requirements. See Wi-Fi Accessory Configuration (page 737).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Both configuration options are required to allow configuration of the AirPlay Accessory via WAC on iOS and OS X devices that support that feature, and via a web interface for older iOS, OS X and Windows Accessories that do not support WAC.

iOS 5 onward provides the ability for the AirPlay accessory to gain access to the user's current Wi-Fi network credentials through the iPod Accessory Protocol (iAP). The use of this feature is detailed in Wi-Fi Information Sharing (page 761). Be aware that the use of this feature requires the AirPlay accessory to include a button or other manual way of triggering the feature. Automatic triggers are not allowed.

When configuring the Accessory on any network interface, it is required that the AirPlay accessory be pre-configured to use DHCP but allow for manual IP address assignment on the setup page.

14.4.1 Network configuration example for Wi-Fi networks

1. Turn on or reset the AirPlay accessory.
    - The AirPlay accessory creates a default network with a unique network name such as "PRODUCT_1234".
2. From an iOS 7.0 or later device's Wi-Fi Settings page, select the device.
3. Verify that the target network and accessory name are assigned as desired.
4. Optionally assign a PIN lock to the accessory.
5. Click "Next".
6. The accessory will confirm its connection to the desired network.

14.4.2 Network configuration example for wired networks

1. Connect the AirPlay accessory to the local network router using an Ethernet cable.
2. The AirPlay accessory joins the network, obtains an IP address using DHCP, and is ready to be seen from the iTunes user interface.

14.5 Power States and Accessory Availability

While the audio stream session is active, the AirPlay accessory must remain in a power state that allows it to play audio and receive TCP and UDP packets. When the AirPlay accessory is in a low-power state, it must continue to maintain its Bonjour advertisements, respond to Bonjour queries, and accept TCP connection requests to start an audio stream session.

If the AirPlay accessory is in a power state that is too low to allow full compliance with the above requirements, it must cleanly deregister its Bonjour record and power down its Wi-Fi radio. No other network functions are permitted in such a low power state (HTTP web serving, etc.).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.6 Bonjour

The AirPlay accessory must only advertise its Bonjour service when that service is actively available. This means that a device which is in a mode where it cannot accept an AirPlay stream, it must not advertise its service. Additionally, when a device is powered down it must deregister its Bonjour service on the network. By broadcasting its Bonjour service, the AirPlay accessory is indicating it is actively prepared and capable of accepting an AirPlay stream at that time.

Model Name

All Accessories must have a vendor defined model name. This name should be unique within the vendors' product line.

- `Field: am=<Vendor model name>`
- `example: am=AppleTV2,1`

Metadata Support

AirPlay accessories that choose to implement metadata must declare their support in their Bonjour record. Accessories must not declare any metadata support unless they plan to actively present that data to the end user.

- field: md=<Metadata Supported>
  - `0 Text; If set, textual metadata supported.`
  - `1 Artwork; If set, artwork metadata supported.`
  - `2 Progress; If set, progress metadata supported.`
- example: md=0,1,2

Firmware Version

All Accessories must indicate the firmware version that the AirPlay accessory is based upon.

- `field: fv=<AirPlay Firmware Version>.<MCU Firmware Version>.<Vendor Custom>`

Accessories must declare the version of the AirPlay code base that their technology provider has supplied in the <AirPlay Firmware Version> field.

For accessories based on the POSIX source release, this field can be found in the third field of the release number of the POSIX source drop, and is composed of the letter "p" followed by a number. For example, for the POSIX source drop release "AirPlay Audio POSIX Receiver 190.9.p7", the value of the <AirPlay Firmware Version> field is "p7". The official POSIX source releases are:

- 190.9.p6

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- 190.9.p7
- 211.1.p8

For accessories based on the Microchip DM8XX module, this field value will be provided by the technology provider and is composed of the letter "s" followed by a number.

If an Accessory has a separate MCU, they must declare the version of the firmware in the <MCU Firmware Version> field. If no MCU is used, this field should be filled with a 0. The Accessory may include whatever other firmware versioning information desired in the <Vendor Custom> field. If no custom data is used, this field should be filled with a 0.

```
example: fv=p101.0515.0
```

14.7 Web Interface

AirPlay Accessories must support the use of a Bonjour HTTP record and a web based configuration page that is compatible with the Safari browser on Mac, PC, iPhone, and iPad. Any additional methodologies to configure user settings are allowed. The required list of features accessible through this web configuration page are:

- **Friendly Product Name** User assignable name.
- **Wi-Fi Network** Selection of available networks.
- **Network Passphrase** Input of the associated network passphrase.
- **Product PIN** Set a password to prevent others from streaming to the AirPlay accessory.
- **DHCP Setting** Ability to turn DHCP on/off.
- **Manual IP Settings** Ability to manually configure the network interface.
- **Firmware Update** Ability to initiate a firmware update for the accessory.

14.8 Firmware Update

All devices are required to be firmware updatable in the field. All firmware that affects the AirPlay functionality in an Accessory must be updatable. This includes but is not limited to: volume control, transport controls, Bonjour, and input switching.

In the case where an AirPlay accessory's firmware update is not successful, the device must not advertise a Bonjour service unless it is in a state where that service can be fully supported.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

AirPlay accessories must minimally be able to upgrade from a locally stored file on the customer's computer through a web based configuration page that is compatible with the Safari browser on both Mac and PC. AirPlay accessories cannot require WAN connectivity for this firmware upgrade method. Accessories can optionally support other upgrade methodologies as long as they do not require specialized tools. Additional methodologies must include Mac OS or iOS support, though other operating systems are allowed. AirPlay devices which choose to use a computer application to update their firmware must minimally include support for Mac OS.

14.9 AirPlay Accessory User Interface

14.9.1 Required Accessory user interface

Accessory UI - All Accessories

Accessories must minimally show a unique status indication for an active network connection, a network connection problem, and a critical firmware issue. All other status indications are optional. See Table 14-1 (page 277).

Table 14-1 UI Status Indicators

	Status Indicator
Required	Network connection
Required	Network problem
Required	Critical Firmware Problem
Optional	Source input label
Optional	AirPlay stream in progress

Accessory UI - GUI Enabled Accessories

Accessories which use a GUI are highly recommended to include the AirPlay logo as defined in the document: AirPlay Logo Guidelines for Made for iPod Licensees. Accessories may not use any imagery other than the AirPlay logo to represent the AirPlay feature. Text indicators are considered to be valid alternatives for the status indicators required of all AirPlay Accessories. See Table 14-2 (page 278).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 14-2 Suggested Text Indicators

Condition	Message
Network connection	Successfully connected to [NETWORK NAME].
Network problem	Cannot connect to [NETWORK NAME]. Please try again.
Critical firmware problem	Please reset device by [RESET METHOD].
Source input label	AirPlay

14.9.2 Metadata Presentation

The latter are optional for Accessories' User Interface

Metadata requirements:

AirPlay accessories with presentation capability (whether on an attached display, a remote control with a display, or an app, web, or voice interface) may choose to present information about the currently-playing song. When choosing to support metadata, the device must minimally support the Song title. The user must also be given the option to cycle through Song title, Artist name, and Album name. Metadata is transmitted using UTF-8 and the presentation mechanism must be able to faithfully represent all UTF-8 characters transmitted.

It is highly recommended that the accessory support both the Song title and Artist name as a minimum on all visible user interfaces. The AirPlay accessory is required to present any metadata it chooses to implement faithfully. All devices with limited display capabilities must scroll a piece of metadata that is too long for its display, ensuring the user can see the entirety of that piece of metadata.

Devices which choose to present metadata for sources other than AirPlay must include metadata display for the AirPlay source as well.

All metadata not listed as required can be implemented as an option in the AirPlay accessory.

Table 14-3 Metadata field

Metadata Field
Album name
Album artwork
Artist name
Elapsed time

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Metadata Field
Song title
Total time

Album artwork display requirements:

Album artwork is only permitted for display on full color screens. The Accessory must be able to render JPEG and PNG images up to at least 512kB in size. If an AirPlay stream reports that it has no associated Album artwork or the Album artwork transmitted is not in a file format the AirPlay accessory can render, the accessory must render a placeholder image in place of the missing album artwork. The placeholder image used is at the discretion of the AirPlay accessory.

iTunes does not cross convert Album artwork so it is recommended that accessories support as many image formats as possible. Common image file formats are: JPEG, PNG, GIF, BMP, JPEG2000, TIFF, and PICT.

14.9.3 Transport controls

Transport controls are optional for AirPlay accessories. AirPlay accessories choosing to implement transport controls must minimally implement Play/Pause toggle (or Play & Pause individually), Next track, and Previous track.

Play/Pause toggle

Toggles the playback state of the currently playing track in iTunes. Other endpoints in the network will retain their ON/OFF settings and audio stream state as previously set.

Play

Starts playback of the currently selected track in iTunes. The AirPlay accessory that initiated the play event must be enabled as a playback endpoint (if not previously enabled in iTunes the command will have no effect). Other endpoints in the network will retain their ON/OFF settings as previously set.

Pause

Pauses the playback of the currently playing track in iTunes. Other endpoints in the network will retain their ON/OFF settings and audio stream state as previously set.

Stop

Stops the playback of the currently playing track in iTunes. If the Apple device is in the middle of playback, the song time will reset to the start of the track. Other endpoints in the network will retain their ON/OFF settings and audio stream state as previously set.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Next track/Previous track

Skips to the next/previous track in the current playlist of iTunes. Has no effect if there is not a currently selected "now playing" track in iTunes. Previous track command automatically skips to the start of the song if more than 3 seconds of the currently playing track has passed.

14.9.4 Shuffle and Repeat State Controls

Shuffle and repeat controls are optional for AirPlay accessories. AirPlay accessories choosing to implement either shuffle or repeat must additionally implement the other command.

Shuffle toggle

Advances the iTunes shuffle setting from it's current state to the next in the progression (... shuffle all songs, shuffle off ...)

Repeat toggle

Advances the iTunes repeat setting from it's current state to the next in the progression (... repeat all songs, repeat single song, repeat off ...)

14.9.5 Volume Controls

Volume controls are optional for AirPlay accessories. AirPlay accessories choosing to implement volume controls in their Accessory must implement both local and remote volume controls and volume state must be communicated both directions. The device must respect the volume setting as commanded by the AirPlay stream and any local volume control should be applied immediately upon the AirPlay accessory and the new volume level reported back to the AirPlay source.

Volume is communicated as a floating-point dB attenuation value where 0.0 is full volume and -144.0 is completely muted. The practical volume range utilized for the AirPlay stream has a basis of -30dB to 0dB with a special cased mute level of -144dB (linear volume of 0) in order to avoid infinities. The following equations can be used to convert between a dB value and a linear volume:

- `linear volume = pow( 10, dB / 20 )`
- `dB value = 20 * log10( linear volume )`

The AirPlay accessory must map the communicated volume range to its own full scale range. Additionally, the AirPlay accessory must implement volume control at the final gain stage available to it in order to minimize audio quality loss through the system.

Local volume control

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Controls the output volume of the local AirPlay accessory immediately upon action. The AirPlay accessory's new volume level must be reported back to the AirPlay server immediately upon setting the system volume.

Remote volume control

Controls the output volume of the local AirPlay accessory from a remote location. Upon receiving a remote volume setting the AirPlay accessory should send the actual volume set to the server.

Mute volume control

All accessories are required to support a true, no volume Mute audio level communicated by the AirPlay streaming protocol by a value of -144dB.

Accessory manufacturers may decide to implement a Mute function within their Accessory. The operation of the mute function should bring the volume level down to -144dB when engaged and back to the pre-mute level when disengaged. It is the responsibility of the Accessory to remember the pre-mute level when entering the mute function and to ensure both that it returns to this level and that it reports the proper level to the AirPlay server when disengaging.

Further, any volume up or down commands that the Accessory receives from the user should disengage the mute function and then act upon the pre-mute level. The exception to this rule is when a volume level is commanded by the AirPlay stream itself. In this case the Accessory should cancel the mute function but set the return volume to that which has been dictated by the AirPlay stream.

14.9.6 Accessories with multiple inputs

14.9.6.1 Playback Prevention

If the AirPlay accessory receives an audio stream, it must automatically switch its audio input source to the audio stream. The AirPlay accessory is required to always be available to accept an AirPlay stream as long as it is in the power mode to do so. This means that an explicit user action selecting the speaker as an output must always interrupt the current activity and start the AirPlay activity.

Allow

The device allows AirPlay audio playback to begin. This is the default state of an AirPlay accessory.

Prevent

The device prevents AirPlay audio playback from beginning. This state is only allowed during an accessory source switch. The state must be set back to Allow immediately upon completing the source switch.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.9.6.2 Audio Source Switching

AirPlay accessories are not required to have multiple modes or inputs other than AirPlay. AirPlay accessories choosing to implement multiple inputs or modes must automatically switch audio input source to the AirPlay stream if the device receives an audio stream. If an audio stream session is active and the audio input source is changed to an alternate source, the AirPlay accessory must send a deselection of the AirPlay accessory output path to iTunes. Similarly, if an audio stream session is inactive and the audio input source is changed to the AirPlay source, the AirPlay accessory must send a selection of the AirPlay accessory output path to iTunes. Audio source switching away from the AirPlay input must be triggered by an explicit user interaction. Actions such as docking an iPod should not be considered explicit whereas pressing play on that iPod after it has been docked is an explicit action.

Available

The device is on the AirPlay input.

Busy

The device is on any source other than AirPlay.

14.10 User Documentation

User documentation, both Quick Start Guides and User Manuals, must include methodologies for using Mac OS to set up and maintain the AirPlay accessory. It is recommended to include instructions on how to use iOS to set up and maintain the AirPlay accessory.

14.11 Special Considerations for Adapter Accessories

- Only proprietary connectors must be used.
- All points listed in the above feature specification must be strictly complied with for the Accessory as a whole.
- The UI for the AirPlay function must be of equivalent quality and detail as other functions included in the Accessory.
    - All AirPlay setup must be able to be adjusted through the Accessory UI as equivalent to other Accessory features.
    - The AirPlay feature must be clearly labeled in the Accessory UI when in use.
- Firmware update for the Accessory must be capable for updating the full system image including the UI.
- Any adapter must be included in-box with the Accessory and must not be packaged or sold separately.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12 Accessory Compliance Test Plan for Audio Streaming Devices

The purpose of this test plan is to specify the testing that vendors must perform to verify that their Accessories conform to the AirPlay specification provided by Apple Inc. Web Based Firmware Upgrade Test (page 287) describes tests that must be run and successfully completed for all AirPlay Accessories. In addition, Web Based Firmware Upgrade Test (page 287) also contains test that are optional but are required to be tested if implemented.

Table 14-4 Terms and Definitions

Term	Definition
IE	Information Element.
AP	Access Point.
AU	AirPort Utility.
STA	Non-AP 802.11 station.
BSSID	Basic Service Set Identifier.
SSID	Service Set Identifier.
MAC	Media Access Control.
BSS	Basic Service Set.
ESS	Extended Service Set.
DS	Distribution System.
DUT	Device Under Test.
URL	Uniform Resource Locator.
PHY	Physical Layer (802.11a, b, g, n, ac).
OOB	Out of the Box.

AirPlay is an evolution of Apple's AirTunes technology, allowing for music to be played seamlessly over any IP connected device. As such, all network interfaces are acceptable to run AirPlay, however it is understood that all AirPlay DUTs will not implement all of the possible network interfaces and thus only the network interface that are implemented will need to be tested.

While not necessarily required to run AirPlay as a feature, the following are **minimum requirements** to complete the test suites that are included in this test plan.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- MacBook or MacBook Pro running OS X 10.8.2 or greater
- iTunes 10.7 or greater
- Apple AirPort Extreme 802.11n (5th Generation) Base Station running 7.6.4 or greater
- Apple AirPort Express 802.11n (2nd Generation) running 7.6.4 or greater
- Apple AirPort Utility 6.3.1 or greater
- Apple iPad/iPhone/iPod Touch running iOS 6.0.1 or greater (iPad is recommended)

Test Tools and Methodology for Safari

Some tests will require the use of Safari. These tests should use the latest version of Safari available for the test platform.

Before performing these tests, ensure that the 'Include Bonjour in the Bookmarks menu' option is enabled in the Safari Preferences pane, under the 'Advanced' tab.

When a test refers to accessing the Bonjour list, the list may be found in the Bookmarks menu bar, under the 'Bonjour' menu bar item.

Test Tools and Methodology for dns-sd

Some tests require the ability to query specific Bonjour records. In order to get the Bonjour information, some tests will require the use of dns-sd, a standard tool provided in OS X. Please refer to the Unix man page provided with dns-sd for usage details not provided in this document.

Test Tools and Methodology for iperf

iPerf is a network bandwidth performance tool used for testing both TCP and UDP performance with various options to test network quality metrics. All necessary options for operation for this test plan will be provided for each test case that iPerf is necessary.

The following test beds are designed and specified for each test case in this test plan. While not all DUTs will need to use every test bed, the minimum requirement for every test bed is a wired OS X computer running the latest version of iTunes and a simultaneous dual band Apple Base Station, running the latest firmware.

All wired performance tests must be conducted from host machine to base station, to DUT.

All wireless performance tests must be conducted on an Apple approved rate vs. range test bed. The test bed may be line of sight or non-line of sight but must be conducted in a clean RF environment as confirmed by a spectrum analyzer. At Apple, these tests are conducted in an office environment with walls and hallways interfering with direct line of sight between the units. The DUT and AP are positioned on the same vertical

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

plane, approximately 1m off the ground. The DUT is rotated about its vertical axis at a rate of 1 RPM during iPerf testing. Tests are required on channels 1 and 11 in the 2.4 GHz band and channels 36 and 161 if the 5 GHz band is supported.

Wired and wireless non-performance tests do not have any particular topology or positioning requirement beyond that required to support connectivity.

The following test beds will be used as references for test environments.

Figure 14-1 Test bed 1

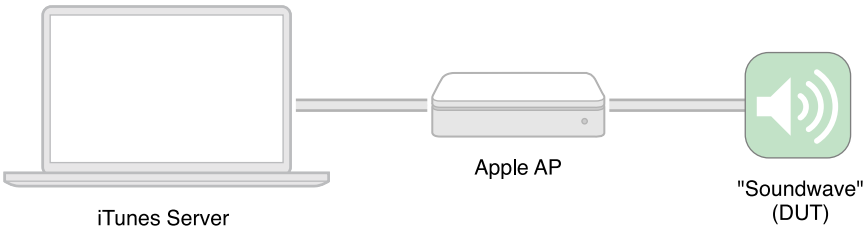


Figure 14-2 Test bed 2

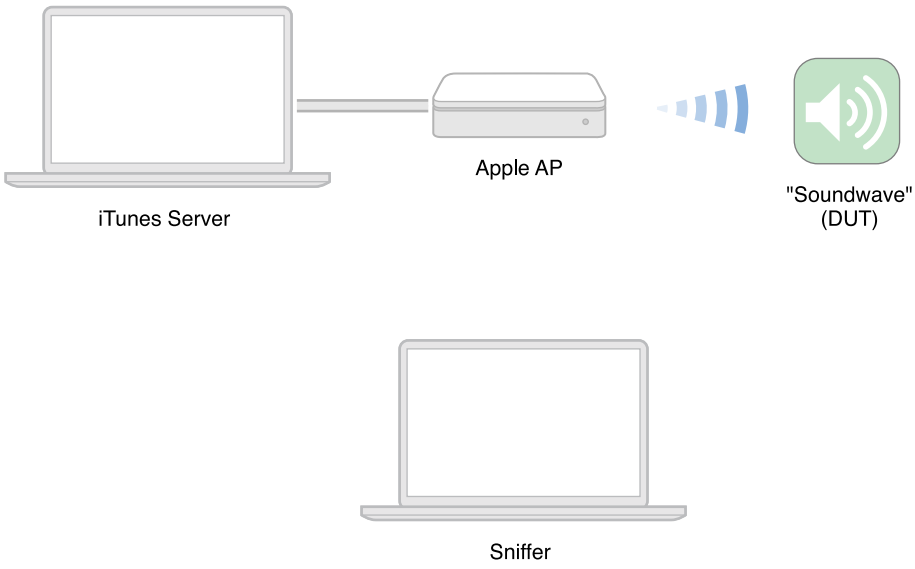


Figure 14-3 Test bed 3

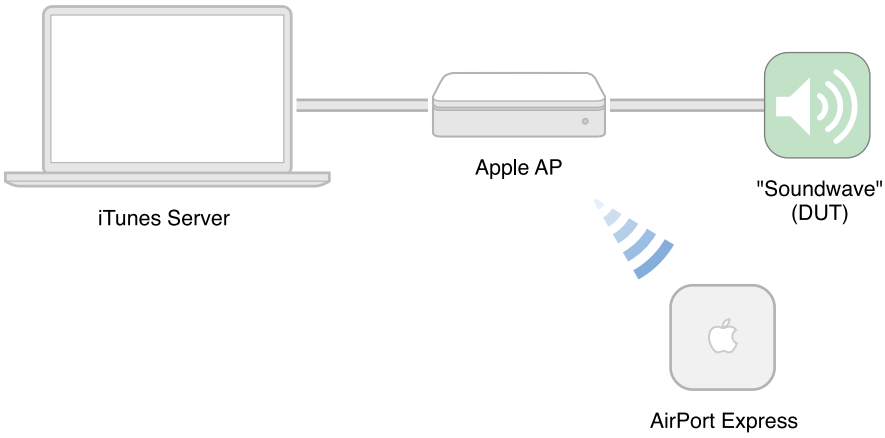


Figure 14-4 Test bed 4

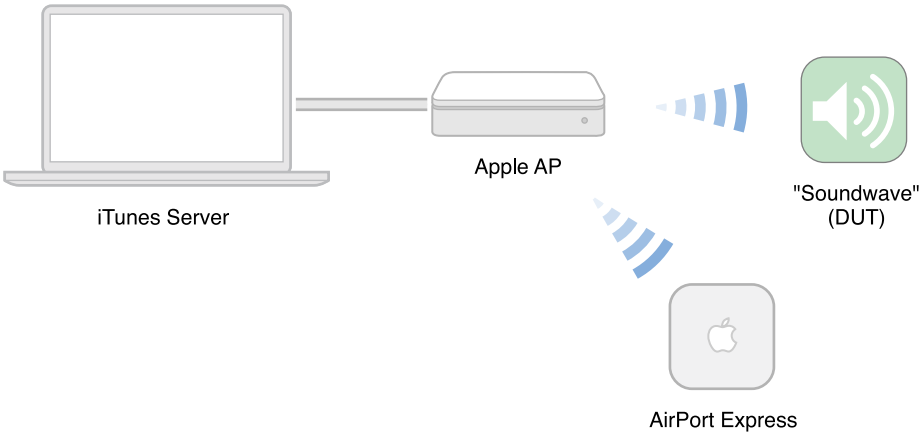
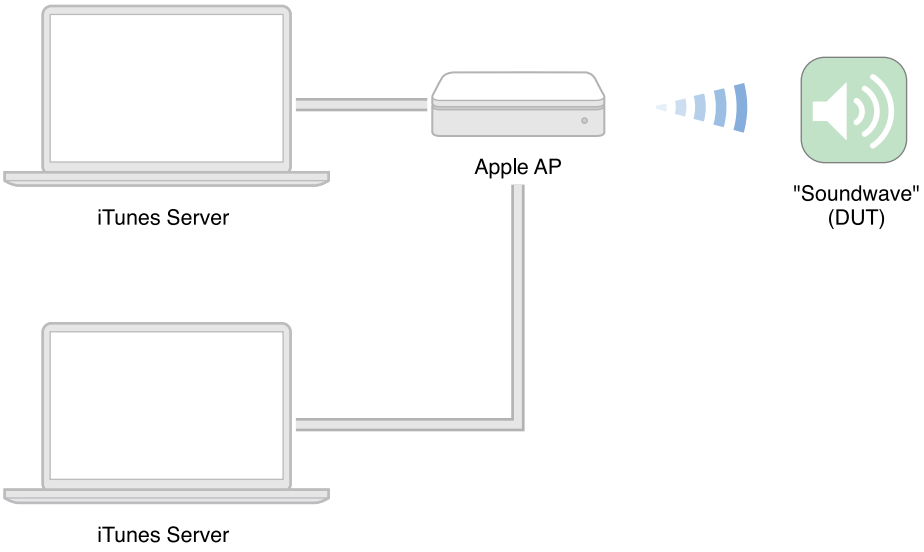


Figure 14-5 Test bed 5



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 14-6 Test bed 6



The following test cases must be completed successfully to ensure compliance with the AirPlay specification and interoperability with Apple Base Station product lines. While it is possible to implement AirPlay without an Apple Base Station, all tests will be done with an Apple Base Station (AirPort Extreme or Time Capsule) product because certain features of the Base Station are required to complete all of the testing.

Customer accessible Accessory firmware upgrade is a required feature of AirPlay Audio devices. All supported firmware upgrade methodologies must be tested and verified as a part of AirPlay Accessory compliance testing.

14.12.1 Web Based Firmware Upgrade Test

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Conduct applicable tests from Link Layer Connection Auto Negotiation for Wired Network (page 288) and Wi-Fi (page 288).
3. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
4. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
5. From Safari, select the DUT from the Bonjour list.
6. The speaker should be able to be selected without and error, or else FAIL.
7. Follow the onscreen instructions to conduct a firmware update to the _telnet image provided.
8. The unit should firmware update and rejoin the network, or else FAIL.
9. Conduct tests from section PHY: Wired Performance (page 294) and PHY: Wireless Performance (100 ft) (page 294).
10. From Safari, select the DUT from the Bonjour list.
11. The speaker should be able to be selected without and error, or else FAIL.
12. Follow the onscreen instructions to conduct a firmware update to the _production image provided.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

13. The unit should firmware update and rejoin the network, or else FAIL.
14. Conduct remainder of test plan.

14.12.2 Generic Firmware Upgrade Test

Note: This test shall be repeated for all available upgrade methodologies available to the end user.

1. Power on the DUT.
2. Conduct applicable tests from sections Link Layer Connection Auto Negotiation for Wired Network (page 288) and Wi-Fi (page 288).
3. Use the web interface to upgrade firmware to debug image provided.
4. Conduct tests from section PHY: Wired Performance (page 294) and PHY: Wireless Performance (100 ft) (page 294).
5. Use the web interface to upgrade firmware to production image provided.
6. Conduct remainder of test plan.

Network connectivity is a requirement for AirPlay and may be implemented either via Ethernet (802.3) or Wi-Fi (802.11). Only the implemented features are required to be tested; however if a DUT contains both interfaces, both must be tested.

14.12.3 Link Layer Connection Auto Negotiation for Wired Network

For the Test Environment, refer to Figure 14-1 (page 285).

If the DUT contains an Ethernet Port, the following test must be completed successfully.

1. Power on the DUT.
2. In the AirPort Utility go to Internet-> Internet connection and make sure the Connection Sharing is set to 'Share a public IP address'. Then set the Ethernet WAN Port to Automatic and click update.
3. Using an Ethernet wire, connect the DUT to the WAN port on the Base Station.
4. The green light on the port that the DUT is connected to should illuminate to indicate link and some indication on the DUT should also indicate a link. (This can be a light on the DUT's Ethernet port, like the Base Station or some other form).

14.12.4 Wi-Fi

If the DUT contains a Wi-Fi (802.11) interface, the following tests must be completed successfully. Each DUT that supports Wi-Fi must, at minimum, support 802.11g. All support PHYs must be tested.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12.4.1 Association Verification

AirPlay DUTs have two options of how to be configured based on whether there is an interface that allows for the SSID and security modes to be configured. Only run the applicable tests.

14.12.4.2 Out of the Box Association Verification

The following tests assume the device has had no configuration done to the DUT and is still in its factory default state.

14.12.4.3 Web Interface

For the Test Environment, refer to Figure 14-2 (page 285).

1. Power on the DUT.
2. Wait for DUT to complete booting.
3. The DUT should begin beaconing 802.11 beacons and advertising as an 802.11 network with a SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
4. Using the iTunes server, associate wirelessly to the DUT.
5. The iTunes server should be able to associate on the 2.4 GHz band using 802.11b/g with no security and receive a network that is on the same IPv4 network as the DUT. If the DUT does not have a DHCP server and instead would like to use the IPv4 link local IP space, this must be specified in manufacturer's documentation.
6. Using Safari on the iTunes server connect to the IP address that is provided by the manufacturer for the DUT.
7. The webpage should appear without problems and if a password is required, this also should be provided in the manufacturer's documentation.
8. Configure the DUT to connect to the Wi-Fi network on the Apple Base Station. Minimally their needs to be away to manually specify the SSID and security, however there can also be a way to select from a list obtained via scanning.
9. The DUT must have a way to manually enter the SSID for the Wi-Fi network and select a security mode. Optionally it can create a list (scan list) for the user pick their network from, if desired. Once the device is connected to the Base Station, there must be some indication on the DUT that device is connected.
10. Open the AU and verify the settings are correct.
11. In the Apple AU, go to Advanced->Logs and Statistics->Wireless Clients. In list box look for the MAC address that was collected from Step 1. Verify that Type that is listed is one of the following options: 802.11b/g, 802.11g, 802.11a, 802.11a/n, 802.11b/g/n, or 802.11g/n. If any other Type is present, FAIL.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

12. Next go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC address is not present, FAIL.
13. Open Safari on the iTunes Server and attempt to connect to the web server on the DUT with the address it just received.
14. Safari should be able to navigate to the web server, or else FAIL. All of the settings that were entered in previous steps should be reflected in the user interface.

14.12.4.4 User Configurable Interface

1. Power on the DUT. Note the MAC address of the DUT.
2. Configure the Base Station to its OOB configuration and note the SSID that is present. It should be something in the form of Apple Network XXXXXX.
3. Use the user interface on the DUT to connect to the Base Station Network with no security. There should be, at a minimum, someway to enter the SSID manually and the security mode. Optionally there can also be a way to select the SSID from a collected list.
4. The DUT must have a way to manually enter the SSID for the Wi-Fi network and select a security mode. Optionally it can create a list (scan list) for the user pick their network from, if desired. Once the device is connected to the Base Station, there must be some indication on the DUT that device is connected.
5. Open the AU and verify the settings are correct.
6. In the Apple AU, go to Advanced->Logs and Statistics->Wireless Clients. In list box look for the MAC address that was collected from Step 1. Verify that Type that is listed is one of the following options: 802.11 b/g, 802.11 g, 802.11 a, 802.11 a/n, 802.11 b/g/n, or 802.11 g/n. If any other Type is present, FAIL.
7. Next go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC address is not present, FAIL.

14.12.5 Security Mode Verification

All AirPlay DUTs must support no security (None) and WPA2-PSK (AES) at minimum. Other security modes are supported by the Apple Base Station and other 3rd party AP Accessories however only those two will be tested.

14.12.5.1 WPA2-PSK

For the Test Environment, refer to Figure 14-2 (page 285).

1. Power on the DUT. Note the MAC address of the DUT.
2. Configure the Base Station to its OOB configuration and note the SSID that is present. It should be something in the form of Apple Network XXXXXX. Configure the Base Station with WPA2-PSK Security with a Pass phrase of 12345678.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Use the user interface on the DUT to connect to the Base Station Network with WPA2-PSK with a Pass Phrase of 12345678. There should be, at a minimum, someway to enter the SSID manually and the security mode. Optionally there can also be a way to select the SSID from a collected list.
4. The DUT must have a way to manually enter the SSID for the Wi-Fi network and select a security mode. Optionally it can create a list (scan list) for the user pick their network from, if desired. Once the device is connected to the Base Station, the DUT must indicate it is connected based on Table 14-5 (page 291).
5. Open the AU and verify the settings are correct.
6. In the Apple AU, go to Advanced->Logs and Statistics->Wireless Clients. In list box look for the MAC address that was collected from Step 1. Verify that Type that is listed is one of the following options: 802.11 b/g, 802.11 g, 802.11 a, 802.11 a/n, 802.11 b/g/n, or 802.11 g/n. If any other Type is present, FAIL.
7. Next go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC address is not present, FAIL.

Table 14-5 Status Indicator

Action		Implementation	
		Accessory with Display	**Accessory without Display**
Required	Network Connection	Message on display	LED 1 illuminates SOLID, COLOR A
Required	Network Problem	Message on display	LED 1 illuminates BLINKING, COLOR B
Optional	Critical Firmware problem	Message on display	LED 1 illuminates BLINKING, COLOR C
Optional	Critical Firmware problem	Message on display	LED 1 illuminates BLINKING faster, COLOR B
Optional	Critical Firmware problem	Message on display	LED 2 illuminates BLINKING, COLOR B
Optional	Source input label	Message on display	N/A
Optional	AirTunes stream in progress	Logo or type on display	LED 2 illuminates SOLID, COLOR A
Optional	AirTunes stream in progress	Logo or type on display	LED 3 illuminates SOLID, COLOR A

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12.5.2 WPA/WPA2-PSK (TKIP Group Key Verification)

For the Test Environment, refer to Figure 14-2 (page 285).

1. Power on the DUT. Note the MAC address of the DUT.
2. Configure the Base Station to its OOB configuration and note the SSID that is present. It should be something in the form of Apple Network XXXXXX. Configure the Base Station with WPA/WPA2-PSK Security with a Pass phrase of 12345678.
3. Use the user interface on the DUT to connect to the Base Station Network with WPA2-PSK with a Pass Phrase of 12345678. There should be, at a minimum, someway to enter the SSID manually and the security mode. Optionally there can also be a way to select the SSID from a collected list.
4. The DUT must have a way to manually enter the SSID for the Wi-Fi network and select a security mode. Optionally it can create a list (scan list) for the user pick their network from, if desired. Once the device is connected to the Base Station, the DUT must indicate it is connected based on Table 14-5 (page 291).
5. Open the AU and verify the settings are correct.
6. In the Apple AU, go to Advanced->Logs and Statistics->Wireless Clients. In list box look for the MAC address that was collected from Step 1. Verify that Type that is listed is one of the following options: 802.11 b/g, 802.11 g, 802.11 a, 802.11 a/n, 802.11 b/g/n, or 802.11 g/n. If any other Type is present, FAIL.
7. Next go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC address is not present, FAIL.

14.12.5.3 WPA/WPA2-PSK (WPA-PSK (TKIP) Verification)

For the Test Environment, refer to Figure 14-2 (page 285).

This needs to be tested if the DUT can select WPA-Personal as a possible security mode.

1. Power on the DUT. Note the MAC address of the DUT.
2. Configure the Base Station to its OOB configuration and note the SSID that is present. It should be something in the form of Apple Network XXXXXX. Configure the Base Station with WPA/WPA2-PSK Security with a Pass phrase of 12345678.
3. Use the user interface on the DUT to connect to the Base Station Network with WPA-PSK with TKIP encryption with a Pass Phrase of 12345678. There should be, at a minimum, someway to enter the SSID manually and the security mode. Optionally there can also be a way to select the SSID from a collected list.
4. The DUT must have a way to manually enter the SSID for the Wi-Fi network and select a security mode. Optionally it can create a list (scan list) for the user pick their network from, if desired. Once the device is connected to the Base Station, the DUT must indicate it is connected based on Table 14-5 (page 291).
5. Open the AU and verify the settings are correct.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

6. In the Apple AU, go to Advanced->Logs and Statistics->Wireless Clients. In list box look for the MAC address that was collected from Step 1. Verify that Type that is listed is one of the following options: 802.11 b/g, 802.11 g, 802.11 a, 802.11 a/n, 802.11 b/g/n, or 802.11 g/n. If any other Type is present, FAIL.
7. Next go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC address is not present, FAIL.

14.12.6 IPv4 Connectivity

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

IP connectivity is required for all AirPlay devices, with a minimum of IPv4 with a DHCP client to be implemented. It is preferred that an IPv6 stack is also implemented; however this is not required, but highly recommended. If a dual stack (IPv4 and IPv6) is implemented, both will need to be verified. (These tests must be repeated for all interfaces, Ethernet and/or Wireless, if both are present.)

DHCP Tests

1. Power on the DUT. Note the MAC address of the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. Additionally, if Wireless is supported this test must be repeated for both No Security and WPA2-PSK.
3. Once connected to the network, open the Apple AU and go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC Address is not present, FAIL. If the MAC address is present, note the IP Address next to the MAC Address and note the 'Lease Time'.
4. On the iTunes server connected to the Base Station ping the IP address that was collected from Step 2-3.
5. If all the pings are successful, then PASS, else FAIL.
6. Reboot the DUT and wait for it to come back up.
7. Once connected to the network, open the Apple AU and go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC Address is not present, FAIL. If the MAC address is present, note the 'Lease Time' and make sure it has been updated than what was in Step 2-3.

14.12.7 IPv6 Connectivity

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

It is highly recommended that IPv6 is also implemented on all AirPlay DUTs and these tests must be run if IPv6 is implemented. (These tests must be repeated for all interfaces, Ethernet and/or Wireless, if both are present.)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Link Local Tests

1. Power on the DUT. Note the MAC address of the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. Additionally, if Wireless is supported this test must be repeated for both No Security and WPA2-PSK.
3. For IPv6 there may be no indication that the DUT has an IPv6 address if it is just using a link local address. If this is the case, the manufacturer must either provide the IPv6 address in the user interface or some documentation on how to obtain this IP address.
4. On the iTunes server connected to the Base Station ping, using the ping6 command, the IP address that was collected from Step 2-3.
5. If all the pings are successful, then PASS, else FAIL.

14.12.8 PHY: Wired Performance

For the Test Environment, refer to Figure 14-1 (page 285).

1. Power on the DUT.
2. Connect the DUT to the Base Station via a wired connection. The DUT should be set to its out of the box configuration.
3. On the DUT start iperf with the following command.

   ```
   iperf –s –u –i 1 –w 128k
   ```

4. On the iTunes server connected to the Base Station start iperf with the following command.

   ```
   iperf –c IP_of_DUT –u –b 10M –t 120 –i 1 –w 128k
   ```

5. The bandwidth output at the end of the test must be greater than 5Mb/s.

14.12.9 PHY: Wireless Performance (100 ft)

For the Test Environment, refer to Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the Base Station and make sure the DUT is 100ft from the base station and is associated at the DUT-s highest PHY rate. The base station must be configured to use WPA2-PSK with a passphrase of 12345678.
3. Verify in the Apple Airport Utility the DUT connected as an RSSI of > -75dB.
4. On the DUT start iperf with the following command.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
iperf –s –u –i 1 –w 128k
```

5. On the iTunes server connected to the Base Station start iperf with the following command.

   ```
   iperf –c IP_of_DUT –u –b 10M –t 120 –i 1 –w 128k
   ```

6. The bandwidth output at the end of the test must be greater than 4.9Mb/s with a packet loss < 0.1% on both channels 1 and 11

14.12.10 Bonjour TXT Records Tests

Some of the tests below check bonjour fields that contain information related to the POSIX Source release version numbers. The valid POSIX Source releases are:

- 190.9.p6
- 190.9.p7
- 211.1.p8

14.12.11 Bonjour ADD and RMV via browsing

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the iTunes server connected to the Base Station, issue the following command in a Terminal and leave it running.

   ```
   dns–sd –B _raop
   ```

4. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

5. A/R should have "Add" below it, if this is true, PASS.
6. Service Type should have "_raop._tcp." below it, if this is true, PASS.
7. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
8. Power off or shut down the DUT in a controlled manner. (Pulling the power out is NOT a controlled manner.)
9. In the terminal opened on the iTunes server, a Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

10. A/R should have "Rmv" below it, if this is true, PASS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

11. Service Type should have "_raop._tcp." below it, if this is true, PASS.
12. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes).

14.12.12 Bonjour TXT Record Lookup

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the iTunes server connected to the Base Station, issue the following command in a Terminal:

   ```
   dns-sd -B _raop
   ```

4. Use Control-C to stop dns-sd at the end of this step.
5. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

6. A/R should have "Add" below it, if this is true, PASS.
7. Service Type should have "_raop._tcp." below it, if this is true, PASS.
8. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes).
9. Note the Instance Name, Domain, and Service Type.
10. On the iTunes server connected to the Base Station, issue the following command in a Terminal:
    - `dns-sd -L Instance Name _raop local.`
    - (Example: `dns-sd -L 78CA3901C341@AirTunes _raop local.`)
11. The Bonjour record needs to contain a minimum of the following TXT records (a value is specified that is the requirement for field):
    - am=<vendor model name>
    - cn=0,1
    - da=true
    - et=0,4

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- fv= <AirPlay firmware version>.<MCU firmware version>.<vendor custom> (unused fields should contain a 0 and the AirPlay firmware version field should start with a "p" or "s" only, followed by a number). See Bonjour TXT Records Tests (page 295) for the list of valid AirPlay POSIX Source release numbers.
- md=(This may only be present if the device displays the meta data options selected)
- pw=(required only if a password is required)
- tp=UDP
- vn=65537
- vs= See Bonjour TXT Records Tests (page 295) for the list of valid AirPlay POSIX Source release versions. (Populated by AirPlay source code) (No field should be left empty, any field that contains no data should not be present.)

12. If all of the fields are present then PASS, else FAIL.

14.12.13 Bonjour IP Lookup

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the iTunes server connected to the Base Station, issue the following command in a Terminal:

   ```
   dns-sd -B _raop
   ```

4. Use Control-C to stop dns-sd at the end of this step.
5. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

6. A/R should have "Add" below it, if this is true, PASS.
7. Service Type should have "_raop._tcp." below it, if this is true, PASS.
8. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes).
9. Note the Instance Name.
10. Obtain the IP address that is listed in the Apple AU under the DHCP clients for this MAC address.
11. On the iTunes server connected to the Base Station, issue the following command in a Terminal:
12. ping IP address.
13. Use Control-C to stop dns-sd at the end of this step.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14. If ping is unsuccessful, FAIL.
15. If the DUT also supports IPv6, on the iTunes server connected to the Base Station, issue the following command in a Terminal:
16. ping6 IP address.
17. Use Control-C to stop dns-sd at the end of this step.
18. If ping is unsuccessful, FAIL.

14.12.14 Bonjour TXT Record Change based on Speaker Name Change

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the iTunes server connected to the Base Station, issue the following command in a Terminal and leave it running.

   ```
   dns-sd -B _raop
   ```

4. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

5. A/R should have "Add" below it, if this is true, PASS.
6. Service Type should have "_raop._tcp." below it, if this is true, PASS.
7. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes).
8. Use the user interface to rename the speaker to "DUTSpeaker" and apply the settings. This may require a soft restart of the DUT.
9. In the terminal opened on the iTunes server, a Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

10. A/R should have "Rmv" below it, if this is true, PASS.
11. Service Type should have "_raop._tcp." below it, if this is true, PASS.
12. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
13. In the terminal opened on the iTunes server, a Bonjour record should appear with the following format:

    ```
    Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
    ```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14. A/R should have "Add" below it, if this is true, PASS.
15. Service Type should have "_raop._tcp." below it, if this is true, PASS.
16. Instance should have "MACADDRESS@DUTSpeaker" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
17. If DUTSpeaker is not the name of the speaker, then FAIL.

14.12.15 Bonjour TXT Record Change based on change Password Protection

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the iTunes server connected to the Base Station, issue the following command in a Terminal:

   ```
   dns-sd -B _raop
   ```

4. Use Control-C to stop dns-sd at the end of this step.
5. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

6. A/R should have "Add" below it, if this is true, PASS.
7. Service Type should have "_raop._tcp." below it, if this is true, PASS.
8. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
9. Note the Instance Name, Domain, and Service Type.
10. On the iTunes server connected to the Base Station, issue the following command in a Terminal:
    - `dns-sd -L Instance Name _raop local.`
    - (Example: `dns-sd -L 78CA3901C341@AirTunes _raop local.`)
11. The Bonjour record needs to contain a minimum of the following TXT records (a value is specified that is the requirement for field):
    - am=<vendor model name>
    - cn=0,1
    - da=true
    - et=0,4

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- fv=<AirPlay firmware version>.<MCU firmware version>.<vendor custom> (unused fields should contain a 0 and the AirPlay firmware version field should start with a "p" or "s" only, followed by a number). See Bonjour TXT Records Tests (page 295) for the list of valid AirPlay POSIX Source release numbers.
- md=(This may only be present if the device displays the meta data options selected)
- pw=(required only if a password is required)
- tp=UDP
- vn=65537
- vs= See Bonjour TXT Records Tests (page 295) for the list of valid AirPlay POSIX Source release versions. (Populated by AirPlay source code) (No field should be left empty, any field that contains no data should not be present.)

12. The pw TXT record should not be present as the out of box behavior.
13. Use the user interface to set a password for the speaker and apply the settings. This may require a soft restart.
14. The Bonjour record needs to contain a minimum of the following TXT records (a value is specified that is the requirement for field):
    - am=<vendor model name>
    - cn=0,1
    - da=true
    - et=0,4
    - fv= <AirPlay firmware version>.<MCU firmware version>.<vendor custom> (unused fields should contain a 0 and the AirPlay firmware version field should start with a "p" or "s" only, followed by a number). See Bonjour TXT Records Tests (page 295) for the list of valid AirPlay POSIX Source release numbers.
    - md=(This may only be present if the device displays the meta data options selected)
    - pw=(required only if a password is required)
    - tp=UDP
    - vn=65537
    - vs= Refer to Bonjour TXT Records Tests (page 295) for the list of valid AirPlay POSIX Source release versions. (Populated by AirPlay source code) (No field should be left empty, any field that contains no data should not be present.)
15. Verify the pw TXT record is now set to true and in iTunes the speaker has a lock next the name of the speaker and when the user attempts to connect, iTunes ask for the password that was entered in the user interface. If all of this is true, PASS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12.16 Single Audio Stream (iTunes) Tests

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. On the iTunes server that is connected to the Apple Base Station, select the newly added speaker.
5. The speaker should be able to be selected without and error, or else FAIL.
6. Once the speaker is selected, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
7. The audio should begin playing within 15 seconds, or else FAIL.
8. Click the stop button within iTunes.
9. The audio should stop playing within 15 seconds, or else FAIL.

14.12.17 Single Audio Stream (iOS) Tests

For the Test Environment, refer to Figure 14-6 (page 287).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iOS within 30 seconds, or else FAIL.
4. On the Apple device that is connected to the Apple Base Station, select the newly added speaker.
5. The speaker should be able to be selected without and error, or else FAIL.
6. Once the speaker is selected, set the volume on the Apple device to 50% and click play. (The 50% does not need to be exact.).
7. The audio should begin playing within 15 seconds, or else FAIL.
8. Click the stop button on the Apple device.
9. The audio should stop playing within 15 seconds, or else FAIL.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12.18 Single Audio Stream to Multiple Devices (iTunes) Tests

For the Test Environment, refer to Figure 14-3 (page 286) and/or Figure 14-4 (page 286).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. On the iTunes server that is connected to the Apple Base Station, confirm the DUT is selectable.
5. The speaker should be able to be selected without and error, or else FAIL.
6. Connect the AirPort Express wirelessly to the Apple Base Station and insure the AirTunes option is selected in the Apple AU.
7. The AirPort Express will show up in iTunes as a speaker.
8. In iTunes select "Multiple Speakers" and check both of the boxes for the DUT and the AirPort Express. Then click Play in iTunes.
9. The audio should begin playing within 15 seconds on both devices and should be in sync, or else FAIL.
10. Click the stop button within iTunes.
11. The audio should stop playing within 15 seconds on both devices, or else FAIL.

14.12.19 Devices with Multiple Inputs Tests

14.12.19.1 Switching to Other Input where the Other Input is active (Manual Switch Available)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the DUT set the device to AirPlay input selection.
4. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.
5. The audio should begin playing within 15 seconds, or else FAIL.
6. On a different input on the DUT begin some activity.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

(Example: For a receiver, attach a CD player and click play so music will be going into the DUT. This will be different based on the type of device.)

7. The audio from the iTunes server should keep playing on the speakers.
8. On the DUT switch the input from the AirPlay input to the input from step 4.
9. The active input should be acting correctly and actively using the DUT.
10. Using the sniffer on the iTunes server, verify three HTTP Get Requests are sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay device-Prevent-playback=1 (Prevent) command. Another request's payload should contain an AirPlay device-busy=1 (Busy) command. Another request's payload should contain an AirPlay device-Prevent-playback=0 (Allow) command. These commands must come in sequence without any additional interspersed commands, or FAIL. The order for of the Device-Prevent-playback=1 (Prevent) and Device-busy=1 (Busy) commands does not matter but the Device-Prevent-playback=0 (Allow) must come after the Device-Prevent-playback=1 (Prevent) command, or FAIL.
11. Continue the activity on the other input for 60 seconds while taking a sniffer trace on the iTunes server.
12. There are two options:
    - No HTTP Get Requests are sent from the DUT to the iTunes.
    - The device-busy=1 (Busy) and device-Prevent-playback=0 (Allow) HTTP Get commands could be sent at regular intervals, which are manufacturer dependent.
13. If a HTTP Get Request with either the Device-Prevent-playback=1 (Prevent) or Device-busy=0 (Available) command is sent, FAIL.
14. Stop the activity on the DUT but leave it on the input other than AirPlay.
15. There are two options:
    - No HTTP Get Requests are sent from the DUT to the iTunes.
    - The device-busy=1 (Busy) and device-Prevent-playback=0 (Allow) HTTP Get commands could be sent at regular intervals, which are manufacturer dependent.
16. If a HTTP Get Request with either the Device-Prevent-playback=1 (Prevent) or Device-busy=0 (Available) command is sent, FAIL.
17. Change the input to the AirPlay input.
18. Using the sniffer on the iTunes server, verify a HTTP Get Request is sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay Device-busy=0 (Available) command.
19. Another HTTP Get Request with a payload of the Device-Prevent-playback=0 (Allow) command could also be sent in addition to the Device-busy=0 (Available) command but if a HTTP Get Request with a payload of either Device-Prevent-playback=1 (Prevent) or Device-busy=1 (Busy) is sent, FAIL.
20. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

21. The audio should begin playing within 15 seconds, or else FAIL.
22. On a different input on the DUT begin some activity.

    (Example: For a receiver, attach a CD player and click play so music will be going into the DUT. This will be different based on the type of device.)
23. The audio from the iTunes server should keep playing on the speakers.
24. On the DUT switch the input from the AirPlay input to the input from 3 steps above.
25. The active input should be acting correctly and actively using the DUT.
26. Using the sniffer on the iTunes server, verify three HTTP Get Requests are sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay device-Prevent-playback=1 (Prevent) command. Another request's payload should contain an AirPlay device-busy=1 (Busy) command. Another request's payload should contain an AirPlay device-Prevent-playback=0 (Allow) command. These commands must come in sequence without any additional interspersed commands, or FAIL. The order for of the Device-Prevent-playback=1 (Prevent) and Device-busy=1 (Busy) commands does not matter but the Device-Prevent-playback=0 (Allow) must come after the Device-Prevent-playback=1 (Prevent) command, or FAIL.
27. Continue the activity on the other input for 60 seconds while taking a sniffer trace on the iTunes server.
28. There are two options:
    - No HTTP Get Requests are sent from the DUT to the iTunes.
    - The device-busy=1 (Busy) and device-Prevent-playback=0 (Allow) HTTP Get commands could be sent at regular intervals, which are manufacturer dependent.
29. If a HTTP Get Request with either the Device-Prevent-playback=1 (Prevent) or Device-busy=0 (Available) command is sent, FAIL.
30. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.
31. The audio should begin playing within 15 seconds, or else FAIL.
32. Using the sniffer on the iTunes server, verify a HTTP Get Request is sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay Device-busy=0 (Available) command.
33. Another HTTP Get Request with a payload of the Device-Prevent-playback=0 (Allow) command could also be sent in addition to the Device-busy=0 (Available) command but if a HTTP Get Request with a payload of either Device-Prevent-playback=1 (Prevent) or Device-busy=1 (Busy) is sent, FAIL.

14.12.19.2 Switching to Other Input where the Other Input is inactive (Manual Switch Available)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the DUT set the device to AirPlay input selection.
4. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.
5. The audio should begin playing within 15 seconds, or else FAIL.
6. On the DUT switch the input from the AirPlay input to an idle input.
7. The idle input should be selected correctly.
8. Using the sniffer on the iTunes server, verify three HTTP Get Requests are sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay device-Prevent-playback=1 (Prevent) command. Another request's payload should contain an AirPlay device-busy=1 (Busy) command. Another request's payload should contain an AirPlay device-Prevent-playback=0 (Allow) command. These commands must come in sequence without any additional interspersed commands, or FAIL. The order for of the Device-Prevent-playback=1 (Prevent) and Device-busy=1 (Busy) commands does not matter but the Device-Prevent-playback=0 (Allow) must come after the Device-Prevent-playback=1 (Prevent) command, or FAIL.
9. Stay on the idle input for 60 seconds while taking a sniffer trace on the iTunes server.
10. There are two options:
    - No HTTP Get Requests are sent from the DUT to the iTunes.
    - The device-busy=1 (Busy) and device-Prevent-playback=0 (Allow) HTTP Get commands could be sent at regular intervals, which are manufacturer dependent.
11. If a HTTP Get Request with either the Device-Prevent-playback=1 (Prevent) or Device-busy=0 (Available) command is sent, FAIL.
12. Change the input to the AirPlay input.
13. Using the sniffer on the iTunes server, verifies a HTTP Get Request is sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay device-busy=0 (Available) command.
14. Another HTTP Get Request with a payload of the Device-Prevent-playback=0 (Allow) command could also be sent in addition to the Device-busy=0 (Available) command but if a HTTP Get Request with a payload of either Device-Prevent-playback=1 (Prevent) or Device-busy=1 (Busy) is sent, FAIL.
15. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.
16. The audio should begin playing within 15 seconds, or else FAIL.
17. On the DUT switch the input from the AirPlay input to an idle input.
18. The idle input should be selected correctly.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

19. Using the sniffer on the iTunes server, verify three HTTP Get Requests are sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay device-Prevent-playback=1 (Prevent) command. Another request's payload should contain an AirPlay device-busy=1 (Busy) command. Another request's payload should contain an AirPlay device-Prevent-playback=0 (Allow) command. These commands must come in sequence without any additional interspersed commands, or FAIL. The order for of the Device-Prevent-playback=1 (Prevent) and Device-busy=1 (Busy) commands does not matter but the Device-Prevent-playback=0 (Allow) must come after the Device-Prevent-playback=1 (Prevent) command, or FAIL.
20. Stay on the idle input for 60 seconds while taking a sniffer trace on the iTunes server.
21. There are two options:
    - No HTTP Get Requests are sent from the DUT to the iTunes.
    - The device-busy=1 (Busy) and device-Prevent-playback=0 (Allow) HTTP Get commands could be sent at regular intervals, which are manufacturer dependent.
22. If a HTTP Get Request with either the Device-Prevent-playback=1 (Prevent) or Device-busy=0 (Available) command is sent, FAIL.
23. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.
24. The audio should begin playing within 15 seconds, or else FAIL.
25. Using the sniffer on the iTunes server, verify a HTTP Get Request is sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay Device-busy=0 (Available) command.
26. Another HTTP Get Request with a payload of the Device-Prevent-playback=0 (Allow) command could also be sent in addition to the Device-busy=0 (Available) command but if a HTTP Get Request with a payload of either Device-Prevent-playback=1 (Prevent) or Device-busy=1 (Busy) is sent, FAIL.

14.12.19.3 Switching to Other Input where the Other Input is active (No Manual Switch Available)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the DUT set the device to AirPlay input selection.
4. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

5. The audio should begin playing within 15 seconds, or else FAIL.
6. On a different input on the DUT begin some activity.

   (Example: For a receiver, attach a CD player and click play so music will be going into the DUT. This will be different based on the type of device.)
7. The DUT should automatically switch to the input selected.
8. Using the sniffer on the iTunes server, verify three HTTP Get Requests are sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay device-Prevent-playback=1 (Prevent) command. Another request's payload should contain an AirPlay device-busy=1 (Busy) command. Another request's payload should contain an AirPlay device-Prevent-playback=0 (Allow) command. These commands must come in sequence without any additional interspersed commands, or FAIL. The order for of the Device-Prevent-playback=1 (Prevent) and Device-busy=1 (Busy) commands does not matter but the Device-Prevent-playback=0 (Allow) must come after the Device-Prevent-playback=1 (Prevent) command, or FAIL.
9. Continue the activity on the other input for 60 seconds while taking a sniffer trace on the iTunes server.
10. There are two options:
    - No HTTP Get Requests are sent from the DUT to the iTunes.
    - The device-busy=1 (Busy) and device-Prevent-playback=0 (Allow) HTTP Get commands could be sent at regular intervals, which are manufacturer dependent.
11. If a HTTP Get Request with either the Device-Prevent-playback=1 (Prevent) or Device-busy=0 (Available) command is sent, FAIL.
12. Stop the activity on the DUT.
13. There are two options:
    - No HTTP Get Requests are sent from the DUT to the iTunes.
    - The device-busy=1 (Busy) and device-Prevent-playback=0 (Allow) HTTP Get commands could be sent at regular intervals, which are manufacturer dependent.
14. If a HTTP Get Request with either the Device-Prevent-playback=1 (Prevent) or Device-busy=0 (Available) command is sent, FAIL.
15. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.
16. The input should switch to AirPlay and audio should begin playing within 15 seconds, or else FAIL.
17. Using the sniffer on the iTunes server, verify a HTTP Get Request is sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay Device-busy=0 (Available) command.
18. Another HTTP Get Request with a payload of the Device-Prevent-playback=0 (Allow) command could also be sent in addition to the Device-busy=0 (Available) command but if a HTTP Get Request with a payload of either Device-Prevent-playback=1 (Prevent) or Device-busy=1 (Busy) is sent, FAIL.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

19. On a different input on the DUT begin some activity.

    (Example: For a receiver, attach a CD player and click play so music will be going into the DUT. This will be different based on the type of device.)

20. The DUT should automatically switch to the input selected.

21. Using the sniffer on the iTunes server, verify three HTTP Get Requests are sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay device-Prevent-playback=1 (Prevent) command. Another request's payload should contain an AirPlay device-busy=1 (Busy) command. Another request's payload should contain an AirPlay device-Prevent-playback=0 (Allow) command. These commands must come in sequence without any additional interspersed commands, or FAIL. The order for of the Device-Prevent-playback=1 (Prevent) and Device-busy=1 (Busy) commands does not matter but the Device-Prevent-playback=0 (Allow) must come after the Device-Prevent-playback=1 (Prevent) command, or FAIL.

22. Continue the activity on the other input for 60 seconds while taking a sniffer trace on the iTunes server.

23. There are two options:

    - No HTTP Get Requests are sent from the DUT to the iTunes.
    - The device-busy=1 (Busy) and device-Prevent-playback=0 (Allow) HTTP Get commands could be sent at regular intervals, which are manufacturer dependent.

24. If a HTTP Get Request with either the Device-Prevent-playback=1 (Prevent) or Device-busy=0 (Available) command is sent, FAIL.

25. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.

26. The audio should begin playing within 15 seconds, or else FAIL.

27. Using the sniffer on the iTunes server, verify a HTTP Get Request is sent from the DUT IP to the iTunes server's IP. The payload of one should contain an AirPlay Device-busy=0 (Available) command.

28. Another HTTP Get Request with a payload of the Device-Prevent-playback=0 (Allow) command could also be sent in addition to the Device-busy=0 (Available) command but if a HTTP Get Request with a payload of either Device-Prevent-playback=1 (Prevent) or Device-busy=1 (Busy) is sent, FAIL.

14.12.19.4 Switching to Other Input where the Other Input is inactive (No Manual Switch Available)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the DUT set the device to AirPlay input selection.
4. On the iTunes server that is connected to the Apple Base Station select the DUT and click on a song in the iTunes library and then click Play in iTunes.
5. The audio should begin playing within 15 seconds, or else FAIL.
6. On the DUT verify there in no way to switch the input from the AirPlay input to an idle input.
7. If there is no manual way to switch away from the AirPlay input test is PASS.

14.12.20 Metadata publishing Tests

14.12.20.1 Metadata TXT Record Verification

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the iTunes server connected to the Base Station, issue the following command in a Terminal:

   ```
   dns-sd -B _raop
   ```

4. Use Control-C to stop dns-sd at the end of this step.
5. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

6. A/R should have "Add" below it, if this is true, PASS.
7. Service Type should have "_raop._tcp." below it, if this is true, PASS.
8. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
9. Note the Instance Name, Domain, and Service Type.
10. On the iTunes server connected to the Base Station, issue the following command in a Terminal:
    - `dns-sd -L Instance Name _raop local.`
    - (Example: `dns-sd -L 78CA3901C341@AirTunes _raop local.`)
11. The Bonjour record needs to contain the following field in the TXT record.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

`md=` (This should only be present if the devices supports one of the metadata options.)

12. This field can be populated with any combination of the following values and the field can contain more than one.
    - `0 = Text is supported`
    - `1 = Artwork is supported`
    - `2 = Progress is supported`
13. In order for a metadata option to be considered supported the device must contain the appropriate mechanism to present that metadata. (i.e. if there is no display, remote control with display, app, or voice interface that can present that metadata, the device can not claim to support it)
14. If the metadata field contains all of the supported options that the device does support then PASS, else FAIL.

14.12.20.2 Song Text Verification (Locally Stored)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. On the iTunes server that is connected to the Apple Base Station, select the newly added speaker.
5. The speaker should be able to be selected without and error, or else FAIL.
6. Once the speaker is selected, select a song that does not have the Artist section in the song info set, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
7. The audio should begin playing within 15 seconds, or else FAIL.
8. Look at the display on the DUT.
9. The Song title should be the same as what is displayed in iTunes, and the Artist should not be displayed.
10. Select a song that does have the Artist section in the song info set, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
11. The audio should begin playing within 15 seconds, or else FAIL.
12. Look at the display on the DUT.
13. The Artist and Song title should be the same as what is displayed in iTunes.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14. Select a song that does not have the Artist section in the song info set, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
15. The audio should begin playing within 15 seconds, or else FAIL.
16. Look at the display on the DUT.
17. The Song title should be the same as what is displayed in iTunes, and the Artist should not be displayed.
18. Click the stop button within iTunes.
19. The audio should stop playing within 15 seconds, or else FAIL.

14.12.20.3 Artwork Verification (Locally Stored)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. On the iTunes server that is connected to the Apple Base Station, select the newly added speaker.
5. The speaker should be able to be selected without and error, or else FAIL.
6. Once the speaker is selected, select a song that does not have any album artwork, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
7. The audio should begin playing within 15 seconds, or else FAIL.
8. Look at the display on the DUT.
9. No artwork should be displayed.
10. Select a song that does have any album artwork, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
11. The artwork should be the same that is displayed in iTunes.
12. Look at the display on the DUT.
13. The artwork should be the same that is displayed in iTunes.
14. Select a song that does not have any album artwork, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
15. The audio should begin playing within 15 seconds, or else FAIL.
16. Look at the display on the DUT.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

17. No artwork should be displayed.
18. Click the stop button within iTunes.
19. The audio should stop playing within 15 seconds, or else FAIL.

14.12.20.4 Progress Bar Verification (Locally Stored)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. On the iTunes server that is connected to the Apple Base Station, select the newly added speaker.
5. The speaker should be able to be selected without and error, or else FAIL.
6. Once the speaker is selected, select a song that does not have any album artwork, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
7. The audio should begin playing within 15 seconds, or else FAIL.
8. Look at the display on the DUT.
9. The Progress bar and time should be the same as what is displayed on the iTunes. Observe Progress bar and displayed time for minimum 60 sec. If all of this is true, PASS.
10. Using either the buttons on the device or on a remote click the "Previous Track" release the button.
11. The audio should begin playing within 15 seconds, or else FAIL.
12. Look at the display on the DUT.
13. The current song should move back to the beginning of the song and restart playing and the slider bar in iTunes should have moved to the far left, indicating time 0. The Progress bar and time should move back and start from 0 in DUT. If all of this is true, PASS.
14. Click the stop button within iTunes.
15. The audio should stop playing within 15 seconds, or else FAIL.

14.12.20.5 Song Text Verification (iOS)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. In the music section of the Apple device that is connected to the Apple Base Station, select the newly added speaker.
5. The speaker should be able to be selected without and error, or else FAIL.
6. Once the speaker is selected, select a song that does not have the Artist section in the song info set, set the volume on the Apple device to 50% and click play. (The 50% does not need to be exact.)
7. The audio should begin playing within 15 seconds, or else FAIL.
8. Look at the display on the DUT.
9. The Song title should be the same as what is displayed on the Apple device, and the Artist should not be displayed.
10. Select a song that does have the Artist section in the song info set, set the volume on the Apple device to 50% and click play. (The 50% does not need to be exact.)
11. The audio should begin playing within 15 seconds, or else FAIL.
12. Look at the display on the DUT.
13. The Artist and Song title should be the same as what is displayed on the Apple device.
14. Select a song that does not have the Artist section in the song info set, set the volume on the Apple device to 50% and click play. (The 50% does not need to be exact.)
15. The audio should begin playing within 15 seconds, or else FAIL.
16. Look at the display on the DUT.
17. The Song title should be the same as what is displayed on the Apple device, and the Artist should not be displayed.
18. Click the stop button within iTunes.
19. The audio should stop playing within 15 seconds, or else FAIL.

14.12.20.6 Artwork Verification (iOS)

For the Test Environment, refer to Figure 14-6 (page 287).

1. Power on the DUT.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iOS within 30 seconds, or else FAIL.
4. In the music section of the Apple device that is connected to the Apple Base Station, select the newly added speaker.
5. The speaker should be able to be selected without and error, or else FAIL.
6. Once the speaker is selected, select a song that does not have any album artwork, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
7. The audio should begin playing within 15 seconds, or else FAIL.
8. Look at the display on the DUT.
9. No artwork should be displayed.
10. Select a song that does have album artwork, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
11. The audio should begin playing within 15 seconds, or else FAIL.
12. Look at the display on the DUT.
13. The artwork should be the same that is displayed in Apple device.
14. Select a song that does not have any album artwork, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
15. The audio should begin playing within 15 seconds, or else FAIL.
16. Look at the display on the DUT.
17. No artwork should be displayed.
18. Click the stop button on the Apple device.
19. The audio should stop playing within 15 seconds, or else FAIL.

14.12.20.7 Progress Bar Verification (iOS)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. In the music section of the Apple device that is connected to the Apple Base Station, select the newly added speaker.
5. The speaker should be able to be selected without and error, or else FAIL.
6. Once the speaker is selected, select a song that does not have the Artist section in the song info set, set the volume on the Apple device to 50% and click play. (The 50% does not need to be exact.)
7. The audio should begin playing within 15 seconds, or else FAIL.
8. Look at the display on the DUT.
9. The Progress bar and time should be the same as what is displayed on the Apple device. Observe Progress bar and displayed time for minimum 60 sec. If all of this is true, PASS.
10. Using either the buttons on the device or on a remote click the "Previous Track" release the button.
11. The audio should begin playing within 15 seconds, or else FAIL.
12. Look at the display on the DUT.
13. The current song should move back to the beginning of the song and restart playing and the slider bar in Apple device should have moved to the far left, indicating time 0. The Progress bar and time should move back and start from 0 in DUT. If all of this is true, PASS.
14. Click the stop button within iTunes.
15. The audio should stop playing within 15 seconds, or else FAIL.

14.12.21 AirPlay Password Tests

14.12.21.1 iTunes streaming after setting/changing/clearing AirPlay password via DUT UI

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678
3. Verify the accessory that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. On the iTunes server connected to the Base Station, issue the following command in a Terminal:

   ```
   dns-sd -B _raop
   ```

5. Use Control-C to stop dns-sd at the end of this step.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

6. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

7. A/R should have "Add" below it, if this is true, PASS.
8. Service Type should have "_raop._tcp." below it, if this is true, PASS.
9. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
10. Note the Instance Name.
11. On the iTunes server connected to the Base Station, issue the following command in a Terminal:
    - `dns-sd -L Instance Name _raop local.`
    - (Example: `dns-sd -L 78CA3901C341@AirTunes _raop local.`)
12. Verify that the Bonjour record does not contain "pw=" field, or else FAIL.
13. On the iTunes server that is connected to the Apple Base Station, select the newly added speaker and then click play.
14. The speaker should be able to be selected without an error, user should not be asked for a password and the audio should begin playing within 15 seconds, or else FAIL.
15. Click the stop button in iTunes.
16. The audio should stop playing within 15 seconds, or else FAIL.
17. Deselect the speaker from iTunes accessory list.
18. Use the DUT user interface to set a password for the speaker and apply the settings. This may require a soft restart.
19. On the iTunes server that is connected to the Apple Base Station, select the speaker again and then click play.
20. The speaker should be able to be selected without an error, user should be asked for the password. Once the user provides the previously set password, the audio should begin playing within 15 seconds, or else FAIL.
21. Click the stop button in iTunes.
22. The audio should stop playing within 15 seconds, or else FAIL.
23. Deselect the speaker from iTunes accessory list.
24. Use the DUT user interface to set a different password for the speaker and apply the settings. This may require a soft restart.
25. On the iTunes server that is connected to the Apple Base Station, select the speaker again and then click play.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

26. The speaker should be able to be selected without an error, user should be asked for the password. Once the user provides the newly set password, the audio should begin playing within 15 seconds, or else FAIL.
27. Click the stop button in iTunes.
28. The audio should stop playing within 15 seconds, or else FAIL.
29. Deselect the speaker from iTunes accessory list.
30. Use the DUT user interface to clear the password for the speaker and apply the settings. This may require a soft restart.
31. On the iTunes server that is connected to the Apple Base Station, select the speaker again and then click play.
32. The speaker should be able to be selected without an error, user should not be asked for a password and the audio should begin playing within 15 seconds, or else FAIL.

14.12.21.2 iOS streaming after setting/changing/clearing AirPlay password via DUT UI

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678
3. Verify the accessory that has just been added to the network appears as a speaker in the Apple device within 30 seconds, or else FAIL.
4. On the iTunes server connected to the Base Station, issue the following command in a Terminal:

   ```
   dns-sd -B _raop
   ```

5. Use Control-C to stop dns-sd at the end of this step.
6. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

7. A/R should have "Add" below it, if this is true, PASS.
8. Service Type should have "_raop._tcp." below it, if this is true, PASS.
9. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
10. Note the Instance Name.
11. On the iTunes server connected to the Base Station, issue the following command in a Terminal:
    - `dns-sd -L Instance Name _raop local.`
    - (Example: `dns-sd -L 78CA3901C341@AirTunes _raop local.`)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

12. Verify that the Bonjour record does not contain "pw=" field, or else FAIL.
13. On the Apple device that is connected to the Apple Base Station, select the newly added speaker and then click play.
14. The speaker should be able to be selected without an error, user should not be asked for a password and the audio should begin playing within 15 seconds, or else FAIL.
15. Click the stop button on the Apple device.
16. The audio should stop playing within 15 seconds, or else FAIL.
17. Deselect the speaker from the Apple device accessory list.
18. Use the DUT user interface to set a password for the speaker and apply the settings. This may require a soft restart.
19. On the Apple device that is connected to the Apple Base Station, select the speaker again and then click play.
20. The speaker should be able to be selected without an error, user should be asked for the password. Once the user provides the previously set password, the audio should begin playing within 15 seconds, or else FAIL.
21. Click the stop button on the Apple device.
22. The audio should stop playing within 15 seconds, or else FAIL.
23. Deselect the speaker from the Apple device accessory list.
24. Use the DUT user interface to set a different password for the speaker and apply the settings. This may require a soft restart.
25. On the Apple device that is connected to the Apple Base Station, select the speaker again and then click play.
26. The speaker should be able to be selected without an error, user should be asked for the password. Once the user provides the newly set password, the audio should begin playing within 15 seconds, or else FAIL.
27. Click the stop button on the Apple device.
28. The audio should stop playing within 15 seconds, or else FAIL.
29. Deselect the speaker from the Apple device accessory list.
30. Use the DUT user interface to clear the password for the speaker and apply the settings. This may require a soft restart.
31. On the Apple device that is connected to the Apple Base Station, select the speaker again and then click play.
32. The speaker should be able to be selected without an error, user should not be asked for a password and the audio should begin playing within 15 seconds, or else FAIL.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12.21.3 iTunes streaming after setting AirPlay password via WAC

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network using Wi-Fi Accessory Configuration (WAC). Ensure that the AirPlay Password has been entered during the WAC setup process. (Refer to WAC Compliance Test documentation for more details)
3. Verify the accessory that has just been added to the network appears as a speaker in iTunes within 60 seconds, or else FAIL.
4. On the iTunes server connected to the Base Station, issue the following command in a Terminal:

   ```
   dns-sd -B _raop
   ```

5. Use Control-C to stop dns-sd at the end of this step.
6. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

7. A/R should have "Add" below it, if this is true, PASS.
8. Service Type should have "_raop._tcp." below it, if this is true, PASS.
9. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
10. Note the Instance Name.
11. On the iTunes server connected to the Base Station, issue the following command in a Terminal:
    - `dns-sd -L Instance Name _raop local.`
    - (Example: `dns-sd -L 78CA3901C341@AirTunes _raop local.`)
12. Verify that the Bonjour record contains "pw=true" field, or else FAIL.
13. On the iTunes server that is connected to the Apple Base Station, select the newly added speaker and then click play.
14. The speaker should be able to be selected without an error, user should be asked for the password. Once the user provides the previously set password, the audio should begin playing within 15 seconds, or else FAIL.

14.12.21.4 iOS streaming after setting AirPlay password via WAC

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Connect the DUT to the network using Wi-Fi Accessory Configuration (WAC). Ensure that the AirPlay Password has been entered during the WAC setup process. (Refer to WAC Compliance Test documentation for more details)
3. Verify the accessory that has just been added to the network appears as a speaker on the Apple device within 60 seconds, or else FAIL.
4. On the iTunes server connected to the Base Station, issue the following command in a Terminal:

   ```
   dns-sd -B _raop
   ```

5. Use Control-C to stop dns-sd at the end of this step.
6. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

7. A/R should have "Add" below it, if this is true, PASS.
8. Service Type should have "_raop._tcp." below it, if this is true, PASS.
9. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
10. Note the Instance Name.
11. On the iTunes server connected to the Base Station, issue the following command in a Terminal:
    - `dns-sd -L Instance Name _raop local.`
    - (Example: `dns-sd -L 78CA3901C341@AirTunes _raop local.`)
12. Verify that the Bonjour record contains "pw=true" field, or else FAIL.
13. On the Apple device that is connected to the Apple Base Station, select the newly added speaker and then click play.
14. The speaker should be able to be selected without an error, user should be asked for the password. Once the user provides the previously set password, the audio should begin playing within 15 seconds, or else FAIL.

14.12.22 Device Commands (DACP) (iTunes)

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

14.12.22.1 Play/Pause:

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. On the iTunes server select DUT as the speaker and select a song in the main music library and click Play.
4. Using either the buttons on the device or on a remote click the "Pause" button.
5. The music should pause immediately on the DUT and the pause button in iTunes should have changed to the play symbol and progress should stop moving. If all of this is true, PASS.
6. Wait 60 seconds.
7. The speaker should not disconnect.
8. Using either the buttons on the device or on a remote click the "Play" button.
9. The music should start playing immediately on the DUT and the play button in iTunes should have changed to the pause symbol and progress should be moving. The metadata information (Artwork, Artist and Song title) should be the same as what is displayed in iTunes. If all of this is true, PASS.

14.12.22.2 Stop:

If the accessory has a "Stop" button on the accessory or on a remote, continue from the Play/Pause test with the following:

1. Using either the buttons on the device or on a remote click the "Stop" button.
2. The music should stop immediately on the DUT and the pause button in iTunes should have changed to the play symbol and progress should stop moving and the progress indicator should have moved back to 0:00. If all of this is true, PASS.
3. Wait 60 seconds.
4. The speaker should not disconnect.
5. Using either the buttons on the device or on a remote click the "Play" button.
6. The music should start playing immediately on the DUT and the play button in iTunes should have changed to the pause symbol and progress should be moving. The metadata information (Artwork, Artist and Song title) should be the same as what is displayed in iTunes. If all of this is true, PASS.
7. Using either the buttons on the device or on a remote click the "Stop" button.
8. The music should stop immediately on the DUT and the pause button in iTunes should have changed to the play symbol and progress should stop moving and the progress indicator should have moved back to 0:00. If all of this is true, PASS.

14.12.22.3 Next Item/Previous Item:

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. On the iTunes server select DUT as the speaker and select a song in the main music library and click play.
4. Using either the buttons on the device or on a remote click the "Next Track" button and release the button immediately.
5. The current song should have moved to the next song in the playlist and should have started playing. The metadata information (Artwork, Artist and Song title) should be the same as what is displayed in iTunes. If all of this is true, PASS.
6. Using either the buttons on the device or on a remote click the "Previous Track" release the button immediately. The button must be pressed within 2 secs of the song starting.
7. The current song should have moved to the previous song in the playlist and should have started playing. The metadata information (Artwork, Artist and Song title) should be the same as what is displayed in iTunes. If all of this is true, PASS.

14.12.22.4 Volume Up/Volume Down:

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the iTunes server load Wireshark and start sniffing all of the traffic on the wired interface.
4. On the iTunes server select DUT as the speaker and select a song in the main music library and click play. Move the Volume indicator bar to approximately 50%. This does not need to be exact.
5. The music should start playing immediately on the DUT and the play button in iTunes should have changed to the pause symbol and progress should be moving. If all of this is true, PASS.
6. Using either the buttons on the device or on a remote click the "Volume Up" button and release the button immediately. This will be the **reference volume** for this test.
7. The volume coming out of the speaker should increase and the slider bar in iTunes indicating the volume should also increase. If true, PASS.
8. The music should never stop playing.
9. On the sniffer verify an HTTP Get Request is sent from the DUT IP to the iTunes server's IP with a payload that indicates a value between -30 and 0. This value must be in a floating-point format and be close to the middle of the scale, i.e. plus or minus 5 db from -15.
10. Using either the buttons on the device or on a remote click the "Volume Down" button and release the button immediately.
11. The volume coming out of the speaker should decrease and the slider bar in iTunes indicating the volume should also decrease. If true, PASS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

12. The music should never stop playing.
13. On the sniffer verify an HTTP Get Request is sent from the DUT IP to the iTunes server's IP with a payload that indicates a value between -30 and 0. This value must be in a floating-point format and be close to the middle of the scale, i.e. plus or minus 5 db from -15. This value should be less than the reference volume.

14.12.22.5 Mute:

If the accessory has a "Mute" button on the accessory or on a remote, continue from the Volume Up/Volume Down test with the following:

1. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
2. No volume should be heard coming from the speaker and the slider bar in iTunes should have moved to the far left, indicating 0 volume. However the progress bar for the song should still be moving forward. If true, PASS.
3. On the sniffer verify an HTTP Get Request is sent from the DUT IP to the iTunes server's IP with a payload that indicates a value of -144. This value must be in a floating-point format.
4. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
5. The volume coming out of the speaker should go back to the reference volume.
6. On the sniffer verify that a HTTP Get Request is sent from the DUT IP to the iTunes server's IP with a payload that indicates a value that is the same as what was sent when setting the reference volume.
7. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
8. No volume should be heard coming from the speaker and the slider bar in iTunes should have moved to the far left, indicating 0 volume. However the progress bar for the song should still be moving forward. If true, PASS.
9. On the sniffer verify an HTTP Get Request is sent from the DUT IP to the iTunes server's IP with a payload that indicates a value of -144. This value must be in a floating-point format.
10. Using either the buttons on the device or on a remote click the "Volume Down" button and release the button immediately.
11. The volume should go to slightly a lower volume than the reference volume.
12. On the sniffer verify an HTTP Get Request is sent from the DUT IP to the iTunes server's IP with a payload that indicates a value slightly lower than the reference volume. This value must be in a floating-point format. If the value is not lower or if it is more than 5 db less, FAIL.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

13. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
14. No volume should be heard coming from the speaker and the slider bar in iTunes should have moved to the far left, indicating 0 volume. However the progress bar for the song should still be moving forward. If true, PASS.
15. On the sniffer verify an HTTP Get Request is sent from the DUT IP to the iTunes server's IP with a payload that indicates a value of -144. This value must be in a floating-point format.
16. Using either the buttons on the device or on a remote click the "Volume Up" button and release the button immediately.
17. The volume coming out of the speaker should go back to the reference volume.
18. On the sniffer verify that a HTTP Get Request is sent from the DUT IP to the iTunes server's IP with a payload that indicates a value that is the same as what was sent when setting the reference volume.
19. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
20. The volume coming out of the speaker should go back to the reference volume.
21. On the sniffer verify that a HTTP Get Request is sent from the DUT IP to the iTunes server's IP with a payload that indicates a value that is the same as what was sent when setting the reference volume.
22. On the iTunes server, adjust the volume to about 50%; this does not need to be exact.
23. The DUT should come out of the mute state and play the music with approximately 50% volume.

14.12.22.6 Repeat/Shuffle:

Note: Only run these tests if the DUT supports Repeat/Shuffle functionality.

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the iTunes server select DUT as the speaker and select a song in the main music library and click play.
4. Using either the buttons on the device or on a remote click the "Repeat" button and release the button immediately.
5. Verify in iTunes the Repeat icon changes from grey to blue and appears highlighted. If true, PASS.
6. The music should never stop playing.
7. Using either the buttons on the device or on a remote click the "Repeat" button and release the button immediately.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8. Verify in iTunes the Repeat icon changes from blue to blue with a little "1" on the icon and appears highlighted. If true, PASS.
9. The music should never stop playing.
10. Using either the buttons on the device or on a remote click the "Repeat" button and release the button immediately.
11. Verify in iTunes the Repeat icon changes from blue with a little "1" on the icon to grey. If true, PASS.
12. The music should never stop playing.
13. Using either the buttons on the device or on a remote click the "Shuffle" button and release the button immediately.
14. Verify in iTunes the Shuffle icon changes from grey to blue and appears highlighted. If true, PASS.
15. The music should never stop playing.
16. Using either the buttons on the device or on a remote click the "Shuffle" button and release the button immediately.
17. Verify in iTunes the Shuffle icon changes from blue to grey. If true, PASS.
18. The music should never stop playing.

14.12.23 Device Commands (DACP) (iOS)

14.12.23.1 Play/Pause:

For the Test Environment, refer to Figure 14-6 (page 287).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the Apple device select the DUT as the speaker and select a song in the main music library and click play.
4. Using either the buttons on the device or on a remote click the "Pause" button.
5. The music should pause immediately on the DUT and the pause button in iOS should have changed to the play symbol and progress should stop moving. If all of this is true, PASS.
6. Wait 10 seconds. Using either the buttons on the device or on a remote click the "Play" button.
7. The music should start playing immediately on the DUT and the play button in iTunes should have changed to the pause symbol and progress should be moving. The metadata information (Artwork, Artist and Song title) should be the same as what is displayed in iOS. If all of this is true, PASS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12.23.2 Stop:

If the accessory has a "Stop" button on the accessory or on a remote, continue from the Play/Pause test with the following:

1. Using either the buttons on the device or on a remote click the "Stop" button.
2. The music should stop immediately on the DUT and the pause button in iTunes should have changed to the play symbol and progress should stop moving and the progress indicator should have moved back to 0:00. The metadata information (Artwork, Artist and Song title) should be the same as what is displayed in iOS. If all of this is true, PASS.

14.12.23.3 Next Item/Previous Item:

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the Apple device select the DUT as the speaker and select a song in the main music library and click play.
4. Using either the buttons on the device or on a remote click the "Next Track" button and release the button immediately.
5. The current song should have moved to the next song in the playlist and should have started playing. The metadata information (Artwork, Artist and Song title) should be the same as what is displayed in iOS. If all of this is true, PASS.
6. Using either the buttons on the device or on a remote click the "Previous Track" release the button immediately. The button must be pressed within 2 secs of the song starting.
7. The current song should have moved to the previous song in the playlist and should have started playing. The metadata information (Artwork, Artist and Song title) should be the same as what is displayed in iOS. If all of this is true, PASS.
8. Using either the buttons on the device or on a remote click the "Rewind" button and release the button immediately. Wait until the song has been playing before pressing rewind.
9. The current song should move back to the beginning of the song and restart playing. If all of this is true, PASS.

14.12.23.4 Volume Up/Volume Down:

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the Apple device select DUT as the speaker and select a song in the main music library and click play. Move the Volume indicator bar to approximately 50%. This does not need to be exact.
4. The music should start playing immediately on the DUT and the play button in iOS should have changed to the pause symbol and progress should be moving. If all of this is true, PASS.
5. Using either the buttons on the device or on a remote click the "Volume Up" button and release the button immediately.
6. The volume coming out of the speaker should increase and the slider bar in iTunes indicating the volume should also increase. If true, PASS.
7. The music should never stop playing.
8. Using either the buttons on the device or on a remote click the "Volume Down" button and release the button immediately. This will be the **reference volume** for this test.
9. The volume coming out of the speaker should decrease and the slider bar in iTunes indicating the volume should also decrease. If true, PASS.
10. The music should never stop playing.

14.12.23.5 Mute:

If the accessory has a "Mute" button on the accessory or on a remote, continue from the Volume Up/Volume Down test with the following:

1. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
2. No volume should be heard coming from the speaker and the slider bar in iTunes should have moved to the far left, indicating 0 volume. However the progress bar for the song should still be moving forward. If true, PASS.
3. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
4. The volume coming out of the speaker should go back to the reference volume.
5. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
6. No volume should be heard coming from the speaker and the slider bar in iTunes should have moved to the far left, indicating 0 volume. However the progress bar for the song should still be moving forward. If true, PASS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

7. Using either the buttons on the device or on a remote click the "Volume Down" button and release the button immediately.
8. The volume should go to slightly a lower volume than the reference volume.
9. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
10. No volume should be heard coming from the speaker and the slider bar in iTunes should have moved to the far left, indicating 0 volume. However the progress bar for the song should still be moving forward. If true, PASS.
11. Using either the buttons on the device or on a remote click the "Volume Up" button and release the button immediately.
12. The volume coming out of the speaker should go back to the reference volume.
13. Using either the buttons on the device or on a remote click the "Mute" button and release the button immediately.
14. No volume should be heard coming from the speaker and the slider bar in iTunes should have moved to the far left, indicating 0 volume. However the progress bar for the song should still be moving forward. If true, PASS.
15. On the iTunes server, adjust the volume to about 50%; this does not need to be exact.
16. The DUT should come out of the mute state and play the music with approximately 50% volume.

14.12.23.6 Repeat/Shuffle:

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. On the Apple device select the DUT as the speaker and select a song in the main music library and click play.
4. Using either the buttons on the device or on a remote click the "Repeat" button and release the button immediately.
5. Verify in iOS the Repeat icon changes from grey to blue and appears highlighted. If true, PASS.
6. The music should never stop playing.
7. Using either the buttons on the device or on a remote click the "Repeat" button and release the button immediately.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8. Verify in iTunes the Repeat icon changes from blue to blue with a little "1" on the icon and appears highlighted. If true, PASS.
9. The music should never stop playing.
10. Using either the buttons on the device or on a remote click the "Repeat" button and release the button immediately.
11. Verify in iTunes the Repeat icon changes from blue with a little "1" on the icon to grey. If true, PASS.
12. The music should never stop playing.
13. Using either the buttons on the device or on a remote click the "Shuffle" button and release the button immediately.
14. Verify in iTunes the Shuffle icon changes from grey to blue and appears highlighted. If true, PASS.
15. The music should never stop playing.
16. Using either the buttons on the device or on a remote click the "Shuffle" button and release the button immediately.
17. Verify in iTunes the Shuffle icon changes from blue to grey. If true, PASS.
18. The music should never stop playing.

14.12.24 Standby State Tests

This section is only required if the DUT has a way to put itself into a low power standby state or sleep state. If the device always stays in an active state, then this section does not need to be tested.

These tests only needs to run if DUT can be put into a standby state, however only choose the applicable test if the DUT supports a full network while in standby state or not. These tests need to be rerun for both wired and wireless if the device contains both interfaces.

14.12.24.1 Full Network Standby State

For the Test Environment, refer to Figure 14-1 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Once connected to the network, open the Apple AU and go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC Address is not present, FAIL. If the MAC address is present, note the IP Address next to the MAC Address.
4. Put the DUT into its standby state. (This maybe done by pushing the power button.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

5. The speaker should remain as a selectable speaker option.
6. On the iTunes server connected to the Base Station ping the IP address that was collected from Step 2-3.
7. If all the pings are successful, then PASS, else FAIL.
8. On the iTunes server connected to the Base Station, issue the following command in a Terminal and leave it running.

   ```
   dns-sd -B _raop
   ```

9. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

10. A/R should have "Add" below it, if this is true, PASS.
11. Service Type should have "_raop._tcp." below it, if this is true, PASS.
12. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)

14.12.24.2 Full Network Standby State Not Supported (Wired)

For the Test Environment, refer to Figure 14-1 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network.
3. Once connected to the network, open the Apple AU and go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC Address is not present, FAIL. If the MAC address is present, note the IP Address next to the MAC Address.
4. Verify the green link light is lit on the Ethernet port that the DUT is plugged into, if not, FAIL.
5. On the iTunes server connected to the Base Station ping the IP address that was collected from Step 2-3.
6. If all the pings are successful, then PASS, else FAIL.
7. Put the DUT into its standby state. (This maybe done by pushing the power button.
8. On the iTunes server connected to the Base Station, issue the following command in a Terminal and leave it running.

   ```
   dns-sd -B _raop
   ```

9. The speaker should be removed as a selectable speaker option.
10. A Bonjour record should appear with the following format:

    ```
    Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
    ```

11. A/R should have "Rmv" below it, if this is true, PASS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

12. Service Type should have "_raop._tcp." below it, if this is true, PASS.
13. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
14. On the iTunes server connected to the Base Station ping the IP address that was collected from Step 2.
15. If all the pings are unsuccessful, then PASS, else FAIL.
16. Verify the green link light is off on the Ethernet port that the DUT is plugged into, if not, FAIL.

14.12.24.3 Full Network Standby State Not Supported (Wireless)

For the Test Environment, refer to Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Once connected to the network, open the Apple AU and go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC Address is not present, FAIL. If the MAC address is present, note the IP Address next to the MAC Address.
4. In the Apple AU, go to Advanced->Logs and Statistics->Wireless Clients. In list box look for the MAC address that was collected from the DHCP table. Verify that Type that is listed is one of the following options: 802.11b/g, 802.11g, 802.11a, 802.11a/n, 802.11b/g/n, or 802.11g/n. If any other Type is present, FAIL.
5. On the iTunes server connected to the Base Station ping the IP address that was collected from Step 2-3.
6. If all the pings are successful, then PASS, else FAIL.
7. Put the DUT into its standby state. (This maybe done by pushing the power button.)
8. The speaker should be removed as a selectable speaker option.
9. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

10. A/R should have "Rmv" below it, if this is true, PASS.
11. Service Type should have "_raop._tcp." below it, if this is true, PASS.
12. Instance should have "MACADDRESS@DeviceName" below it, if this is true, PASS. (Example: 78CA3901C341@AirTunes)
13. On the iTunes server connected to the Base Station ping the IP address that was collected from Step 2-3.
14. If all the pings are unsuccessful, then PASS, else FAIL.
15. In the Apple AU, go to Advanced->Logs and Statistics->Wireless Clients. In list box look for the MAC address that was collected from Step 2-3. The MAC address should not be present, if it is, FAIL.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12.25 Wake on LAN Tests

For the Test Environment, refer to Figure 14-2 (page 285).

This section is only required if the DUT has a way to put itself into a low power standby state or sleep state. If the device always stays in an active state, then this section does not need to be tested.

1. Power on the DUT.
2. Using an Ethernet wire, connect the DUT to one of the LAN ports on the Base Station.
3. On the iTunes server connected to the Base Station, use iTunes and select the DUT as a Speaker that was just added to the network. Once the speaker is selected attempt to start the stream.
4. Music should start playing within 15 seconds of clicking play.
5. Stop the audio stream and select the Computer as the speaker.
6. Music should stop playing within 15 seconds of clicking stop.
7. Put the DUT into its low power or sleep state.
8. The speaker should remain as a selectable speaker option.
9. On the iTunes server connected to the Base Station, use iTunes and select the DUT as a Speaker that was just added to the network. Once the speaker is selected attempt to start the stream.
10. Music should start playing within 15 seconds of clicking play.

14.12.26 Wake on WLAN Tests

For the Test Environment, refer to Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the Wireless network using WPA2-PSK with a Passphrase of 12345678.
3. On the iTunes server connected to the Base Station, use iTunes and select the DUT as a Speaker that was just added to the network. Once the speaker is selected attempt to start the stream.
4. Music should start playing within 15 seconds of clicking play.
5. Stop the audio stream and select the Computer as the speaker.
6. Music should stop playing within 15 seconds of clicking stop.
7. Put the DUT into its low power or sleep state.
8. The speaker should remain as a selectable speaker option.
9. On the iTunes server connected to the Base Station, use iTunes and select the DUT as a Speaker that was just added to the network. Once the speaker is selected attempt to start the stream.
10. Music should start playing within 15 seconds of clicking play.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12.27 Web Configuration Interface

AirPlay Accessories must support the use of a Bonjour HTTP record and a web based configuration page that is compatible with the Safari browser on Mac, PC, and iPhone, and iPad. The accessory will be tested and verified with the latest version of Safari on all three operating systems.

The web configuration interface must provide access to the following accessory settings: All settings must be modifiable from the web interface.

14.12.27.1 Friendly Product Name

For the Test Environment, refer to Figure 14-1 (page 285) and/or Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. From Safari, select the DUT from the Bonjour list
5. The speaker should be able to be selected without and error, or else FAIL.
6. Follow the onscreen instructions to rename the DUT.
7. The unit should update its name and rejoin the network, or else FAIL.
8. From Safari, go to the Bonjour list
9. The unit should be present in the list with its updated name, or else FAIL.
10. From iTunes, verify that the proper name is indicated in the AirPlay list.
11. The unit should indicate its new name, or else FAIL.
12. From Safari, select the DUT from the Bonjour list
13. The speaker should be able to be selected without and error, or else FAIL.
14. From Safari, verify that the proper name is indicated in the Web interface
15. The unit should indicate its new name, or else FAIL.

14.12.27.2 Wi-Fi Network Selection of available networks

For the Test Environment, refer to Figure 14-2 (page 285).

1. Power on the DUT.
2. Wait for DUT to complete booting.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. The DUT should begin beaconing 802.11 beacons and advertising as an 802.11b/g network with a SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
4. Using the iTunes server, associate wirelessly to the DUT.
5. The iTunes server should be able to associate on the 2.4 GHz band using 802.11b/g with no security and receive a network that is on the same IPv4 network as the DUT. If the DUT does not have a DHCP server and instead would like to use the IPv4 link local IP space, this must be specified in manufacturer's documentation.
6. From Safari, select the DUT from the Bonjour list
7. The speaker should be able to be selected without and error, or else FAIL.
8. From Safari, configure the DUT to connect to the Wi-Fi network on the Apple Base Station. Minimally their needs to be a way to manually specify the SSID and security passphrase, however there can also be a way to select from a list obtained via scanning.
9. The DUT must have a way to manually enter the SSID for the Wi-Fi network and select a security mode. Optionally it can create a list (scan list) for the user pick their network from, if desired. Once the device is connected to the Base Station, there must be some indication on the DUT that device is connected.
10. Open the AU and verify the settings are correct.
11. In the Apple AU, go to Advanced->Logs and Statistics->Wireless Clients. In list box look for the MAC address that was collected from Step 1. Verify that Type that is listed is one of the following options: 802.11b/g, 802.11g, 802.11a, 802.11a/n, 802.11b/g/n, or 802.11g/n. If any other Type is present, FAIL.
12. Next go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC address is not present, FAIL.
13. Open Safari on the iTunes Server and attempt to connect to the web server on the DUT with the address it just received.
14. Safari should be able to navigate to the web server, or else FAIL. All of the settings that were entered in previous steps should be reflected in the user interface.

14.12.27.3 DHCP Setting / Manual IP Settings

For the Test Environment, refer to Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. Using the iTunes server, associate wirelessly to the DUT.
5. The iTunes server should be able to associate on the 2.4 GHz band using 802.11b/g with no security and receive a network that is on the same IPv4 network as the DUT. If the DUT does not have a DHCP server and instead would like to use the IPv4 link local IP space, this must be specified in manufacturer's documentation.
6. From Safari, select the DUT from the Bonjour list
7. The speaker should be able to be selected without and error, or else FAIL.
8. From Safari, configure the DUT to connect to the Wi-Fi network on the Apple Base Station. Minimally their needs to be a way to manually specify the SSID and security passphrase, however there can also be a way to select from a list obtained via scanning.
9. The DUT must have a way to manually enter the SSID for the Wi-Fi network and select a security mode. Optionally it can create a list (scan list) for the user pick their network from, if desired. Once the device is connected to the Base Station, there must be some indication on the DUT that device is connected.
10. Open the AU and verify the settings are correct.
11. In the Apple AU, go to Advanced->Logs and Statistics->Wireless Clients. In list box look for the MAC address that was collected from Step 1. Verify that Type that is listed is one of the following options: 802.11b/g, 802.11g, 802.11a, 802.11a/n, 802.11b/g/n, or 802.11g/n. If any other Type is present, FAIL.
12. Next go to Advanced->Logs and Statistics->DHCP clients, and in the list box look for the MAC address that was collected from Step 1. If the MAC address is not present, FAIL.
13. Open Safari on the iTunes Server and attempt to connect to the web server on the DUT with the address it just received.
14. Safari should be able to navigate to the web server, or else FAIL. All of the settings that were entered in previous steps should be reflected in the user interface.
15. From Safari, select the DUT from the Bonjour list
16. The speaker should be able to be selected without and error, or else FAIL.
17. From Safari, turn off DHCP on the DUT. Manually configure the DUT to connect to the AP with a known good IP address.
18. Open Safari on the iTunes Server and attempt to connect to the web server on the DUT with the address it just received.
19. Safari should be able to navigate to the web server, or else FAIL. All of the settings that were entered in previous steps should be reflected in the user interface.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14.12.28 Audio Delay

AirPlay audio modules are tightly controlled for audio delay characteristics. The software on AirPlay audio modules can be adjusted to compensate for inherent system audio delays such as DSP processing. Please see vendor specific AirPlay module documentation on how to achieve this.

All DUTs must be tested to ensure additional delay has not been introduced into the system. DUTs will be tested by taking an oscilloscope trace with one channel connected to the line out audio jack of an AirPort Express 802.11n (2nd Generation) and a second channel connected to a microphone input axially aligned with the DUT's right speaker driver. Both the AirPort Express and the DUT shall be connected via Wi-Fi if available, otherwise both shall be connected via ethernet directly to an AirPort Extreme Base Station.

A Mac will be used as an iTunes server with both the AirPort Express and DUT selected as speaker outputs. The signal delay between the oscilloscope channels must measure no more than 5 ms between the DUT and the AirPort Express.

14.12.28.1 DUT Audio Delay Measurement

For the Test Environment, refer to Figure 14-2 (page 285).

1. Power on the DUT.
2. Connect the DUT to the network. There are two options here, either Ethernet or Wireless. If the device supports both interfaces then the test must be done for both. For Wireless only use WPA2-PSK with a Passphrase of 12345678.
3. Verify the device that has just been added to the network appears as a speaker in iTunes within 30 seconds, or else FAIL.
4. Using the iTunes server, associate wirelessly to the DUT.
5. The iTunes server should be able to associate on the 2.4 GHz band using 802.11b/g with no security and receive a network that is on the same IPv4 network as the DUT. If the DUT does not have a DHCP server and instead would like to use the IPv4 link local IP space, this must be specified in manufacturer's documentation.
6. On the iTunes server that is connected to the Apple Base Station, select multiple speaker output: AirPort Express & the newly added DUT.
7. The speaker should be able to be selected without and error, or else FAIL.
8. Once the speakers are selected, set the volume in iTunes to 50% and click play. (The 50% does not need to be exact.)
9. Start playing a click track from the iTunes server.
10. The audio should begin playing within 15 seconds, or else FAIL.
11. Measure the time delay from the rising edge of the signal from the AirPort Express and the DUT.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

12. The time delay must be less than 5 ms, or else FAIL.
13. Click the stop button within iTunes.
14. The audio should stop playing within 15 seconds, or else FAIL.

14.12.29 Certification Procedure

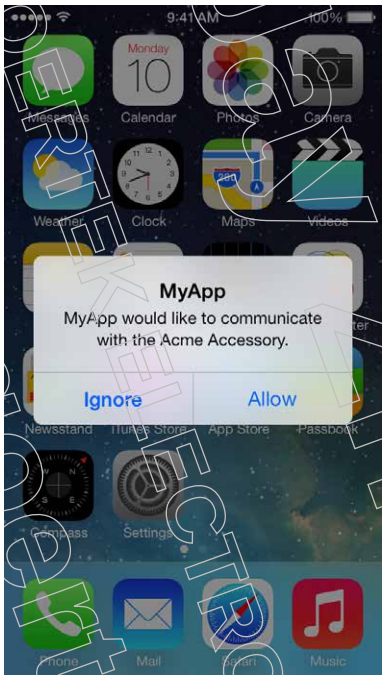
- Accessory submission must be accompanied by proof of Wi-Fi Certification for the Accessory.
- 2 Production samples are required.
- Two firmware versions are required for certification. The firmware shipped on both units should be the production candidate firmware as would ship to the end customer. The submission should additionally include a copy of the production firmware for reflashing the units (use name convention `<filename>_production`) and a copy of firmware that is identical to the production firmware but enables a telnet debug console and is enabled to run the iPerf tests required in PHY: Wired Performance (page 294) and PHY: Wireless Performance (100 ft) (page 294) of this document (use name convention `<filename>_telnet`). The firmware update test of Web Based Firmware Upgrade Test (page 287) requires that the Accessory firmware be updated to the debug version and then back.
- Full Accessory documentation must be provided in both printed and soft copy form (all customer facing documents such as user manuals and quick start guides).
- The submission must be accompanied by a completed "AirPlay Accessory Product Questionnaire R6".

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

15. App Launch

An accessory that supports the App Launch feature can request that an Apple device launch an app on its behalf.

Figure 15-1 App Launch Alert



15.1 App Launch Requirements

All accessories that support the App Launch feature via iAP2 must send or receive the following iAP2 control session message(s):

RequestAppLaunch (page 820)

Note: This message must only be sent in response to direct user action, such as pushing a button on the accessory. Autonomous launch of apps is explicitly prohibited unless the accessory is an automotive headunit.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The launched app must communicate with the accessory via the External Accessory Protocol (page 535).

15.2 App Launch Usage

If an Apple device does not support the app launch feature it will reject all related accessory identification attempts that call for RequestAppLaunch (page 820) to be sent by the accessory.

Accessories that establish an iAP connection via the Lightning connector may set `LaunchAlert` to 'no alert'. All other accessories must set `LaunchAlert` to 'alert'.

Accessories must not assume that the app has launched, because the request may be denied by iOS or the user. If positive confirmation of app launch is needed, the app must open an External Accessory Protocol (page 535) session with the accessory.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

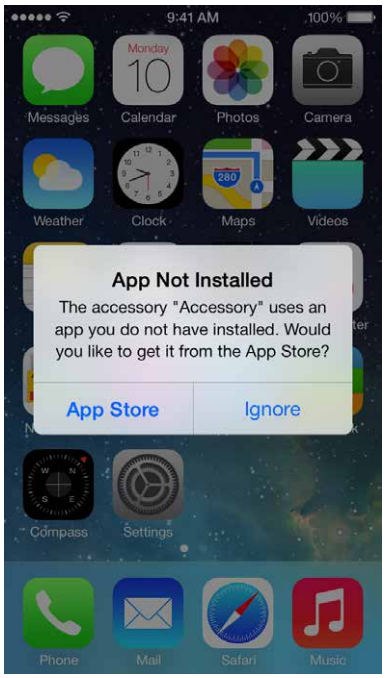
16. App Match

The App Match feature enables accessories that support the External Accessory Protocol feature (see External Accessory Protocol (page 535)) to match with compatible apps on the App Store.

When connected for the first time, the Apple device will ask the user if they would like to visit the App Store and view apps that work with the accessory. Subsequently, this action may be repeated by the user via *Settings > General > About > 'Accessory Name' > 'Find App for this Accessory'*.

Matched apps are listed in alphabetical order with one exception. If the accessory will work with apps from multiple development teams/companies, the accessory may provide a preferred Team ID that places apps from that development team at the top of the list.

Figure 16-1 App Match Alert



16.1 App Match Requirements

To support App Match during Accessory Identification (see Accessory Identification (page 265)), the accessory:

- Must declare one or more supported External Accessory Protocols (see External Accessory Protocol (page 535)) that matches against one or more apps that are allowed to declare support for the same External Accessory Protocol(s).
- May additionally declare a preferred Team ID that matches against one or more apps from a single app development team.

16.1.1 External Accessory Protocol

Unless otherwise specified, accessories must fully implement at least one External Accessory Protocol (see External Accessory Protocol (page 535)) and use it to communicate with iOS apps in order to support App Match.

Accessories may use the App Match feature with more than one External Accessory Protocol. If multiple protocols are listed, the App Store will display a list of apps that support one or more of those protocols.

16.1.2 Team ID

- The Team ID used for App Match must belong to the accessory developer.
- A maximum of one Team ID can be declared for App Match purposes.
- The Team ID also must not change dynamically.

16.2 App Match Usage

16.2.1 External Accessory Protocol

Accessories must support App Match for a particular External Accessory Protocol by setting the MatchAction parameter (see Table 59-12 (page 810)) in the protocol's parameter group (see Table 59-11 (page 810)) to an appropriate value during accessory identification.

16.2.2 Team ID

Accessories may declare compatibility with apps by providing a AppMatchTeamID parameter in the IdentificationInformation message (see IdentificationInformation (page 807)). The parameter's value must be a valid iOS Developer Program Team ID.

The Team ID is supplied by Apple and is unique to a specific development team. More information on the Team ID can be found in the iOS Developer Library (see https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/AppID.html).

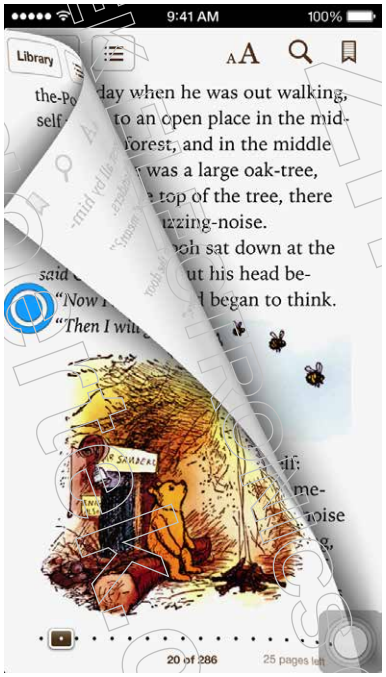
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

17. AssistiveTouch

The AssistiveTouch feature lets users generate iOS Multi-Touch gestures using one finger, a stylus, or a compatible accessory. Accessory developers should consider supporting AssistiveTouch if the accessory is intended for use by users with special needs who can see the display and manipulate more complex input peripherals, such as joysticks or touchpads. If the user cannot see the display, consider supporting the VoiceOver feature instead.

When an AssistiveTouch-compatible accessory connects to an Apple device and the feature is enabled via the iOS Settings app, an accessory-controlled virtual cursor is displayed. All AssistiveTouch interaction then proceeds as if the user's own fingers were being used to interact with the device.

Figure 17-1 AssistiveTouch Pointer



17.1 AssistiveTouch Requirements

All accessories supporting AssistiveTouch must be targeted at users with special needs. The accessory must also implement a `HIDComponent` that complies with all HID AssistiveTouch Pointer Requirements (page 588).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

All accessories that support the AssistiveTouch feature via iAP2 must send or receive the following iAP2 control session message(s):

StartAssistiveTouch (page 820)

StopAssistiveTouch (page 821)

StartAssistiveTouchInformation (page 821)

AssistiveTouchInformation (page 821)

StopAssistiveTouchInformation (page 821)

17.2 AssistiveTouch Usage

Accessories supporting AssistiveTouch must identify themselves as sending and receiving all messages associated with the feature. There are no optional messages. If the device does not support AssistiveTouch then it will reject all identification attempts that include AssistiveTouch messages. The accessory must also use the HID feature to register the HID Mouse descriptor with the device.

After identification has been completed and accepted, the accessory must send a StartAssistiveTouchInformation (page 821) message to the device. While these notifications are active, an AssistiveTouchInformation (page 821) message will be sent from the device to inform the accessory whether the feature is currently enabled or disabled. Accessories must be prepared to have the AssistiveTouch feature disabled or enabled at any time, whether by the user or by another accessory.

The AssistiveTouch feature can be disabled and enabled via the StartAssistiveTouch (page 820) and StopAssistiveTouch (page 821) messages while the accessory is still attached to an Apple device. This can be useful if the accessory needs to enter a different operating mode that does not provide an external pointing input to AssistiveTouch.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18. Bluetooth

Accessories that integrate Bluetooth technology must comply with the requirements stated in this chapter.

18.1 Conformity With Bluetooth Specifications

Every accessory that is compatible with an Apple product must support the *Bluetooth Core Specification* Version 2.1 + EDR or higher. This specification introduced the important security feature Secure Simple Pairing as well as Extended Inquiry Response.

18.1.1 Enhanced Data Rate

The Enhanced Data Rate (EDR) feature introduced in the *Bluetooth 2.0* specification enables accessories to communicate more efficiently. Every accessory must use EDR for the following reasons:

- It provides higher data rates compared to Basic Data Rate (BDR).
- It communicates more efficiently, transferring more data bits per unit of time.
- It reduces the power consumption used per bit transferred.
- It improves coexistence with Wi-Fi and other connected Bluetooth devices because it frees up more air time.
- It improves performance in multipoint configurations.

18.1.2 Adaptive Frequency Hopping

The Adaptive Frequency Hopping (AFH) feature introduced in the *Bluetooth 1.2* specification improves coexistence with Wi-Fi and other connected Bluetooth devices. Every accessory must use AFH.

18.1.3 Sniff Mode for Low Power Consumption

Minimizing power consumption is critical for all mobile devices. Therefore, every accessory that is compatible with an Apple product:

- Must support and must request Bluetooth sniff mode.
- Must accept requests for sniff mode and support all valid parameters listed in the Bluetooth specification.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Must support a sniff interval of 15 ms.
- Should use the following recommended sniff mode values:
  - Max Interval: 15 ms
  - Min Interval: 15 ms
  - Sniff Attempt: 1
  - Sniff Timeout: 0
- Must not renegotiate sniff after being established.
- Must support sniff subrating.

Accessories that are compatible with Apple products must also use sniff mode as much as possible, especially when there is little or no data being transmitted over the Bluetooth link. Besides its power consumption advantages, sniff mode enables better antenna sharing with Wi-Fi.

The sniff mode parameters are specific to the usage model and Bluetooth Profile. The Apple product expects the accessory to request sniff mode with appropriate parameters for a specific usage. If the accessory does not send such a request, the Apple product may send a sniff mode request. When the Apple product sends a request for sniff mode, the remote device must accept the request and its parameters without negotiation.

If the accessory sets the sniff mode parameters, the accessory must set the sniff interval to less than a third of the Bluetooth baseband Link Supervision Timeout. This makes the Bluetooth link less susceptible to interference. To improve link robustness, the accessory must use a shorter sniff interval instead of multiple sniff attempts.

Links with a sniff interval of 1 second or more make the slave device open up a large correlation window, which has to be taken into account when calculating the number of sniff attempts. With sniff intervals shorter than 1 second, multiple sniff attempts can improve link robustness but will increase power consumption.

18.1.4 Role and Topology Management

Every accessory that is compatible with an Apple product must:

- Accept a request for Role Switch from an Apple product.
- Continue with the connection when the Apple product rejects a request for Role Switch.

In a Bluetooth connection, one device is the master and the other the slave. The master can have multiple slaves, thus forming a piconet. The master device can also be a slave role to another master, creating a scatternet.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Such a scenario creates complications since the device has to alternate between the two piconets and thus wastes valuable bandwidth. Managing the topology of the network is therefore important for maximum performance. The Apple product may request a Role Switch, depending on its current topology, and the remote device must accept the request. The Apple product may also reject a request for a Role Switch because of topology concerns. Having a suboptimal topology may degrade the audio quality and the user's experience.

Only when it is maintaining multiple links, either Bluetooth or Wi-Fi, will the Apple product request or deny role switches. Hence, it will grant a role switch if there is no reason for the Apple product to be master. It is expected that the accessory will behave the same, only trying to be master when there is a legitimate reason.

The accessory must not always request to be master by default if there is no need in the system topology to do so. If later the accessory needs to be master in order to maintain multiple links, it must ask to be master at that time.

18.1.5 Extended Inquiry Response

Every accessory that is compatible with an Apple product must provide the following information in its Extended Inquiry Response packet:

- The Local Name of the accessory (Complete or Shortened).
- The TX Power Level.
- The Service Class UUID for the iAP2 protocol, if applicable (see iAP2 (page 357)).

During the Bluetooth discovery process, the Apple product prefers to display the Friendly Name of discovered accessories. Before the 2.1 version of the Bluetooth specification the Apple product would have to set up a connection to the accessory and do a Remote Name Request, which takes power, antenna time, and user's time. The Extended Inquiry Response feature, introduced in Bluetooth 2.1, lets an accessory send its Local Name and other information as part of the Inquiry Response and thereby increase the speed and efficiency of the discovery process.

The Local Name must match the accessory's markings and packaging and not contain ':' or ';'.

18.1.6 Secure Simple Pairing

Every accessory that is compatible with an Apple product must:

- Use Secure Simple Pairing.
- Use the Numerical Comparison method if it has a display and input device supporting it.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Secure Simple Pairing greatly increases security and is a mandatory security feature introduced in the Bluetooth 2.1 specification. To protect against a man-in-the-middle attack, the Numerical Comparison association model must be used whenever feasible. See Volume 1, Section 5.4 in the *Bluetooth Core Specification*, Version 2.1 + EDR.

18.2 Profiles

The Apple knowledge base article support.apple.com/kb/ht3647 provides a complete list of the Bluetooth profiles that certain Apple devices support. The Bluetooth specifications are the starting point for designing accessories that are compatible with these products. The following sections add information and requirements for some profiles, which can help accessory developers achieve superior results.

18.2.1 Device ID Profile (DID)

Every accessory that is compatible with an Apple product must:

- Support the Bluetooth Device ID Profile, version 1.3 or higher.
- Use the company identifier from the Assigned Numbers specification assigned by the Bluetooth SIG as its Vendor ID value (VID). See http://www.bluetooth.org/Technical/AssignedNumbers/identifiers.htm (requires login). Bluetooth HID Profile accessories may use a VID assigned by the USB Implementers Forum (USB-IF at http://www.usb.org) if the manufacturer does not have a Bluetooth SIG company identifier.
- Use its VID value for the end product manufacturer.
- Use the Vendor ID Source field to identify which organization assigned the value used in Vendor ID field value. See Section 5.6 of the *Bluetooth Device ID Profile Specification*.
- Use a ProductID value that uniquely identities the product.
- Use a Version value that uniquely identifies the software version. If the accessory supports iAP2, then this value must match the Firmware Version used in the Accessory Identification message (see Accessory Identification (page 265)).

The Device ID profile lets the Apple product identify the implementation of the remote accessory. This is valuable information and can be used to bridge alternate interpretations of the Bluetooth specification when communicating with a remote accessory. Therefore it is important that the information in the Device ID record uniquely identify the implementation.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

In the case of Bluetooth car kit devices, for instance, the same car kit might go into two different car models. Ideally the two car kits must have different Product IDs. However, it is acceptable for them to have the same ProductID as long as they have identical hardware, software, and features. If the implementations differ at all, they must have different Product IDs. The accessory can also use a secondary Device ID to uniquely identify the product ID or model number.

18.2.2 Hands-Free Profile (HFP)

Every accessory that is compatible with an Apple product and supports the Handsfree Profile must meet the requirements of the Bluetooth *Hands-Free Profile* specification, Version 1.5 or higher. Additional Apple requirements are specified in this section.

Remote accessories can use the Bluetooth *Hands-Free Profile* for phone communications. To achieve the best user experience, the remote accessory must support the following features, which are optional in the Bluetooth specification.

18.2.2.1 Remote Audio Volume Control

Every accessory that is compatible with an Apple product and supports HFP must:

- Support Remote Audio Volume Control so the speaker volume on the Hands-Free accessory can be controlled from the Apple product as described in Section 4.28 in the Bluetooth *Hands-Free Profile* specification version 1.5.
- Set the Remote volume control bit in the Supported Features bitmap sent with the `AT+BRSF=` command.

In some situations it is easier for the user to control the output volume through the Apple product instead of directly on the remote accessory. For example, a passenger (or-if the car is parked-the driver) in a car could use the volume slider on the phone to control the audio volume. Volume control synchronization is outlined in Section 4.48.2 in the Bluetooth *Hands-Free Profile* specification version 1.5.

18.2.2.2 Indicator Event Reporting

Every accessory that is compatible with an Apple product and supports HFP must use indicator events reporting and not perform repetitive polling of status.

Apple products support all mandatory and optional indicators specified in HFP version 1.5 (service, call, callsetup, callheld, signal, roam, battchg). To minimize unnecessary polling of status using the `AT+CIND?` command, the remote accessory must enable indicator events reporting by sending an `AT+CMER` command. The Apple product will then send a `+CIEV` event when there is a change in status of an indicator. The remote accessory must request the initial status using the `AT+CIND=?` and `AT+CIND?` commands, according to the HFP specification.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.2.2.3 Voice Recognition Activation

Every accessory that is compatible with an Apple product and supports HFP must:

- Support Voice Recognition Activation, both AG and HF initiated as described in Section 4.25 in the Bluetooth *Hands-Free Profile* specification version 1.5.
- Set the "Voice recognition activation" bit in the "SupportedFeatures" bitmap sent with the `AT+BRSF=` command.

Apple products support voice recognition initiated by remote (Hands-Free) accessories and iOS (Audio Gateway) accessories.

18.2.2.4 Echo Cancellation and Noise Reduction

When echo cancellation and noise reduction are performed locally on a Hands-Free accessory, it must turn off echo cancellation and noise reduction on the Apple product by sending an `AT+NREC` command, as described in Section 4.24 in the Bluetooth *Hands-Free Profile* specification version 1.5.

Apple products support echo cancellation and noise reduction; these features are active by default. If a Hands-Free accessory also does echo cancellation and noise reduction it needs to turn these features off on the Apple product (the Audio Gateway). This avoids unnecessary degradation of audio quality due to double audio processing.

18.2.2.5 In-Band Ringing

Every accessory that is compatible with an Apple product and supports HFP must also support In-Band Ringing as specified in Section 4.13.1 in the Bluetooth *Hands-Free Profile* specification version 1.5. If the user sets a ring tone on the Apple product, the same ring tone must sound on the hands-free accessory.

18.2.2.6 Synchronous Connection

Every accessory that is compatible with an Apple product and supports HFP must:

- Support eSCO parameter set S2 and S3 and accept requests for these settings. See Section 5.6 of the Bluetooth *Hands-Free Profile* specification version 1.5.
- Request eSCO parameter set S2 or S3 when setting up a Synchronous Connection. Note that eSCO parameter set S1 must not be requested.
- Render audio within 40 ms after the SCO/eSCO connection has been set up.

The eSCO packet types offers retransmission of packets; traditional SCO packets are not retransmitted. This improves audio quality and the user's experience. The eSCO packet types 2-EV3 and 3-EV3 offer a greater time interval between packets, which can improve Wi-Fi performance and allow time for other concurrent Bluetooth

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

connections to send data. Apple strongly recommends the use of 2-EV3 and 3-EV3 packets for SCO connections. Using HV3 packets is highly discouraged. HV3 packets require more link time and does not allow for retransmission of audio packets which impacts the audio performance in presence of RF interference.

18.2.2.7 Wide Band Speech

Every accessory that is compatible with an Apple product and supports HFP must support a Wide Band Speech Connection as described in Section 5.7.4 of the Bluetooth *Hands-Free Profile* specification version 1.6. If Wide Band Speech Connection is supported, it must support the T2 link parameter settings.

All Apple devices running iOS 5 or later support Wide Band Speech. If both the Apple device and the accessory support Wide Band Speech then Wide Band Speech link will be used for eSCO connection for use cases like cellular calls, FaceTime and Siri.

18.2.3 Message Access Profile (MAP)

Every accessory that is compatible with an Apple product and supports MAP must:

- Support Message Notification as described in Section 4.1 of the Bluetooth *Message Access Profile* specification, version 1.0.
- Register for notifications immediately after the connection is established, as described in Section 4.5 in the *Message Access Profile* specification, version 1.0.
- Not expect the TEL property to be present in the originator VCARD (the properties N and FN will be included). See Section 3.1.3 in the *Message Access Profile* specification, version 1.0.
- Not provide a user interface for sending messages. Apple devices do not support sending messages using MAP.

All Apple devices running iOS 6.0 or later support MAP.

18.2.4 Audio/Video Remote Control Profile (AVRCP)

Every accessory that is compatible with an Apple product and supports the Audio/Video Remote Control Profile must meet the requirements of the Bluetooth *Audio/Video Remote Control Profile* specification, Version 1.4. Additional Apple requirements are specified in this section.

18.2.4.1 Supported Operations

Apple products support the following operation_IDs in Pass Through commands:

- Play
- Stop

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Pause
- Fast Forward
- Rewind
- Forward
- Backward

18.2.4.2 Repeat and Shuffle Modes

Every Apple device supports Repeat and Shuffle modes in the role of an AVRCP target. An AVRCP controller may use `SetPlayerApplicationSettingValue` to set a value on the Apple device and `GetPlayerApplicationSettingValue` to read a value, as described in Sections 6.5.4 and 6.4.3 of the Bluetooth *Audio/Video Remote Control Profile* specification version 1.4.

18.2.4.3 Notifications

Every accessory that is compatible with an Apple product and supports AVRCP must register for notifications and not perform repetitive polling to determine the status of the Apple product.

Every Apple device supports registering for notifications in the role of an AVRCP Target, as described in Section 6.7 of the Bluetooth *Audio/Video Remote Control Profile* specification version 1.4. The commands `RegisterNotification` and `GetPlayStatus` are supported for these notifications:

- EVENT_PLAYBACK_STATUS_CHANGED
- EVENT_TRACK_CHANGED
- EVENT_NOW_PLAYING_CONTENT_CHANGED
- EVENT_AVAILABLE_PLAYERS_CHANGED
- EVENT_ADDRESSED_PLAYER_CHANGED
- EVENT_VOLUME_CHANGED

Accessories must use AVRCP notifications to present Apple device state to the user in accordance with the requirements in Presentation of Apple Device Updates (page 61).

18.2.4.4 Play/Pause Button

All accessories that support AVRCP and implement a Play/Pause button must confirm the playback status of the Apple device via AVRCP notifications (see Notifications (page 351) before sending a Play or Pause command (see Supported Operations (page 350)). Specifically:

- If the Apple device has notified the accessory that it is paused, pressing the accessory's Play/Pause button must send a Play command.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- If the Apple device has notified the accessory that it is playing, pressing the accessory's Play/Pause button must send a Pause command.
- The accessory must not infer Apple device playback status based on the number of times the Play/Pause button has been pressed.

18.2.4.5 Volume Handling

Every accessory that is compatible with an Apple product and supports AVRCP must support Absolute Volume, as described in Section 6.13 of the Bluetooth *Audio/Video Remote Control Profile* specification version 1.4.

Every Apple device supports volume handling in the role of AVRCP Controller.

18.2.4.6 Browsing

Every accessory that is compatible with an Apple product and supports Browsing (in controller role) as part of AVRCP must:

- Not try to index or cache the entire library upon connection. The Apple product may contain tens of thousands of media items, each present multiple times in the hierarchy.
- When browsing a specific folder, do not fetch all its items. Only fetch those that are displayed to the user. It may prefetch a few items to improve the responsiveness of the user interface.
- Not reorder items (e.g. alphabetically).
- Not assume UIDs to be statically defined, especially in the root folder. The ordering and UIDs of folders and items may change at any point in future releases.
- Send the `SetBrowsedPlayer` command after receiving an EVENT_UIDS_CHANGED notification.
- Not assume that the UID passed to the PlayItem command will result in the media player playing that UID.

Currently only the built-in Music app supports browsing. When switching between players, an EVENT_AVAILABLE_PLAYERS_CHANGED notification and an EVENT_ADDRESSED_PLAYER_CHANGED notification will be generated. The UI then needs to look at the feature bit mask of the listed player to determine whether browsing is currently available.

All Apple devices running iOS 6.0 or later support AVRCP Browsing.

18.2.4.7 iOS App-Provided Metadata

An audio app running on an Apple device may use the iOS MediaPlayer Framework APIs to provide metadata about the current audio stream. The Apple device supplies this metadata to the accessory using AVRCP. For more information, see the `MPNowPlayingInfoCenter` class in Apple's MediaPlayer Framework documentation.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.2.5 Advanced Audio Distribution Profile (A2DP)

Every accessory that is compatible with an Apple product and supports the Advanced Audio Distribution Profile must meet the requirements of the Bluetooth *Advanced Audio Distribution Profile* specification, Version 1.2. Additional Apple requirements are specified in this section.

18.2.5.1 SubBand Codec (SBC)

The SBC Codec Specific Information Elements, defined in Section 4.3.2 of the A2DP specification, that are applicable to Apple products are listed in Table 18-1 (page 353).

Table 18-1 SubBand Codec Information Elements for Apple products

Element	Value
Sampling Frequency	44,100 Hz
Channel Mode	Stereo
Block Length	16
Subbands	8
Allocation Method	Loudness
Bitpool range	2 to 53. Accessories for Apple products must support 53.

18.2.5.2 MPEG 2/4 AAC Codecs

Apple devices support the non-mandatory codec MPEG-2/4 AAC, as defined in Section 4.5 of the *Advanced Audio Distribution Profile* specification, Version 1.2. Accessories must use the AAC codec in addition to SBC, because it provides higher audio quality for a given bit rate.

> **Note:** The following specifications provide details of Apple's implementation of the MPEG-2/4 AAC codec. In case of conflicts, the A2DP specification governs.

The MPEG 2/4 AAC Codec Specific Information Elements, defined in Section 4.5 of the A2DP specification, that are applicable to Apple devices are listed in Table 18-2 (page 353).

Table 18-2 MPEG-2/4 AAC Codec Information Elements for Apple devices

Element	Value
Object Type	MPEG-2 AAC LC

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Element	Value
Sampling Frequency	44,100 Hz
Channels	2
Bit rate	264,630 bps
VBR	0

AAC audio stream packets in Apple devices have the structure shown in Table 18-3 (page 354).

Table 18-3 AAC audio packet for Apple devices

L2CAP	AVDTP	MPEG-4 LATM	MPEG-4 AAC
Header	Header	AudioMuxElement	Audio Payload

The AAC Media Payload Format, as defined in Section 4.5.4 of the A2DP specification, is formatted using LATM, as defined in Section 4 of *IETF RFC 3016* . The following notes apply to the packet fields shown in Table 18-3 (page 354).

- The suggested L2CAP MTU value for each Apple device's AAC streaming channel is 885 bytes.
- The AVDTP Header is shown as the RTP header in Figure 4 of RFC 3016, and is the header defined in Section 7.2.1 of the Bluetooth *Audio/Video Distribution Transport Protocol* , Version 1.2.
- The `AudioMuxElement` is the same as the RTP payload in RFC 3016. It is defined in Section 1.7.3, Table 1.32 in ISO/IEC 13818-3:2005, subpart 1. The `muxConfigPresent` argument to the `AudioMuxElement` is set to 1 (in-band mode), as recommended in Section 4.1 of RFC 3016. As recommended in Section 4.3 of RFC 3016, only one `AudioMuxElement` is put into each AVDTP packet.
- The audio payload is encoded using MPEG-4, as recommended in Section 4.5.4 of the A2DP specification.
- For AAC-LC support, the accessory must support VBR capability. The Apple device will be varying AAC bit rate depending on the content and the accessory must be able to handle the variation without causing gap in the audio.

18.3 Audio Routing

This section describes how an accessory can differentiate between various audio contents coming from an Apple device and use this information to decide playback behavior.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

An accessory can receive audio data from the Apple device via either of two Bluetooth profiles:

- HFP using eSCO channel
- A2DP using ACL channel

The Apple device picks which channel to use depending on how the audio content is used. An audio path created for two way communication (such as phone calls or FaceTime) always uses the HFP (eSCO) route for sending audio data. Music and similar content uses the A2DP route. In the absence of a defined route, audio playback will default to the Apple device.

If the accessory is already connected to the Apple device using the Lightning or 30-pin connector, connecting using Bluetooth must not pause any playing audio.

18.3.1 Audio Data Received via HFP Profile

Most of the audio content sent via HFP (eSCO) routes requires two way communication. Cases where HFP (eSCO) is used include (but are not limited to) cellular calls, FaceTime, and voice mail.

For any audio content that is being received via the HFP (eSCO) route, it is expected that both the speaker and the microphone of the accessory are dedicated to the Bluetooth link and must not handle any other audio content.

18.3.2 Audio Data Received via A2DP Profile

Audio content transferred via A2DP profiles can be broadly classified into two categories:

- Audio content from music, video, or game-like applications.
- System-generated sound for alerts and notifications.

18.3.2.1 Differentiating Audio Content from System Sounds

Music-like content can be differentiated from system sound by adding support for Audio/Video Remote Control Profile (AVRCP) version 1.3 or later. The AVRCP profile allows an accessory to be aware of the audio playback state in the Apple device, using notifications. See Audio/Video Remote Control Profile (AVRCP) (page 350).

When an Apple device initiates audio playback over an A2DP channel for playing music content, an AVRCP notification EVENT_PLAYBACK_STATUS_CHANGED is sent to indicate that playback status has changed to play state. See Section 6.7.2 of the *Audio/Video Remote Control Profile* specification, version 1.4. This indicates that audio data via the A2DP profile contains music. When an Apple device initiates audio playback over an A2DP channel for playing system sound, no AVRCP notification is sent.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 18-1 (page 356) and Figure 18-2 (page 356) show the difference between the notifications for music playback and for system sounds.

Figure 18-1 Initiate Audio Playback (e.g. music)

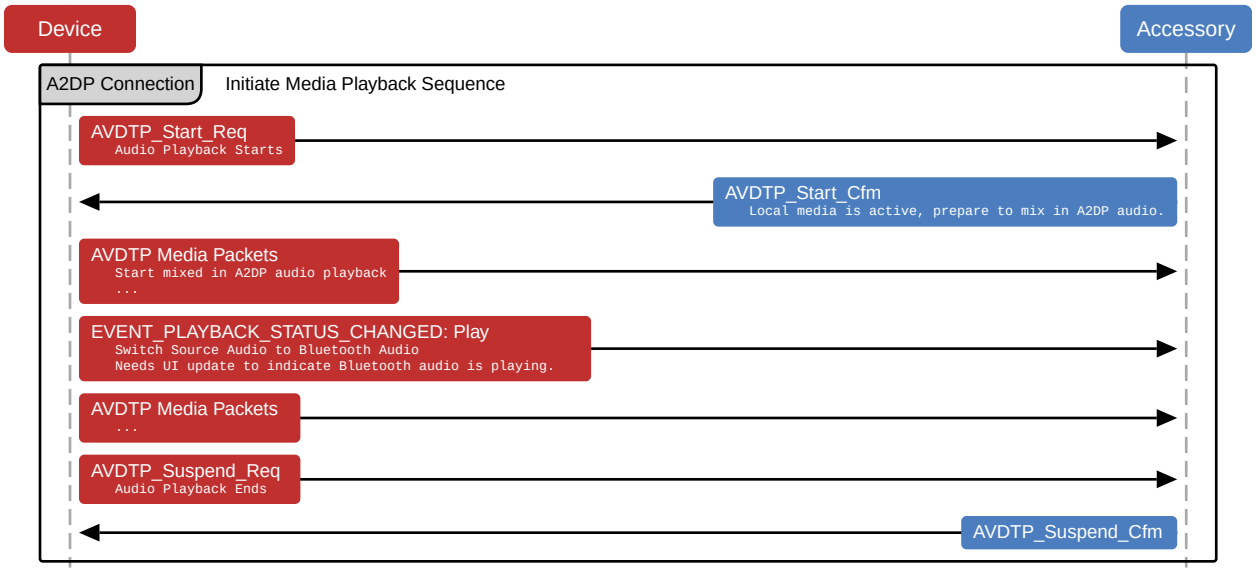
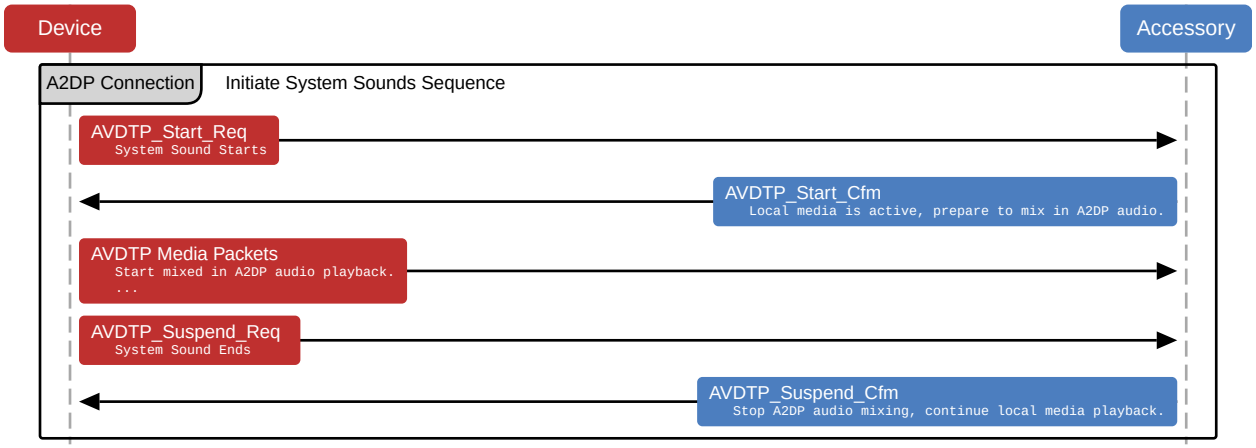


Figure 18-2 Initiate System Sound (e.g. turn-by-turn directions)



18.3.2.2 Expected Audio Routing Behavior for A2DP

The accessory must tune its audio routing behavior based on audio content over A2DP channel.

If audio data contains music, then it is expected that the accessory speakers are dedicated to audio data coming via the Bluetooth link and any other audio playback is paused. If audio data contains system sound, then it is expected that the accessory can render audio as desired. If the accessory is playing audio from a different source, then system sound data can be mixed with the existing track for playback; it is not necessary to pause existing audio playback on the device.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.4 iAP2

Accessories that implement iAP2 over the Bluetooth transport must additionally meet the following requirements:

- The Bluetooth Service Discovery Protocol (SDP) must be supported.
- The SDP data Maximum Transmission Unit (MTU) must be at least 672 bytes.
- SDP records must not be fragmented.
- Extended Inquiry Response (EIR) must be supported.
- A Service Class UUID of `0xFFCACADEAFDECADEDEFACADE00000000` must be declared in both SDP and EIR.
- The EIR Local Name must be the same as the `Name` parameter in the IdentificationInformation (page 807) message.

Unlike iAP1, there is no requirement for a specific Class of Device (CoD) or Major Service.

Apple recommends the following for a successful Bluetooth connection:

- Implement Bluetooth Sniff Mode, or Sniff Subrating if Bluetooth 2.1 is being used.
- Let the Apple device be the master device.
- For large packet transfers, stay within a Maximum Transmission Unit (MTU) size of 1000 bytes and a minimum of 658 bytes at the iAP2 layer.

Unlike wired transports, Apple devices will enter a hibernate state regardless of how many active Bluetooth connections are present. Bluetooth traffic from the accessory to the device will cause the device to exit the hibernate state. Therefore, accessories must generate Bluetooth traffic to an Apple device only in response to a direct user action. Autonomous generation of Bluetooth traffic for the purpose of keeping an Apple device out of hibernate is grounds for failure to pass self certification.

To re-establish a Bluetooth connection to the Apple device, the accessory must make a Bluetooth SDP query to find the RFCOMM channel associated with the Service Class UUID of `0xFECACADEAFDECADEDEFACADE00000000`, then connect to that channel. The accessory must not assume that the channel will remain the same between connections.

The Apple device may refuse a Bluetooth connection under these circumstances:

- There are too many Bluetooth connections made to the Apple device. In this case, the accessory will receive an error indicating that a resource is unavailable.
- There are too many RFCOMM protocol connections to the Apple device. In this case, the accessory will receive a Resource Denied error.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The accessory must not expect that the Apple device will try to re-establish a broken Bluetooth connection.

> **Note:** Service Class UUID and RFCOMM UUID are listed in little-endian order.

18.5 Test Procedures

18.5.1 Pairing and Connection Establishment

18.5.1.1 Apple Device to Accessory

1. Search for accessory from Apple device and initiate paring.
2. Pair should be successful after exchanging pin code.

18.5.1.2 Car Kit to Apple Device

1. Search for Apple device from accessory and initiate paring (Apple device needs to be in Bluetooth menu).
2. Pair should be successful after exchanging pin code.

18.5.1.3 Sniff Mode

1. Leave the connection idle for 30 seconds.
2. Device should enter sniff mode upon request.

18.5.2 Reconnection

18.5.2.1 Reconnect from Apple Device

1. Toggle Airplane Mode and reconnect BT from Apple device.
2. Reconnection should be successful.

18.5.2.2 Reconnect from Accessory

1. Toggle Airplane Mode and reconnect BT from accessory.
2. Reconnection should be successful.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.5.2.3 Reconnect During Active Call

1. Power cycle accessory during active call.
2. Reconnection should be successful. Audio should be available in uplink and downlink.

18.5.2.4 Range

1. Take Apple device out of range during active call. Return to BT range after 2 minutes and initiate BT connection from Apple device/accessory.
2. Audio should be routed to Apple device after disconnection. Audio should be available in uplink and downlink after reconnection.

18.5.3 Indicators

1. Check the Human Machine Interface (HMI) after BT connection.
2. Support indicators should display correct values for: Battery, Signal Strength, Roaming Indicator, and Carrier Name.

18.5.4 Outgoing Call

18.5.4.1 Dialed from Accessory

1. Enter phone number on accessory's HMI.
2. Dialing tone should be heard over SCO/eSCO link before call is answered by the remote party. Audio should be available in uplink and downlink once call is answered.

18.5.4.2 Cancel Call from Accessory

1. Enter phone number on accessory's HMI. Wait for dialing to start and press End Call.
2. Dialing tone should be heard over SCO/eSCO link.

18.5.4.3 Dialed from Apple Device

1. Enter phone number on Apple device.
2. Dialing tone should be heard over SCO/eSCO link before call is answered by the remote party. Audio should be available in uplink and downlink once call is answered.

18.5.4.4 Dialed from Downloaded Call History

1. On accessory, dial to an entry in downloaded call history.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Dialing tone should be heard over SCO/eSCO link before call is answered by the remote party. Audio should be available in uplink and downlink once call is answered.

18.5.4.5 Audio Transfer

1. From accessory/Apple device, transfer audio between Apple device and accessory.
2. Audio should be routed to the correct source after each action.

18.5.4.6 Audio Quality

1. Maintain a call for 10 minutes and check audio quality.
2. Audio quality should be decent in uplink and downlink.

18.5.5 Incoming Call

18.5.5.1 In-Band Ringtone

1. Make incoming call and check audio quality for in-band ringtone.
2. Ring tone should be complete and should sound clear.

18.5.5.2 Caller ID

1. Make incoming call and check caller ID on accessory's HMI.
2. Caller ID should be displayed in a good format.

18.5.5.3 Call Rejection

1. Make incoming call and reject from accessory/Apple device.
2. Call should be rejected.

18.5.5.4 Answer Call from Accessory

1. Make incoming call and answer from accessory.
2. Audio should be available in uplink and downlink once call is answered.

18.5.5.5 Answer Call from Apple Device

1. Make incoming call and answer from Apple device.
2. Audio should remain on Apple device (Privacy mode).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.5.5.6 Audio Transfer

1. From accessory/Apple device, transfer audio between Apple device and accessory.
2. Audio should be routed to the correct source after each action.

18.5.5.7 Audio Quality

1. Maintain a call for 10 minutes and check audio quality.
2. Audio quality should be decent in uplink and downlink.

18.5.6 FaceTime Audio

18.5.6.1 Incoming FaceTime Audio Call

1. Use the same steps as Incoming Call (page 360).

18.5.6.2 Outgoing FaceTime Audio Call

1. Use the same steps as Outgoing Call (page 359).

18.5.7 Three Way Call

18.5.7.1 Single Call Hold/Resume

1. During a call, press Hold / Resume on HMI.
2. Call should be held / resumed after each action.

18.5.7.2 Ignore Second Incoming Call

1. During an active call, press Reject on HMI to ignore the second incoming call.
2. Second call should be rejected. First call should remain unchanged.

18.5.7.3 Replace Call

1. During an active call, press Replace to end current call and answer the second call.
2. Second call should be answered. First call should be ended.

18.5.7.4 Answer Second Call

1. During an active call, press Answer to the second incoming call.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Second call should be answered. First call should be put on hold.

18.5.7.5 Call Swap

1. Swap between active and held calls from HMI.
2. Call status should change in response to each action.

18.5.7.6 Conference

1. Join 2 calls from HMI. Make sure Hold/Resume/End functions work for conference calls.
2. Each party should be able to hear audio from the other two. Hold/Resume/End functions should work for conference calls.

18.5.8 Enhanced Call Control

18.5.8.1 Specify a Call to End in a Three Way Call

1. Form a conference call and specify to end one of the calls.
2. The specified call should end.

18.5.8.2 Specify a Call to Hold in a Three Way Call

1. Form a conference call and specify to put on call on hold (split the conference).
2. The specified call should become held from active.

18.5.9 Visual Voice Mail

18.5.9.1 VVM Playback

1. Start VVM playback.
2. Virtual call should be displayed by HMI. Audio should be clear.

18.5.9.2 VVM Playback During A2DP

1. Start VVM playback during A2DP streaming.
2. Virtual call should start and should remain undisrupted. Audio should be clear.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.5.9.3 Call Back During VVM Playback

1. Play VVM and press Call Back on Apple device.
2. Transition from VVM to outgoing call should be smooth.

18.5.9.4 Incoming Call During VVM Playback

1. Play VVM and make an incoming call.
2. Transition from VVM to incoming call should be smooth.

18.5.10 Siri

1. Trigger Siri from either Apple device or accessory.
2. Beeps should be complete and sound clear.
3. Ask Siri a question (time, weather, etc.).
4. Audio should be complete and sound clear. Virtual call should end soon after Siri provides the answer.
5. Ask Siri to call someone.
6. Transition from Siri to outgoing call should be smooth.
7. Ask Siri to play certain tracks.
8. Siri should find the right track to play.
9. Ask Siri for directions (Starbucks, Whole Foods, etc).
10. Siri should find the directions and start turn-by-turn navigation.

18.5.11 A2DP

18.5.11.1 Audio Quality

Assess the quality of the audio signal in each of the following scenarios:

1. Stream music from Music app.
2. Stream music using iTunes Radio.
3. Stream audio using Podcast app.
4. Stream audio using Beats Music app.
5. Stream audio using 3rd party apps (Pandora, Spotify, etc).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.5.11.2 Audio Switching

1. During A2DP streaming, switch audio back to Apple device and switch back to accessory.
2. Audio should be routed to the intended source. Audio quality should be good switching back to BT.

18.5.11.3 HFP Interaction

1. Make incoming / outgoing call during A2DP.
2. Audio should be suspended during the call and resume after the call.

18.5.11.4 Siri

1. Trigger Siri during A2DP.
2. Audio should be resumed after the Siri session.

18.5.11.5 Video Playback

1. Stream A2DP while watching a video.
2. Audio / video synchronization and quality should be good.

18.5.12 AVRCP

18.5.12.1 AVRCP Media Control

1. Play/Pause, Next/Previous, FF/RW.
2. All controls should work when selected on the HMI.

18.5.12.2 Absolute Volume

1. Volume Up/Down.
2. Volume on Apple device and accessory should be synchronized.

18.5.12.3 AVRCP Settings Control

1. Shuffle, Repeat.
2. All controls should work when selected on the HMI.

18.5.12.4 Metadata

1. Switch between Music app, iTunes Radio and 3rd party apps.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Metadata should always work after each transition, no matter which app is being used.

18.5.12.5 AVRCP Browsing

1. Browse the folder structure and media from HMI. Select a track and a station in iTunes Radio to play.
2. Apple device's media folder structure should be browsable on HMI.

18.5.12.6 AVRCP Version

1. Connect accessory with phones that support higher AVRCP versions (1.5, 1.6, etc).
2. AVRCP functionalities should remain (Control, Metadata, browsing, etc).

18.5.13 Turn-by-turn Navigation

18.5.13.1 Turn-by-turn Navigation During A2DP Streaming

1. Stream A2DP. Start turn-by-turn navigation.
2. Turn-by-turn prompts should be audible during A2DP streaming. Prompts should be complete and sound clear.

18.5.13.2 Turn-by-turn Navigation when A2DP is Paused

1. Stream A2DP. Pause music. Start turn-by-turn navigation.
2. Turn-by-turn prompts should be audible with music in pause state. Prompts should be complete and sound clear.

18.5.13.3 Turn-by-turn Navigation Over HFP

1. Enable Maps Prompts over HFP. While listening to FM, start turn-by-turn navigation.
2. Virtual call should appear for each turn-by-turn prompt. Prompts should be complete and sound clear.

18.5.14 Phonebook

18.5.14.1 Contacts

1. Download phonebook from accessory.
2. Size and contents of the downloaded phonebook should match with Apple device phonebook.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.5.14.2 Call History

1. Download call history from accessory.
2. Size and contents of the downloaded call history should match with Apple device call history (ich, och, mch).

18.5.14.3 Update

1. Add/Delete a contact/call history from Apple device. Download again from accessory.
2. Updated should be seen in the downloaded phonebook/call history on HMI.

18.5.15 iAP

The following reviews how to setup and capture iAP-over-Bluetooth with ATS for MFi accessory audits. This section covers only procedures and test cases for iAP1 and iAP2 over the Bluetooth transport. It does not cover any additional Bluetooth profile testing.

18.5.15.1 Equipment

To run iAP-over-Bluetooth accessory audits, you will need the following:

- Mac with ATS 4.1 or later
- ATS Utility 1.1.1 or later installed on an Apple device running iOS 8.0 or later
- ComProbe BPA 100
- RF shielding cloth
- Prop or stand for RF shielding cloth

18.5.15.2 Setting up the test environment

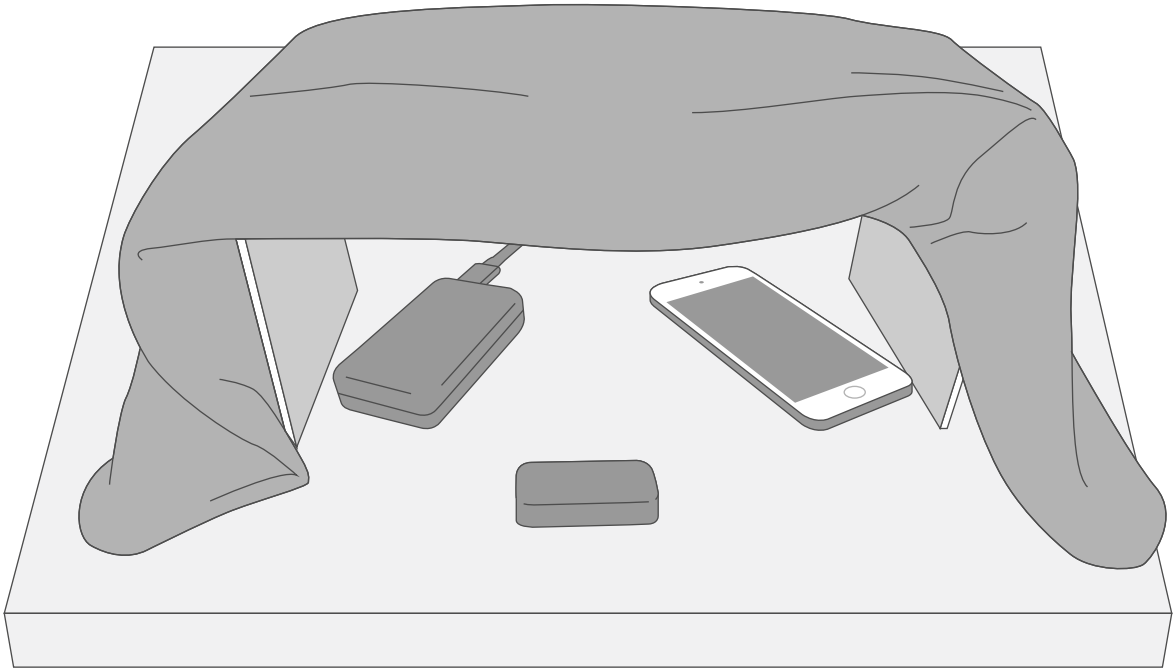
Due to the inherent nature of capturing data wirelessly, ATS may not be able to always correctly capture the communication between the Apple device and accessory. If there is too much RF interference, or the Apple device and accessory are placed incorrectly with respect to the ComProbe you may see missing Bluetooth data in the ATS trace. To help alleviate these issues:

1. Make sure that the ComProbe, Apple device and the accessory are placed to form an equilateral triangle during capture.
2. Use RF cloth to shield the ComProbe, Apple device and the accessory from RF interference. Use a stand or prop to hold the cloth above the hardware. This will allow you to continue to interact with the Apple device and accessory during the capture.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Make sure that the RF environment is free of noise by turning off other devices that may use the 2.4 GHz frequency range such as Wi-Fi and other Bluetooth devices.

Figure 18-3 RF Interference Test Setup



18.5.15.3 Capturing iAP-over-Bluetooth

The first time you setup an iAP-over-Bluetooth capture, you will have to install drivers to your Mac. Additionally, you may have to update the firmware on your ComProbe.

1. Upon selecting the ComProbe in the Capture Configuration Assistant, you will be promoted to install the ComProbe driver. Select Install.
2. Run the installer to completion. Once the driver installation has completed you must restart your computer.
3. After restart, launch ATS and select the ComProbe. If the connected ComProbe contains older firmware, you will be prompted to update it. If promoted to do so, select Update.
4. Select "Install" to install the driver for the ComProbe firmware updater app.
5. The firmware updater app will launch in a separate window. Follow the on-screen instructions to update the ComProbe's firmware. This may take a few minutes.
6. When finished select "Done" and the firmware updater app will close. You are now ready to start capturing iAP-over-Bluetooth with ATS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.5.15.4 Capturing iAP-over-Bluetooth with ATS

Follow these instructions to setup an iAP-over-Bluetooth capture with the Capture Configuration Assistant. You may also setup Bluetooth captures from the Advanced Capture Configuration Assistant.

1. Select ComProbe BPA 100. Note: if more than one ComProbe is connected to your Mac, only one will be displayed in the Capture Configuration Assistant screen.
2. Select "New Bluetooth capture configuration".
3. Make sure your Apple device's Bluetooth is on and the device is discoverable. Select Start Inquiry to allow the ComProbe to look for all discoverable Bluetooth devices. Select your Apple device from the list of devices.
4. Make sure your accessory's Bluetooth is on and the device is discoverable. Select "Start Inquiry" to allow the ComProbe to look for all discoverable Bluetooth devices. Select your accessory from the list of devices.
5. Follow the on-screen instructions. Note that only accessories that use iAP will show a Link Key in ATS Utility app.
6. Select the iAP version the accessory supports.
7. Follow the on-screen instructions.
8. After selecting "Start Capture", switch the Apple devices Bluetooth to "On" and tap on the accessory's name to Connect. You will see iAP traffic in the capture window. Continue to interact with the Apple device and accessory as you would in accessory audits with wired accessories.

18.5.15.5 Special Considerations

Some accessories will only utilize iAP over Bluetooth for specific accessory interactions (for example, EA Session over iAP). For accessories such as these, a Link Key will not be viewable in ATS Utility until the iAP communication is initialized.

This behavior has been observed in automotive head units, speakers and receivers.

If you are unable to view a Link Key for an accessory which claims support for both iAP over Bluetooth and EA Session, please attempt the following.

1. After connecting the device to the accessory, launch the associated application and use the app.
2. Reopen ATS Utility and check for a Link Key.
3. With the Link Key, proceed with BT audit as instructed above.

Note that, similar to the way you will not get a Link Key until the EA Session has been triggered, you will not see iAP traffic until the application is launched.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18.5.15.6 EA string matching

Verify that the application protocol strings match the accessory protocol strings SDK Apps (page 540).

18.5.15.7 Autonomous app launch

Verify that the accessory only sends RequestAppLaunch (page 820)after the user has initiated the launch of the iOS app. A nag screen asking the user to "Allow" the use of the app must be presented after direct user action.

1. Verify that RequestAppLaunch (page 820) is only sent from the accessory in response to direct user action, such as pushing a button on the accessory App Launch (page 338). Do note that the automotive head units do not respond the same way.
2. If the accessory is iAP2-over-Bluetooth and sends RequestAppLaunch (page 820) LaunchAlert, verify that AppLaunchMethod is set to 0.

18.5.15.8 Power parameter test cases

For Bluetooth accessories that do not draw or provide power to an Apple device, verify that the accessory does not claim support for power via IdentificationInformation messages.

- Table 59-10 (page 810) must be set to None.
- MaximumCurrentDrawnFromDevice parameter must be set to 0. IdentificationInformation (page 807).

18.5.15.9 Standard test cases

Execute all additional applicable test procedures within this document.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

19. Bluetooth Accessory Identification

This chapter describes Apple-specific Bluetooth commands that extend accessory capabilities beyond those supported by standard Bluetooth profiles.

To enable Apple-specific features, the accessory must support HFP Command AT+XAPL (page 370), which provides accurate information about the accessory's supported features. The Apple device will use the information sent by this command to enable and disable custom commands.

The accessory must send the following AT+XAPL command after making a successful HFP Service Level Connection (SLC) to the Apple device. The accessory must send an AT+XAPL command first, before sending any of the additional Apple-specific commands described below.

19.1 HFP Command AT+XAPL

Description: Enables custom AT commands from an accessory.

Initiator: Bluetooth accessory

Format: AT+XAPL=*vendorID*-*productID*-*version*,*features*

Parameters:

- *vendorID*: A string representation of the hex value of the vendor ID from the manufacturer, without the `0x` prefix.
- *productID*: A string representation of the hex value of the product ID from the manufacturer, without the `0x` prefix.
- *version*: The revision of the software.
- *features*: A base-10 representation of a bit field. Available features are:
  - Bit 0 = reserved
  - Bit 1 = The accessory supports battery reporting (reserved only for battery operated accessories).
  - Bit 2 = The accessory is docked or powered (reserved only for battery operated accessories).
  - Bit 3 = The accessory supports Siri status reporting.
  - Bit 4 = the accessory supports noise reduction (NR) status reporting.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- All other values are reserved.

Example: AT+XAPL=ABCD-1234-0100,10 (Supports battery reporting and Siri status)

Response: +XAPL=iPhone,*features*

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

20. Bluetooth Connection

Accessories can automate the Bluetooth pairing process between an Apple device and any number of integrated Bluetooth accessory components if they also have a wired connection to the Apple device.

Additionally, some Apple devices can report changes in the Bluetooth connection status for an accessory's Bluetooth components. These notifications can be useful when the accessory wishes to determine whether it is connected to a particular Apple device via both Bluetooth and a Lightning connector. Accessories must implement this feature if they can maintain audio transport connections over both Bluetooth A2DP and other transports (see Multiple Audio Connections (page 528)).

20.1 Bluetooth Connection Requirements

The accessory must declare at least one Bluetooth component during identification to use this feature. The component(s) must comply with the requirements in Bluetooth (page 344).

The accessory must send an initial BluetoothComponentInformation (page 822) message within 2 seconds of receiving a IdentificationAccepted (page 816) message. Additionally, the accessory must also send a BluetoothComponentInformation (page 822) message to the device whenever the state of an identified accessory Bluetooth component changes. The accessory must report status for every identified Bluetooth component and not just status for the changed component. Conversely, after the initial BluetoothComponentInformation (page 822) message is sent, the accessory must not send additional ones unless there is a state change in one of its identified Bluetooth components.

In the case of Bluetooth components, the accessory must report the `ComponentIdentifier` of the identified component and `ComponentEnabled` parameter value of `true` if the Bluetooth component is ready for connections. Conversely, the accessory must send a BluetoothComponentInformation (page 822) message with `ComponentEnabled` set to `false` if the Bluetooth component is not ready for connections.

All accessories that support the Bluetooth Connection feature via iAP2 must send or receive the following iAP2 control session message(s):

- BluetoothComponentInformation (page 822)
- StartBluetoothConnectionUpdates (page 822)
- BluetoothConnectionUpdate (page 823)
- StopBluetoothConnectionUpdates (page 824)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

20.2 Bluetooth Connection Usage

Pairing of Bluetooth components with an Apple device will automatically initiate if the accessory has successfully authenticated and the Apple device has not previously paired with the component. The accessory's Bluetooth component must be prepared to complete the pairing process as soon as the accessory has started identification.

The accessory must send a StartBluetoothConnectionUpdates (page 822) message to start the generation of component updates from the device. More than one component identifier may be specified if the accessory has multiple Bluetooth components with different MAC addresses. If the accessory later wishes to stop receiving component status notifications it may send a StopBluetoothConnectionUpdates (page 824) message to the device.

When updates are first started, one or more initial BluetoothConnectionUpdate (page 823) messages will be sent to the accessory to inform it of the starting connection state of all requested components. Subsequent notifications for a particular component always completely override prior ones. It is possible for the device to connect to the same component profile multiple times in a row. For example, this may occur when the device temporarily moves out of range for a period of time that is not long enough to trigger an automatic device disconnect and reconnects upon rediscovering the accessory component. Therefore, accessories must handle duplicate BluetoothConnectionUpdate (page 823) messages gracefully.

Accessories must send StopBluetoothConnectionUpdates (page 824) when they no longer have a need for them. For example, if an accessory makes use of Bluetooth connection updates purely to automate pairing of the accessory's Bluetooth component, it must stop asking for updates and not start them again upon subsequent connections to the Apple device.

20.3 Test Procedures

20.3.1 iAP2 Tests

1. Verify that accessories which maintain audio transport connections over both Bluetooth A2DP and other transports implement this feature.
2. Verify that the following iAP2 control session message(s) are sent or received:
   - BluetoothComponentInformation (page 822)
   - StartBluetoothConnectionUpdates (page 822)
   - BluetoothConnectionUpdate (page 823)
   - StopBluetoothConnectionUpdates (page 824)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

21. Bluetooth Headset Battery Level Indication

Any Hands-Free Bluetooth headset accessory can show its battery level to the user as an indicator icon in the Apple device status bar. This feature is supported on all Apple devices that support the Hands-Free Profile, including iPhone, iPod touch, and iPad.

Headset battery indication is implemented by two Apple-specific Bluetooth HFP AT commands, HFP Command AT+XAPL (page 370) and HFP Command AT+IPHONEACCEV (page 374)

21.1 HFP Command AT+IPHONEACCEV

Description: Reports a headset state change.

Initiator: Headset accessory

Format: `AT+IPHONEACCEV=`*Number of key/value pairs* `,`*key1* `,`*val1* `,`*key2* `,`*val2* `,...`

Parameters:

- *Number of key/value pairs*: The number of parameters coming next.
- *key*: the type of change being reported:
  - 1 = Battery Level
  - 2 = Dock State
- *val*: the value of the change:
  - Battery Level: string value between '`0`' and '`9`'
  - Dock State: `0` = undocked, `1` = docked

Example: `AT+IPHONEACCEV=1,1,3`

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

22. Bluetooth Low Energy

The *Bluetooth 4.0* specification introduces Bluetooth Low Energy, a new wireless technology targeted for accessories with limited battery resources. If Bluetooth Low Energy is supported, the accessory must follow the guidelines in this section.

22.1 Role

The accessory must implement either the Peripheral role as defined in the *Bluetooth 4.0* specification, Volume 3, Part C, Section 2.2.2.3 or the Broadcaster role, as defined in Section 2.2.2.1.

22.2 Advertising Channels

The accessory must advertise on all three advertising channels (37, 38, and 39) at each advertising event. See the *Bluetooth 4.0* specification, Volume 6, Part B, Section 4.4.2.1.

22.3 Advertising PDU

The accessory must use one of the following advertising PDUs:

- ADV_IND
- ADV_NOCONN_IND
- ADV_SCAN_IND

ADV_DIRECT_IND must not be used. See the *Bluetooth 4.0* specification, Volume 6, Part B, Section 2.3.1.

22.4 Advertising Data

The advertising data sent by the accessory must contain at least the following information as described in the *Bluetooth Core Specification Supplement*, Part A:

- Flags

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- TX Power Level
- Local Name
- Services

The Local Name must match the accessory's markings and packaging and not contain ':' or ';'.

The accessory may put the Local Name and the TX Power Level data in the SCAN_RSP PDU if, for example, it needs to reduce power consumption or not all of the advertising data fit into the advertising PDU. Note that, depending on its state, the Apple product may not always perform active scanning.

The primary services must always be advertised in the advertising PDU. Secondary services must not be advertised. Services not significant to the primary use case of the accessory may be omitted if space is limited in the Advertising PDU.

The advertising data and the scan response data in the SCAN_RSP PDU must comply with the formatting guidelines in the *Bluetooth 4.0* specification, Volume 3, Part C, Section 18: it starts with a length field, followed by AD Type and AD Data.

22.5 Advertising Interval

The advertising interval of the accessory must be carefully considered, because it affects the time to discovery and connect performance. For a battery-powered accessory, its battery resources must also be considered.

To be discovered by the Apple product, the accessory must first use the recommended advertising interval of 20 ms for at least 30 seconds. If it is not discovered within the initial 30 seconds, Apple recommends using one of the following longer intervals to increase chances of discovery by the Apple product:

- 152.5 ms
- 211.25 ms
- 318.75 ms
- 417.5 ms
- 546.25 ms
- 760 ms
- 852.5 ms
- 1022.5 ms
- 1285 ms

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: Longer advertising intervals usually result in longer discovery and connect times.

22.6 Connection Parameters

The accessory is responsible for the connection parameters used for the Low Energy connection. The accessory must request connection parameters appropriate for its use case by sending an L2CAP Connection Parameter Update Request at the appropriate time. See the *Bluetooth 4.0* specification, Volume 3, Part A, Section 4.20 for details. The connection parameter request may be rejected if it does not comply with all of these rules:

- *Interval Max * (Slave Latency + 1) ≤ 2 seconds*
- *Interval Min ≥ 20 ms*
- *Interval Min + 20 ms ≤ Interval Max*
- *Slave Latency ≤ 4*
- *connSupervisionTimeout ≤ 6 seconds*
- *Interval Max * (Slave Latency + 1) * 3 < connSupervisionTimeout*

If Bluetooth Low Energy HID is one of the connected services of an accessory, connection interval down to 11.25 ms may be accepted by the Apple product.

The Apple product will not read or use the parameters in the Peripheral Preferred Connection Parameters characteristic. See the *Bluetooth 4.0* specification, Volume 3, Part C, Section 12.5.

22.7 Privacy

The accessory must be able to resolve a Resolvable Private Address in all situations. Due to privacy concerns, the Apple product will use a Random Device Address as defined in the *Bluetooth 4.0* specification, Volume 3, Part C, Section 10.8.

22.8 Permissions

The accessory must not require special permissions, such as pairing, authentication, or encryption to discover services and characteristics. It may require special permissions only for access to a characteristic value or a descriptor value. See the *Bluetooth 4.0* specification, Volume 3, Part G, Section 8.1, fifth paragraph.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

22.9 Pairing

The accessory must not request pairing until an ATT request is rejected using the Insufficient Authentication error code. See the *Bluetooth 4.0* specification, Volume 3, Part F, Section 4 for details.

If, for security reasons, the accessory requires a bonded relationship with the Central, the Peripheral must reject the ATT request using the Insufficient Authentication error code, as appropriate. As a result, the Apple product may proceed with the necessary security procedures.

Similarly, if the Apple device acts as a Central and a GATT server, it may reject an ATT request using the Insufficient Authentication error code. The accessory must initiate the security procedure for pairing in response.

Pairing may require user authorization depending on Apple product. Once an accessory is paired with an Apple product, it must retain the distributed keys of both central and peripheral for future use. If the pairing is no longer required, the accessory must delete both sets of keys.

22.10 MTU Size

The Apple device supports and requests MTU size larger than default MTUs during the Exchange MTU Request handshake. See the *Bluetooth 4.0* specification, Volume 3, Part F, Section 3.2.8. The accessory may use this mechanism to negotiate larger MTU sizes.

22.11 Services

22.11.1 Generic Access Profile Service

The accessory must implement the Device Name characteristic per the *Bluetooth 4.0* specification, Volume 3, Part C, Section 12.1. The Device Name characteristic must be writeable.

22.11.2 Generic Attribute Profile Service

The accessory must implement the Service Changed characteristic only if the accessory has the ability to change its services during its lifetime.

The Apple product may use the Service Changed characteristic to determine if it can rely on previously read (cached) information from the device. See the *Bluetooth 4.0* specification, Volume 3, Part G, Section 7.1.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

22.11.3 Device Information Service

The accessory must implement the Device Information Service. The service UUID for this service must not be advertised in the Advertising Data. The following characteristics must be supported:

- Manufacturer Name String
- Model Number String
- Firmware Revision String
- Software Revision String

22.11.4 Available Services

With iOS 7.0, any Apple device makes Battery Service, Current Time Service and Apple Notification Center Service (ANCS) available to an accessory. The Current Time Service supports the current time and local time information characteristics. The service does not provide an "Adjust Reason" when the current time changes. ANCS uses 7905F431-B5CE-4E99-A40F-4B1E122D00D0 as its UUID.

These services are not guaranteed to be available immediately after connection and the accessory must support Characteristic Value Indication of the Service Changed characteristic (see *Bluetooth 4.0* specification, Volume 3, Part G, Section 7.1) to be notified when the services become available. The Apple device will maintain a connection to an accessory as long as it is paired and uses one of the available services.

22.12 GATT Server

With iOS 6.0, applications may contribute services and characteristics to the GATT server that the Apple device makes available to the accessory. The recommendations in this section apply to the accessory in this case.

The following services are implemented internally by iOS and must not be published by third party iOS applications:

- Generic Attribute Profile Service
- Generic Access Profile Service
- Bluetooth Low Energy HID Service
- Battery Service
- Current Time Service
- Apple Notification Center Service

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The Apple device implements the GAP Service Changed characteristic, because the database contents can change at any time. The accessory must therefore support the Characteristic Value Indication of this characteristic and, upon receiving indications, invalidate its database cache accordingly. See the *Bluetooth 4.0* specification, Volume 3, Part G, Section 7.1.

The accessory must minimize the use of ATT/GATT requests and commands and only send what is necessary. For example, do not use GATT Discover All Services when the accessory is looking for specific services. Use Discover Primary Service By Service UUID instead. Less airtime equals less power consumption and better performance for both the accessory and the Apple device.

When third party iOS applications discover services on the accessory, the following services are used internally by iOS and are filtered out from the list of discovered services:

- Generic Attribute Profile Service
- Generic Access Profile Service
- Bluetooth Low Energy HID Service
- Apple Notification Center Service

The accessory must be robust enough to handle any error gracefully. Pairing and Characteristic Value reads/writes may fail if the application that owns the service is not in the foreground and is not entitled to run in the background.

If an ATT Prepare Write Request is used, all queued attributes are contained within the same GATT Service.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23. CarPlay

This chapter defines how a vehicle infotainment system (head unit) may receive and interact with digital content (audio, video, and metadata) streamed from an Apple device running iOS. It covers accessories that can receive streamed digital content from Apple devices.

To work with Apple CarPlay™, accessories must support the CarPlay feature.

CarPlay is only allowed for dashboard-integrated vehicle infotainment systems. It is not intended for rear-seat or non-vehicle integrations. The accessory must be connected to the vehicle audio system and must use an external microphone.

CarPlay accessories must only be marketed in countries where the CarPlay feature is available. For a list of countries where the CarPlay feature is available, see http://www.apple.com/ios/feature-availability/#applecarplay-applecarplay.

All CarPlay accessories must comply with the *Apple CarPlay Identity Guidelines* .

In the context of CarPlay, the term accessory is used to refer to either a vehicle and its head unit or an after-market head unit.

After-market head units must be integrated systems with a built-in display with a minimum 6 inch diagonal screen size. After-market head units that rely on the customer, an installer, or another third party to ensure spec compliance should make those requirements clear in their installation manual, documentation, etc. Examples of this include instructions on connecting vehicle speed sensors and external GNSS antennas, microphone placement, and placement of the Apple device for thermal considerations.

23.1 Additional Specifications

For clarifications and detailed specifications, this chapter cites portions of the following external specifications:

- USB Specifications
  - USB 2.0 Specification (http://www.usb.org/developers/docs/usb20/)
  - Universal Serial Bus Communications Class Subclass Specifications for Network Control Model Devices (http://www.usb.org/developers/docs/devclass_docs/)
  - Universal Serial Bus Class Definitions for Communication Devices (http://www.usb.org/developers/docs/devclass_docs/)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- IETF RFCs
  - 793 (http://www.ietf.org/rfc/rfc793.txt)
  - 768 (http://www.ietf.org/rfc/rfc768.txt)
  - 2460 (http://www.ietf.org/rfc/rfc2460.txt)
  - 2474 (http://www.ietf.org/rfc/rfc2474.txt)
  - 3966 (http://www.ietf.org/rfc/rfc3966.txt)
  - 5905 (http://www.ietf.org/rfc/rfc5905.txt)
  - "Internet Protocol (IP), DARPA Internet Program, Protocol Specification", RFC 791, September 1981
  - "An Ethernet Address Resolution Protocol (ARP)", RFC 826, November 1982
  - "Unique Local IPv6 Unicast Address", RFC 4193, October 2005
  - "Dynamic Host Configuration Protocol (DHCP)", RFC 2131, March 1997
  - "Internet Control Message Protocol (ICMP), DARPA Internet Program, Protocol Specification", RFC 792, September 1981
  - "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006
  - "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 1007
  - K. McCloghrie, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", RFC 1213, March 1991
  - F. Kastenholz, "Definition of Managed Objects for the Ether-like Interface Types", RFC 1398, January 1993
- IEEE 802.11 Standards
  - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11-2012
  - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification: Enhancements for Very High Throughput in Bands below 6 GHz, IEEE 802.11ac-2013
- Wi-Fi Alliance Specifications
  - Wi-Fi 802.11 with WPA2 System Interoperability Test Plan for IEEE 802.11a, b & g Devices, Version 2.4.2, March 2005
  - Wi-Fi 802.11n System Interoperability Test Plan, Version 2.0.1
  - Wi-Fi CERTIFIED(tm) ac Interoperability Test Plan, Version 1.0.0
  - Wi-Fi WMM System Interoperability Test Plan, Version 1.3.2, June 2006
  - Wi-Fi Voice Personal Interoperability Test Plan Version 2.3.3
- Bluetooth SIG Specifications

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

    - Specification of the Bluetooth System, "Bluetooth v2.1 + EDR", February 2007
    - "Bluetooth Specification Version 4.1", December 2013
    - Supplement to the Bluetooth Core Specification, Version 4, Dec 2013
- Apple Bonjour Support Site (https://www.apple.com/support/bonjour/)
- ITU Specifications
    - ITU H.264 Specification (http://www.itu.int/rec/T-REC-H.264)
    - ITU-T P.1100: Narrow-band hands-free communication in motor vehicles (http://www.itu.int/rec/T-REC-P.1100-201103-I/)
    - ITU-T P.1110: Wideband hands-free communication in motor vehicles (http://www.itu.int/rec/T-REC-P.1110-200912-I/)

23.2 General Requirements

23.2.1 Overview

Apple CarPlay lets an Apple device stream interactive digital content to an accessory with one or more audio/video endpoints and controls.

CarPlay is intended for use with a single front seat display. An accessory must not replicate digital video content to multiple displays.

For more information on how to integrate the Apple CarPlay identity in vehicle hardware and software and in marketing communications, refer to the *Apple CarPlay Identity Guidelines* .

23.2.2 Hardware Requirements

This section contains physical requirements for an accessory that supports CarPlay.

23.2.2.1 High Resolution Display

The accessory must provide a high-resolution display capable of rendering the User Interface (UI) stream without scaling and with all pixels visible to the end user. The display must have a minimum resolution of 800 x 480, support 24 bits of RGB color per pixel, and 30 Hz refresh rate (60 Hz recommended). Note that future specifications may require at least 60 Hz refresh rate. Displays with square pixels are recommended, but the Apple device can accommodate for non-square pixel displays as well. See User Interface (UI) Stream (page 411).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.2.2.2 Processing

The accessory must be capable of:

- Hardware decoding of H.264 video, see User Interface (UI) Stream (page 411).
- Running the Communication Plug-in and other software clients, see Software Clients (page 409).
- Providing different audio processing based on the audio mode, see Audio (page 413).
- Synchronizing the audio output and input sample clocks, see Synchronization (page 447).

23.2.2.3 User Input Devices

The accessory must provide a touchscreen, or a combination of a rotary knob and at least select and back buttons. See User Input (page 456).

23.2.2.4 Speakers and Microphone

The accessory must provide audio output and input via the vehicle's speakers and a microphone in the vehicle cabin. See Audio (page 413).

23.2.2.5 Siri Button

Siri is an integral part of the CarPlay experience. Siri can be used in conjunction with the display to call people, select and play music, hear and compose text messages, and get directions. Siri can also be used to access Apple device features that do not appear on the display, including notifications, calendar information, reminders and more. Siri is one of the most important ways that users interact with CarPlay so it's critical that the behavior of Siri is consistent with what users are familiar with on Apple devices.

The accessory must have a tactile button for Siri activation. The Siri button cannot be a soft button that appears on a vehicle display. The Siri button is used to activate a Siri session and to continue a Siri session when there is an on-going conversation. The Siri button must be accessible at all times. Immediately after the Apple device has been connected to the accessory, users expect to be able to activate Siri at any time, regardless of what state the vehicle is in and even if CarPlay is not showing on the display. There are some exceptions, such as when the vehicle is performing a safety critical function or if there is already an active voice recognition session using the accessory's built-in system.

The Siri button must be easily accessible. It is expected that the Siri button is located on the steering wheel, although in some exceptional cases the Siri button may be located elsewhere. In many cases the Siri button will be the same button that is used to activate the accessory's built-in voice recognition system. In this case, a short press on the button should trigger the accessory's built-in voice recognition system and a long press on the button should trigger Siri. The accessory is responsible for defining what constitutes a long press on the button, but it must not be greater than 1000 ms. Apple recommends a period of 600 ms since it closely

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

mimics the behavior of the Apple device home button. Once the user activates Siri by holding down the button, a Siri session is initiated. The accessory knows that a Siri session is active by observing the application state which will be "Speech". While the Siri session is active, the accessory must send all Siri button events (every time the Siri button is either pressed or released) to the Apple device. The Apple device monitors both button press events and button release events and manages the Siri session accordingly. The accessory must send raw events and should not attempt to interpret the button press and button release events. When the Siri session is terminated, the accessory may stop sending button events and return to a default state.

If the Siri button is a dedicated hardware button used exclusively for Siri and is never used for built-in voice recognition activation or for other features using other devices or systems, then you may choose to label the button with Siri branding. In this case, please consult with Apple for guidelines on using the Siri brand. Otherwise Siri should not appear on the button and it should be labeled with standard iconography, such as the icon used for the built-in voice recognition system.

If the accessory supports CarPlay over wireless, the Siri button must also function as a trigger to start the pairing process. When no device is connected, a long press on the Siri button must activate the accessory's wireless pairing user interface and the accessory must immediately become discoverable by an Apple device. However, if the accessory is actively connected to a second Bluetooth device, a long press on the Siri button may be used to trigger voice recognition activation instead of activating the accessory's wireless pairing user interface.

See requestSiri (page 482).

23.2.2.6 "Apple CarPlay" Button

To show the CarPlay UI, the accessory must provide one of the following:

- A physical button.
- A soft button in a top-level menu.
- A menu item in a top-level menu.

The Apple CarPlay icon may be used on physical buttons that show the CarPlay UI. The label "Apple CarPlay" should appear in text below the icon. If the icon is not used, the text-only label "Apple CarPlay" must be used.

The Apple CarPlay logo may be used on soft buttons and menu items that show the CarPlay UI. If color graphics are featured in the accessory screen interface, the accessory should use the color version of the Apple CarPlay logo. If black-and-white graphics or other monochromatic themes are featured in the accessory screen interface, use either the white version or the black version of the Apple CarPlay logo, matching the visual style of the interface. The logo must be accompanied by the label "Apple CarPlay". If the logo is not used, the text-only label "Apple CarPlay" must be used.

Whenever the CarPlay session is not active, the Apple CarPlay logo and/or label should appear disabled or should be hidden.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

When using the label "Apple CarPlay", the accessory should match the font, size, style, and placement of other text used in the interface; do not imitate Apple typography. For example, if all capital letters are used on other items, then Apple CarPlay can also appear in all capital letters: "APPLE CARPLAY".

It is preferred that "Apple CarPlay" is placed on one line. If space is limited and other titles are stacked, "Apple CarPlay" can be stacked, with "Apple" on one line and "CarPlay" on the next line.

Any use of the Apple CarPlay icon, logo, or trademark within native UI requires Apple review and approval.

See the *Apple CarPlay Identity Guidelines* for more details.

See requestUI (page 483) to show the CarPlay UI.

23.2.2.7 Sensors

The accessory must provide location information to the Apple device as described in Location Information (page 637).

23.2.2.8 Connection to the Apple device

To connect to the Apple device over USB, the accessory must provide a USB-A receptacle or a Lightning connector, see CarPlay over USB (page 388).

To connect to the Apple device wirelessly, the accessory must provide a Bluetooth connection and a wireless access point, see CarPlay over Wireless (page 395).

Refer to the *Apple CarPlay Identity Guidelines* for labeling any receptacles or connectors.

23.2.2.9 Device Mount Location

The Apple device should be mounted in an open holder that is away from direct sunlight or any heat source. Air ventilation is recommended but not required, however an enclosed compartment with no air circulation, such as the glove box, should be avoided.

23.2.3 Architecture

To support CarPlay, an accessory must support USB as a USB device where the Apple device is the host, see USB Host Mode (page 712).

To support CarPlay over wireless, an accessory must support Bluetooth and Wi-Fi, see CarPlay over Wireless (page 395).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The accessory communicates with the Apple device using protocols defined by the Internet Protocol (IP) suite. An IP model is employed to abstract data transport away from the chosen physical layer where possible, although some aspects of the overall implementation will differ per transport. In addition, the accessory must establish an iAP2 link with the Apple device for authentication, identification, and exchange of metadata.

The major building blocks of this architecture are illustrated in Figure 23-1 (page 387), with Apple-provided software components highlighted in blue.

Figure 23-1 Topology of an automotive head unit using CarPlay

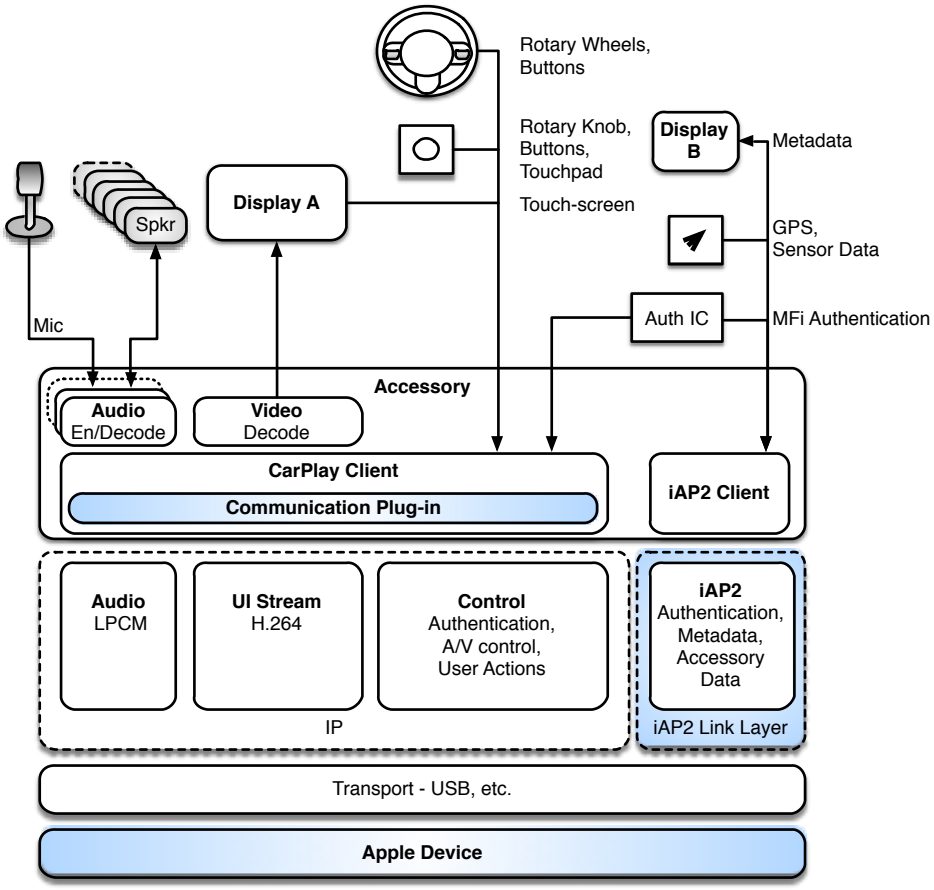


Figure 23-1 (page 387) shows an automotive head unit and its associated instrument cluster and displays with:

- Digital content that is independently driven by an Apple device
- User interactions that are controlled by one or more discrete controllers
- Multichannel audio, both input and output

23.2.4 CarPlay over USB

At a minimum, the accessory must support Hi-Speed USB; see the *USB 2.0 Specification*. Because of the bandwidth and low-latency requirements for transferring interactive audio/visual content, the use of a dedicated USB controller for the interface between the Apple device and accessory is recommended.

The accessory must also support the USB Host Mode feature (where the accessory is a USB device and the Apple device is the host), as specified in USB Host Mode (page 712).

USB Network Control Model (NCM) is defined in the following specifications:

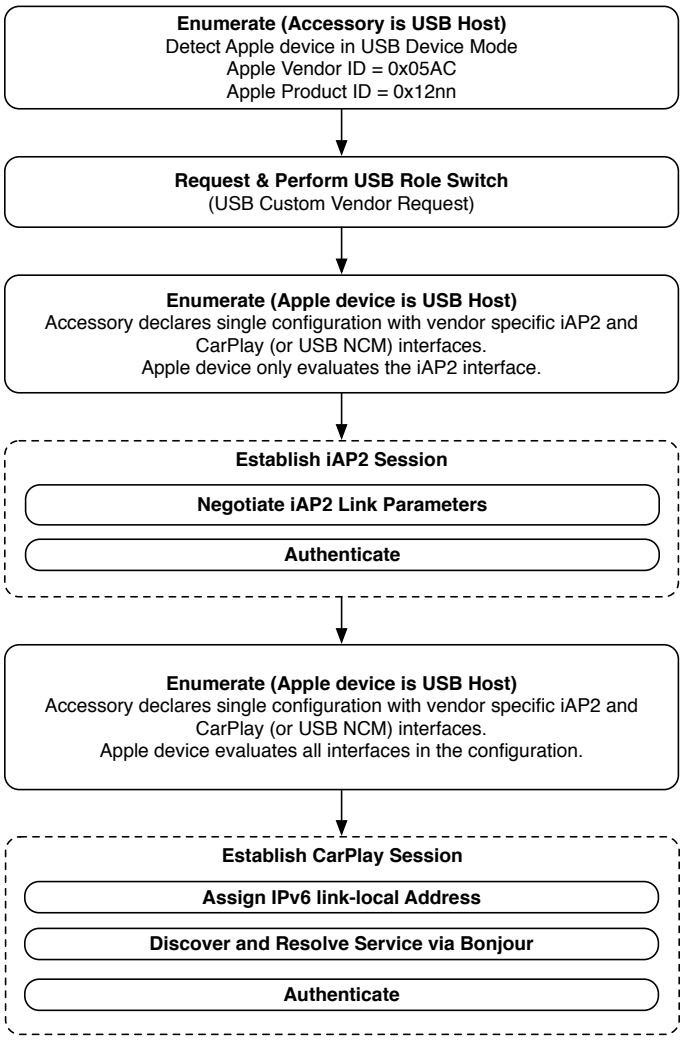
- *Universal Serial Bus Communications Class Subclass Specifications for Network Control Model Devices, Revision 1.0*
- *Universal Serial Bus Class Definitions for Communication Devices, Version 1.2*

The accessory must establish the CarPlay session automatically upon physical connection to the Apple device.

For an accessory that normally acts as a USB host, the steps shown in Figure 23-2 (page 389) are required.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 23-2 Process for establishing a CarPlay session



23.2.4.1 Role Switch

If the accessory normally acts as a USB host, it must perform a host-to-device role switch as specified in USB Role Switch (page 714).

An accessory that presents directly as a USB device without undergoing the host-to-device role switch, will require the use of a custom USB cable or dock solution, incorporating the Lightning (C10A), Lightning (C11A), Lightning (C48A), or Lightning (C68A) connector (USB Host Mode). The standard Apple-provided white USB-to-Lightning cable is incompatible with a USB accessory presenting directly as a USB device.

23.2.4.2 iAP2 / NCM Interface Configuration

Once role switch is completed, the accessory must continue by enumerating the supported USB interfaces. At a minimum the accessory must present a single configuration which includes:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- An iAP2 interface (see USB Host Mode (page 712)).
- A USB NCM control interface (see Table 23-1 (page 390) and Table 23-2 (page 390)).
- A USB NCM data interface (see Table 23-3 (page 391)).

The accessory must proceed to establish an authenticated iAP2 Session as described in Accessory Authentication (page 261).

> **Note:** Until authentication is completed successfully, the Apple device will access the iAP2 interface unless it is using control session version 2 (see Control Session (page 789)). With version 2, the Apple device may access other USB interfaces whether or not authentication is complete.

Table 23-1 USB NCM Control Interface Descriptor

USB Descriptor	Value	Description
Interface Number	0xNN	Must be different from the iAP2 interface and USB NCM data interface numbers. Must match the USBHostTransportCarPlayInterfaceNumber (see Table 59-15 (page 812)).
Interface Class	0x02	USB Communication Interface Class
Interface Subclass	0x0D	Network Control Model
Interface Protocol	0x00	No encapsulated commands / responses
Number of Endpoints	1	1 Interrupt IN (optional): This is typically used to convey changes in link status. Since link is expected to be maintained at all times, we will synthesize link up if there is a read completion via the data interface.

The accessory must also publish the additional descriptors outlined in Table 23-2 (page 390).

Table 23-2 USB NCM Communication Interface Descriptor Requirements

USB Descriptor	Description
Header	CDC Header functional descriptor
Union	CDC Union functional descriptor
Ethernet	CDC Ethernet Networking functional descriptor

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

USB Descriptor	Description
NCM	NCM functional descriptor

Accessory MAC addresses must be properly assigned by the manufacturer, see http://www.iana.org/assignments/ethernet-numbers/ethernet-numbers.xhtml. The accessory must provide this MAC address for the Apple device to use via the CDC Ethernet Networking functional descriptor. The accessory must not use the same MAC address provided to the Apple device for its own network interface. The accessory may use a different assigned MAC address, or it may modify the MAC address for its own use by setting the locally administered bit of the Organizationally Unique Identifier (OUI) portion of the MAC address. The locally administered bit is the second-least-significant bit of the most significant byte of the address, see http://standards.ieee.org/develop/regauth/tut/macgrp.pdf. This will ensure that there are no collisions between the Apple device and the head unit.

Table 23-3 USB NCM Data Interface Descriptor

USB Descriptor	Value	Description
Interface Number	0xNN	Must be different from the iAP2 interface and USB NCM control interface numbers.
Interface Class	0x0A	USB Data Interface Class
Interface SubClass	0x00	
Interface Protocol	0x01	NCM Data Class
Number of Endpoints	0	(for Alternate Setting 0)
Number of Endpoints	2	(for Alternate Setting 1) 1 Bulk IN; and 1 Bulk OUT

The accessory must implement USB Hi-Speed NCM on this interface. The interface must support transfers of encapsulated datagrams up to 64 KB (i.e., up to 40 1514-byte Ethernet frames) and 16-bit NCM Transfer Blocks (i.e., NTB-16).

Accessories using the USB NCM interface for CarPlay must support at least 100 Mbps of bandwidth with a latency less than 5 ms over both TCP and UDP protocols and a packet loss of less than 1% (as measured with iperf) over UDP.

When the Apple device is connected or disconnected, the accessory must reflect this change in the connection state of the NCM interface. The network stack on the head unit must mark the NCM interface as available only while the Apple device is connected.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

An example configuration for an accessory implementing the USB Communications Class NCM Subclass descriptors is as follows:

```
Device Release                                          :   r2.00
USB Spec Release                                        :   v2.00
Serial Number                                          :   ABC-0123456799
Class                                                   :   0xff
(Vendor-specific)
Subclass                                                :   0x00
Protocol                                                :   0x00
Configurations                                          :   1
    Configuration                                       :   1 (Default
Configuration)
        Attributes                                      :   0xc0
(Self-powered)
        Max Power                                       :   0 mA
        Interfaces                                      :   3
            Interface                                   :   0 / 0 (iAP2
Interface)
                Class                                   :   0xff
                Subclass                                :   0xf0
                Protocol                                :   0x00
                Endpoints                               :   2
                    Endpoint                            :   0x81
                        Attributes                      :   Bulk/IN
                        Max Packet Size                 :   512
                    Endpoint                            :   0x1
                        Attributes                      :   Bulk/OUT
                        Max Packet Size                 :   512
            Interface                                   :   1 / 0 (NCM
Communication Interface)
                Class                                   :   0x02
                Subclass                                :   0x0d
                Protocol                                :   0x00
                Endpoints                               :   1
                    Endpoint                            :   0x82
                        Attributes                      :   Interrupt/IN
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
                    Max Packet Size                         :   64
                    Interval                                :   1 mframe
                CDC Header Functional Descriptor            :
                    bcdCDC                                  :   v1.10
                CDC Union Functional Descriptor             :
                    Control Interface                       :   1
                    Subordinate Interface                   :   2
                CDC Ethernet Networking Functional Descriptor   :
                    iMacAddress                             :   7
                    Ethernet Statistics                     :   0
                    Max Segment Size                        :   1514
                    Number MC Filters                       :   0
                    Number Power Filters                    :   0
                NCM Functional Descriptor                   :
                    NCM Version                             :   v1.00
                    Network Capabilities                    :   0x3A
            Interface                                       :   2 / 0 (NCM
Data Interface Alternate 0)
                Class                                       :   0x0a
                Subclass                                    :   0x00
                Protocol                                    :   0x01
                Endpoints                                   :   0
            Interface                                       :   2 / 1 (NCM
Data Interface Alternate 1)
                Class                                       :   0x0a
                Subclass                                    :   0x00
                Protocol                                    :   0x01
                Endpoints                                   :   2
                    Endpoint                                :   0x83
                        Attributes                          :   Bulk/IN
                        Max Packet Size                     :   512
                    Endpoint                                :   0x3
                        Attributes                          :   Bulk/OUT
                        Max Packet Size                     :   512
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.2.4.3 Authentication

CarPlay is an authenticated solution, requiring the use of an MFi authentication coprocessor (2.0 B or later) obtained through Apple. Apple devices will stream only to authorized accessories.

Communication over both the iAP2 and CarPlay interfaces requires authentication and each interface provides a discrete authentication API. To expedite these dual authentication steps, local caching of the X.509 Certificate provided by the Apple Authentication Coprocessor is permitted.

All authenticated accessories are required to be certified under the Apple MFi program. The CarPlay accessory must successfully pass compliance tests to assure that all digital content from an Apple device will be correctly decoded and displayed and that all electrical requirements described in this specification are met.

23.2.4.4 Networking and Service Discovery

To establish a CarPlay session, the accessory must be networked with the Apple device by using the communication protocols defined by the Internet Protocol (IP) documentation; see *IETF RFC 2460, Internet Protocol, Version 6 (Ipv6) Specification* , December 1998. IP connectivity must be provided by IPv6 link-local addressing and the accessory must support Apple Bonjour zero-configuration networking over this interface; see the *Apple Bonjour Support Site* . The accessory will be identified uniquely by its MAC address and by a TXT record with additional data needed to resolve or use the service.

For more details, see Discovery (page 424).

23.2.4.5 Session Establishment

Once a connection has been established, setup and content transfer will start after the accessory completes authentication over the CarPlay interface.

The accessory must not send a 'Play' command automatically upon connection of a CarPlay enabled device. See Media Library Playback Requirements (page 650).

For more information on setting up a CarPlay session refer to Setup and Control (page 427).

The accessory must be able to establish a CarPlay session within 3 seconds of Apple device connection.

23.2.4.6 Session Termination

The accessory must be able to detect a disconnect and terminate the session within 1 sec of a physical detachment of the Apple device.

When the accessory also supports CarPlay over wireless (see CarPlay over Wireless (page 395)), if the Apple device is disconnected, the accessory must follow the requirements outlined in USB to Wireless (page 409).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.2.5 CarPlay over Wireless

CarPlay can automatically connect the car wirelessly, without needing to handle or plug in the Apple device - perfect for short drives.

Setup for CarPlay over wireless begins with Bluetooth discovery between the Apple device and the accessory. The user initiates pairing on the accessory either through a long press on the accessory's Siri button, or by using the accessory's user interface. On the Apple device, the user initiates pairing on the Apple device through Settings. The accessory and the Apple device may choose to show all detected Bluetooth devices, or they may choose to show only devices that support CarPlay over wireless by querying the Bluetooth Extended Inquiry Response (EIR). A Bluetooth Classic pairing flow generates a record of trust between the Apple device and the accessory.

After Bluetooth pairing is complete, the Apple device requests the accessory's Wi-Fi credentials using iAP2 over Bluetooth. The Apple device uses the Wi-Fi credentials and the Apple Device Information Element provided by the accessory's access point to associate with the accessory's Wi-Fi network.

Once associated with the network, the Apple device advertises a CarPlay Control Bonjour Service to indicate that CarPlay is available. The accessory is responsible for requesting initiation of the CarPlay session using CarPlay Control. Once the CarPlay session starts, the user can interact with CarPlay through the accessory's controls. Finally, the accessory transitions from iAP2 over Bluetooth to iAP2 over CarPlay client.

Once initial pairing is completed, Apple CarPlay reconnects seamlessly the next time the user enters the vehicle. Reconnection starts with the accessory reestablishing the Bluetooth connection. This is used as a trigger for the Apple device to associate with the already known Wi-Fi network and to start advertising the CarPlay Control Bonjour service. Once the accessory discovers a CarPlay-enabled device though the CarPlay Control Bonjour service, it requests a CarPlay session from that device.

For accessories that support multiple Apple devices with CarPlay, including CarPlay over USB and CarPlay over wireless, the user selects which Apple device to use for CarPlay from a list presented in the accessory’s user interface. The accessory can add a CarPlay device to the list by following the same steps to reconnect each previously paired Apple device: a Bluetooth connection is established, the Apple device associates with the Wi-Fi network, the Apple device advertises CarPlay availability through the CarPlay Control Bonjour service. If an Apple device is connected over USB and over wireless, it appears only once in the list.

23.2.5.1 Bluetooth

CarPlay over wireless requires the accessory to provide Bluetooth connection, service discovery, and pairing.

During initial pairing, the accessory must support standard Bluetooth Secure Simple Pairing using Numeric Comparison. Once a secure Bluetooth link is established, the accessory must negotiate the iAP2 profile and establish an iAP2 session for exchanging the Wi-Fi credentials, see iAP2 Client over Bluetooth (page 409).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

After starting iAP2, the accessory may negotiate all of the relevant Bluetooth profiles such as HFP, A2DP, AVRCP, etc. However, once a CarPlay session is established, the Apple device will notify the accessory to disconnect all active profiles, see disableBluetooth (page 478).

If CarPlay over wireless is available on the Apple device as indicated by the Bluetooth EIR then the accessory must start iAP2 prior to any additional Bluetooth profiles.

Accessories may use the WirelessCarPlayUpdate (page 842) message to receive notifications about the availability of CarPlay over wireless on an Apple Device. If legacy Bluetooth profiles, such as HFP, A2DP, etc. are supported and not already connected, the accessory must re-establish the connection when CarPlay over wireless is no longer available on the Apple device.

See Bluetooth (page 344) for additional requirements.

23.2.5.1.1 Accessory CarPlay Bluetooth EIR

The Apple device determines whether the accessory supports CarPlay over wireless by examining the CarPlay UUID included in the accessory's Bluetooth EIR. Accessories that support CarPlay over wireless must respond with the CarPlay Service UUID defined below.

Once the accessory is discoverable, it must run periodic inquiry scans and respond to an inquiry from the Apple device with an FHS packet with the BT EIR bit set as defined in the *Core Specification Supplement, Part A* . It must then return an Extended Inquiry Response packet 1250 microseconds after the FHS of the packet.

The EIR packet must include a 128-bit CarPlay Service UUID, `0xEC884348CD4140A29727575D50BF1FD3`, in one of the following EIR data types:

- Incomplete List of 128-bit Service Class UUIDs
- Complete List of 128-bit Service Class UUIDs

The Extended Inquiry Response may contain additional data as defined in the *Core Specification Supplement, Part A* . DM1 or DM3 packets, which support FEC, are recommended for transmitting the EIR data for maximizing the range and increasing the robustness of the packet.

For more information on Bluetooth EIR, see Additional Specifications (page 381).

23.2.5.1.2 Apple Device CarPlay Bluetooth EIR

Apple devices which support CarPlay over wireless and are discoverable via Bluetooth will advertise the following 128-bit UUID: `0x2D8D2466E14D451C88BC7301ABEA291A`, in one of the following EIR data types:

- Incomplete List of 128-bit Service Class UUIDs
- Complete List of 128-bit Service Class UUIDs

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Accessories may use this information to distinguish between devices with Apple CarPlay enabled and general Bluetooth devices.

23.2.5.1.3 Accessory iAP2 Bluetooth EIR

The accessory must support iAP2 over Bluetooth, see iAP2 Client over Bluetooth (page 409).

23.2.5.2 Wi-Fi Access Point

CarPlay accessories must operate as a standard Wi-Fi access point allowing Apple devices to join as a standard Wi-Fi client.

During initial pairing, the Apple device requests the Wi-Fi access point's Wi-Fi credentials (SSID and Passphrase) using iAP2 over Bluetooth, see iAP2 Client over Bluetooth (page 409). These Wi-Fi credentials are stored on the Apple device and will be used to join the Wi-Fi access point automatically during reconnection. To speed up the reconnection scenario, the Apple device will initiate Wi-Fi scanning based on a re-connect of any Bluetooth profile.

The CarPlay accessory must ensure that the Wi-Fi access point is fully operational before it initiates pairing or reconnection of the device over Bluetooth.

Once the Apple device joins the Wi-Fi access point, it advertises the CarPlay Control Bonjour service and waits for the accessory to establish the CarPlay session, see Networking and Service Discovery (page 402).

The accessory Wi-Fi access point implementation must transmit a de-authentication packet to the Apple device before the Wi-Fi access point goes offline (accessory is being turned off).

The accessory must not implement any policy that tracks or filters Apple devices based on Wi-Fi MAC addresses.

The Wi-Fi access point must support at least 25 Mbps of bandwidth (as measured with iperf) with a latency less than 16 ms (as measured with the ping command) over both TCP and UDP protocols and a packet loss of no more than 1% on a 25 Mbps UDP uplink stream.

The Wi-Fi access point may support multiple Wi-Fi connected devices for other applications, but those connections must not impact the required CarPlay performance.

The Wi-Fi access point must be configured for a power save operation using a DTIM value of one (1).

The Wi-Fi access point must not use a hidden SSID.

23.2.5.2.1 Hardware Requirements

At a minimum the accessory must support one of the following physical level data rates and modulations as defined in the *IEEE 802.11-2012* standard:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- 802.11n 2.4 GHz HT20
- 802.11n 5 GHz HT20 or HT40

It is recommended that the accessory supports the 802.11ac VHT20, VHT40, or VHT80 physical level data rates and modulations as defined in the *IEEE 802.11ac-2013* standard.

For more information on the IEEE 802.11-2012 and 802.11ac-2013 standards, see Additional Specifications (page 381).

23.2.5.2.2 Frequency Bands

The Wi-Fi access point may operate in either the 2.4 GHz or the 5 GHz frequency bands.

When hosting a wireless network in the 2.4 GHz frequency band, the Wi-Fi access point must operate on one of the channels in Table 23-4 (page 398).

Table 23-4 Operational Channels for 2.4 GHz Wi-Fi

Channel	Frequency
1	2.412 GHz
6	2.437 GHz
11	2.462 GHz

When hosting a wireless network in the 5 GHz frequency band, the Wi-Fi access point must operate on one of the channels in Table 23-5 (page 398).

Table 23-5 Operational Channels for 5 GHz Wi-Fi

Band	Channel	Frequency
UNII-I (Lower Band)	36	5.180 GHz
UNII-I (Lower Band)	40	5.200 GHz
UNII-I (Lower Band)	44	5.220 GHz
UNII-I (Lower Band)	48	5.240 GHz
UNII-III (Upper Band)	149	5.745 GHz
UNII-III (Upper Band)	153	5.765 GHz

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Band	Channel	Frequency
UNII-III (Upper Band)	157	5.785 GHz
UNII-III (Upper Band)	161	5.805 GHz

It is highly recommended that the accessory Wi-Fi access point operate in the 5 GHz frequency band. Operating in the 5 GHz frequency band will avoid interference and coexistence with Bluetooth traffic that only exists in the 2.4 GHz frequency band.

Wi-Fi operation in the 5 GHz frequency band depends on regulatory domain rules and regulations. It is understood that some regulatory domains may prohibit operation in the 5 GHz frequency band, therefore the accessory Wi-Fi access point will have to operate in the 2.4 GHz frequency band.

Channel switching should be limited as it increases connection time. Channel switching should be avoided across accessory power cycles and should not change during a CarPlay over wireless session.

23.2.5.2.3 Basic Wi-Fi Requirements

The accessory must support the Software Access Point (SWAP) Wi-Fi Operational Mode.

The accessory must support Distributed Coordination Functions (DCF).

The accessory must support the following frame types:

- Association Request and Response
- Re-association Request and Response
- Probe Request and Response
  - Broadcast Probe Requests
  - Directed Probe Requests
- Beacons
- Disassociation
- De-authentication
- RTS/CTS
- ACK
- Data Frames
- Null Frames

The accessory must support the following frame reception and transmission functionality:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Reception and Transmission of Data Frames
- Reception and Transmission of Management/Control Frames
- Reception and Transmission of Public Action Frames
- Receive Defragmentation
- Transmit Fragmentation (optional)

The accessory may implement the mandatory tallies and counters defined in the *IEEE 802.11 Management Information Base (MIB)*.

The accessory must support standard power management and power save functions as defined in the IEEE 802.11-2012 standard, see Additional Specifications (page 381).

The accessory may support short guard interval (400 ns) for the defined Data Rates and MCS indices.

The accessory must support at least the following OFDM Data Rates: 6, 9, 12, 18, 24, 36, 48, and 54 Mbps.

23.2.5.2.4 Advanced Wi-Fi Requirements

The accessory must support the WFA Wireless Multimedia (WMM) Quality of Service (QOS) mechanism, see Additional Specifications (page 381).

The CarPlay protocol uses AC_VO for Voice data traffic, AC_VI for Screen data traffic, and AC_VO for Control data traffic.

The accessory may support Universal Advanced Power Save Delivery (U-APSD) as defined in the IEEE 802.11-2012 standard, see Additional Specifications (page 381). It is strongly recommended that implementations support this power save mode in order to optimize power consumption on the Apple device.

When the Apple device transmits a null data packet with the PM Bit set (entering 802.11 power save mode), the accessory must ACK the null data packet and must flush the Tx hardware queue for that wireless client (STA) therefore not transmitting any additional packets to the Apple device.

23.2.5.2.5 Security

The accessory must use the WPA2 Personal security mode as defined by the Wi-Fi Alliance and IEEE.

The accessory may support the mandatory security related tallies/counters defined in the *IEEE 802.11 Management Information Base (MIB)*.

All support encryption algorithms/functions must be executed in hardware unless otherwise described and approved by Apple.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The AES/CCMP encryption may comply with the *Government Security Requirements for Cryptographic Modules FIPS PUB 140-2*.

23.2.5.2.6 Performance

The following table defines recommended throughput performance targets for the different IEEE 802.11 MAC/PHY modes for the TCP and UDP protocols.

Table 23-6 Expected Throughput for IEEE 802.11 MAC/PHY Modes

IEEE 802.11 MAC/PHY Radio Modes	1SS		2SS	
TCP	**UDP**	**TCP**	**UDP**	
802.11n 2.4 GHz - HT20	55	65	100	130
802.11n 5 GHz - HT20	55	65	100	130
802.11n 5 GHz - HT40	120	135	210	270
802.11ac 5 GHz - VHT20	60	70	120	140
802.11ac 5 GHz - VHT40	140	160	280	320
802.11ac 5 GHz - VHT80	300	350	600	700

23.2.5.2.7 Wi-Fi Alliance Compliance and Conformance

The CarPlay accessory Wi-Fi implementation must be in compliance with the mandatory to implement requirements defined in the *Wi-Fi CERTIFIED(tm) 802.11 ac Interoperability Test Plan*.

The CarPlay accessory Wi-Fi implementation must be in compliance with the mandatory to implement requirements defined in the *Wi-Fi 802.11n System Interoperability Test Plan*.

23.2.5.2.8 Interworking Information Element (IE)

The accessory must include the IEEE 802.11 Interworking IE in Beacon, Probe response, and Association response frames at all times during the operation of the Wi-Fi access point.

The accessory must set the following fields:

- "Access Network Options" field - must be set based on the availability of Internet connectivity, see Enabling Internet Data Connectivity (page 403).
- "Venue Info" field - must be set to 10 (Vehicular). The Venue Type code must be set to one of the values that correspond to Venue Group code 10 (0-255) as defined in *IEEE Std. 802.11-2012*.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

For more information on the Interworking IE, see Additional Specifications (page 381).

23.2.5.2.9 Apple Device Information Element (IE)

The accessory must support and include the Apple Device IE, see Apple Device Information Element (IE) (page 743).

The accessory must include the Apple Device IE in Beacon, Probe response, and Association response frames at all times during the operation of the Wi-Fi access point.

The Features flags must set the flags as specified in Table 23-7 (page 402) (see Table 55-8 (page 746)):

Table 23-7 Features flag settings

Flag	Value
Supports CarPlay over Wireless	1
Provides Internet access	1 if Internet data connectivity is supported, provisioned, and enabled.
Supports 2.4 GHz Wi-Fi networks	1 if the 2.4 GHz frequency band is supported and used for CarPlay.
Supports 5 GHz Wi-Fi networks	1 if the 5 GHz frequency band is supported and used for CarPlay.

The following parameters are required, see Payload (page 745):

- Name
- Manufacturer
- Model
- OUI
- Bluetooth MAC Address
- Device ID

23.2.5.3 Networking and Service Discovery

To establish a CarPlay over wireless session, the accessory must be networked with the Apple device using the communication protocols defined by Internet Protocol (IP) documentation, see *IETF RFC 791, Internet Protocol (IP), DARPA Internet Program, Protocol Specification*, September 1981.

Copyright © 2015 Apple Inc. All Rights Reserved.

IP connectivity must be provided by IPv4 DHCP addressing and the accessory must support TCP and UDP transmission modes; see *IETF RFC 2131, Dynamic Host Configuration Protocol (DHCP)* , *IETF RFC 793, Transmission Control Protocol (TCP)* and *IETF RFC 768, User Data Gram Protocol (UDP)* . In addition, the accessory must support the following networking protocols: ARP and ICMP; see *IETF RFC 826, Address Resolution Protocol (ARP)* and *IETF RFC 792, Internet Control Message Protocol (ICMP)* .

The accessory must support Apple Bonjour zero-configuration networking over this interface; see the *Apple Bonjour Support Site* . The accessory will be identified uniquely by its MAC address and by a TXT record with additional data needed to resolve or use the service. For more details, see Discovery (page 424).

In order to reduce reconnection time, the accessory must store the DHCP IP address lease information across multiple reboots. The DHCP IP address lease time must be at least 3 days with an address pool size of at least 100; a lease time of 7 days with a pool size of 250 is recommended.

23.2.5.4 Enabling Internet Data Connectivity

Internet connection is defined as an active data connection to servers that reside on the Internet. In case of a cellular based Internet connection the SIM must be valid, provisioned, and enabled for data. The accessory must indicate if it provides an active Internet connection that can route data.

Based on availability of Internet data connection, the accessory must update the following fields:

- "Access Network Options" flag in the Interworking IE, see Interworking Information Element (IE) (page 401)
- "Provides Internet Access" flag in the Apple device IE, see Apple Device Information Element (IE) (page 402)
- DHCP server gateway IP address - a valid DHCP server gateway IP address if Internet connectivity is provided, otherwise the gateway IP address must be set to 0.0.0.0

23.2.5.5 Session Establishment

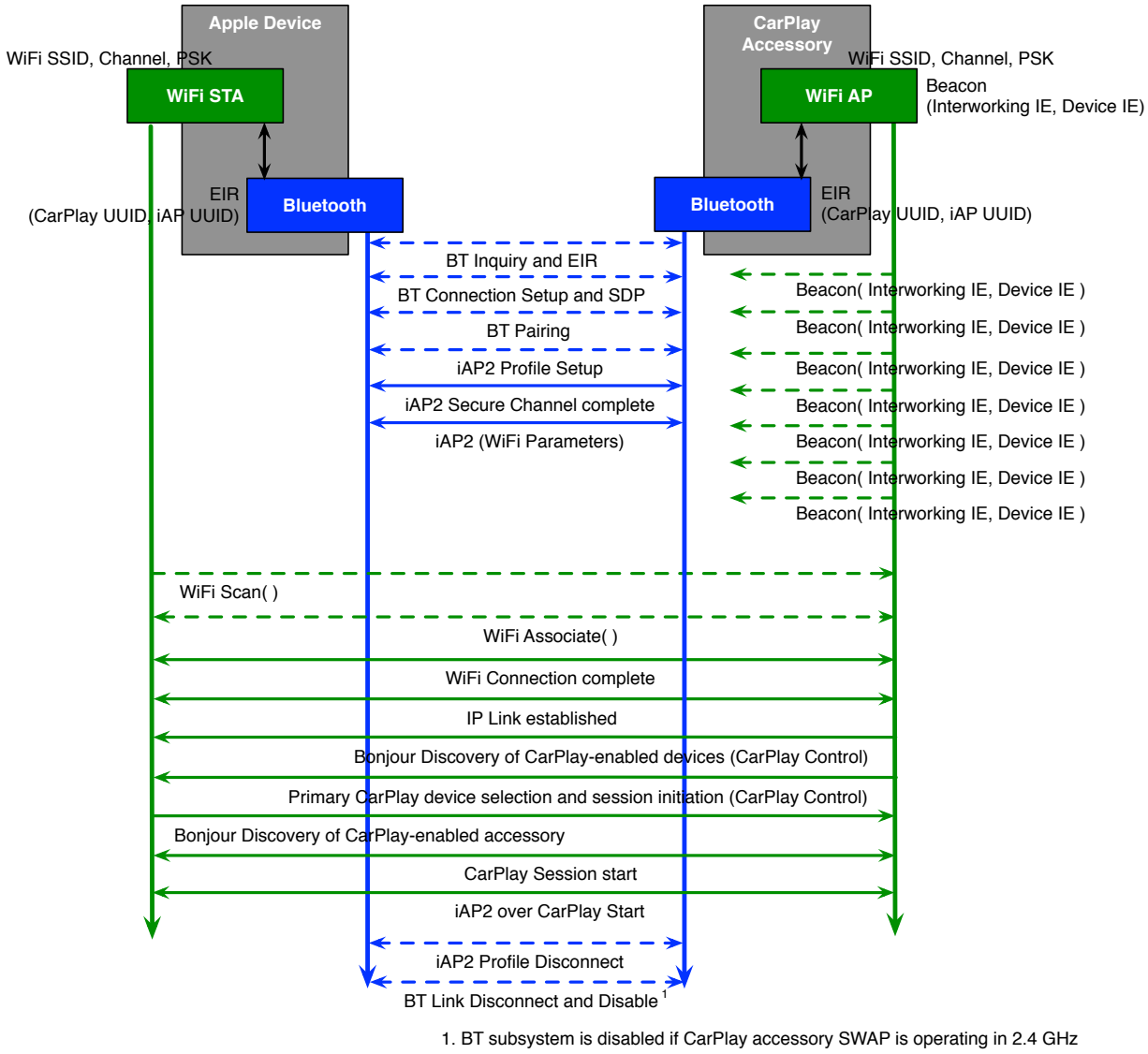
Once the Apple device has successfully received the Wi-Fi credentials over iAP2, the Apple device will scan for the accessory Wi-Fi access point and associate using the provided parameters. Once the Wi-Fi connection is complete, IP will be brought up, Bonjour discovery and device selections occurs, and the CarPlay session can be initiated as defined in Discovery (page 424).

Once the CarPlay session is successfully initiated, if the accessory Wi-Fi access point is operating in 2.4 GHz, Bluetooth link with the Apple device must be disconnected and the Bluetooth subsystem must be disabled (see disableBluetooth (page 478)). If the accessory Wi-Fi access point is operating in 5 GHz, Bluetooth link with the Apple device must be disconnected and the Bluetooth subsystem must go into idle mode, available for connections with other devices.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 23-3 (page 404) defines the message flow when using Bluetooth and iAP2 over Bluetooth as the initial connection and pairing mechanism.

Figure 23-3 Initial Connection and Pairing using Bluetooth and iAP2



1. BT subsystem is disabled if CarPlay accessory SWAP is operating in 2.4 GHz

23.2.5.6 Session Reconnection

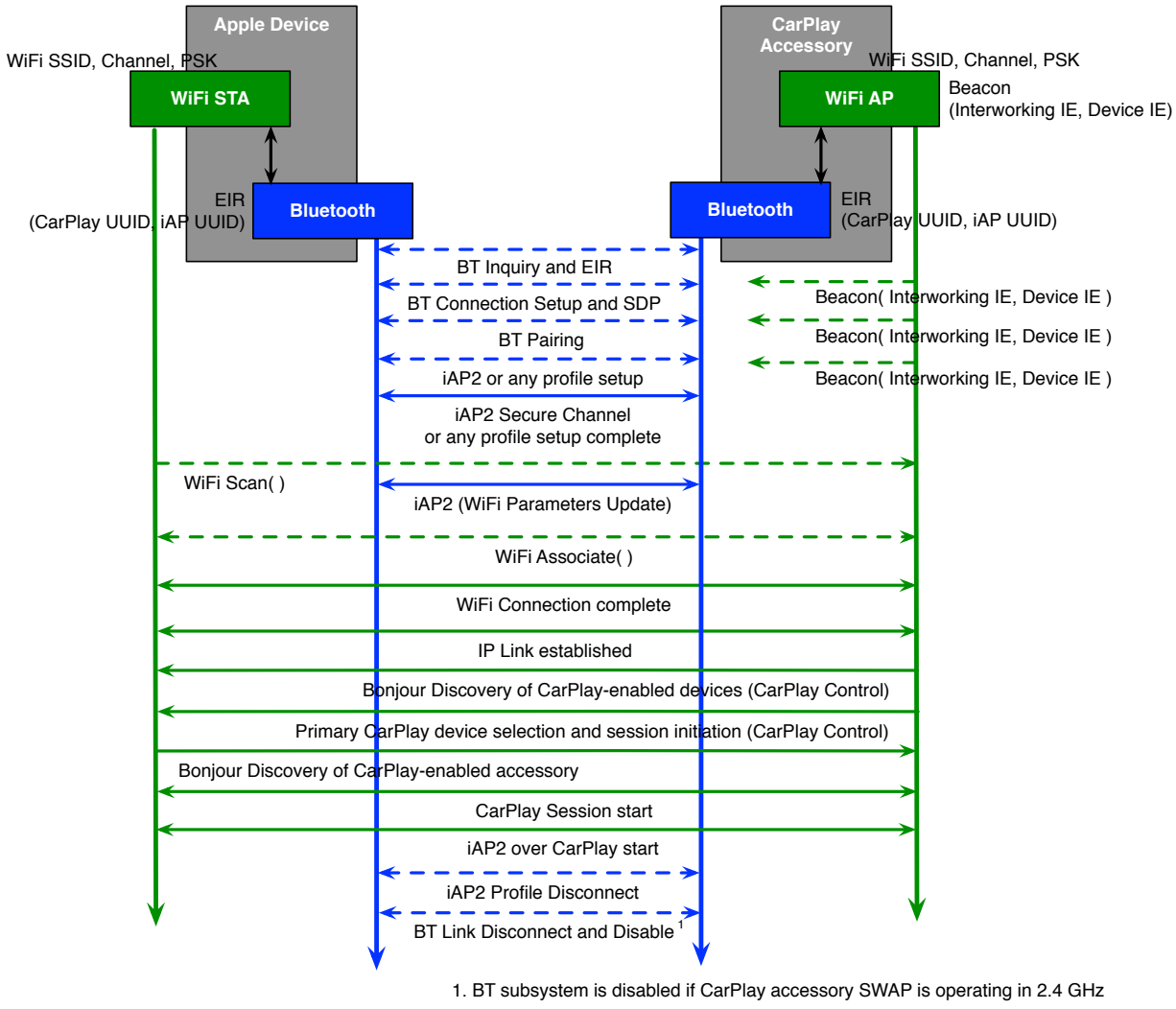
This section defines the reconnection procedure and requirements when using Bluetooth.

The accessory must initiate reconnection to a previously paired Apple device using Bluetooth. The accessory Wi-Fi access point must be fully operational before initiating reconnection. Once the Bluetooth link is established, the Apple device will scan and attempt to associate to the accessory's Wi-Fi access point using the Wi-Fi credentials mapped to the BD_ADDR of the accessory in question. Just like with the Initial Connection and Pairing procedure, the accessory must exchange the Wi-Fi credentials using iAP2 over Bluetooth.

Once the CarPlay session is successfully initiated, if the accessory Wi-Fi access point is operating in 2.4 GHz, Bluetooth link with the Apple device must be disconnected and the Bluetooth subsystem must be disabled (see disableBluetooth (page 478)). If the accessory Wi-Fi access point is operating in 5 GHz, Bluetooth link with the Apple device must be disconnected and the Bluetooth subsystem must go into idle mode, available for connections with other devices.

Figure 23-4 (page 405) defines the message flow when using Bluetooth for reconnection.

Figure 23-4 Reconnect using Bluetooth



For discovery and connection establishment, the reconnect latency must be 8 seconds or less. Reconnect latency is defined as the period from the when the accessory's Bluetooth subsystem transmits the first Bluetooth connection packet (Bluetooth Connection and SDP Setup) to the Apple device to when the Apple device starts the CarPlay session (CarPlay Session Start).

In the event the Apple device with the active CarPlay session is disconnected, the accessory must enable the Bluetooth subsystem and attempt to reconnect to the Apple device. This requirement is meant to re-establish the CarPlay session with an Apple device when a sudden Wi-Fi disconnect occurs.

23.2.5.7 Supporting Multiple Devices

At any time, there is only one active CarPlay session. However, accessories operating in the 5 GHz frequency band may support multiple Apple devices connected to the accessory's Wi-Fi access point by providing a user interface to select which Apple device to actively use for CarPlay. If the accessory provides a user interface to select the active CarPlay device, it must display a list of available Apple devices obtained by actively scanning for all previously paired Bluetooth devices, and using the CarPlay Control Bonjour Service to obtain the status of each Apple device.

Upon ignition, the accessory must scan for all previously paired Apple devices. When Apple devices are discovered, the accessory must attempt to reconnect to the last device used for CarPlay, or to the preferred device that is set up by the user. If neither is found, the accessory must connect an available Apple device.

When there is an active CarPlay session, the accessory must never switch to another Apple device without explicit user action. If the active CarPlay device is disconnected for any reason, the accessory must rescan for available devices and attempt to reconnect the previously active CarPlay device.

Accessories operating in the 2.4 GHz band must support only a single Apple device for CarPlay over wireless. The accessory must not allow Bluetooth pairing while there is an active CarPlay session.

23.2.5.8 Wireless Coexistence

Accessories incorporating other RF technologies such as Bluetooth, multiple Wi-Fi access points, LTE, wireless audio systems, etc. must first consult with Apple to plan co-existence scenarios and testing.

23.2.5.8.1 Wi-Fi and Bluetooth Coexistence

CarPlay over wireless Wi-Fi and Bluetooth protocol traffic may interfere with each other when operating in the 2.4 GHz frequency band.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

If the accessory Wi-Fi access point is operating in the 2.4 GHz frequency band, the Bluetooth subsystem must be disabled. Once the Apple device has established a Wi-Fi connection and successfully initiated a CarPlay session, the Bluetooth link must be terminated with the Apple device, the accessory's Bluetooth subsystem must be disabled, and all Bluetooth connections to other devices must be terminated.

If the accessory Wi-Fi access point is operating in the 5 GHz frequency band, Bluetooth protocols and profiles can be supported and enabled with other devices that are not engaged in an active CarPlay session. Once the Apple device has established a Wi-Fi connection and successfully initiated a CarPlay session, the Bluetooth link must be terminated with the Apple device. A Bluetooth connection and profiles may be established with other devices that are not engaged in a CarPlay session.

23.2.5.8.2 Wi-Fi and Cellular Coexistence

CarPlay over wireless and Internet Sharing Services using LTE on Band 40 may interfere with each other when operating in the 2.4 GHz frequency band. In order to maintain the required performance and user experience, the accessory Wi-Fi access point will be limited to a set of 2.4 GHz operating channels.

If the accessory provides Internet sharing services and uses LTE on Band 40 as the WWAN communication, then the accessory Wi-Fi access point must use the operating channels in Table 23-8 (page 407).

Table 23-8 Operational Channels for Cellular Coexistence

Channel	Frequency
6	2.437 GHz
11	2.462 GHz

If the accessory can demonstrate a minimum of 30 dBm of isolation between the Wi-Fi antenna and Cellular antenna, then the accessory Wi-Fi access point may use any 2.4 GHz channels specified in Table 23-4 (page 398).

23.2.5.8.3 Wi-Fi and Coexistence with Other RF Technologies

If the accessory supports other RF technologies in the vehicle, the manufacturer must communicate to Apple the operating properties of these RF technologies:

- Frequency Band
- Channel Width
- Protocol

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Other RF technologies operating in the vehicle must not interfere with the Wi-Fi access point providing CarPlay over wireless services since they may cause a performance degradation.

23.2.5.8.4 Multiple Wi-Fi Access Points

If the accessory supports multiple Wi-Fi access point in the vehicle, the manufacturer must communicate to Apple the operating parameters of other Wi-Fi access points that are not intended for CarPlay over wireless.

- Frequency Band
- Channel Width
- Wi-Fi Protocol
- Security Mode
- SSID

Other Wi-Fi access points must not interfere with the Wi-Fi access point providing CarPlay over wireless service since they may cause a performance degradation.

If other Wi-Fi access point(s) are configured with a different SSID and WPA2 passphrase, the non-CarPlay Wi-Fi access points must operate in a different channel.

The non-CarPlay Wi-Fi access points must operate in 2.4 GHz and 5 GHz channels that are not used by the CarPlay Wi-Fi access point.

23.2.6 Transitioning Between Wireless and USB

This section defines requirements for accessories supporting CarPlay over both wired and wireless transports.

23.2.6.1 Wireless to USB

Upon detection of a USB connection with an Apple device, the accessory must use the Get Supported Capabilities USB Vendor Request to determine if CarPlay is enabled. If CarPlay is enabled, the accessory must determine if there is already an active CarPlay session over wireless using Device Notifications Usage (page 526) and DeviceUUIDUpdate (page 842).

If the device has CarPlay enabled and there is no active CarPlay session over wireless, the accessory must initiate USB role switch to start CarPlay. See Role Switch (page 389).

If a CarPlay session over wireless is already established with the Apple device, the accessory must perform USB role switch and initiate an iAP2 session for charging only. The accessory must complete accessory authentication and declare their capability to provide power, see Providing Power to the Apple Device (page 661).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

If a CarPlay session over wireless is already established with another Apple device, or if CarPlay is not enabled on the device, the accessory may connect to the device for music browsing.

23.2.6.2 USB to Wireless

When any Apple device is disconnected from the accessory via USB, the accessory must initiate Bluetooth discovery, see Bluetooth (page 395). This ensures that the Apple device can be used wirelessly.

23.2.7 Software Clients

The accessory must implement iAP2 and CarPlay clients to support the message protocols and states specific to each interface.

Portions of both clients are provided as reference source code by Apple, namely:

- An iAP2 link layer
- A CarPlay Communication Plug-in, see CarPlay Communication Plug-in (page 485)

23.2.7.1 iAP2 Client over USB

For accessories which support CarPlay over USB, at a minimum, the iAP2 client must support:

- Accessory Authentication (page 261)
- Accessory Identification (page 265) with VehicleInformationComponent (and WirelessCarPlayTransportComponent if the accessory supports CarPlay over wireless)
- Power (page 660)
- Location Information (page 637) and provide location information from GPS and other sensor data

The location information will be used to augment the Apple device's built-in location services and help conserve device power.

Additionally, the iAP2 interface enables transfer of metadata and state information to the accessory from select iOS apps running in CarPlay, such as Music and Telephony. This information may be used to complement the CarPlay experience on an additional accessory screen that is not the primary CarPlay screen; for instance, to display call information on a vehicle instrument cluster display.

Accessories may also provide sensor data and other vehicle status to the Apple device, see Vehicle Status (page 718).

23.2.7.2 iAP2 Client over Bluetooth

For accessories which support CarPlay over wireless, at a minimum, the iAP2 client must support:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Accessory Authentication (page 261)
- Accessory Identification (page 265) with BluetoothTransportComponent, VehicleInformationComponent, and WirelessCarPlayTransportComponent
- RequestAccessoryWiFiConfigurationInformation (page 877) and AccessoryWiFiConfigurationInformation (page 877)

The iAP2 client over Bluetooth is used only to provide Wi-Fi credentials and must be disconnected once the CarPlay session is established.

23.2.7.3 CarPlay Client

The CarPlay client must implement the logic described in Resource Management (page 448) for resource sharing between the accessory and Apple device. It must also manage user events and the transfer of digital UI and audio between the accessory and the Apple device.

Payload data for the CarPlay client may be encoded according to a variety of standards dependent on the content type, as described in Media Types and Formats (page 411), but the underlying transport layer employed by the Communication Plug-in will consist of one or more channels implementing either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). For TCP, see *IETF RFC 793, Transmission Control Protocol*, September 1981. For UDP, see *IETF RFC 768, User Datagram Protocol*, August 1980.

The Communication Plug-in will open a number of additional UDP and TCP channels in parallel to handle synchronization, retransmission requests, usage reports for user actions and other custom commands and controls.

The accessory must provide a networking stack that supports multiple concurrent UDP and TCP channels for:

- audio/video transfer
- command and control
- clock synchronization and retransmission

23.2.7.4 iAP2 Client over CarPlay Client

For accessories which support CarPlay over wireless, the CarPlay Communication Plug-in enables iAP2 over CarPlay, see iAPSendMessage (page 484).

At a minimum, the iAP2 over CarPlay client must support:

- Accessory Authentication (page 261)
- Accessory Identification (page 265) with VehicleInformationComponent, LocationInformationComponent, and WirelessCarPlayTransportComponent

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Location Information (page 637) and provide location information from GPS and other sensor data

For CarPlay over wireless, the accessory must provide location information using GNSS Mode, see Global Navigation Satellite System (GNSS) Mode (page 638), as the Apple device may be positioned in a location without any satellite visibility.

Additionally, the iAP2 interface enables transfer of metadata and state information to the accessory from select iOS apps running in CarPlay, such as Music and Telephony. This information may be used to complement the CarPlay experience on an additional accessory screen that is not the primary CarPlay screen; for instance, to display call information on a vehicle instrument cluster display.

Accessories may also provide sensor data and other vehicle status to the Apple device, see Vehicle Status (page 718).

23.2.8 Media Types and Formats

Video content (a User Interface stream) will stream from the Apple device on a dedicated channel. Two additional dedicated channels are required for audio: main audio (input and output) and alternate audio.

All control and video data packets are encrypted (AES-128). The task of encryption/decryption is abstracted for the accessory maker within the Apple-provided Communication Plug-in code.

Apple devices will not stream content subject to DRM (Digital Rights Management) over CarPlay.

23.2.8.1 User Interface (UI) Stream

The UI is streamed from the Apple device as individual, H.264 NAL units configured with AVCC header data and an additional presentation timestamp for scheduling and synchronization.

The AVCC Format is specified in *ISO/IEC STANDARD 14496-15 (AVC file format)* section 5.2.4.1.1. If the accessory decoder does not natively support AVCC format, but instead requires Annex-B format, a set of utility functions are provided in the Communication Plug-in to assist with format conversion. Note however, that this conversion, if required, is software based, not hardware accelerated, and may impact performance and latency.

The H.264 Sequence Parameter Set (SPS) and Picture Parameter Set (PPS) are only sent once, at the beginning of the stream.

In order to minimize video encoding and decoding latency, no frame reordering is used (i.e. no B-frames). The accessory must support the SPS Video Usability Information (VUI) header with the bitstream_restriction_flag set, max_num_reorder_frames of 0, and max_dec_frame_buffering set to the maximum Decoded Picture Buffer (DPB) size.

Video is encoded using full range Y'CbCr, black is 0, white is 255. The accessory must support the SPS video_full_range_flag to reflect this. See Annex E of the *H.264* specification for more information.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The accessory's target resolution, max frame rate, and physical dimensions for the CarPlay content must be reported via the Info Message (page 430). Resolutions natively supported by the Apple device and its set of CarPlay compatible applications are listed in Table 23-9 (page 412).

The accessory must implement an H.264 decoder capable of supporting at least High Profile 3.1 and content with a 800 x 480 resolution at 30 fps using YCbCr 4:2:0 chroma subsampling, but higher frame rates are strongly recommended. The decoder must be capable of supporting the H.264 level corresponding to the resolution and frame rate reported to CarPlay. See Table 23-10 (page 412).

Table 23-9 CarPlay H.264 Video Stream Standard Resolutions

Aspect Ratio	Pixel Dimensions	Frame Rate
16:9	960 x 540, 1280 x 720	60 fps (recommended), 30 fps
15:9	800 x 480	60 fps (recommended), 30 fps
24:9	1920 x 720	60 fps (recommended), 30 fps

Table 23-10 CarPlay H.264 Levels

H.264 Level	Max Bit Rate Wired	Max Bit Rate Wireless	Examples
High Profile 3.1	17.5 Mbps	10 Mbps	800x480@30fps 800x480@60fps 960x540@30fps 1280x720@30fps
High Profile 3.2	25 Mbps	10 Mbps	960x540@60fps 1280x720@60fps
High Profile 4.0	25 Mbps	10 Mbps	1920x720@30fps
High Profile 4.2	25 Mbps	10 Mbps	1920x720@60fps

The accessory's digital video display must natively support one of the standard resolutions encoded by the Apple device and display the CarPlay content in a full screen configuration. The accessory may optionally use a larger display with a resolution higher than one of the standard resolutions and display the CarPlay content in a windowed configuration. Windowed configurations may have additional design requirements and must be reviewed by Apple. The CarPlay content must always be rendered pixel for pixel without resorting to scaling or stretching. See High Resolution Display (page 383) for display requirements.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Accessory makers wishing to offer different target resolutions for the UI stream and/or nonsquare pixel displays must first consult with Apple for compatibility.

The CarPlay UI must appear on the accessory display within 500 ms of either:

- The accessory receiving a screen stream setup request.
- The accessory receiving a session configuration that includes a screen stream.

23.2.8.2 Audio

The accessory must support audio input and output streams encoded in the form of uncompressed LPCM. Volume control and display of volume control UI are the responsibility of the accessory because the Apple device provides only unattenuated line-level audio output.

See Volume Management (page 423) for specific requirements and recommendations.

The accessory must support one main (input and output) and one alternate audio stream from the Apple device as described in Resources (page 448). Alternate audio must be mixed with main audio regardless of whether the source of main audio is the Apple device (for example, the iOS Music app) or the accessory (for example, the FM tuner on a vehicle head unit). In some use cases output on alternate audio might be accompanied by a request to duck the main audio channel (see Ducking (page 423)).

The audio sample rates supported by the accessory must be reported using the Info message, see Info Message (page 430).

Accessories that implement the CarPlay feature must respect the requirements from Multiple Audio Connections (page 528). In addition, they must use the CarPlay audio stream when using the feature and cannot use USB Host Mode Audio Transport in parallel.

The Apple device will at times simultaneously employ different sample rates for the main (input and output) and alternate audio streams, or may require sample rate transitions, for instance when transitioning main audio from music to a phone call. Main and alternate audio must be capable of maintaining independent sample rates. For example, if the main audio channel is operating at 16 kHz for telephony, alternate audio should continue to operate at a higher sample rate, e.g. 44.1 kHz, to preserve the fidelity of alerts and other alternate audio that would mix with phone call audio. All sample rate transitions will be communicated to the accessory by the Apple device using the Setup Message (page 440).

CarPlay over USB uses LPCM for audio. CarPlay over wireless uses raw AAC-LC for high latency audio (Main High Audio) and either OPUS or raw AAC-ELD for low latency audio (Main Audio except "media" and Alt Audio). Accessories supporting CarPlay over wireless must support multiple decode instances and concurrent decode/encode instances.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Also, the Apple device will communicate the intended use of the audio stream to allow the accessory to: (i) adjust its microphone audio processing (such as noise reduction, echo cancellation, frequency response, and gain), and (ii) suppress or eliminate additional noise sources (for example, temporarily turning down the defroster in a vehicle to reduce ambient noise).

After the Apple device sends a Setup Message (page 440) to request main (input and output) or alternate audio streams, the accessory must reply to the setup message within 100 ms and only after:

- It is ready to begin receiving and outputting audio samples (if main audio with input and output was requested). For output, the accessory must output all samples given to it after an acknowledgement of the setup message. For input the accessory must be ready to send audio samples to the Apple device after the acknowledgment of the setup message.
- It is ready to begin to continuously receive audio from the Apple device (if main or alternate audio output was requested). For output, the accessory must output all samples given to it after an acknowledgement of the setup message.

The accessory must support full-duplex audio on the main audio stream at the same sample rate in both input (microphone) and output directions. In addition, input and output must be driven from the same clock to avoid the need for asynchronous sample rate conversion.

The accessory must not buffer more than 10 ms of audio at a time to ensure that transfer delays do not cause gaps in the audio stream.

At no time during interaction with the Apple device must any popping, clicking, or other audible artifacts occur on the accessory, whether it be during transitions between shared resource ownership or mode changes, starting and stopping IO, initiation or cessation of ducking, or any other circumstance. Additionally, volume changes and ducking must be free any zippering or other audible artifacts. The output to the DAC for an unencoded LPCM audio signal sent from the Apple device should be bit-identical to that of an audio signal sent from other sources, assuming those sources allow direct digital transfer.

23.2.8.2.1 Main Audio - Entertainment

For wired sessions, entertainment audio is sent via the low latency stream (Main Audio). For wireless sessions, entertainment audio is sent via the high latency stream (Main High Audio).

Sample Rate

Table 23-11 (page 415) details the audio stream requirements. The accessory must support both 44.1 kHz 16-bit and 48 kHz 16-bit audio. At least one of these two sample rates must be natively supported, that is, a sample rate at which the accessory can perform its audio processing without sample rate conversion. See Table 23-28 (page 435) for a list of audio configurations.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-11 Entertainment Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
44.1 kHz	16 bit	stereo	n/a	Required
48 kHz	16 bit	stereo	n/a	Required

Latency

Latency from the receipt of a buffer of LPCM audio by the accessory from the CarPlay Communication Plug-in to the time it is emitted by the accessory's speaker must not exceed 35 ms.

23.2.8.2.2 Main Audio - Telephony

Unless otherwise specified, the accessory must meet the recommendations and test criteria in International Telecommunication Union Recommendations *ITU-T P.1100: Narrowband hands-free communication in motor vehicles* and *ITU-T P.1110: Wideband hands-free communication in motor vehicles*, whichever applies to the current use case.

Proof of compliance with the *ITU-T P.1100* and *ITU-T P.1110* specifications must be submitted to Apple as part of the self-certification audit process.

Sample Rate

Table 23-12 (page 415) details the audio stream requirements. The accessory must accept and provide the required sample rates with equal or higher bit depth.

Table 23-12 Telephony Audio Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
8 kHz	16 bit	mono	full-duplex	Required (wired only)
16 kHz	16 bit	mono	full-duplex	Required
32 kHz	16 bit	mono	full-duplex	Recommended

Apple recommends implementing support for 32 kHz sampling. This is a key feature of VoLTE. Future voice communication standards may incorporate 48 kHz sample rates and stereo.

Audio Processing

The accessory must perform echo cancellation and noise processing tuned for telephony and provide full-duplex audio that is free of echoes, noise, or other artifacts.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Uplink Output Level

In the nominal test arrangement, the accessory must provide a constant average uplink output level to the device. The use of automatic gain control (AGC) targeting the specified level is recommended. The nominal level measured at the uplink output of the accessory must be A-weighted -30 dB ±2 dB root-mean-square (RMS), expressed in units relative to full-scale (dBFS(A)). Alternatively, the nominal level may be 13 dB ±2 dB SLR if using the ITU measurement procedure.

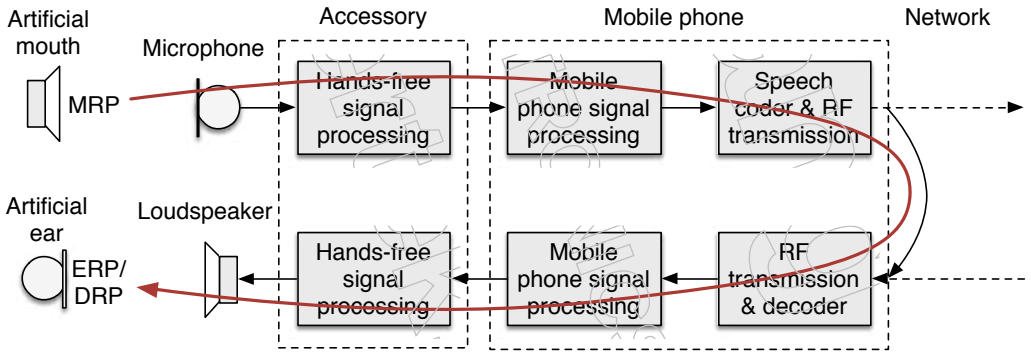
Signal-to-Noise Ratio

The accessory must maintain adequate signal-to-noise ratio (SNR) in its uplink output for voice communication purposes. The absolute SNR must be over 12 dB in all typical driving conditions as specified in Annex D of *ITU-T P.1100* / *ITU-T P.1110*. Higher values are preferred and the SNR should be much higher in a stationary vehicle and at slow driving speeds.

Round-Trip Delay

The definition of the combined round-trip delay of the accessory and the device follows *3GPP TS 26.132*, consisting of the sum of uplink and downlink processing latency of all acoustic, analog, and digital components of the accessory and the device, including any digital buffering. The round-trip path is illustrated in Figure 23-5 (page 416).

Figure 23-5 Telephony Audio Round-Trip Path



The combined round-trip delay of the accessory and the device must not exceed 275 ms for wired transports and 415 ms for wireless transports. Smaller values are preferred and ideally the delay should be under 200 ms.

Send Frequency Response

The send frequency response must meet the requirements in *ITU-T P.1100*, Clause 11.4.1 for narrowband telephony and *ITU-T P.1110*, Clause 11.4.1 for wideband telephony.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.2.8.2.3 Main Audio - FaceTime Audio

Unless otherwise specified, the accessory must meet the recommendations and test criteria specified in *ITU-T P.1110*.

FaceTime Audio operates at 24 kHz and uses high-quality audio codecs.

Sample Rate

Table 23-13 (page 417) details the audio stream requirements. The accessory must accept and provide the following sample rate with equal or higher bit depth.

Table 23-13 FaceTime Audio Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
24 kHz	16 bit	mono	full-duplex	Required

In the future, even higher sample rates such as 32 kHz and 48 kHz and stereo may be used.

Audio Processing

The accessory must perform echo cancellation and noise processing tuned for telephony and provide full-duplex audio that is free of echoes, noise, or other artifacts. Due to high-quality audio coding, it is important to preserve the quality of the uplink signal. Less aggressive noise suppression than in telephony should be used if it results in better speech quality in the send direction (see Clause 12.5.1 in *ITU-T P.1110*).

Uplink Output Level

In the nominal test arrangement, the accessory must provide a constant average uplink output level to the device. The use of an AGC targeting the specified level is recommended, but the device should preserve linear operation for any constant speech level. The nominal level measured at the uplink output of the accessory must be A-weighted -30 dB ±2 dB root-mean-square (RMS), expressed in units relative to full-scale (dBFS(A)). Alternatively, the nominal level may be 13 dB ±2 dB SLR if using the ITU measurement procedure.

Signal-to-Noise Ratio

The accessory must maintain adequate signal-to-noise ratio (SNR) in its uplink output for voice communication purposes. The absolute SNR must be over 12 dB in all typical driving conditions as specified in Annex D of *ITU-T P.1110*. Higher values are preferred and the SNR should be much higher in a stationary vehicle and at slow driving speeds.

Round-Trip Delay

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The round-trip delay must meet the same requirements as defined for Telephony (see Main Audio - Telephony (page 415)).

Send Frequency Response

The normalized send frequency response is measured from the mouth reference point (MRP) to the uplink output of the accessory (input to the device). The normalized send frequency response must be within the limits of the tolerance mask shown in Table 23-14 (page 418). The mask (see Figure 23-6 (page 419)) is drawn by straight lines between the breaking points in the table on a linear (dB sensitivity) - logarithmic (frequency) scale. Ideally, the response characteristics of the accessory should be flat in the frequency range of 100 Hz - 11 kHz. Extending the audio bandwidth to above 8 kHz is strongly recommended.

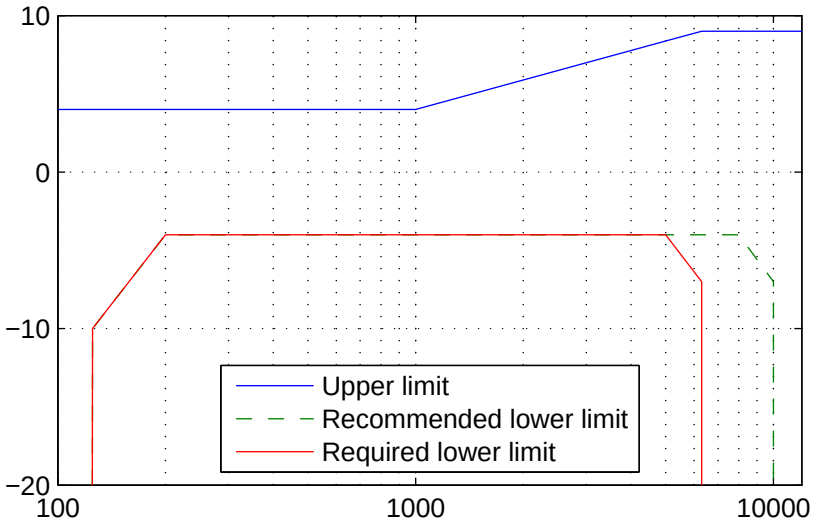
Table 23-14 Audio Tolerance Mask Limits For Send Frequency Response

Frequency	Upper Limit	Recommended Lower Limit	Required Lower Limit
100 Hz	4 dB	-∞ dB	-∞ dB
125 Hz	4 dB	-10 dB	-10 dB
200 Hz	4 dB	-4 dB	-4 dB
1,000 Hz	4 dB	-4 dB	-4 dB
5,000 Hz	(* See note.)	-4 dB	-4 dB
6,300 Hz	9 dB	-4 dB	-7 dB
8,000 Hz	9 dB	-4 dB	-∞ dB
10,000 Hz	9 dB	-7 dB	-∞ dB
12,000 Hz	9 dB	-∞ dB	-∞ dB

* Note: The limits for intermediate frequencies lie on a straight line drawn between the given values on a linear (dB) - logarithmic (Hz) scale. See Figure 23-6 (page 419)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 23-6 Audio Tolerance Mask For Send Frequency Response (Sensitivity (dB) vs. Frequency (Hz))



23.2.8.2.4 Main Audio - Siri

Unless otherwise specified, the accessory must meet the recommendations and test criteria specified in *ITU-T P.1110* .

Sample Rate

Table 23-15 (page 419) details the audio stream requirements. The accessory must accept and provide the following sample rate with equal or higher bit depth.

Table 23-15 Siri Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
24 kHz	16 bit	mono	full-duplex	Required

Audio Processing

Directional microphones and linear beamforming with microphone arrays giving improved SNR are recommended. Linear echo cancellation for reducing unwanted audio sources (such as audio output from the system) without having any other effect on the speech signal are also recommended. However, single channel noise reduction methods (such as spectrum subtraction) must not be applied, as they will be detrimental to the speech recognition accuracy. Similarly, automatic gain control, residual echo suppression and attempts to blank out non-speech periods in the waveform must not be applied.

Uplink Output Level

In the nominal test arrangement, the accessory must provide a constant uplink output level to the device. The use of an AGC is not recommended. If the accessory adjusts signal gain, the gain must be held constant across each spoken utterance. The nominal level measured at the uplink output of the accessory must be A-weighted -30 dB ±2 dB root-mean-square (RMS), expressed in units relative to full-scale (dBFS(A)). Alternatively, the nominal level may be 13 dB ±2 dB SLR if using the ITU measurement procedure.

Signal-to-Noise Ratio

The accessory must maintain adequate signal-to-noise ratio (SNR) in its uplink output for speech recognition purposes. To maintain high speech recognition accuracy, the absolute SNR should be over 20 dB and all reasonable efforts should be made to maintain high SNR in all typical driving conditions as specified in Annex D of *ITU-T P.1110*. Values above 20 dB are recommended, and the SNR should be at least 25 dB in a stationary vehicle and at slow driving speeds.

Latency

Latency from the reception of sound at the accessory's microphone and the receipt by the CarPlay Communication Plug-in of a buffer of LPCM audio representing that sound must not exceed 60 ms. Likewise, latency from the receipt of a buffer of LPCM audio by the accessory from the CarPlay Communication Plug-in to the time it is emitted by the accessory's speaker must not exceed 35 ms.

Send Frequency Response

The normalized send frequency response for Siri must meet the same requirements as defined for FaceTime. See Main Audio - FaceTime Audio (page 417)

23.2.8.2.5 Main Audio - Alert

The alert audio type is a low latency audio stream.

Sample Rate

Table 23-16 (page 420) details the audio stream requirements. The accessory must support at least one of 44.1 kHz 16-bit or 48 kHz 16-bit audio natively, that is, a sample rate at which the accessory can perform its audio processing without sample rate conversion. See Table 23-28 (page 435) for a list of audio configurations.

Table 23-16 Alert Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
44.1 kHz	16 bit	mono or stereo	n/a	Required (if 48 kHz 16 bit audio is not supported)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
48 kHz	16 bit	mono or stereo	n/a	Required (if 44.1 kHz 16 bit audio is not supported)

Latency

Latency from the receipt of a buffer of LPCM audio by the accessory from the CarPlay Communication Plug-in to the time it is emitted by the accessory's speaker must not exceed 35 ms.

23.2.8.2.6 Main Audio - Default

The default audio type is a duplex audio stream.

Sample Rate

Table 23-17 (page 421) details the audio stream requirements. The accessory must accept and provide the required sample rates with equal or higher bit depth.

Table 23-17 Default Audio Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
16 kHz	16 bit	mono	full-duplex	Required
24 kHz	16 bit	mono	full-duplex	Required
32 kHz	16 bit	mono	full-duplex	Recommended
44.1 kHz	16 bit	mono	full-duplex	Recommended
48 kHz	16 bit	mono	full-duplex	Recommended

Audio Processing

The accessory must not perform echo cancellation or noise processing and must provide full-duplex audio.

Latency

For full duplex audio, the accessory must meet the same latency requirements as defined in Main Audio - Telephony (page 415) and Main Audio - FaceTime Audio (page 417).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.2.8.2.7 Main High Audio - Entertainment

Main High audio is a high latency audio stream for entertainment audio used only for CarPlay over wireless. It replaces the main audio "entertainment" stream used for CarPlay over USB.

Sample Rate

Table 23-18 (page 422) details the audio stream requirements. The accessory must support at least one of 44.1 kHz 16-bit or 48 kHz 16-bit audio natively, that is, a sample rate at which the accessory can perform its audio processing without sample rate conversion. See Table 23-28 (page 435) for a list of audio configurations.

Table 23-18 Main High Audio Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
44.1 kHz	16 bit	mono or stereo	n/a	Required (if 48 kHz 16 bit audio is not supported)
48 kHz	16 bit	mono or stereo	n/a	Required (if 44.1 kHz 16 bit audio is not supported)

Latency

Latency from the receipt of a buffer of AAC-LC audio by the accessory from the CarPlay Communication Plug-in to the time it is emitted by the accessory's speaker must not exceed 35 ms.

23.2.8.2.8 Alternate Audio

Sample Rate

Table 23-19 (page 422) details the audio stream requirements. The accessory must support at least one of 44.1 kHz 16-bit or 48 kHz 16-bit audio natively, that is, a sample rate at which the accessory can perform its audio processing without sample rate conversion. See Table 23-28 (page 435) for a list of audio configurations.

Table 23-19 Alternate Audio Stream Requirements

Sample Rate	Bit Depth	Channels	Duplexing	Requirement
44.1 kHz	16 bit	mono or stereo	n/a	Required (if 48 kHz 16 bit audio is not supported)
48 kHz	16 bit	mono or stereo	n/a	Required (if 44.1 kHz 16 bit audio is not supported)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Latency

Latency from the receipt of a buffer of LPCM audio by the accessory from the CarPlay Communication Plug-in to the time it is emitted by the accessory's speaker must not exceed 35 ms.

23.2.8.2.9 Volume Management

The accessory's volume controls (physical or on-screen) must control the volume of audio from the Apple device.

If the accessory supports maintaining separate volumes for different types of audio, it is recommended that it does so using the following volume categories, with corresponding mappings to main audio stream audioTypes (see Table 23-40 (page 443)), and alternate audio:

- Entertainment - main audio "media"
- Ringtone - main audio "alert"
- Phone Calls - main audio "telephony"
- Voice Prompts and Notifications - main audio "speech recognition" and alternate audio
- Other - main audio "default"

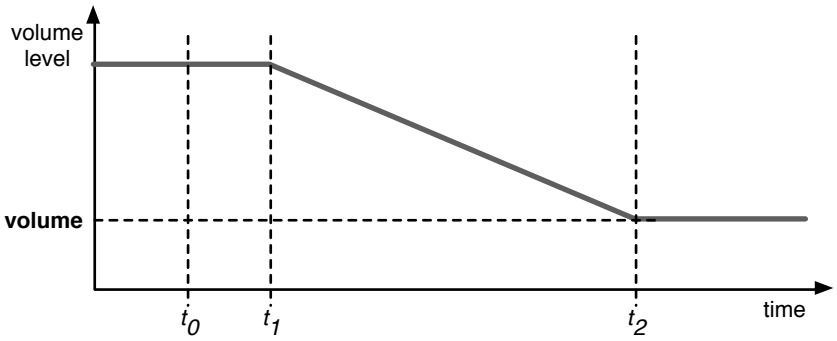
For example, were the user to change the volume while listening to music from the device, the accessory would remember that volume for the Entertainment volume category, and subsequently restore that volume the next time the main audio audioType was configured for "default" or "media".

23.2.8.2.10 Ducking

The accessory must support ducking, the process of ramping the main audio volume down so the user can more clearly hear an announcement or alert from the alternate audio channel. This ramping must be free of any zippering or other audible artifacts. For use cases where ducking is required, the device will send a "duckAudio" command to the accessory. For example, if the user is listening to music and they are getting close to their destination, the music may be ducked (attenuated) so that another stream can say "Arriving at your destination." Ducking fades the main audio volume down to a lower level so it is not an abrupt change. When the announcement is done, the original stream is unducked by fading back to the original level.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 23-7 Event timeline for audio ducking/unducking



t_0: duckAudio command received by accessory
t_1: accessory audio engine ready to begin volume ramp
t_2: volume ramp ends; t_2 - t_0 = **durationMs**

The duck/unduck ramp (t1 - t0) must start within a very short time after the message has been received. Typical and recommended times are 10-20 ms; response within 75 ms is required.

23.3 CarPlay Communication Protocol

The CarPlay communication protocol provides a mechanism for an accessory and an Apple device to:

- Discover and connect to one another on a network
- Setup and control audio and video (UI) streaming
- Arbitrate ownership and use of the accessory's shared resources
- Communicate the Apple device's app state to avoid conflicts that would lead to undesirable user experiences.
- Exchange user input events

Implementation of this protocol must incorporate the reference implementation provided in the CarPlay Communication Plug-in.

The CarPlay communication protocol is flexible in its parsing of parameter values. CarPlay message parameters that are numbers, enum values, or booleans may be provided as strings that parse as those data types.

23.3.1 Discovery

This section contains requirements for Apple device discovery by accessories as well as accessory discovery by Apple devices.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.3.1.1 Apple Device Discovery by Accessories

Apple devices supporting CarPlay will advertise a Bonjour service to the accessory. The advertised services is of a server type `_carplay-ctrl._tcp`. The service provides an HTTP-based protocol to initiate a CarPlay session with an Apple device.

Accessories must use this service to discover CarPlay enabled devices and to initiate the CarPlay session to a specific device. The accessory may use the service to display a list of CarPlay enabled devices to the user.

> **Note:** Apple devices which do not support the CarPlay Control service (iOS 8.2 or older), will initiate the CarPlay session automatically.

The CarPlay Communication Plug-in implements the Bonjour discovery and provides a command protocol to the accessory. See CarPlay Control (page 425).

The name of the Bonjour service is the user-visible name of the Apple device (e.g. "Tim's iPhone"). The name may contain any Unicode character and is encoded using UTF-8. It has a maximum length of 63 bytes (which may be fewer than 63 characters as a single Unicode character may require multiple bytes). Additional metadata needed at discovery-time is advertised via a TXT record as defined in Table 23-20 (page 425).

Table 23-20 CarPlay Control Bonjour Service Keys

Key	Description
id	Bluetooth MAC address for the Apple device; e.g., "00:11:22:33:44:55".
srcvers	Apple CarPlay source version assigned by Apple, in the form "x.y.z".

23.3.1.1.1 CarPlay Control

When the accessory wants to send a CarPlay Control command, it must do the following:

1. Look up the DNS name of the Apple device with Bonjour by resolving the _carplay-ctrl._tcp service.
2. Resolve the DNS name to the IP address(es) of the CarPlay Control Apple device. If the accessory platform's `getaddrinfo` is Bonjour aware then it may be used. Otherwise, Bonjour's asynchronous DNSServiceGetAddrInfo API should be used.
3. Connect to the CarPlay Control Apple device's IP address(es). There may be multiple IP addresses associated with the DNS name. The accessory should try each IP address until it makes a successful TCP connection (or runs out of IP addresses and fails). IPv6 addresses are generally preferred because in some environments, the Apple device and the accessory may be on different IPv4 subnets, even if they are on the same physical

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

link. Prioritizing link-local IPv6 addresses over IPv4 addresses often means a faster and more reliable connection. Link-local addresses should not be used as they will not be able to wake an Apple device that is in a sleep state.

4. Send the command using an HTTP request.

CarPlay Control requests look like normal HTTP requests. The request line of the request uses the following format to specify the command:

```
GET /ctrl-int/1/<command> HTTP/1.1
```

The command must be one of the following:

Table 23-21 CarPlay Control Commands

Command	Description
connect	Requests that the Apple device begin a CarPlay session with the accessory issuing the request.

All command requests must include the `AirPlay-Receiver-Device-ID` header field. This header field specifies the accessory that the Apple device should connect to. The value of this field is the primary MAC address of the accessory as a 48-bit integer.

For example, if an accessory wanted a Apple device to initiate a CarPlay session to it then it would send the following HTTP request to the Apple device:

```
GET /ctrl-int/1/connect HTTP/1.1
Host: bb15.local.
AirPlay-Receiver-Device-ID: 18838586676582
```

23.3.1.2 Accessory Discovery by Apple Devices

CarPlay accessories must advertise a Bonjour service to the Apple device. The advertised service must be of a server type `_airplay._tcp`. Apple devices browse for these accessories and offer them to the user if they are found and determined to be compatible.

Additional metadata needed at discovery-time is advertised via a TXT record containing fields for feature detection, model information, versions, etc. as defined in Table 23-22 (page 427). The CarPlay Communication Plug-in will interface to the accessory's Bonjour implementation to configure and broadcast these values.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-22 CarPlay Bonjour Service Keys

Key	Required?	Description
deviceid	Y	Globally unique ID for the accessory; e.g., the primary MAC address, such as "00:11:22:33:44:55".
features	Y	Internally defined for Apple CarPlay. Value must not be modified.
model	Y	Model name of the accessory; must be globally unique and uniquely identify the product. This should be the model name of the vehicle for OEM integrations or an aftermarket unit's model name.
protovers	N	Protocol version string <major>.<minor> (e.g., "1.0"). Missing key defaults to "1.0".
srcvers	Y	Apple CarPlay source version assigned by Apple, in the form "x.y.z".
flags	Y	Status of an operation. Value must not be modified. See Table 23-23 (page 427).

Table 23-23 Status flags bitfield enum values

Value	Bit	Description
0x01	0	Problem has been detected.
0x02	1	Device is not configured.
0x04	2	Audio cable is attached.

The accessory must enable the Bonjour DirectLink optimization. This will optimize the Bonjour discovery process and speed up the connection establishment time over the NCM interface.

23.3.2 Setup and Control

The Communication Plug-in in the CarPlay Client establishes a control channel for the primary tasks of:

- Configuring the UI stream to one of the target screen resolutions listed in Media Types and Formats (page 411).
- Scheduling the video signal to a synchronized clock between both accessory and Apple device (a microsecond resolution clock or better is required on the accessory).
- Configuring and managing the main and alternate audio streams, including callbacks to schedule audio delivery to the accessory and notifications from the Apple device whenever sample rate or format changes are required.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Synchronizing shared resources such as the accessory's screen and speakers, plus determining which media source currently has audio and/or video focus on those resources.

HTTP/RTSP is used to set up, control, and monitor a CarPlay session. When an Apple device discovers a compatible accessory that is either already known by the Apple device or is user selected, the Apple device initiates a connection to the accessory and starts a CarPlay session. The following summarizes a typical session lifetime:

1. The Apple device establishes a TCP control connection to the accessory.
   - This connects to the TCP port advertised via Bonjour.
2. The Apple device authenticates the accessory and sets up a secure channel.
   - This process requires the accessory to perform the message exchange described in MFI-SAP (page 429).
   - After this step, all subsequent control requests and responses are encrypted.
3. The Apple device sends an info request to provide its info and get the accessory's info.
   - The request message is a plist providing the Apple device's model, version, etc.
   - The response message is a plist returning the accessory's audio formats, display info, model, current modes, etc.
   - See Info Message (page 430).
4. The Apple device sends a setup request to set up streams for control and, if needed, audio, screen, etc.
   - This describes each stream to set up, including any TCP/UDP port numbers, configuration options, etc.
   - The accessory responds with details about the streams it set up, such as listening TCP/UDP port numbers.
   - For input or output TCP streams, the accessory will start listening for TCP connections.
   - For input UDP streams, the Apple device will start listening for UDP packets.
   - For output UDP streams, the accessory will start listening for UDP packets.
   - See Setup Message (page 440).
5. The Apple device makes a TCP connection to the accessory for each TCP-based stream.
   - For both input and output streams, the Apple device initiates the connection.
   - For output streams, the Apple device sends data to the accessory (e.g., music to play on the accessory).
   - For input streams, the accessory will send data to the Apple device (e.g., HID reports). See Table 23-40 (page 443) and Table 23-43 (page 444).
6. The Apple device sends a record request to start a session on the accessory.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- The accessory performs time sync negotiation with the Apple device. See Synchronization (page 447).
- The accessory starts corresponding streams, creating threads to process audio/video data, timers, etc.

7. The Apple device sends setup requests to set up streams for audio and/or screen, and makes TCP connections as needed.
8. The Apple device and accessory exchange data.
   - The Apple device sends audio and video output data to the accessory for playback.
   - The accessory sends audio input data (if any) to the Apple device for input.
9. The Apple device sends a TEARDOWN requests to end one or more streams on the accessory.
   - The accessory stops processing for specified streams and releases corresponding resources.
10. The Apple device sends a TEARDOWN request to end the session on the accessory.
    - The accessory stops all audio and video processing and releases all session-specific resources.
11. The Apple device closes all TCP connections to the accessory.
    - This resets the Apple device back to the point before it started the session.

23.3.2.1 MFI-SAP

Authentication over the CarPlay interface requires use of the MFi Authentication Coprocessor for authenticating an encrypted channel. The protocol uses Curve25519 Elliptic-Curve Diffie-Hellman technology for key exchange, as described in cr.yp.to/ecdh.html. It also uses RSA for signing and verifying and AES-128 in counter mode for encryption.

Authentication starts when the Apple device generates a new, random Curve25519 key pair and sends its Curve25519 public key X to the accessory. The accessory responds by sending its public key Y to the Apple device, followed by its MFi Certificate and an encryption of Signature(Y,X).

Messages are exchanged using RTSP/HTTP POST requests and responses to the `/auth-setup` URL. The content of the message is delivered via the body of the RTSP/HTTP message.

When the accessory receives the authentication request, it must:

1. Generate a new, random Curve25519 key pair.
2. Generate the shared secret using its Curve25519 private key and the streamer's Curve25519 public key.
3. Sign the two public keys with its RSA private key and encrypt the result with the AES master key derived from the Curve25519 shared secret.
4. Send its Curve25519 public key, its MFi certificate, and its AES-encrypted signature to the Apple device.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

When the Apple device receives the response, it verifies that the accessory's MFi certificate is valid, decrypts the accessory response and verifies that the signature was signed by that certificate. If the signature verifies, the accessory is considered authenticated. The AES key derived from the Curve25519 shared secret is then used to encrypt all future content.

All Curve25519 keys must be generated from cryptographically strong random numbers. New Curve25519 keys must be generated for each authentication attempt; reuse of keys by the accessory is not permitted.

Each media stream must use a unique key and IV for encrypting content to avoid reusing the same key/IV for different pieces of data. The following are used to derive stream keys and IVs:

- ekey = Bytes 0-15 of SHA-512("AirPlayStreamKey" + streamConnectionID + session key (master))
- eiv = Bytes 0-15 of SHA-512("AirPlayStreamIV" + streamConnectionID + session key (master))

The steam connection ID is contained in Table 23-43 (page 444), and the ekey, eiv, and encryption type are contained in Table 23-36 (page 441). Table 23-24 (page 430) details the values used to specify the encryption type.

Table 23-24 Encryption type enum values

Value	Description
0x00	Invalid.
0x01	Unencrypted.
0x10	MFi-SAP-encrypted AES key.

23.3.2.2 Info Message

The info message provides Apple device information to the accessory and returns accessory information to the Apple device. It is an HTTP GET to the `/info` URL (see Table 23-50 (page 446)) with binary plist request and response payloads. The Apple device may include any information it wants to provide to the accessory in the request plist. The request (see Table 23-25 (page 431)) may contain the "qualifier" key to limit the properties being requested from the accessory; for example, the Apple device may only want to get the accessory's model. The response (see Table 23-26 (page 431)) contains the properties requested by the Apple device.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-25 Info Message request keys

Key	Type	Required?	Description
qualifier	array	N	Array of property strings from Table 23-26 (page 431) to request from the accessory. If this key is missing, the accessory must return all of its properties. If this key is empty, the accessory must not return any properties.

Table 23-26 Info Message response keys

Key	Type	Required?	Description
audioFormats	array	Y	Array of dictionaries for audio formats supported by the accessory. Accessories must provide the supported sample rates for each audioType for all audio streams. In order to maintain compatibilty with versions of iOS that do not support CarPlay over wireless, the first two entries in this array must be: 1) type=100, audioType="compatibility", and must only contain PCM formats 2) type=101, audioType="compatibility", and must only contain PCM formats. These first two entries will be used by earlier versions of iOS that do not directly support the "audioType" key. Following the previous two entries, there must be entries for all valid combinations of type and audioType keys. See Table 23-27 (page 435).
audioLatencies	array	Y	Array of dictionaries specifying the fixed audio latencies within the accessory. See Table 23-29 (page 437). Accessory must specify default latencies for all audio streams, audio types and audio formats (some entries can be omitted based on the optional specifiers). A latter entry in the array overrides an earlier one if they both apply for a given stream type/audio type/format (last one wins).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Key	Type	Required?	Description
bluetoothIDs	array	N (*)	Array of MAC address strings for the Bluetooth modules in the accessory. MAC addresses must be provided in colon separated format, for example "00:11:22:33:44:55". See disableBluetooth (page 478). (*) Required when accessory supports Bluetooth.
deviceID	string	Y	Globally unique device ID: e.g., "00:11:22:33:44:55". See Table 23-22 (page 427).
displays	array	Y	An array of dictionaries for display information supported by the accessory. See Table 23-30 (page 437).
extendedFeatures	array	N (*)	Array of strings indicating support for additional features. See Table 23-33 (page 439). (*) Required when accessory supports CarPlay over wireless.
features	bitfield	Y	Internally defined for Apple CarPlay. Value must not be modified. See Table 23-22 (page 427).
firmwareRevision	string	N	Firmware revision of the accessory, e.g. "FirmwareRevision0.1".
hardwareRevision	string	N	Hardware revision of the accessory, e.g. "HardwareRevision0.1".
hidDevices	array	Y	An array of dictionaries for HID device information. See User Input (page 456).
hidLanguages	array	N	List of BCP-47 language code strings for languages supported by the accessory's character recognizer.
keepAliveLowPower	boolean	Y	Indicates if the accessory supports session idle UDP keepalive.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Key	Type	Required?	Description
keepAliveSendStatsAsBody	boolean	Y	Indicates whether the accessory supports statistics as part of the keep alive message body. This is used for debugging.
limitedUIElements	array	N (*)	Array of UI elements affected by limited UI mode. See Table 23-34 (page 439). (*) Required when limitedUI.
limitedUI	boolean	N	Indicates whether or not certain UI elements are limited. See setLimitedUI (page 484).
macAddress	string	N	MAC address of the accessory's local network interface. See iAP2 / NCM Interface Configuration (page 389). This address must be different from the one which is provided to the Apple device via the CDC Ethernet Networking functional descriptor.
manufacturer	string	Y	Name of the accessory manufacturer (user-visible).
model	string	Y	The model name of the accessory, which must be globally unique and must uniquely identify the product. This should be the model name of the vehicle for OEM integrations or an aftermarket unit's model name. See Table 23-22 (page 427).
modes	group	Y	The modes describing the current state of the accessory, including appState, mainScreen and mainAudio as applicable. See the technical notes in the CarPlay DevKit for more details. The modes dictionary uses the same keys as the "changeModes" command, see Table 23-67 (page 479).
name	string	Y	This must be set to "CarPlay".

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Key	Type	Required?	Description
nightMode	boolean	N	Indicates if it is dark outside and appearance should change to suit night time use. If the accessory does not support night mode, then it should not respond to this property at all (return an unsupported error). The Apple device can then use other metrics (such as time of day) to decide on its own when to enter night mode. See setNightMode (page 484).
oemIcons	array	N (*)	An array of dictionaries for icons representing the accessory manufacturer's logo. Icons should be provided for the following pixel sizes: 120x120, 180x180, and 256x256. The Apple device will use the image sized most appropriately for the usage and accessory screen size. See Table 23-35 (page 440). (*) Required when oemIconVisible is True.
oemIconLabel	string	N (*)	Label shown underneath the oemIcon. (*) Required when oemIconVisible is True.
oemIconVisible	boolean	N	Whether or not the oemIcon is visible on the home screen.
OSInfo	string	N	Operating system information including name, version, and architecture (e.g. "Darwin 13.0.0 x86_64").
protocolVersion	string	Y	Protocol version string major.minor ; e.g., "1.0". The default value is "1.0". See Table 23-22 (page 427).
rightHandDrive	boolean	Y	True if the vehicle is right hand drive.
sourceVersion	string	Y	CarPlay source version assigned by Apple, in the form x.y.z . See Table 23-22 (page 427).
statusFlags	bitfield	Y	Status of an operation. Value must not be modified. See Table 23-22 (page 427).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-27 Audio format keys

Key	Type	Required?	Description
type	enum	Y	The stream type to which the input and/or output formats apply. One of Table 23-45 (page 444).
audioInputFormats	bitfield	Y (*)	The audio input formats supported by the given type. See Audio (page 413) for requirements. (*) Not required if the stream type does not support input. Combination of Table 23-28 (page 435).
audioOutputFormats	bitfield	Y	The audio output formats supported by the given type. See Audio (page 413) for requirements. Combination of Table 23-28 (page 435).
audioType	string	N	The audio type to which the format(s) apply. See Table 23-46 (page 445).

The audio formats for wired and wireless transports must be combined in a single entry. PCM formats must be used for wired and compressed formats must be used for wireless. Multiple compression type (e.g. AAC-ELD and OPUS) must not be offered simultaneously.

The audio input formats supported by the accessory must be a subset of the audio output formats supported. When using full-duplex audio, the Apple device will use the same format for both directions and will choose from the supported input formats.

Table 23-28 Audio formats bitfield enum values

Value	Bit	Description
0x00000004	2	PCM, 8000 Hz, 16-Bit, Mono
0x00000008	3	PCM, 8000 Hz, 16-Bit, Stereo
0x00000010	4	PCM, 16000 Hz, 16-Bit, Mono
0x00000020	5	PCM, 16000 Hz, 16-Bit, Stereo
0x00000040	6	PCM, 24000 Hz, 16-Bit, Mono
0x00000080	7	PCM, 24000 Hz, 16-Bit, Stereo
0x00000100	8	PCM, 32000 Hz, 16-Bit, Mono
0x00000200	9	PCM, 32000 Hz, 16-Bit, Stereo

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Bit	Description
0x00000400	10	PCM, 44100 Hz, 16-Bit, Mono
0x00000800	11	PCM, 44100 Hz, 16-Bit, Stereo
0x00001000	12	Reserved
0x00002000	13	Reserved
0x00004000	14	PCM, 48000 Hz, 16-Bit, Mono
0x00008000	15	PCM, 48000 Hz, 16-Bit, Stereo
0x00010000	16	Reserved
0x00020000	17	Reserved
0x00040000	18	Reserved
0x00080000	19	Reserved
0x00100000	20	Reserved
0x00200000	21	Reserved
0x00400000	22	AAC-LC, 44100 Hz, Stereo
0x00800000	23	AAC-LC, 48000 Hz, Stereo
0x01000000	24	AAC-ELD, 44100 Hz, Stereo
0x02000000	25	AAC-ELD, 48000 Hz, Stereo
0x04000000	26	AAC-ELD, 16000 Hz, Mono
0x08000000	27	AAC-ELD, 24000 Hz, Mono
0x10000000	28	OPUS, 16000 Hz, Mono
0x20000000	29	OPUS, 24000 Hz, Mono
0x40000000	30	OPUS, 48000 Hz, Mono
0x80000000	31	Reserved

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The PCM formats must be only used for CarPlay over USB. For CarPlay over wireless, the AAC-LC compression must only be used for the high latency "main high audio - entertainment" stream, whereas AAC-ELD or OPUS compression must be used for the low latency "main audio" streams (except for "entertainment" audio) and alternate audio. See Audio (page 413).

Table 23-29 Audio latencies keys

Key	Type	Required?	Description
type	enum	Y	The stream type to which the input and/or output formats apply. One of Table 23-45 (page 444).
audioType	string	Y	Type of audio content (e.g. telephony, media, etc.). See Table 23-46 (page 445).
sr	number	N	Number of samples per second (e.g. 44100).
ss	number	N	Bit size of each audio sample (e.g. "16").
ch	number	N	Number of audio channels (e.g. 2 for stereo).
inputLatencyMicros	number	Y (*)	Input latency in microseconds. (*) Only required for duplex audio streams.
outputLatencyMicros	number	Y	Output latency in microseconds.

Table 23-30 Display keys

Key	Type	Required?	Description
edid	data	N	Raw EDID of the display, if available.
features	bitfield	Y	Features of the display as a bitmask. Combination of Table 23-31 (page 438).
maxFPS	number	N	Max frames per second the display supports.
heightPixels	number	Y	Height of the display in pixels. Must be non-zero.
widthPixels	number	Y	Width of the display in pixels. Must be non-zero.
heightPhysical	number	Y	Height of the display in millimeters. Must be non-zero.
widthPhysical	number	Y	Width of the display in millimeters. Must be non-zero.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Key	Type	Required?	Description
uuid	string	Y	UUID of the display.
primaryInputDevice	enum	Y	Primary input device to be used for navigating the user interface. One of Table 23-32 (page 438).

Table 23-31 Display features bitfield enum values

Value	Bit	Description
0x02	1	Supports interacting via knobs.
0x04	2	Supports interacting via low-fidelity touch.
0x08	3	Supports interacting via high-fidelity touch.
0x10	4	Supports interacting via touchpad.

The display features value is a bitmask, so combinations are possible. It is required to set the appropriate bit for all input devices supported by the accessory.

To claim support for high-fidelity touch, the time between a user touch input to the time a frame updates on the display must be less than 140 ms.

Table 23-32 Primary input device enum values

Value	Description
0x01	Accessory uses touchscreen as primary input.
0x02	Accessory uses touchpad as primary input.
0x03	Accessory uses knob as primary input.

Note that the choice of the primary input device, along with the display features supported by the accessory, will directly impact the look and feel of the user interface presented by the Apple device on the accessory's display:

- An accessory that reports supporting low-fidelity touch (display feature bit 2) with touchscreen as the primary input device (value 0x01) will be presented with a limited touch interface with other input methods as secondary inputs. For instance, the user interface for a map will be augmented with soft arrow-keys as overlays to enable panning.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- An accessory that reports supporting high-fidelity touch (display feature bit 3) with touch as the primary input device (value 0x01) will be presented with a touch interface that allows for scrolling and panning gestures with other input methods as secondary inputs.
- An accessory that reports touchpad (display feature bit 4) as primary input device (value 0x02) will be presented with a user interface that may be interacted with primarily via the use of touchpad, select, and back buttons with other input methods as secondary inputs.
- An accessory that reports knob (display feature bit 1) as primary input device (value 0x03) will be presented with a user interface that may be interacted with primarily via the use of knob, select, and back buttons with other input methods as secondary inputs.

Table 23-33 Extended features string values

Value	Description
"enhancedRequestCarUI"	Indicates accessory support for the "url" request key of the requestUI command. See requestUI (page 483).
"vocoderInfo"	Provides the native sample rate of the telephony output stream. As an optimization, the Apple device may prefer to encode the stream at a sample rate other than that used by the cellular network. The accessory may use this parameter to optimize telephony audio playback to the appropriate sample rate. Required when accessory supports CarPlay over wireless.

Table 23-34 Limited UI elements string values

Value	Description
"softKeyboard"	Touch keyboard that appears on screen.
"softPhoneKeypad"	Touch phone keypad that appears on screen.
"nonMusicLists"	Lists of non-music items.
"musicLists"	Lists of music items.
"japanMaps"	Minor roads in Japan.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-35 Icon keys

Key	Type	Required?	Description
imageData	data	Y	PNG data for an icon. Icon can be either a mask or a full-bleed image. Mask icons should be solid white with a transparent, alpha channel background. Full-bleed icons should be 8 bits per channel and must not include an alpha channel. Avoid including icon outlines, shine, or other effects. The Apple device will clip the icon to the appropriate shape and provide any additional styling required to match the design principles of the current version of iOS. The icon should follow the design principles of the current version of iOS. For example, see the Mail envelope icon.
heightPixels	number	Y	Height in pixels of the image.
widthPixels	number	Y	Height in pixels of the image.
prerendered	boolean	Y	Indicates if the icon is full-bleed. False if the icon is a mask icon.

The `/info` URL may also be used via an HTTP GET to read accessory information. The URL may contain a query string to limit the properties being requested (for example, `GET /info?deviceID&model` to get the accessory's device ID and model properties).

23.3.2.3 Setup Message

The setup message is sent by the device to configure and set up all streams between the accessory and the device. The CarPlay streams are logical flows of data, such as a control stream for sending commands and getting responses, a screen stream for sending H.264 frames, an audio stream for sending audio, etc. The setup request message includes descriptions of all the streams the Apple device wants to set up. The response includes details about the streams that were set up, i.e. control only, main audio and screen streams, alternate audio only, etc. Both the request and the response payloads are binary plists.

The initial setup message after the device connects to the accessory contains information to set up the control stream. If needed, it may also setup the audio and screen streams.

Table 23-36 (page 441) lists the specifics of the initial setup request keys sent by the Apple device to the accessory.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-36 Initial Setup request keys

Key	Type	Required?	Description
deviceID	string	Y	Globally unique device ID (e.g. "11:22:33:44:55").
eiv	data	Y	AIS Encryption Initialization Vector for the accessory. This is sent in the clear. See MFI-SAP (page 429)
ekey	data	Y	AIS Encryption Key for the accessory. Encrypted with master key from encryption setup. See MFI-SAP (page 429)
et	enum	Y	Encryption type. One of Table 23-24 (page 430).
macAddress	string	Y	MAC address of the iOS network interface used for the connection.
model	string	Y	Model name of the Apple device (e.g. "Device1,1").
name	string	Y	User-customizable name of the Apple device.
osBuildVersion	string	Y	Operating system build version string (e.g. "11A200").
sessionUUID	string	Y	UUID of the session.
sourceVersion	string	Y	Apple device side CarPlay source version string (e.g. "101.7").
streams	array	Y (*)	Array of stream descriptions. See Table 23-40 (page 443) and Table 23-43 (page 444). (*) The initial setup message may not include "streams" if it only sets up the control stream.
timingPort	number	Y	Port the Apple device is listening on for time sync requests.

Table 23-37 (page 441) lists the specifics of the initial setup response keys sent by the accessory to the Apple device.

Table 23-37 Initial Setup response keys

Key	Type	Required?	Description
eventPort	number	Y	TCP/UDP port number for events.
keepAlivePort	number	Y	UDP port number for session idle UDP keepalive.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Key	Type	Required?	Description
streams	array	Y (*)	Array of stream descriptions. See Table 23-40 (page 443) and Table 23-43 (page 444). (*) The initial setup message may not include "streams" if it only sets up the control stream.
timingPort	number	Y	Port the Apple device is listening on for time sync requests.

Once a session is established, the setup message is sent to set up and configure the audio and screen streams, as they are required by the device.

Table 23-38 (page 442) lists the specifics of the setup request keys sent by the Apple device to the accessory after a session has been started.

Table 23-38 Setup request keys

Key	Type	Required?	Description
streams	array	Y	Array of stream descriptions. See Table 23-40 (page 443) and Table 23-43 (page 444).

Table 23-39 (page 442) lists the specifics of the setup response keys sent by the accessory to the Apple device after a session has been started.

Table 23-39 Setup response keys

Key	Type	Required?	Description
streams	array	Y	Array of stream descriptions. See Table 23-40 (page 443) and Table 23-43 (page 444).

23.3.2.3.1 Streams

Each stream is set up using a stream descriptor with a type, parameters, and behaviors associated with it, as listed in Table 23-40 (page 443) for main/alternate audio stream and Table 23-43 (page 444) for screen stream.

The timestamp and sampleTime response parameters are used for media clock synchronization of the main and alternate audio streams as described in Synchronization (page 447).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-40 Main/alternate audio stream descriptors request keys

Key	Type	Required?	Description
audioFormat	bitfield	Y	Format of the audio data (e.g. 44100 Hz, Stereo PCM). For main audio applies to both input and output. Combination of Table 23-28 (page 435).
audioLatencyMs	number	Y	Desired milliseconds of audio latency. This value is used to set the size of the network jitter buffer and is therefore the largest possible size that can be accomodated by a read or write of audio data. The IO block size used by the accessory to read and write audio data must be no larger than this size and it is strongly recommended that it be smaller in order to avoid under or overflowing the jitter buffer.
audioLoopback	boolean	N	True if audio output must be looped back and sent as audio input. Debug use only.
audioType	string	Y (*)	Type of audio content (e.g. telephony, media, etc.). See Table 23-46 (page 445). (*) Main audio only.
dataPort	number	Y (*)	UDP port the Apple device is listening on for data. (*) Only required when audio input is enabled (main audio only).
input	boolean	N	True if input must be enabled for the stream (main audio only).
type	enum	Y	Type of stream. One of Table 23-45 (page 444).
vocoderInfo	group	Y	Telephony vocoder information. Contains dictionary of Table 23-41 (page 443).

Table 23-41 Vocoder Information keys

Key	Type	Required?	Description
sampleRate	number	Y	The native sample rate of the telephony output stream.

Table 23-42 Main/alternate audio stream descriptors response keys

Key	Type	Required?	Description
dataPort	number	Y	UDP port the accessory is listening on for data.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Key	Type	Required?	Description
type	enum	Y	Type of stream. One of Table 23-45 (page 444).
sampleTime	number	N	Media timestamp at the same moment as "timestamp".
timestamp	number	N	Wall clock timestamp at the same moment as "sampleTime" using synchronized wall time.

Table 23-43 Screen stream descriptors request keys

Key	Type	Required?	Description
latencyMs	number	Y	Desired milliseconds of screen latency.
type	enum	Y	Type of stream. One of Table 23-45 (page 444).
streamConnectionID	number	Y	Unique stream connection ID. Used for deriving encryption key and IV. See MFI-SAP (page 429)

Table 23-44 Screen stream descriptors response keys

Key	Type	Required?	Description
dataPort	number	Y	TCP or UDP port the accessory is listening on for data.
type	enum	Y	Type of stream. One of Table 23-45 (page 444).

Table 23-45 Stream ID enum values

Name	Value	Protocol	Description
Invalid	0	n/a	Reserved for an invalid stream ID.
Main Audio	100	UDP	Low-latency audio input/output. Value is also the RTP payload type.
Alt Audio	101	UDP	Low-latency UI sounds, alerts, etc., output. Value is also the RTP payload type.
Main High Audio	102	UDP	High-latency audio output. Value is also the RTP payload type.
Screen	110	TCP/UDP	H.264 screen output. Value is also the RTP payload type.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-46 Audio type string values

Value	Stream ID	Description
"default"	Main Audio, Alt Audio	Unspecified or unknown audio type.
"alert"	Main Audio	Ringtones, alarms, and other high-priority sounds.
"media"	Main Audio, Main High Audio	Entertainment (e.g.: music playback).
"telephony"	Main Audio	Telephony.
"speechRecognition"	Main Audio	Speech recognition.
"compatibility"	Main Audio, Alt Audio	Special value used only in the Info Message response for specifying audio formats compatible with older versions of iOS. Not a valid audio type for stream descriptors.

Table 23-47 Channels

Name	Transport	Port	QoS	Direction	Description
RTSP Controller control	TCP	5000	BE	device to accessory	iOS control channel
RTSP Accessory control	TCP	dynamic	BE	accessory to device	Accessory control channel
RTP Main Audio Output	UDP	dynamic	VO	device to accessory	Audio data
RTP Alternate Audio Output	UDP	dynamic	VO	device to accessory	Audio data
RTP Main Audio Input	UDP	dynamic	VO	accessory to device	Audio data
Time Sync Client	UDP	dynamic	CTL	accessory to device	Audio time sync requests
Time Sync Server	UDP	dynamic	CTL	device to accessory	Audio time sync responses
RTP Screen	TCP	dynamic	VI	device to accessory	User interface (H.264 frames)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.3.2.4 Feedback Message

The feedback message exchanges timing information and statistics to and from an accessory. The Apple device does an HTTP POST to the `/feedback` URL and the accessory responds with its feedback info. The request and response payloads are binary plists. These plists may be omitted or empty if they don't have feedback to report in that direction.

This message is used for media clock synchronization of the main and alternate audio streams as described in Synchronization (page 447).

Table 23-48 Feedback response keys

Key	Type	Required?	Description
streams	array	N	Streams reporting feedback. Contains dictionaries of Table 23-49 (page 446).

Table 23-49 Feedback stream keys

Key	Type	Required?	Description
sr	number	N (*)	Estimated consumption rate in samples per second of the stream. This is calculated based on sampleTime and timestamp. (*) Required when type is an audio stream.
type	enum	Y	Type of stream. One of Table 23-45 (page 444).
sampleTime	number	N	Media timestamp at the same moment as "timestamp".
timestamp	number	N	Wall clock timestamp at the same moment as "sampleTime" using synchronized wall time.

23.3.2.5 URLs

Table 23-50 (page 446) lists the URLs used with CarPlay. All URLs are accessed via HTTP/RTSP.

Table 23-50 URLs for CarPlay

URL	Method	Description
n/a	OPTIONS	Standard HTTP options message to return the supported methods, etc.
n/a	RECORD	RTSP message to start playback.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

URL	Method	Description
n/a	SETUP	RTSP message to set up a session and streams.
n/a	TEARDOWN	RTSP message to tear down a session.
`/auth-setup`	POST	Performs MFi-SAP authentication; see MFI-SAP (page 429). Request and response are opaque security data.
`/command`	POST	For performing command requests and their responses; see Commands (page 477). Requests and responses are binary plists.
`/feedback`	POST	Exchanges information with the accessory; see Feedback Message (page 446). Requests and responses are binary plists.
`/info`	GET	Exchanges information with the accessory; see Info Message (page 430). Requests and responses are binary plists.
`/perf`	POST	Performs network performance tests. Requests and responses are binary plists.

23.3.2.6 Synchronization

The clocks between the Apple device and the accessory must be synchronized to present media data at the correct time. The synchronization process is managed by the Communication Plug-in in the CarPlay Client and includes:

- Synchronizing wall clocks so both entities have an accurate notion of absolute presentation times (for example, when the user should hear an audio sample). Note that the synchronized clocks are maintained internally and do not affect the system time of either the Apple device or the accessory.
- Mapping media clocks to wall clocks so that a media timestamp from one device can be mapped to a media timestamp on another device. This allows the media production or consumption rate of one device's media to be clocked off the other device.

The accessory must drive all audio streams, regardless of type (input or output) from the same media clock that is used for synchronization with the Apple device.

23.3.2.6.1 Wall Clock Synchronization

Synchronizing wall clocks is done by exchanging Network Time Protocol (NTP) packets. The Apple device acts as the NTP server, providing the reference clock, and the accessory acts as the NTP client, syncing its clock to the reference. There are two parts to this process: rate and offset synchronization. Rate synchronization tracks how fast the reference clock is running relative to the local clock. This is done by measuring the difference between timestamps over an increasingly large interval to minimize measurement noise (such as network

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

jitter). The interval is capped to allow it to adapt to real changes in the reference clock rate (for example, from temperature changes affecting the rate). Offset synchronization tracks the absolute difference from the reference clock. This is measured with each packet exchange and smoothed to minimize measurement noise.

23.3.2.6.2 Media Clock Synchronization

Synchronizing media clocks is done by periodically exchanging mapping relationships between media sample time and wall clock time. When the accessory is providing the clock, it must sample its media clock and synchronized wall clock at the same moment. These samples are used to calculate media clock rate. The synchronized wall clock is used so the calculated rate is based on time relative to the Apple device. This allows the Apple device to adapt its media production rate to match the accessory's consumption rate and avoid the need for sample rate conversion. The media clock and wall clock tuple also allows the Apple device to determine the absolute media clock offset between devices. Media clock timing relationships are exchanged via the Feedback Message (page 446).

The parameters exchanged in the Feedback message are calculated by the Communication Plug-in on behalf of the accessory. To do so, the accessory must provide local and media clocks using a high resolution tick counter to measure time intervals (such as a processor cycle counter). These counters must be stable (less than 50 PPM of variance) and precise (10 MHz or higher) with minimal and consistent overhead (less than 1 microsecond to sample each clock).

23.3.2.7 Keepalive

CarPlay uses a combination of methods to monitor the session status. When either screen stream or audio stream is active, the controller uses feedback messages sent over the control channel as "keepalive" messages. When the screen stream is active, the controller also sends screen-specific keepalive messages via the screen data connection. If the accessory supports low power keepalive mode (as indicated by "keepAliveLowPower" feature flag), the controller switches to UDP-only keepalive messages when there are no active media streams to save power.

23.3.3 Resource Management

23.3.3.1 Resources

In its typical embodiment in an automobile, the CarPlay architecture includes three resources that must be managed:

- **Screen**: The automotive head unit screen.
- **Main Audio**: The full-duplex (input and output) audio stream. There can be only one source (either accessory or Apple device) for main audio. Voice control, Siri, telephony, and media are examples of audio that uses this stream.

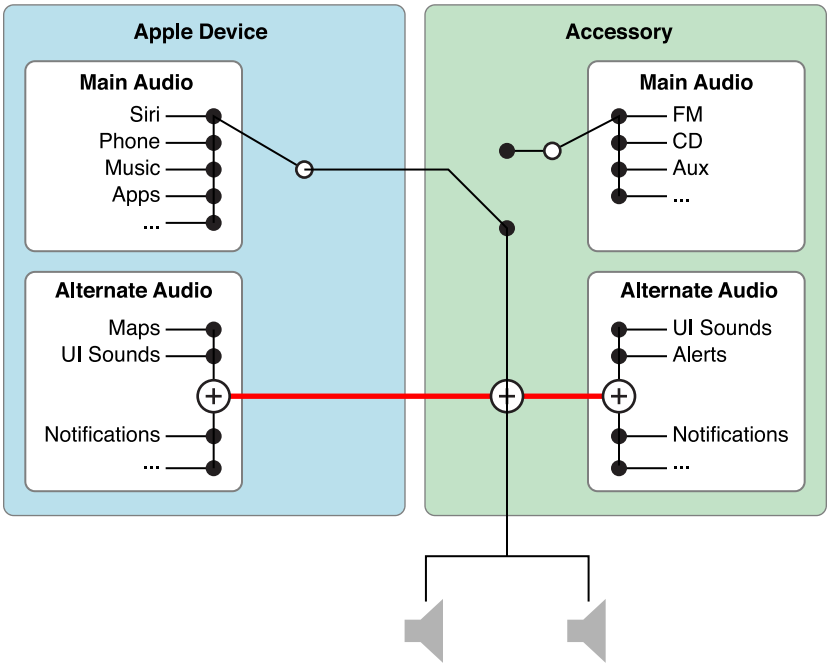
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- **Alternate Audio**: An output-only audio stream. There can be multiple sources of alternate audio, which must be mixed together (possibly with Ducking (page 423)). UI sounds, alerts, and navigation prompts are examples of audio that uses this stream.

Screen and main audio are managed as a shared resource and both the accessory and the device must request an ownership (see Resource Ownership (page 450)), where as the alternate audio stream must be available to the device at all times.

Both accessory and Apple device must distinguish between main and alternate audio, and route audio accordingly. Figure 23-8 (page 449) illustrates conceptually the various audio sources and shows how they must be mixed.

Figure 23-8 Mixing CarPlay audio



23.3.3.2 App States

Three app-specific states must be communicated between the accessory and the Apple device:

- **Speech**: An entity (accessory or Apple device) is performing a speech operation (such as, speaking or recognizing speech). All audio output from the accessory must be disabled while speech is being recorded to prevent interference with recognition algorithms. When an entity is speaking to the user, other entities must avoid playing speech. For example, during this time navigation prompts can be replaced with tones or other indicators.
- **Phone Call**: An entity is on a phone call. Only a single entity must be in this state at a time.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- **Turn-by-turn Navigation**: An entity is performing turn-by-turn navigation. Only a single entity must be in this state at a time.

Note that additional app states may be added in future versions of CarPlay. Also note that the values of the app state properties are purely informative, to allow both the accessory and Apple device to make policy decisions that improve the user experience. For example:

- While one entity is performing speech recognition, all audio from the accessory must be silenced to avoid interfering with the speech recognition algorithms.
- While one entity is speaking to the user, other speech must be suppressed or conveyed as tones rather than speech; multiple, simultaneous speakers can be confusing to the user.
- While one entity is engaged in a phone call, the other entity cannot initiate a phone call (first caller wins).
- If one entity starts performing turn-by-turn navigation, the other entity must stop doing its turn-by-turn navigation, if applicable (last system wins).

The accessory must respect the current values of the app state properties. These properties must not be ignored unless it is essential to give critical safety or emergency information to the user.

23.3.3.3 Resource Ownership

Only one entity (accessory or Apple device) can claim ownership of a resource at any given time. When one entity claims ownership of a resource, it also constrains the circumstances under which ownership can be transferred. The three constraint modifiers are:

- **Anytime**: The other entity may take/borrow the resource at any time. If the screen is showing unimportant information, it would use this value.
- **User Initiated**: The other entity may take or borrow this resource, but doing so would disrupt the current user experience, and must only happen in response to a user-initiated action, and only if acquiring the resource is crucial to the user experience of the new mode. While music is playing, for example, the main audio channel must not be taken or borrowed unless the user has switched to a different audio source.

  A user-initiated action is defined as one where the user taps a user interface element, presses a button, turns a knob, and so forth, in order to instruct an entity to set a new context, change source, and so forth. The accessory must not initiate any 'user-initiated' actions on the user's behalf unless these are required to give critical safety or emergency information to the user.
- **Never**: The other entity may not take or borrow this resource under any circumstances. The owner/borrower of the resource would set this during a phone call or while a safety alert is being shown.

The accessory must set the borrow constraint to Anytime except for UI that requires direct user interaction or is safety related.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Ownership of a resource can be permanently transferred (taken) or temporarily transferred (borrowed).

Take: Ownership of the resource is permanently transferred to the other entity. If the user switches from the accessory's FM radio to the Apple device's Music app, for example, the Apple device would take the main audio channel from the accessory.

Borrow: Ownership is temporarily transferred. When ownership is returned, the owner should resume whatever it was doing with the resource. If the user is listening to the accessory's FM radio and starts a Siri session, for example, the Apple device would borrow the main audio channel. When the Siri session ends, ownership returns to the accessory and the FM radio must continue.

When a resource is borrowed, the other entity still retains ownership and must maintain its state. Maintaining the original ownership state (with take constraints) is important to support the following three possible ways for a borrow state to end:

- The entity borrowing the resource has finished using it and returns it to the owner. This is the typical case. For example, when a phone call ends the main audio channel may return to the accessory, which continues playing FM radio.
- The owner requires the resource and prematurely ends the borrowing state, using the take constraint of the borrow state. For example, the Apple device may borrow the main audio channel to start a Siri session, but the accessory prematurely ends the session because the main audio channel is needed for an incoming call from a phone in the vehicle that does not support CarPlay.
- The entity borrowing the resource wishes to convert from borrowing to ownership, using the original owner's take constraints. For example, the user may ask Siri to switch from FM radio to the Apple device's music player. The Siri session initially borrows the main audio channel, but when the music player starts it will want to own it.

Taking a resource with a constraint of Never is not recommended. Instead, the accessory should borrow the resource with an unborrow constraint of Never and then unborrow the resource when finished with it.

23.3.3.4 Examples of Resource Management

For examples of mode changes, please refer to the *Resource Management* document in the CarPlay DevKit.

23.3.4 Modes

Modes provide a mechanism to manage shared resources (e.g. the main screen), and state (e.g. on a phone call) during a CarPlay session. They are represented by the values in the following tables:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-51 Entity enum values

Name	Value	Description
None	0	Only used for appStates to indicate that no entity is in one of the App States. May not be used to describe ownership for a given resource (Main Screen or Main Audio).
Controller	1	The Apple device owns the given resource (Main Screen or Main Audio) or is in the given app state.
Accessory	2	The accessory owns the given resource (Main Screen or Main Audio) or is in the given app state.

Table 23-52 Resource ID enum values

Name	Value	Description
Main Screen	1	The main display for the system.
Main Audio	2	The main audio stream used for input, output, or full-duplex audio, used by telephony and media. This mode requires exclusive access; either the Apple device or the accessory may use it but not both at once.

Table 23-53 Resource constraint enum values

Name	Value	Description
Anytime	100	Resource may be taken or borrowed at any time.
User Initiated	500	Resource may be taken or borrowed if user requests.
Never	1000	Resource may never be taken or borrowed.

Table 23-54 Resource ownership transfer type enum values

Name	Value	Description
Take	1	Acquire ownership of a resource permanently. For example, if the user switches from the accessory's FM radio to the Apple device's music app, the Apple device would "take" main audio.
Untake	2	Indicate that a resource is no longer needed. Ownership of a resource does not change immediately, it changes the next time the Apple device requests it. The accessory is notified using a "modesChanged" command (see modesChanged (page 480)).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	Value	Description
Borrow	3	Transfer ownership temporarily. For example, if the user is listening to the accessory's FM radio and then starts a Siri session, the Apple device would "borrow" main audio. When the Siri session ends, the Apple device would "unborrow" main audio.
Unborrow	4	Release ownership of a resource that was acquired temporarily.

Table 23-55 Resource transfer priority enum values

Name	Value	Description
Nice to Have	100	Transfer succeeds only if constraint is Anytime. Must only be used with a constraint of Anytime.
User Initiated	500	Transfer succeeds only if constraint is User Initiated or Anytime.

Table 23-56 App state enum values

Name	Value	Description
Speech	1	An entity is performing a speech operation. All audio output must be disabled while speech is being recorded to prevent interference with recognition algorithms. When an entity is speaking to the user, other entities must avoid playing speech. During this time, navigation prompts, etc. can be replaced with tones or other indicators.
PhoneCall	2	An entity is engaged in a phone call. Only a single entity must be in this state at a time.
TurnByTurn	3	An entity is performing turn-by-turn navigation. Only a single entity must be in this state at a time.

Table 23-57 Speech mode enum values

Name	Value	Description
None	-1	No speech-related states are active.
Speaking	1	An entity is speaking to the user.
Recognizing Speech	2	An entity is recording audio to recognize speech from the user.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.3.4.1 Initial Mode

When a session starts (on connection, for example), the Apple device queries the accessory for its current mode to synchronize the states of the two devices. This is done via the Info Message (page 430). The accessory must respond with its current mode, including the type, priority and constraints for each resource, as well as current app states and speech mode. The Apple device interprets this initial state as a resource transfer to the accessory and will only then evaluate whether any mode changes are required and permissible. For example, if music was already playing on the Apple device prior to connection to the accessory, the Apple device would then "take" main audio, provided that this action was compatible with the accessory's initial mode constraints.

An accessory must not take audio with a constraint of Never and should not take the screen with a constraint of Never.

Synchronization of the Initial Mode may be required beyond just the start of a CarPlay session: the accessory must be ready to respond to an Info message requesting an update of its current mode at any time.

23.3.4.2 Mode Changes

To modify the current mode, the accessory sends the Apple device a mode change request communicating the intent of the mode change. This is done via the `/command` URL (See Commands (page 477)). The Apple device takes the request and determine a new mode based on it. The Apple device will reject badly-formed requests or requests that are not compatible with the current resource constraints; in either case an error will be returned to the accessory. Once the Apple device has evaluated a new mode, it communicates it to the accessory (see Current Mode (page 456)). Mode changes originating from the Apple device are communicated directly to the accessory as an update to the current mode; they are not preceded by a mode change request from the Apple device to the accessory.

The accessory must honor any updates to the current mode within 100 ms of receiving them from the Apple device. For example, it must relinquish control of the screen within 100 ms if screen ownership is transferred to the Apple device.

A mode change request consists of the following information:

Main Screen:

- Transfer Type: Take, Untake, Borrow, or Unborrow
- Priority: Nice to Have or User Initiated
- Constraints: (1 or 2 constraints, depending on the Transfer Type)
- For Transfer Type Take, Apple device may Take back Main Screen: Anytime or User Initiated or Never
- For Transfer Type Take, Apple device may Borrow back Main Screen: Anytime or User Initiated or Never
- For Transfer Type Borrow, Apple device may Unborrow Main Screen: Anytime or User Initiated or Never

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Main Audio:

- Transfer Type: Take, Untake, Borrow, or Unborrow
- Priority: Nice to Have or User Initiated
- Constraints: (1 or 2 constraints, depending on the Transfer Type)
- For Transfer Type Take, Apple device may Take back Main Audio: Anytime or User Initiated or Never
- For Transfer Type Take, Apple device may Borrow back Main Audio: Anytime or User Initiated or Never
- For Transfer Type Borrow, Apple device may Unborrow Main Audio: Anytime or User Initiated or Never

App States:

- Speech: Speaking, Speech Recognizing, or Neither
- Phone Call: True or False
- Turn-by-turn navigation: True or False

A change request must include at least one of the foregoing properties. If an action requires changes to multiple properties of a mode, they should be sent in a single transaction.

Not all mode change requests will require changes to all properties, in which case unchanged properties can be left out.

It is important that the accessory always update the Apple device with its current state, regardless of the current mode. For example, if the user switches from FM radio to the CD player, ownership of the main audio will not change. But the accessory should still send a mode change request indicating that it wants to take ownership of main audio.

The accessory must specify whether it wishes to own or borrow a resource, as described in the Resource Ownership (page 450). When the accessory has finished borrowing a resource, it must send a mode change request to release it. Each successful borrow must have a matching unborrow, for example, if an accessory borrows main audio 3 times, it must unborrow 3 times to release it to the previous owner. The release transfer type indicates to the Apple device that the accessory no longer needs the resource, and the resource constraints will be reset to "Other may take anytime". A successful taking or borrowing of a resource currently borrowed by the other entity will terminate the borrow, as described in Resource Ownership (page 450).

The accessory can use a priority value to indicate to the Apple device how badly it wants a resource. A priority of "nice to have" will succeed if the take/borrow constraint is "anytime." A priority of "user-initiated" will succeed if the take/borrow constraint is "user-initiated" or "anytime."

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

In the case of a non-routine safety or emergency situation (for example, a flat tire), the accessory should still send a changeModes request, but it does not have to wait for a modesChanged notification before taking over video or audio. In all other routine cases, the accessory must wait for the notification.

The accessory must be ready to handle setup or record events from the Apple device within 200 ms of the time that the accessory issues the changeModes command (see changeModes (page 479)) or receives a modesChanged command (see modesChanged (page 480)).

23.3.4.3 Current Mode

The current mode that the Apple device communicates to the accessory includes these parameters:

- **Accessory** or **Apple device** has Screen
- **Accessory** or **Apple device** has Main Audio
- **Accessory** or **Apple device** or **Neither** is performing speech recognition or speech
- **Accessory** or **Apple device** or **Neither** is engaged in a phone call
- **Accessory** or **Apple device** or **Neither** is performing turn-by-turn navigation

Note that if ownership of a resource is returned to the accessory when the accessory didn't explicitly ask for it, the accessory should resume whatever it was last doing with that resource, if possible.

23.3.5 User Input

This section covers the delivery of user input events for CarPlay and assumes familiarity with the Device Class for Human Interface Devices (HID) specification, the HID Usage Tables and addendum that are published by the USB Implementors Forum. For further information please visit http://www.usb.org/developers/hidpage/.

See additional requirements in HID Requirements (page 580).

Each HID device dictionary contains a HID descriptor (see USB HID specification), device UUID, and a display UUID directing the HID events to a specific display. Table 23-58 (page 456) lists the possible HID device description keys stored in an HID device dictionary.

Table 23-58 HID device description keys

Key	Type	Required?	Description
displayUUID	string	Y	UUID of a display associated with the HID device or a null UUID if not associated to a display.
hidCountryCode	number	Y	USB-style HID country code.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Key	Type	Required?	Description
hidDescriptor	data	Y	USB-formatted HID descriptor.
hidProductID	number	Y	USB-style HID product ID. Must be non-zero.
hidVendorID	number	Y	USB-style HID vendor ID. Must be non-zero.
name	string	Y	User-friendly name of the HID device.
uuid	string	Y	UUID to uniquely identify the HID device.

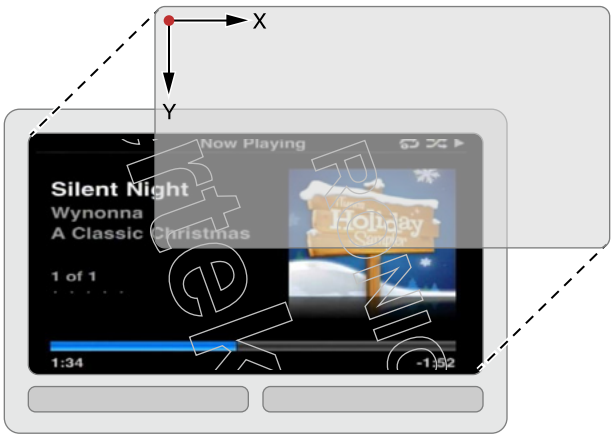
23.3.5.1 Digitizer Support (Touchscreen and Touchpad)

A digitizer is a device that measures absolute spatial position, typically in two or more dimensions. CarPlay supports the following types of digitizers:

23.3.5.1.1 Touchscreen

A touchscreen is a digitizer with an integrated display that allows the use of a finger for direct interaction with a presented interface.

Figure 23-9 Touchscreen



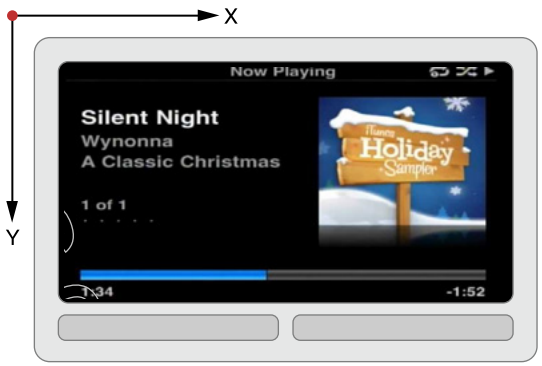
The touch surface must be sampled at the same rate as the refresh rate of the video stream on the integrated display. A sampling and refresh rate of 60 Hz is recommended. See High Resolution Display (page 383) and User Interface (UI) Stream (page 411).

23.3.5.1.2 Touchpad

A touchpad is a digitizer without an integrated display that allows the use of a finger for indirect interaction with a presented interface.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 23-10 Touchpad



Generally, reports originating from this class of device would entailed relative movements synonymous with a mouse. These devices must only convey absolute transducer movement.

Accessories using a touchpad as the primary user interface must support the following:

- Single Touch as a touchpad, see Single Touch (page 459)
- Button 1 (Primary Button) to convey selection
- AC Back button

If the accessory supports character recognition, it must provide character input gesture support (see Character Input Gesture Support (page 461)) using hidSetInputMode (see hidSetInputMode (page 482)).

The accessory must use touchpad as a primary input device only with supported iOS versions.

See hidSetInputMode (page 482) to configure the input mode of a HID device and hidSendReport (page 481) for delivering HID reports.

23.3.5.1.3 HID Descriptor Support

The accessory may support the following HID usages:

Table 23-59 Digitizer Support HID Usages

Page ID	Page Name	Usage ID	Usage Name	Usage Type
0x01	Generic	0x30	X	Dynamic Value
0x01	Generic	0x31	Y	Dynamic Value
0x0D	Digitizer	0x04	Touch Screen	Application Collection

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Page ID	Page Name	Usage ID	Usage Name	Usage Type
0x0D	Digitizer	0x05	Touch Pad	Application Collection
0x0D	Digitizer	0x22	Finger	Logical Collection
0x0D	Digitizer	0x32	In Range	Momentary Control
0x0D	Digitizer	0x33	Touch	Momentary Control
0x0D	Digitizer	0x37	Data Valid	Momentary Control
0x0D	Digitizer	0x38	Transducer Index	Dynamic Value
0x0D	Digitizer	0x42	Tip Switch	Momentary Control
0x0D	Digitizer	0x51	Contact Identifier	Dynamic Value

When the Apple device does not own the screen, the accessory must not send these HID usages.

23.3.5.1.4 Single Touch

Single touch implies the ability to convey the movement of only one finger over a digitizer surface.

In order to ensure proper detection, an accessory must declare an application collection using either one of the following usages:

- Touch Screen
- Touch Pad

The accessory must declare a logical collection using the following usage:

- Finger

Each logical collection must also contain absolute X and Y axes:

- X
- Y

And either one of the following to convey a touch down:

- Touch
- Tip Switch

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

In addition, the accessory must supply the following:

- Physical Minimum
- Physical Maximum
- Unit Exponent
- Unit

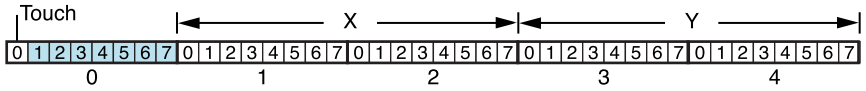
The following is an example report descriptor and format for a single-touch screen:

```
0x05, 0x0D,          // Usage Page (Digitizer)
0x09, 0x04,          // Usage (Touch Screen)
0xA1, 0x01,          // Collection (Application)
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x22,          //   Usage (Finger)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x33,          //     Usage (Touch)
0x15, 0x00,          //     Logical Minimum......... (0)
0x25, 0x01,          //     Logical Maximum......... (1)
0x35, 0x00,          //     Physical Minimum........ (0)
0x46, 0x64, 0x00,    //     Physical Maximum........ (100)
0x55, 0x0F,          //     Unit Exponent (-1)
0x65, 0x11,          //     Unit (cm)
0x75, 0x01,          //     Report Size............. (1)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0x75, 0x07,          //     Report Size............. (7)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x01,          //     Input...................(Constant)
0x05, 0x01,          //     Usage Page (Generic Desktop)
0x09, 0x30,          //     Usage (X)
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0x20, 0x03,    //     Logical Maximum......... (800)
0x75, 0x10,          //     Report Size............. (16)
0x95, 0x01,          //     Report Count............ (1)
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
0x81, 0x02,          //     Input..................(Data, Variable, Absolute)
0x09, 0x31,          //     Usage (Y)
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0xE0, 0x01,    //     Logical Maximum......... (480)
0x75, 0x10,          //     Report Size.............. (16)
0x95, 0x01,          //     Report Count............. (1)
0x81, 0x02,          //     Input..................(Data, Variable, Absolute)
0xC0,                //   End Collection
0xC0,                // End Collection
```

Figure 23-11 Example Input Report Layout for Single-Touch Screen



23.3.5.2 Character Input Gesture Support

We are seeking to add functional enhancements to the HID Digitizer Page (0x0D) to convey the accessory's ability to process character generating gestures. Though there already exists a HID unicode page, it is limited to USC-2 (UTF16-LE). As a result, this prevents accessories from fully supporting certain Asian character sets.

We are proposing the addition of support for the transmitting of character strings with alternate encodings such as UTF8, UTF16 and UTF32.

The accessory may support the following HID usages:

Table 23-60 Character Input Gesture Support HID Usages

Page ID	Page Name	Usage ID	Usage Name	Usage Type
0x0D	Digitizer	0x23	Device Settings	Logical Collection
0x0D	Digitizer	0x24	Character Gesture	Logical Collection
0x0D	Digitizer	0x60	Character Gesture Enable	Dynamic Flag
0x0D	Digitizer	0x61	Character Gesture Quality	Dynamic Value
0x0D	Digitizer	0x62	Character Gesture Data Length	Buffered Bytes
0x0D	Digitizer	0x63	Character Gesture Data	Dynamic Value
0x0D	Digitizer	0x64	Character Gesture Encoding	Named Array

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Page ID	Page Name	Usage ID	Usage Name	Usage Type
0x0D	Digitizer	0x65	UTF8 Character Gesture Encoding	Selector
0x0D	Digitizer	0x66	UTF16 Little Endian Character Gesture Encoding	Selector
0x0D	Digitizer	0x67	UTF16 Big Endian Character Gesture Encoding	Selector
0x0D	Digitizer	0x68	UTF32 Little Endian Character Gesture Encoding	Selector
0x0D	Digitizer	0x69	UTF32 Big Endian Character Gesture Encoding	Selector

When the Apple device does not own the screen, the accessory must not send these HID usages.

23.3.5.2.1 Basic Character Gesture Recognition

Basic character gesture recognition allows an accessory to convey a single character string as a result of interpreting transducer movement on a digitizer surface.

In order for the Apple device to properly detect support for these gestures, the accessory must declare a logical collection with the following usages:

- Character Gesture
- Character Gesture Data Length
- Character Gesture Data

Additionally, the accessory must also include string encoding information. If more than one encoding type is supported, they must be placed in a selector array. Otherwise, the accessory may declare individual encoding support via a static item. Use the following usages:

- Character Gesture Encoding
- UTF8 Character Gesture Encoding
- UTF16 Little Endian Character Gesture Encoding
- UTF16 Big Endian Character Gesture Encoding
- UTF32 Little Endian Character Gesture Encoding
- UTF32 Big Endian Character Gesture Encoding

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The Apple device will notify the accessory of the input mode using the hidSetInputMode as described in hidSetInputMode (page 482).

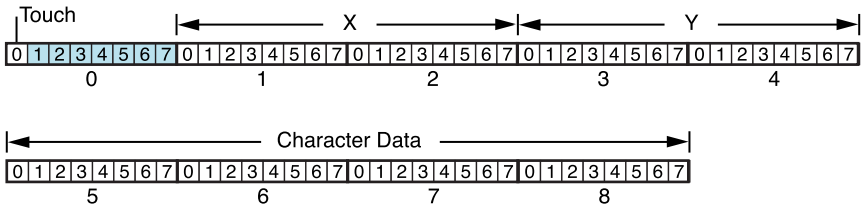
The following is an example report descriptor and format for a single-touch touchpad with basic character gesture recognition:

```
0x05, 0x0D,          // Usage Page (Digitizer)
0x09, 0x05,          // Usage (Touch Pad)
0xA1, 0x01,          // Collection (Application)
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x22,          //   Usage (Finger)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x33,          //     Usage (Touch)
0x15, 0x00,          //     Logical Minimum......... (0)
0x25, 0x01,          //     Logical Maximum......... (1)
0x35, 0x00,          //     Physical Minimum........ (0)
0x46, 0x64, 0x00,    //     Physical Maximum........ (100)
0x55, 0x0F,          //     Unit Exponent (-1)
0x65, 0x11,          //     Unit (cm)
0x75, 0x01,          //     Report Size............. (1)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0x75, 0x07,          //     Report Size............. (7)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x01,          //     Input...................(Constant)
0x05, 0x01,          //     Usage Page (Generic Desktop)
0x09, 0x30,          //     Usage (X)
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0x00, 0x04,    //     Logical Maximum......... (1024)
0x35, 0x00,          //     Physical Minimum........ (0)
0x46, 0x64, 0x00,    //     Physical Maximum........ (100)
0x55, 0x0F,          //     Unit Exponent (-1)
0x65, 0x11,          //     Unit (cm)
0x75, 0x10,          //     Report Size............. (16)
0x95, 0x01,          //     Report Count............ (1)
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0x09, 0x31,          //     Usage (Y)
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0x00, 0x04,    //     Logical Maximum......... (1024)
0x35, 0x00,          //     Physical Minimum........ (0)
0x46, 0x64, 0x00,    //     Physical Maximum........ (100)
0x55, 0x0F,          //     Unit Exponent (-1)
0x65, 0x11,          //     Unit (cm)
0x75, 0x10,          //     Report Size............. (16)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0xC0,                //   End Collection
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x24,          //   Usage (Gesture Character)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x63,          //     Usage (Gesture Character Data)
0x75, 0x20,          //     Report Size............. (32)
0x95, 0x01,          //     Report Count............ (1)
0x82, 0x02, 0x01,    //     Input...................(Data, Variable, Absolute,
Buffered bytes)
0x09, 0x65,          //     Usage (Gesture Character Encoding UTF8)
0x09, 0x62,          //     Usage (Gesture Character Data Length)
0x75, 0x08,          //     Report Size............. (8)
0x95, 0x02,          //     Report Count............ (2)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0xC0,                //   End Collection
0xC0,                // End Collection
```

Figure 23-12 Example Input Report Layout for Single-Touch Touchpad with Basic Character Gesture Recognition



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

When reporting character data, care must taken to ensure proper processing of repeated characters. This must be accomplished by issuing a subsequent report that clears gesture data and data length for a given character collection. Using the report format from the example above, the following sequence illustrates how this is accomplished ignoring current finger touch and location states:

First HID report dispatched containing gesture event for the character 'A':

```
XX XX XX XX XX 41 00 00 00 01 65
```

Second HID report dispatched clearing the gesture:

```
XX XX XX XX XX 00 00 00 00 00 65
```

23.3.5.2.2 Character Gesture Recognition with Alternate Interpretations

There can be situations in which the recognition system generates more than one interpretation of a gesture motion. We are proposing the ability to convey alternate gesture interpretations from a single accessory which will allow the Apple device to select the appropriate string based on its current application context.

Each gesture item follows the requirements detailed in Basic Character Gesture Recognition, but must also include a declaration for Character Gesture Quality in each logical collection. This will give the Apple device additional qualitative information so that it can select the appropriate interpretation.

In addition, if no alternative interpretations are available, the recognition system must inform the Apple device by ensuring that only the first character is populated and all subsequent characters are cleared.

The following is an example report descriptor and format for a single-touch touchpad with character gesture recognition with alternate interpretations:

```
0x05, 0x0D,          // Usage Page (Digitizer)
0x09, 0x05,          // Usage (Touch Pad)
0xA1, 0x01,          // Collection (Application)
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x22,          //   Usage (Finger)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x33,          //     Usage (Touch)
0x15, 0x00,          //     Logical Minimum......... (0)
0x25, 0x01,          //     Logical Maximum......... (1)
0x35, 0x00,          //     Physical Minimum........ (0)
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
0x46, 0x64, 0x00,    //     Physical Maximum........ (100)
0x55, 0x0F,          //     Unit Exponent (-1)
0x65, 0x11,          //     Unit (cm)
0x75, 0x01,          //     Report Size............. (1)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0x75, 0x07,          //     Report Size............. (7)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x01,          //     Input...................(Constant)
0x05, 0x01,          //     Usage Page (Generic Desktop)
0x09, 0x30,          //     Usage (X)
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0x00, 0x04,    //     Logical Maximum......... (1024)
0x35, 0x00,          //     Physical Minimum........ (0)
0x46, 0x64, 0x00,    //     Physical Maximum........ (100)
0x55, 0x0F,          //     Unit Exponent (-1)
0x65, 0x11,          //     Unit (cm)
0x75, 0x10,          //     Report Size............. (16)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0x09, 0x31,          //     Usage (Y)
0x15, 0x00,          //     Logical Minimum......... (0)
0x26, 0x00, 0x04,    //     Logical Maximum......... (1024)
0x35, 0x00,          //     Physical Minimum........ (0)
0x46, 0x64, 0x00,    //     Physical Maximum........ (100)
0x55, 0x0F,          //     Unit Exponent (-1)
0x65, 0x11,          //     Unit (cm)
0x75, 0x10,          //     Report Size............. (16)
0x95, 0x01,          //     Report Count............ (1)
0x81, 0x02,          //     Input...................(Data, Variable, Absolute)
0xC0,                //   End Collection
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x24,          //   Usage (Gesture Character)
0xA1, 0x02,          //   Collection (Logical)
```

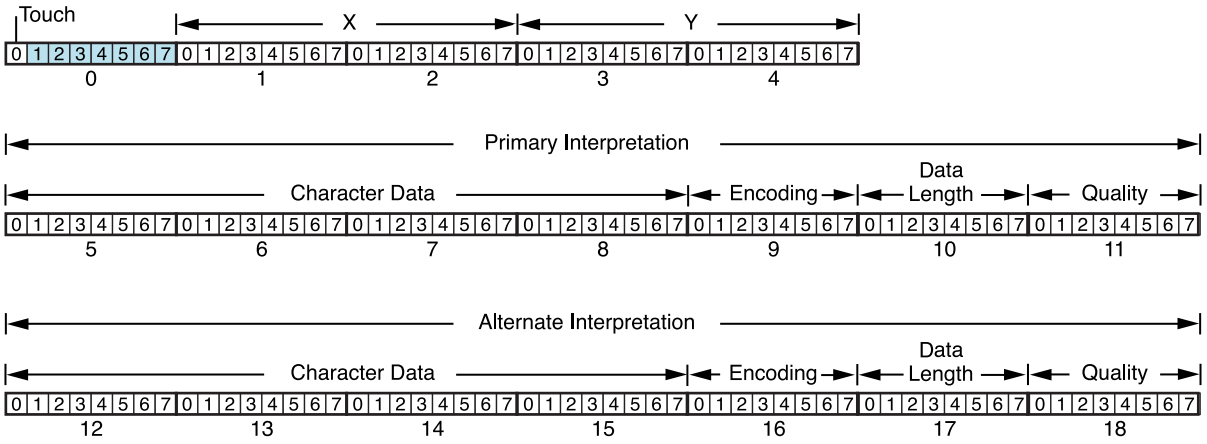
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x63,          //     Usage (Gesture Character Data)
0x75, 0x20,          //     Report Size..... (32)
0x95, 0x01,          //     Report Count..... (1)
0x82, 0x02, 0x01,    //     Input.................(Data, Variable, Absolute,
Buffered bytes)
0x09, 0x65,          //     Usage (Gesture Character Encoding UTF8)
0x09, 0x62,          //     Usage (Gesture Character Data Length)
0x75, 0x08,          //     Report Size..... (8)
0x95, 0x02,          //     Report Count..... (2)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0x09, 0x61,          //     Usage (Gesture Character Quality)
0x15, 0x00,          //     Logical Minimum..... (0)
0x25, 0x64,          //     Logical Maximum..... (100)
0x75, 0x08,          //     Report Size..... (8)
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0xC0,                //   End Collection
0x05, 0x0D,          //   Usage Page (Digitizer)
0x09, 0x24,          //   Usage (Gesture Character)
0xA1, 0x02,          //   Collection (Logical)
0x05, 0x0D,          //     Usage Page (Digitizer)
0x09, 0x63,          //     Usage (Gesture Character Data)
0x75, 0x20,          //     Report Size..... (32)
0x95, 0x01,          //     Report Count..... (1)
0x82, 0x02, 0x01,    //     Input.................(Data, Variable, Absolute,
Buffered bytes)
0x09, 0x65,          //     Usage (Gesture Character Encoding UTF8)
0x09, 0x62,          //     Usage (Gesture Character Data Length)
0x75, 0x08,          //     Report Size..... (8)
0x95, 0x02,          //     Report Count..... (2)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0x09, 0x61,          //     Usage (Gesture Character Quality)
0x15, 0x00,          //     Logical Minimum..... (0)
0x25, 0x64,          //     Logical Maximum..... (100)
0x75, 0x08,          //     Report Size..... (8)
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
0x95, 0x01,          //     Report Count..... (1)
0x81, 0x02,          //     Input.................(Data, Variable, Absolute)
0xC0,                //   End Collection
0xC0,                // End Collection
```

Figure 23-13 Example Input Report Layout for Single-Touch Touchpad with Character Gesture Recognition with Alternate Interpretations



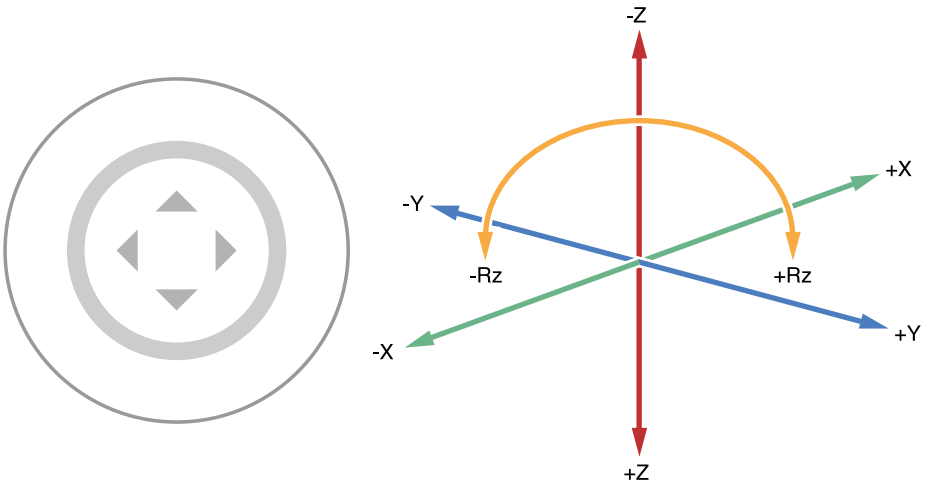
23.3.5.3 Knob Support (Multi-axis Controller)

Knobs refer to user input devices, commonly found on a vehicle center console, which allow for indirect interaction with a media display.

A multi-axis controller is similar to a joystick but minimally consists of three variable axes (X, Y and Z). Additionally, rotational axes for X, Y, and Z may also be included.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 23-14 Multi-axis Controller



The accessory may support the following HID usages:

Table 23-61 Knob Support HID Usages

Page ID	Page Name	Usage ID	Usage Name	Usage Type
0x01	Generic	0x30	X	Dynamic Value
0x01	Generic	0x31	Y	Dynamic Value
0x01	Generic	0x32	Z	Dynamic Value
0x01	Generic	0x35	Rz	Dynamic Value
0x01	Generic	0x37	Dial	Dynamic Value
0x01	Generic	0x38	Wheel	Dynamic Value
0x09	Button	0x01	Button 1 (Primary Button)	(see USB HID)
0x0D	Consumer	0x224	AC Back	One Shot Control

When the Apple device does not own the screen, the accessory must not send these HID usages.

The preferred state of the axes should signify that the knob rests at the center when not in use.

The accessory must support the following to convey translational movement:

- X
- Y

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The accessory must support either one of the following to convey selection:

- Z
- Button 1 (Primary Button)

When implementing the Z axis, the Apple device only requires movement in the +Z (down) direction and will translated it to a primary button.

The accessory must support either one of the following to allow scrolling:

- Rz
- Dial
- Wheel

The accessory must support the AC Back button.

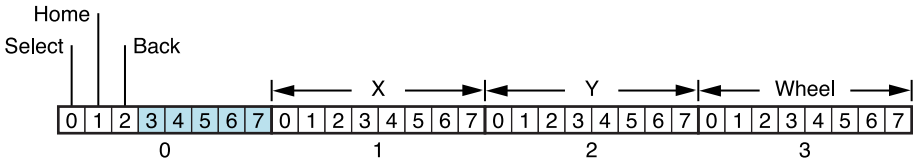
The following is an example report descriptor and format for a multi-axis controller:

```
0x05, 0x01,          // Usage Page (Generic Desktop)
0x09, 0x08,          // Usage (MultiAxisController)
0xA1, 0x01,          // Collection (Application)
0x05, 0x09,          //   Usage Page (Button)
0x09, 0x01,          //   Usage 1 (0x1)
0x15, 0x00,          //   Logical Minimum......... (0)
0x25, 0x01,          //   Logical Maximum......... (1)
0x75, 0x01,          //   Report Size............. (1)
0x95, 0x01,          //   Report Count............ (1)
0x81, 0x02,          //   Input...................(Data, Variable, Absolute)
0x05, 0x0C,          //   Usage Page (Consumer)
0x0A, 0x23, 0x02,    //   Usage 547 (AC Home)
0x0A, 0x24, 0x02,    //   Usage 548 (AC Back)
0x15, 0x00,          //   Logical Minimum......... (0)
0x25, 0x01,          //   Logical Maximum......... (1)
0x75, 0x01,          //   Report Size............. (1)
0x95, 0x02,          //   Report Count............ (2)
0x81, 0x02,          //   Input...................(Data, Variable, Absolute)
0x75, 0x05,          //   Report Size............. (5)
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
0x95, 0x01,          //   Report Count..... (1)
0x81, 0x01,          //   Input.................(Constant)
0x05, 0x01,          //   Usage Page (Generic Desktop)
0x09, 0x30,          //   Usage (X)
0x09, 0x31,          //   Usage (Y)
0x15, 0x81,          //   Logical Minimum..... (-127)
0x25, 0x7F,          //   Logical Maximum..... (127)
0x75, 0x08,          //   Report Size..... (8)
0x95, 0x02,          //   Report Count..... (2)
0x81, 0x02,          //   Input.................(Data, Variable, Absolute)
0x05, 0x01,          //   Usage Page (Generic Desktop)
0x09, 0x38,          //   Usage (Wheel)
0x15, 0x81,          //   Logical Minimum..... (-127)
0x25, 0x7F,          //   Logical Maximum..... (127)
0x75, 0x08,          //   Report Size..... (8)
0x95, 0x01,          //   Report Count..... (1)
0x81, 0x06,          //   Input.................(Data, Variable, Relative)
0xC0,                // End Collection
```

Figure 23-15 Example Input Report Layout for Multi-Axis Controller



23.3.5.4 Buttons

The CarPlay feature supports various buttons commonly found within the center console or steering wheel of a vehicle.

The accessory may support the following HID usages:

Table 23-62 Button Support HID Usages

Page ID	Page Name	Usage ID	Usage Name	Usage Type	Apple Function
0x0C	Consumer	0xB0	Play	On/Off Control	Play

Page ID	Page Name	Usage ID	Usage Name	Usage Type	Apple Function
0x0C	Consumer	0xB1	Pause	On/Off Control	Pause
0x0C	Consumer	0xB5	Scan Next Track	One Shot Control	Scan Next Track
0x0C	Consumer	0xB6	Scan Previous Track	One Shot Control	Scan Previous Track
0x0C	Consumer	0xCD	Play/Pause	One Shot Control	Toggle Play/Pause
0x0C	Consumer	0x223	AC Home	One Shot Control	CarPlay
0x0C	Consumer	0x224	AC Back	One Shot Control	Back
0x07	Keyboard	0x2A	Keyboard Delete (Backspace)	One Shot Control	Clear
0x0B	Telephony	0x20	Hook Switch	On/Off Control	Accept call
0x0B	Telephony	0x21	Flash	Momentary Control	Toggle Accept or Reject/End call
0x0B	Telephony	0x26	Drop	One Shot Control	Reject/End call
0x0B	Telephony	0x2F	Phone Mute	One Shot Control	Mute the microphone
0x0B	Telephony	0xB0	Phone Key 0	Selector	Phone Key 0
0x0B	Telephony	0xB1	Phone Key 1	Selector	Phone Key 1
0x0B	Telephony	0xB2	Phone Key 2	Selector	Phone Key 2
0x0B	Telephony	0xB3	Phone Key 3	Selector	Phone Key 3
0x0B	Telephony	0xB4	Phone Key 4	Selector	Phone Key 4
0x0B	Telephony	0xB5	Phone Key 5	Selector	Phone Key 5
0x0B	Telephony	0xB6	Phone Key 6	Selector	Phone Key 6

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Page ID	Page Name	Usage ID	Usage Name	Usage Type	Apple Function
0x0B	Telephony	0xB7	Phone Key 7	Selector	Phone Key 7
0x0B	Telephony	0xB8	Phone Key 8	Selector	Phone Key 8
0x0B	Telephony	0xB9	Phone Key 9	Selector	Phone Key 9
0x0B	Telephony	0xBA	Phone Key Star	Selector	Phone Key Star
0x0B	Telephony	0xBB	Phone Key Pound	Selector	Phone Key Pound

Accessories with non-touch UIs must implement AC Back.

Accessories must implement one of the following:

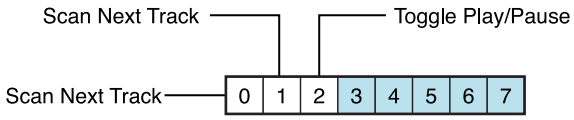
- Flash, if the accessory has a single telephony button.
- Hook Switch & Drop, if the accessory has separate telephony accept and reject buttons.

The following is an example report descriptor and format for a simple media buttons accessory:

```
0x05, 0x0C,          // Usage Page (Consumer)
0x09, 0x01,          // Usage 1 (0x1)
0xA1, 0x01,          // Collection (Application)
0x05, 0x0C,          //   Usage Page (Consumer)
0x09, 0xB5,          //   Usage 181 (Scan Next Track)
0x09, 0xB6,          //   Usage 182 (Scan Previous Track)
0x09, 0xCD,          //   Usage 205 (Toggle Play / Pause)
0x15, 0x00,          //   Logical Minimum......... (0)
0x25, 0x01,          //   Logical Maximum......... (1)
0x75, 0x01,          //   Report Size............. (1)
0x95, 0x03,          //   Report Count............ (3)
0x81, 0x02,          //   Input...................(Data, Variable, Absolute)
0x75, 0x05,          //   Report Size............. (5)
0x95, 0x01,          //   Report Count............ (1)
0x81, 0x01,          //   Input...................(Constant)
0xC0,                // End Collection
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 23-16 Example Input Report Layout for Simple Media Buttons



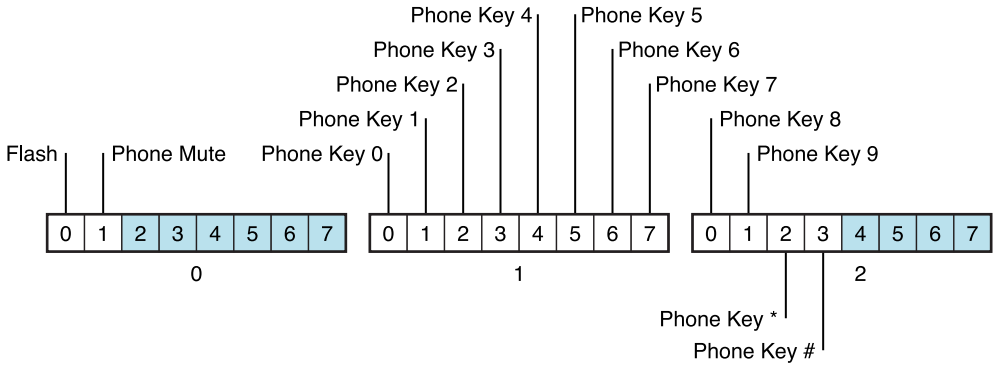
The following is an example report descriptor and format for a simple telephone buttons accessory with flash and numeric keys:

```
0x05, 0x0B,          // Usage Page (Telephony Device)
0x09, 0x01,          // Usage 1 (0x1)
0xA1, 0x01,          // Collection (Application)
0x05, 0x0B,          //   Usage Page (Telephony Device)
0x09, 0x21,          //   Usage 33 (Flash)
0x09, 0x2F,          //   Usage 47 (Phone Mute)
0x15, 0x00,          //   Logical Minimum......... (0)
0x25, 0x01,          //   Logical Maximum......... (1)
0x75, 0x01,          //   Report Size............. (1)
0x95, 0x02,          //   Report Count............ (2)
0x81, 0x02,          //   Input...................(Data, Variable, Absolute)
0x75, 0x06,          //   Report Size............. (6)
0x95, 0x01,          //   Report Count............ (1)
0x81, 0x01,          //   Input...................(Constant)
0x05, 0x0B,          //   Usage Page (Telephony Device)
0x09, 0xB0,          //   Usage 176 (Phone Key 0)
0x09, 0xB1,          //   Usage 177 (Phone Key 1)
0x09, 0xB2,          //   Usage 178 (Phone Key 2)
0x09, 0xB3,          //   Usage 179 (Phone Key 3)
0x09, 0xB4,          //   Usage 180 (Phone Key 4)
0x09, 0xB5,          //   Usage 181 (Phone Key 5)
0x09, 0xB6,          //   Usage 182 (Phone Key 6)
0x09, 0xB7,          //   Usage 183 (Phone Key 7)
0x09, 0xB8,          //   Usage 184 (Phone Key 8)
0x09, 0xB9,          //   Usage 185 (Phone Key 9)
0x09, 0xBA,          //   Usage 186 (Phone Key *)
0x09, 0xBB,          //   Usage 187 (Phone Key #)
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
0x15, 0x00,          //   Logical Minimum......... (0)
0x25, 0x01,          //   Logical Maximum......... (1)
0x75, 0x01,          //   Report Size............. (1)
0x95, 0x0C,          //   Report Count............ (12)
0x81, 0x02,          //   Input...................(Data, Variable, Absolute)
0x75, 0x04,          //   Report Size............. (4)
0x95, 0x01,          //   Report Count............ (1)
0x81, 0x01,          //   Input...................(Constant)
0xC0,                // End Collection
```

Figure 23-17 Example Input Report Layout for Simple Telephone Buttons



23.3.6 Siri User Input

Accessories may support the ability for a user to activate Siri. This is communicated to the Apple device via the requestSiri command (see requestSiri (page 482)), with the parameter being one of the following Siri actions:

Table 23-63 Siri actions enum values

Name	Value	Description
Prewarm	1	Indicate that the Apple device should begin preparing Siri. At this point, no audio or video resources will be taken.
ButtonDown	2	Indicate to the Apple device that the Siri button has been depressed.
ButtonUp	3	Indicate to the Apple device that the Siri button has been released.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Accessories may wish to overload their existing "speech recognition" button, where a short press activates the accessory's native speech recognition system, while a longer press activates Siri. The recommended value for the longer press timeout is 600 ms, but it must not be greater than 1000 ms. It is important that the accessory prewarm Siri as early as possible (for example, when the physical button is pressed) to ensure the best possible response time from Siri in the event that it is activated. After the inital Siri start up sequence, the accessory must forward all physical button presses to the device without any additional delays. Once the Siri session is finished, the accessory must return to the button's default behavior.

Short press (accessory speech recognition):

```
Physical button depressed
|        Physical button released
|        |
+--------+-------------------------------------------------------> time
|
|
Prewarm action sent
```

For short presses, where the physical button is released before the accessory-determined timeout interval, only the Prewarm action is sent to the Apple device, and the accessory's native speech recognition system is activated.

Long press (Siri):

```
Physical button depressed
|                Long Press Timeout       Physical button released
|                |                        |
+----------------+------------------------+-----------------------> time
|                |                        |
|                ButtonDown action sent   ButtonUp action sent
Prewarm action sent
```

For long presses, where the physical button is held down past the accessory-determined timeout interval, the Prewarm and ButtonUp actions are sent to the Apple device, just like in the previous scenario, but when the timeout interval is reached, the accessory sends an additional ButtonDown action when the timeout interval has elapsed.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

For accessories that have a dedicated Siri button, sending a Prewarm action is unnecessary and ButtonDown / ButtonUp actions are sent when the button is physically depressed and released.

23.3.7 Commands

Commands are messages exchanged between the Apple device and the accessory to perform an action. They are sent as HTTP requests using the `/command` URL on the control stream. The Apple device sends its commands over the Controller control channel while the accessory sends its commands over the Accessory control channel. Each command receives an HTTP response. The request and response payloads are binary plists. Each request contains a key, type, to indicate the command to perform. Additional keys may be included for command-specific parameters.

For example, an hidSendReport command would look like this (in XML for readability, the actual request is a binary plist):

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>
    <key>hidReport</key>
    <data>
    ABEiM0Q=
    </data>
    <key>type</key>
    <string>hidSendReport</string>
    <key>uuid</key>
    <string>cbb63f31-c020-4db9-ac16-2283dda2a079</string>
</dict>
</plist>
```

23.3.7.1 duckAudio

Sender: Apple device

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Description: Ramps volume down (for example, to fade down music to play a voice prompt). See Ducking (page 423). Table 23-64 (page 478) lists the specifics of duckAudio request keys.

Table 23-64 duckAudio request keys

Key	Type	Required?	Description
durationMs	number	Y	Number of milliseconds the ramp down should last.
volume	number	Y	Suggested final dB attenuation of the audio at the end of the duck (-144 to 0 dB).

23.3.7.2 unduckAudio

Sender: Apple device

Description: Ramps volume back to the pre-duck level. See Ducking (page 423). Table 23-65 (page 478) lists the specifics of unduckAudio request keys.

Table 23-65 unduckAudio request keys

Key	Type	Required?	Description
durationMs	number	Y	Number of milliseconds the ramp up should last.

23.3.7.3 disableBluetooth

Sender: Apple device

Description: Disable Bluetooth connectivity to specified device (i.e. the Apple device). If the accessory supports Bluetooth, it must provide its bluetoothIDs (see Info Message (page 430)) in order to receive a disableBluetooth command from the device. At a minimum, on receipt of this command, the accessory must immediately disable any existing connections with the specified Apple device for the following Bluetooth profiles:

- Hands-Free Profile
- Advanced Audio Distribution Profile
- A/V Remote Control Profile

In addition, the accessory must not attempt to connect any of the above-listed Bluetooth profiles with the same Apple device while the two are already connected over CarPlay (that is, a session is already in progress). Table 23-66 (page 479) lists the specifics of disableBluetooth request keys.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-66 disableBluetooth request keys

Key	Type	Required?	Description
deviceID	string	Y	MAC address of Bluetooth device to disable connectivity to (that is, the Apple device).

23.3.7.4 changeModes

Sender: Accessory

Description: Change resource states and/or app states. See Modes (page 451). Table 23-67 (page 479) lists the specifics of changeModes request keys and Table 23-70 (page 480) lists the specifics of changeModes response keys.

Table 23-67 changeModes request keys

Key	Type	Required?	Description
appStates	array	N	Application state(s) to change. Contains dictionaries of Table 23-68 (page 479).
resources	array	N	Resource ownership(s) to change. Contains dictionaries of Table 23-69 (page 479).
reasonStr	string	N	A textual description of the reason for the mode change.

Table 23-68 appState keys

Key	Type	Required?	Description
appStateID	enum	Y	One of Table 23-56 (page 453).
speechMode	enum	N (*)	(*) Required if appStateID is Speech or PhoneCall. One of Table 23-57 (page 453).
state	boolean	N (*)	(*) Required if appStateID is PhoneCall or TurnByTurn.

Table 23-69 resource keys

Key	Type	Required?	Description
resourceID	enum	Y	One of Table 23-52 (page 452).
transferType	enum	Y	One of Table 23-54 (page 452).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Key	Type	Required?	Description
transferPriority	enum	N (*)	(*) Required if transferType is take or borrow. One of Table 23-55 (page 453).
takeConstraint	enum	N (*)	(*) Required if transferType is take. One of Table 23-53 (page 452).
borrowConstraint	enum	N (*)	(*) Required if transferType is take. One of Table 23-53 (page 452).
unborrowConstraint	enum	N (*)	(*) Required if transferType is borrow. One of Table 23-53 (page 452).

Table 23-70 changeModes response keys

Key	Type	Required?	Description
params	group	N	Updated modes if the mode change was successful, otherwise absent. The modes dictionary uses the same keys as the "changeModes" command.
status	number	Y	Result of performing the mode change. If the value is nonzero, the change failed.

23.3.7.5 modesChanged

Sender: Apple device

Description: Updates the accessory's current mode after a mode change by the Apple device. Table 23-71 (page 480) lists the specifics of modesChanged request keys.

Table 23-71 modesChanged request keys

Key	Type	Required?	Description
appStates	array	N	Application state(s) to change. Contains dictionaries of Table 23-72 (page 481).
resources	array	N	Resource ownership(s) to change. Contains dictionaries of Table 23-73 (page 481).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-72 appState keys

Key	Type	Required?	Description
appStateID	enum	Y	One of Table 23-56 (page 453).
entity	enum	Y	One of Table 23-51 (page 452).
speechMode	enum	N (*)	(*) Required if appStateID is Speech or PhoneCall. One of Table 23-57 (page 453).

Table 23-73 resource keys

Key	Type	Required?	Description
resourceID	enum	Y	One of Table 23-52 (page 452).
entity	enum	Y	Value must not be 'None'. One of Table 23-51 (page 452).

23.3.7.6 forceKeyFrame

Sender: Accessory

Description: The accessory may force a key frame to be sent for the screen stream. This should only be used to recover from a decoder problem and must not be sent periodically during normal operation.

23.3.7.7 hidSendReport

Sender: Accessory

Description: When an HID event occurs on the accessory, such as the user turning a knob, it sends it to the Apple device over the control stream by sending a command with a type of hidSendReport to the `/command` URL. The command contains the USB HID report and UUID of the HID device. Table 23-74 (page 481) lists the HID event keys.

Table 23-74 hidSendReport request keys

Key	Type	Required?	Description
hidReport	data	Y	USB-formatted HID report.
timestamp	number	N	NTP timestamp when the event occurred (synchronized to the Apple device's clock).
uuid	string	Y	UUID to uniquely identify the HID device.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

23.3.7.8 hidSetInputMode

Sender: Apple device

Description: Sets input mode on a HID.

Table 23-75 hidSetInputMode request keys

Key	Type	Required?	Description
hidInputMode	enum	Y	One of Table 23-76 (page 482).
uuid	string	Y	UUID to uniquely identify the HID device.

Table 23-76 HID input modes enum values

Name	Value	Description
Default	0	Default mode for non-character input (e.g. panning). See Single Touch (page 459).
Character	1	Optimize for entering characters. See Single Touch (page 459) and Character Input Gesture Support (page 461).
Scrolling	2	Optimize for non-character input (e.g. panning). See Single Touch (page 459).
ScrollingWithCharacters	3	Optimize for non-character input (e.g. panning) and entering characters. See Single Touch (page 459) and Character Input Gesture Support (page 461).
DialPad	4	Optimize for non-character input (e.g. panning) and entering dial-pad character events only (i.e., 0-9, #, *). See Single Touch (page 459) and Character Input Gesture Support (page 461), using numerical characters only.

23.3.7.9 requestSiri

Sender: Accessory

Description: Requests that Siri be invoked with a specified action.

The requestSiri command must only be sent as a result of direct user action on a physical or virtual control surface. The behavior should match that of hidSendReport (page 481), see HID Requirements (page 580).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-77 requestSiri request keys

Key	Type	Required?	Description
siriAction	enum	Y	One of Table 23-63 (page 475).

23.3.7.10 requestUI

Sender: Either Apple device or accessory

Description: For the Apple device to ask for the accessory UI to be shown, or for the accessory to ask for the CarPlay UI to be shown. The accessory must not send requestUI upon connection; the Apple device will send requestUI, if appropriate.

The requestUI command must only be sent as a result of direct user action on a physical or virtual control surface. The accessory must not send requestUI upon connection; the Apple device will request resources based on the modes provided in the Info message. See HID Requirements (page 580).

Where the sender is the accessory, a URL request key (see Table 23-78 (page 483)) is available to specify the desired CarPlay compatible app to be shown.

> **Note:** Only URLs that do not require user interaction on the Apple device are allowed.

Table 23-78 Accessory to Apple device requestUI request keys

Key	Type	Required?	Description
url	string	N	URL identifier of the desired CarPlay UI application to launch: • no url - the CarPlay screen will be shown. • "maps:" - the CarPlay Maps application will be shown. • "mobilephone:" - the CarPlay Phone application will be shown. • "music:" - the CarPlay Music application will be shown. • "nowplaying:" - the Now Playing screen will be shown. • "tel:xxx-xxx-xxxx" - the CarPlay Phone application will be shown and a phone call with the desired number will be placed (for support phone number formats see *IETF RFC 3966* .

Where the sender is the Apple device, a URL request key (see Table 23-79 (page 484)) is available to specify the desired accessory UI to be shown. This is only sent when the accessory supports the `enhancedRequestCarUI extendedFeatures`. See Table 23-33 (page 439).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-79 Apple device to accessory requestUI request keys

Key	Type	Required?	Description
url	string	N	URL identifier of the desired accessory UI to show: • no url - show an accessory screen which allows for re-entering the CarPlay UI. • "oem:back" - show the last accessory screen that was visible before showing the CarPlay UI.

23.3.7.11 setNightMode

Sender: Accessory

Description: The accessory may indicate whether it is dark outside.

Table 23-80 setNightMode request keys

Key	Type	Required?	Description
nightMode	boolean	Y	True if it is dark outside, false otherwise.

23.3.7.12 setLimitedUI

Sender: Accessory

Description: The accessory may indicate whether or not to limit certain UI elements. See Table 23-34 (page 439).

Table 23-81 setLimitedUI request keys

Key	Type	Required?	Description
limitedUI	boolean	Y	True if certain UI elements should be limited.

23.3.7.13 iAPSendMessage

Sender: Either Apple device or accessory

Description: Sends an iAP protocol message. This command must only be used for CarPlay over wireless sessions.

Copyright © 2015 Apple Inc. All Rights Reserved.

Table 23-82 iAPSendMessage request keys

Key	Type	Required?	Description
data	data	Y	The content of the iAP message.

23.3.8 CarPlay Communication Plug-in

The CarPlay Communication Plug-in, source code available to accessory developers with approved CarPlay product plans, implements the core functionality described in the preceding sections of CarPlay Communication Protocol (page 424). It interfaces with Bonjour to configure and advertise the CarPlay service and establishes the communication and data links between the accessory and Apple device.

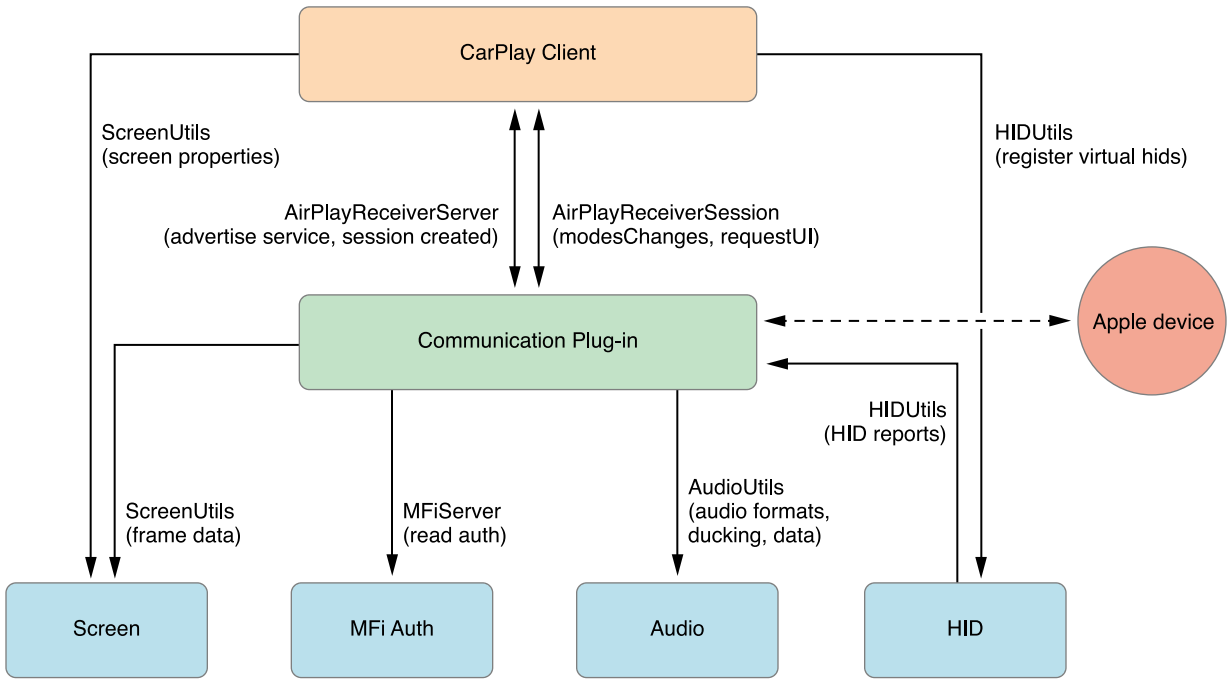
The accessory maker is not required to implement their own TCP and UDP channels for either setup and control or audio and UI transfer. Instead, these are automatically created by the Communication Plug-in which also manages authentication and encryption-decryption of all CarPlay channels.

For simplicity, the core info, setup and feedback messages are abstracted into a platform API, as they are any commands and user event. Figure 23-18 (page 486) shows an architectural overview of the CarPlay Communication Plug-in. To integrate the plug-in on a specific platform, a set of utility functions has been provided:

- **MFiServer** provides integration with platform specific installation of an MFi authentication chip, e.g. over I2C bus. The plug-in will automatically read out authentication data for encrypting the audio and video streams.
- **ScreenUtils** provides integration with the platform specific video decoding interfaces. During setup, a CarPlay client application can configure the specific screen properties. The plug-in will then directly send video frames through the provided custom APIs.
- **AudioUtils** provides integration with the platform specific audio interfaces. On each audio stream setup the plug-in will provide information about the audio formats and data, as well as forward requests to duck the current audio playback.
- **HIDUtils** provides integration with the platform specific HID devices (touchscreens, knob-based controls, etc.). A CarPlay client application can first register the available HID devices and then post HID reports to the plug-in.
- **AirPlayReceiverServer** / **AirPlayReceiverServerDelegate** - A CarPlay client application can use the delegate to send and receive server-level commands to the device, as well as obtain a reference to an active AirPlayReceiverServerSession.
- **AirPlayReceiverServerSession** / **AirPlayReceiverServerSessionDelegate** - A CarPlay client application can use the delegate to send and receive session-level commands to the device. Typical examples will be notifications for modeChanged, or commands as requestUI.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 23-18 CarPlay Communication Plug-in Reference Architecture



See the technical notes provided with the CarPlay Communication Plug-in for more details on this platform abstraction and use cases.

23.4 Test Procedures

Test procedures for self certification are available for accessory manufacturers with approved product plans.

24. Cases

Accessories that substantially enclose Apple devices must comply with the requirements stated in this chapter unless the accessory supports other features in this specification whose requirements conflict with the requirements in this chapter. For example, the requirements listed in Apple Lightning Connector (page 128) may override some requirements listed in this chapter.

If the accessory has multiple user-detachable components that substantially enclose the Apple device, the requirements and/or overrides must be applied to each component separately.

24.1 Product Design

A well-designed case will securely house an Apple device while not interfering with the device's operation. Significant factors in mechanical design include access to the device's sensors, controls, and connectors. The case developer should take into account the minimum and maximum dimensional tolerances of the Apple device(s) it claims compatibility with.

24.1.1 Device Layouts and Dimensions for Apple Devices

Cases must be designed to accommodate the full range of Apple device sizes within each product's dimensional variation. Dimensional drawings with tolerances for all Apple devices can be found in Device Dimensional Drawings (page 880).

24.1.2 Access to Controls

The case must readily permit the user to access and operate the device's mechanical controls such as, but not limited to:

- volume
- ring/silent controls
- sleep/wake control
- home button

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.1.3 Access to the Headset Jack and 30-pin or Lightning Connector

The case must provide ready access to an Apple device's headset jack. The headset jack opening must be at least 6.0 mm in diameter and at most 14.0 mm deep. At least 6.5 mm in diameter and at most 10.0 mm deep is recommended for best compatibility with a range of headsets.

The case must also provide unobstructed access to either the 30-pin connector or the Lightning connector.

If the case is for an Apple device with the Lightning connector, the opening must be at least 12.05 mm by 6.30 mm with full radii rounded edges. 13.65 mm by 6.85 mm is recommended for best compatibility with a range of cables and docks.

In addition, the headset jack and 30-pin or Lightning connector openings must be designed with enough margin to compensate for shifting or dimensional changes of the case material.

24.1.4 Access to the Smart Connector

Cases that do not make use of the Smart Connector must not expose it.

24.1.5 Device Protection

Cases must protect the Apple device from a 1 m drop onto a hard paved surface in any device orientation.

Specifically, exposed glass on the Apple device must not come within 1 mm of a flat surface, such as a table or floor, in any orientation when the case is attached. This may be achieved by either covering the exposed glass or creating features around it that will space the exposed glass at least 1 mm away from the flat surface.

24.1.6 Cover Glass Contact

Cases that claim compatibility with devices below should not contact the cover glass as defined in their dimensional drawings.

- iPhone 6s Plus
- iPhone 6s
- iPhone 6 Plus
- iPhone 6

See Device Dimensional Drawings (page 880).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.1.7 Dock Compatibility

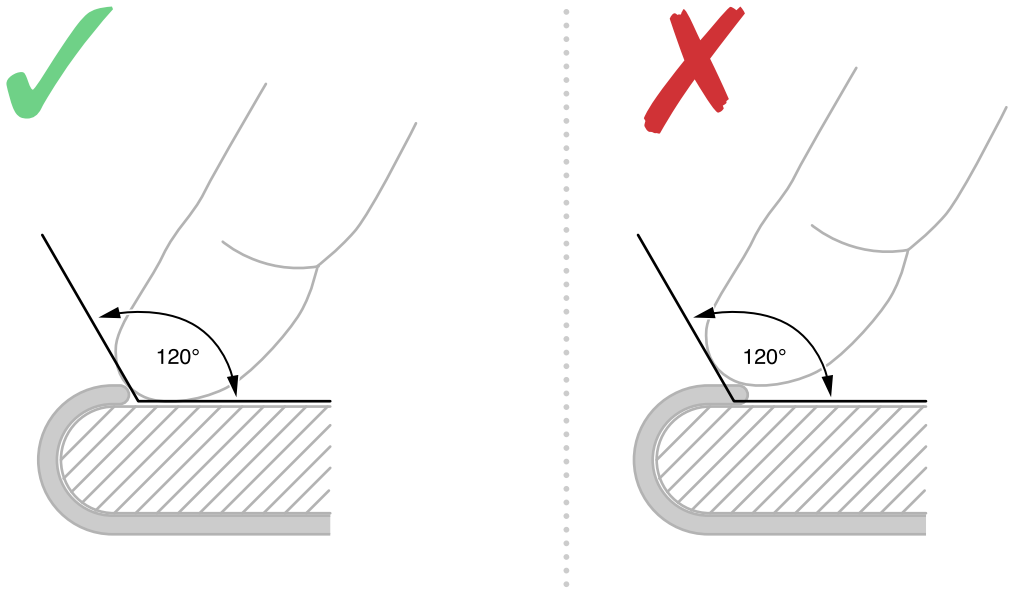
For compatibility with docks, the distance from bottom of the Apple device to the outside of a case should not exceed 1.8 mm.

24.1.8 Touchscreen

The case should permit water droplets to freely roll off the screen when the Apple device is held at a 30 degree angle relative to the horizon.

Cases must allow a 120° opening along the edges of the active area of the touchscreen to ensure compatibility with the Apple device touchscreen features. See Figure 24-1 (page 489) for more information on the keep-out and Device Dimensional Drawings (page 880) device specific active display areas.

Figure 24-1 Touchscreen Keep-out Area



24.2 Acoustics

The case must not impair or degrade the acoustical performance of an Apple device.

24.2.1 Speaker and Microphone Openings

When Apple devices have speakers or microphones, their locations may vary from model to model. Refer to the dimensional drawings for various Apple devices cited in Device Layouts and Dimensions for Apple Devices (page 487). The case must not obstruct the speaker or microphone ports.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.2.2 Speaker to Microphone Coupling

The case must not facilitate the conduction of sound from the speaker to any microphone. Such sound conduction may cause echoing in phone calls.

24.2.3 Call Quality

The case must not impair or degrade the user's experience making and receiving both audio calls over a cellular network or audio/video calls using Apple's FaceTime software. User testing must be conducted in all modes of operation (handset, speakerphone, and headset modes) supported by the Apple device, to confirm that the case does not change the loudness or frequency response of the speakers or microphones. In addition, the user must not be able to detect any sound distortion resulting from enclosing the Apple device in the case.

24.3 Sensors

Cases must be designed so they do not interfere with the operation of the sensors on an Apple device, such as:

- ambient light sensor
- magnetic compass
- proximity sensor
- accelerometer
- three-axis gyroscope

24.3.1 Ambient Light and Proximity Sensor Interference

The ambient light sensor and proximity sensor locations for various Apple devices are shown in the dimensional drawings cited in Device Dimensional Drawings (page 880). Some of the dimensional drawings specify a recommended keep-out area around these sensors. No material must be allowed to cover these sensors or their keep-out areas, this includes films and privacy screens. Cases that allow the Apple device to slide around must not obstruct any sensors.

24.3.2 Magnetic Interference

Apple recommends avoiding the use of magnets and metal components in cases.

Cases for Apple devices must not affect the device's built-in magnetic compass (if present).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Additionally, iPhone 6s Plus and iPhone 6 Plus have an autofocus rear camera equipped with optical image stabilization that may be affected by magnets and metal components in cases and rear camera accessories. Cases and rear camera accessories that claim compatibility with iPhone 6s Plus and iPhone 6 Plus must not affect operation of the autofocus rear camera.

24.3.3 Touch ID Sensor

Cases for Apple devices that are equipped with Touch ID must not inhibit the use of the device's Touch ID sensor. See Device Dimensional Drawings (page 880) for specific device keep-outs.

24.4 Camera

The field of view (FOV) of the camera and the illumination provided by the flash is designed for each Apple product. It is imperative that manufacturers consult technical specifications released for each product and do not assume these parameters are shared between products.

Images from the camera may be affected by the geometry, color and surface finish of the camera opening in the case.

24.4.1 Geometry

The camera lens FOV must not be blocked. Making the case opening around the camera too small may block the FOV of the lens. This may cause vignetting in the image, where the corners of are darker than the center. Blocking marginal rays just outside the FOV of the lens may also reduce the sharpness of the image. See Device Layouts and Dimensions for Apple Devices (page 487) for the detailed mechanical keep-out.

The case opening must not be designed in a way that directs stray light into the camera. If the opening is too narrow or too steep, it may reflect light into the camera, washing out the image or adding an unwanted color cast. Adding a chamfer to the case opening trim may help to direct stray light away from the camera. Additionally, where the product is equipped with a flash, a narrow or steep opening may reflect light from the case opening back into the camera or scene. This may cause the image to appear washed out or contain unwanted artifacts. Designers should ensure that the mechanical keep-outs outlined in the device dimensional drawings (Device Dimensional Drawings (page 880)) are maintained with worst-case X-Y placement tolerances.

24.4.2 Color

Any light reflected from the case may pick up the color of the case. Black material or black coating may help avoid color bleeding into the camera from an external light source or the flash. The darker the color, the less light may be reflected from the source into the camera.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: Apple recommends a semi-gloss black material or coating around the camera and flash opening.

24.4.3 Surface Finish

The flash is a strong source of light and reflections from the camera case opening edge should be managed so they do not reflect back into the camera or the scene. Semi-gloss material may direct light away from the camera. Matte or diffuse materials scatter light in all directions and will increase the likelihood that light from the flash or strong sources in the scene is reflected into the camera.

24.4.4 Image Degradation Examples

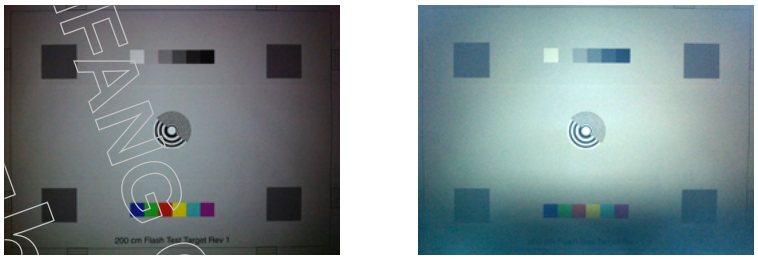
Figure 24-2 Image degradation by color shifting



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.4.4.1 Contrast Decrease

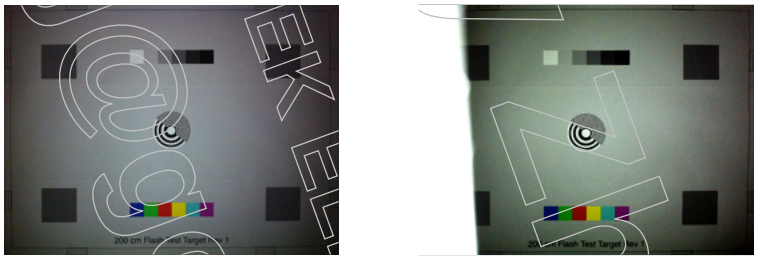
Figure 24-3 Image degradation by decrease of contrast



Reference **Degraded**

24.4.4.2 Image Blocking

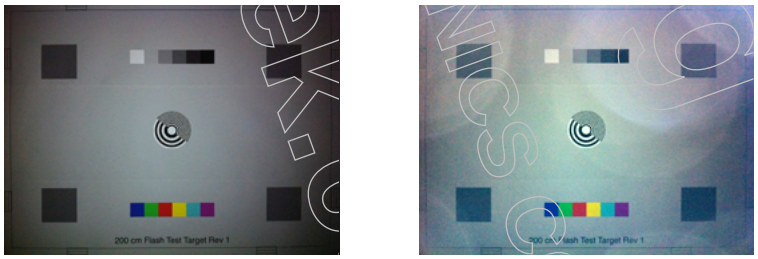
Figure 24-4 Image degradation by blocking



Reference **Degraded**

24.4.4.3 Flash Interference

Figure 24-5 Image degradation by flash interference



Reference **Degraded**

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.5 Reliability

Cases for Apple devices must be tested to verify that they will withstand long-term use under typical use conditions, and that they do not impair or degrade the functionality of the device, damage it or its immediate surroundings, or adversely affect the user.

24.5.1 Device Insertion and Removal

The case must hold the Apple device securely while permitting its easy insertion and removal. The case and the enclosed device must not be damaged by the repeated insertion and removal of the device from the case under conditions representative of long-term use in a variety of environments.

24.5.2 Colorfastness

Any dyes, inks, or coatings in or on the case must not bleed color onto either the device or its user, particularly while the case is in contact with common substances such as water or sunscreen.

24.6 Environmental

Cases for Apple devices must comply with applicable environmental regulations in the regions in which such cases are to be sold, and any applicable substance or material restrictions, including applicable restrictions on the following substances:

- organic tin compounds, PFOS, PFOA, phthalates, azo dyes, polybrominated biphenyls (PBBs) and PAHs, per requirements of the EU REACh regulation EC 1907/2006
- Nickel leach rate on surfaces in prolonged skin contact, per requirements of the EU REACh regulation EC 1907/2006
- Cadmium, lead, hexavalent chromium, and nickel, per requirements of EU Directive 2009/48/EC
- Natural rubber latex, per requirements of EU Directive EC 93/42/EEC
- Dimethylfumarate (DMFu), per requirements of EU Regulation 412/2012
- pH and Formaldehyde, per requirements of China GB 18401 for textiles and China GB 20400 for leather
- Endangered species of flora and fauna in products or packaging (US Lacey Act) Polybrominated diphenyl ethers (PBDE)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.7 RF

24.7.1 Materials and Coatings

Cases for Apple devices must not contain materials or coatings that absorb radio frequency energy. Such materials may impair or degrade the performance of cellular communication antennas or GPS, Wi-Fi, or Bluetooth antennas. Examples include (but are not limited to) the following:

- Metals (e.g. steel, aluminum, magnesium, titanium, etc.)
- Plastics with any carbon content
- Plastics with any glass content
- Plastics with metallic plating
- Metallic paints
- Black paints with high carbon loading
- White paints with high titanium dioxide loading Metallic
- Physical Vapor Deposition (PVD) coatings

24.7.2 Near Field Communication (NFC)

Cases that claim compatibility with NFC enabled Apple devices must not degrade device NFC transaction performance. The following Apple devices are NFC enabled:

- iPhone 6s Plus
- iPhone 6s
- iPhone 6 Plus
- iPhone 6

24.8 Touchscreen

The touch interface in an Apple device senses the presence of one or more fingers on its surface. Any material between the surface and the user's hand, even a very thin sheet of plastic, may affect the performance of the touch interface.

24.8.1 Overlay

If a case design requires the Apple device's touchscreen to be covered with an overlay, the overlay must not:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- exceed 0.3 mm in thickness
- introduce air gaps between the touchscreen and overlay
- be electrically conductive

24.8.2 Edge Swipe Gestures

A case must allow the user to easily use edge swipe gestures. These gestures include bringing up Control Center, Notification Center, and swiping back from apps that may use edge swipe gestures (such as the Messages app).

24.8.3 Edge Press Gestures

Cases that claim compatibility with iPhone 6s Plus or iPhone 6s must allow the user to easily use the edge press gesture. This gesture is used to bring up the task switcher in iOS 9.0 and later.

24.9 Test Procedures

24.9.1 Apple Device Compatibility

Case testing procedures vary depending on the Apple device they enclose. Cases must be tested for every Apple device that they claim compatibility with.

24.9.1.1 iPhone 6s Plus/iPhone 6 Plus

The following tests must be run for cases that claim compatibility with iPhone 6s Plus/iPhone 6 Plus:

- Product Design (page 501) using an iPhone 6s Plus and iPhone 6 Plus.
- RF (OTA) (page 504) using an iPhone 6s Plus for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

It is not possible for a case to claim compatibility with only the iPhone 6s Plus or only the iPhone 6 Plus.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.9.1.2 iPhone 6s/iPhone 6

The following tests must be run for cases that claim compatibility with iPhone 6s/iPhone 6:

- Product Design (page 501) using an iPhone 6s and iPhone 6.
- RF (OTA) (page 504) using an iPhone 6s for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

It is not possible for a case to claim compatibility with only the iPhone 6s or only the iPhone 6.

24.9.1.3 iPhone 5/iPhone 5s

The following tests must be run for cases that claim compatibility with iPhone 5/iPhone 5s:

- Product Design (page 501) using an iPhone 5s.
- RF (OTA) (page 504) using an iPhone 5s for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

It is not possible for a case to claim compatibility with only the iPhone 5 or only the iPhone 5s.

24.9.1.4 iPhone 5c

The following tests must be run for cases that claim compatibility with iPhone 5c:

- Product Design (page 501) using an iPhone 5c.
- RF (OTA) (page 504) using an iPhone 5c for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4 GHz Wi-Fi 1, 7, 13

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

24.9.1.5 iPad Pro

The following tests must be run for cases that claim compatibility with iPad Pro:

- Product Design (page 501) using an iPad Pro.
- RF (OTA) (page 504) using an iPad Pro for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

24.9.1.6 iPad mini 4

The following tests must be run for cases that claim compatibility with iPad mini 4:

- Product Design (page 501) using an iPad mini 4.
- RF (OTA) (page 504) using an iPad mini 4 for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

24.9.1.7 iPad mini/iPad mini 2/iPad mini 3

The following tests must be run for cases that claim compatibility with iPad mini/iPad mini 2/iPad mini 3:

- Product Design (page 501) using an iPad mini 3.
- RF (OTA) (page 504) using an iPad mini 3 for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

It is not possible for a case to claim compatibility with only the iPad mini or only the iPad mini 2 or only the iPad mini 3.

24.9.1.8 iPad Air 2

The following tests must be run for cases that claim compatibility with iPad Air 2:

- Product Design (page 501) using an iPad Air 2.
- RF (OTA) (page 504) using an iPad Air 2 for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

24.9.1.9 iPad Air

The following tests must be run for cases that claim compatibility with iPad Air:

- Product Design (page 501) using an iPad Air.
- RF (OTA) (page 504) using an iPad Air for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - FDD-LTE 7, 17
  - TD-LTE 40
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

24.9.1.10 iPad (4th generation)

The following tests must be run for cases that claim compatibility with iPad (4th generation):

- Product Design (page 501) using an iPad (4th generation).
- RF (OTA) (page 504) using an iPad (4th generation) for the following bands/channels:
  - UMTS I, II, IV, V, VIII
  - LTE 17
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.9.1.11 iPod touch (5th generation)/iPod touch (6th generation)

The following tests must be run for cases that claim compatibility with iPod touch (5th generation)/iPod touch (6th generation):

- Product Design (page 501) using an iPod touch (6th generation).
- RF (OTA) (page 504) using an iPod touch (6th generation) for the following bands/channels:
  - 2.4 GHz Wi-Fi 1, 7, 13
  - 5.0 GHz Wi-Fi 36, 60, 100, 120, 165

It is not possible for a case to claim compatibility with only the iPod touch (5th generation) or only the iPod touch (6th generation).

24.9.2 Apple Device Configurations

Specific Apple device configurations must be used depending on the test being performed and the device being tested against.

Table 24-1 Required device configurations for accessory case test procedures

Test	Device	Model	Color
Product Design (page 501)	iPhone 6s Plus	Any	Any
Product Design (page 501)	iPhone 6s	Any	Any
Product Design (page 501)	iPhone 6 Plus	Any	Any
Product Design (page 501)	iPhone 6	Any	Any
Product Design (page 501)	iPhone 5s	Any	Any
Product Design (page 501)	iPhone 5c	Any	Any
Product Design (page 501)	iPad Pro	Any	Any
Product Design (page 501)	iPad mini 4	Any	Any
Product Design (page 501)	iPad Air 2	Any	Any
Product Design (page 501)	iPad mini 3	Any	Any
Product Design (page 501)	iPad Air	Any	Any
Product Design (page 501)	iPad (4th generation)	Any	Any

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Test	Device	Model	Color
Product Design (page 501)	iPod touch (6th generation)	Any	Any
RF (OTA) (page 504)	iPhone 6s Plus	Unlocked A1634 or Unlocked A1687	Any
RF (OTA) (page 504)	iPhone 6s	Unlocked A1633 or Unlocked A1688	Any
RF (OTA) (page 504)	iPhone 6 Plus	Unlocked A1524	Any
RF (OTA) (page 504)	iPhone 6	Unlocked A1586	Any
RF (OTA) (page 504)	iPhone 5s	Unlocked A1453 and Unlocked A1530	Any
RF (OTA) (page 504)	iPhone 5c	Unlocked A1456 and Unlocked A1529	Any
RF (OTA) (page 504)	iPad Pro	Unlocked A1652	Any
RF (OTA) (page 504)	iPad mini 4	Unlocked A1550	Any
RF (OTA) (page 504)	iPad Air 2	Unlocked A1567	Any
RF (OTA) (page 504)	iPad mini 3	Unlocked A1600 and Unlocked A1601	Any
RF (OTA) (page 504)	iPad Air	Unlocked A1475 and Unlocked A1476	Any
RF (OTA) (page 504)	iPad (4th generation)	Unlocked A1459	Any
RF (OTA) (page 504)	iPod touch (6th generation)	Any	Any

24.9.3 Product Design

24.9.3.1 Equipment

- Apple device. See Table 24-1 (page 500).
- Apple Lightning Digital AV Adapter
- Apple 30-pin to USB Cable

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Vernier calipers
- 1.00 mm plastic feeler gauge
- Apple EarPods with Remote and Mic
- Touchscreen test block (3D CAD available for download at https://mfi.apple.com/)

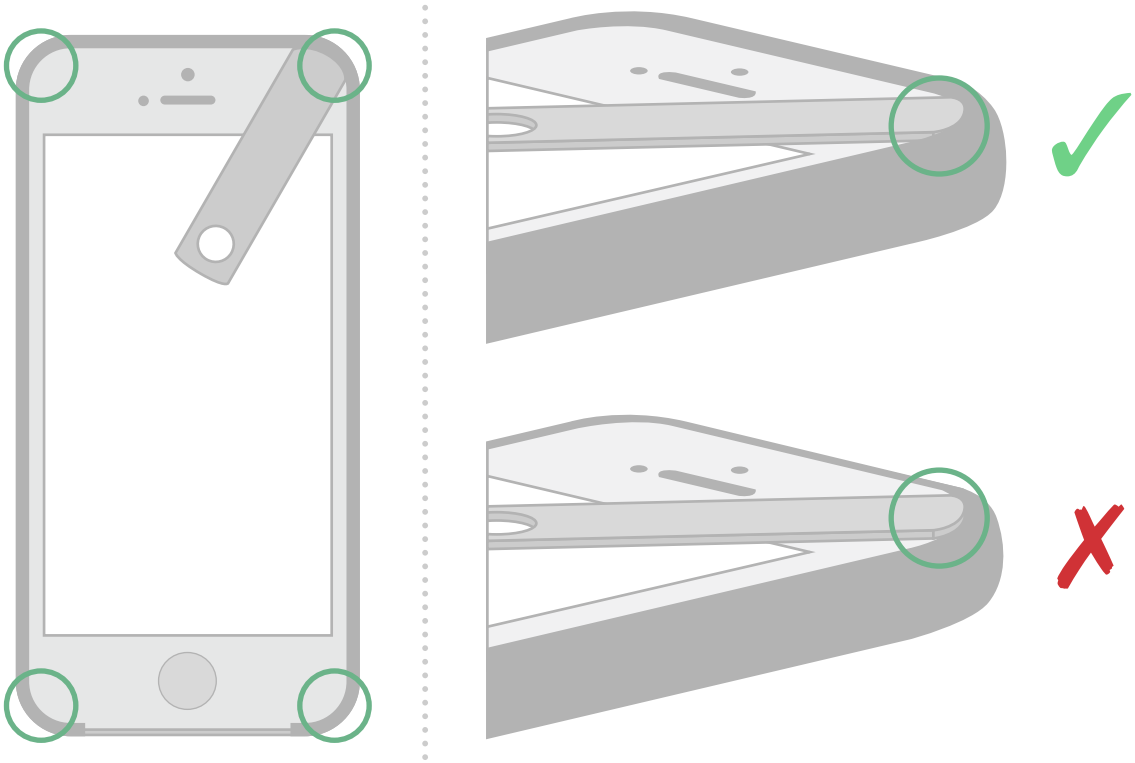
24.9.3.2 Procedure

1. Insert the Apple device into the case.
2. Verify that the Apple device completely fits inside the case. The Apple device must not be loose.
3. Verify that all buttons are accessible.
4. Inspect for button feel. The buttons must not be too hard to press or take a lot of effort to press.
5. For Apple devices that have an Apple Lightning connector:
   a. Insert the Apple Lightning Digital AV Adapter into the Lightning receptacle and verify that it fits.
   b. Using vernier calipers, measure the Lightning connector opening on the case. Verify that the opening is measured to be at least 12.05 mm by 6.30 mm.
6. For Apple devices that have an Apple 30-pin connector:
   a. Insert the Apple 30-pin to USB Cable into the 30-pin receptacle and verify that it fits.
   b. Using vernier calipers, measure the 30-pin connector opening on the case. Verify that the opening is measured to be at least 22.1 mm by 2.8 mm.
7. Insert Apple EarPods with Remote and Mic into the headset jack of the Apple device and verify it fits.
8. Using vernier calipers, measure the headset jack opening on the case. Verify that the opening is measured to be at least 6 mm in diameter and no more than 14 mm deep.
9. For Apple devices with Touch ID, use vernier calipers to verify that the case is at least 2 mm away from the Touch ID sensor.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

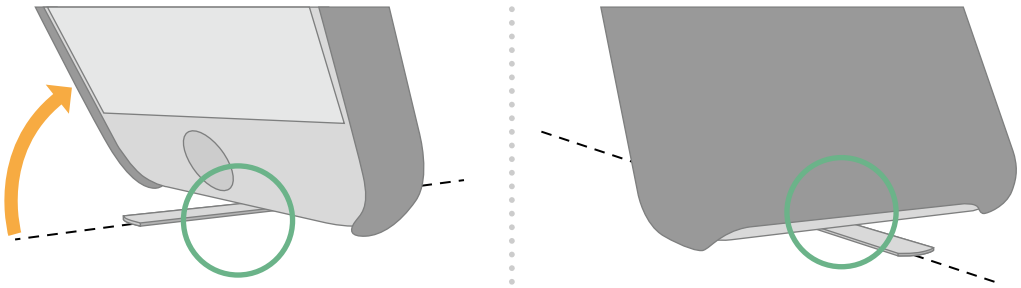
10. Verify that the case is always proud of the feeler gauge when the gauge is placed at each corner of the Apple device. See Figure 24-6 (page 503).

Figure 24-6 Apple Device Proudness Test



11. Set the Apple device flat on its face (screen facing down).
12. Roll the device towards any side that is not enclosed by the case until the gap between the Apple device's exposed glass and flat surface is smallest.
13. Verify that the feeler gauge fits into the gap between the Apple device's exposed glass and flat surface.

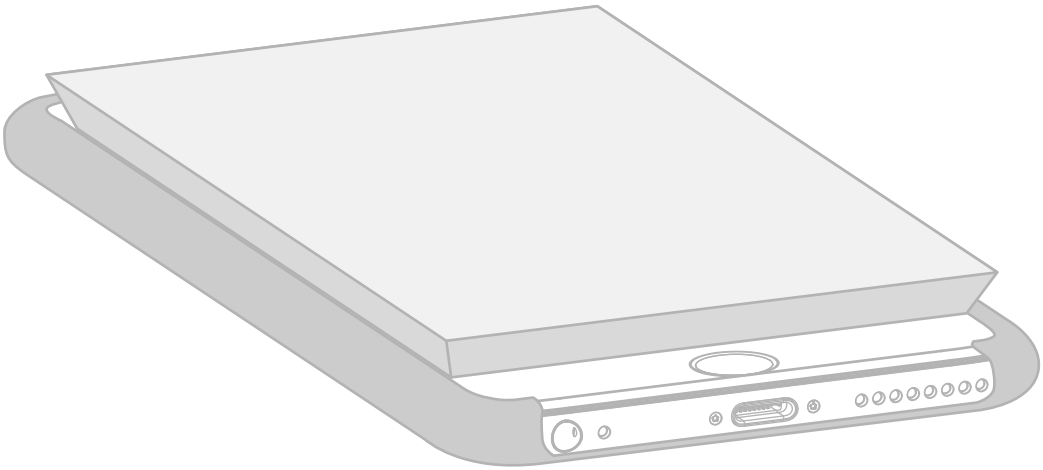
Figure 24-7 Apple Device Gap Test



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14. Place the touchscreen test block onto the touchscreen of the Apple device. Verify the test block is sitting flush on the touchscreen.

Figure 24-8 Apple Device Touchscreen Keep-out Test



15. If the case has an overlay, verify that there are no air gaps introduced between it and the touchscreen.

24.9.4 RF (OTA)

This section contains test procedures for accessory cases that contain a Lightning connector.

24.9.4.1 Definitions

- TRP: Total Radiated Power (dBm)
- TIS: Total Isotropic Sensitivity (dBm)
- EIS: Effective Isotropic Sensitivity (dBm)
- EIRP: Effective Isotropic Radiated Power (dBm)
- Electric Permittivity: $\varepsilon = \varepsilon_0 k_e$, where k_e is dielectric constant and ε_0 is permittivity of free space.

24.9.4.2 Labs

Testing must be performed by CTIA Test Labs (http://www.ctia.org/policy-initiatives/wireless-device-certification/ctia-authorized-test-labs) with the authorization and equipment needed for the following tests:

- EIRP Measurement for cellular (UMTS/LTE) and Wi-Fi (802.11a/b)
- EIS Measurement for cellular (UMTS/LTE)
- TRP for Wi-Fi (802.11a/b)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.9.4.3 Reference Device Setup

Figure 24-9 Apple Device Setup

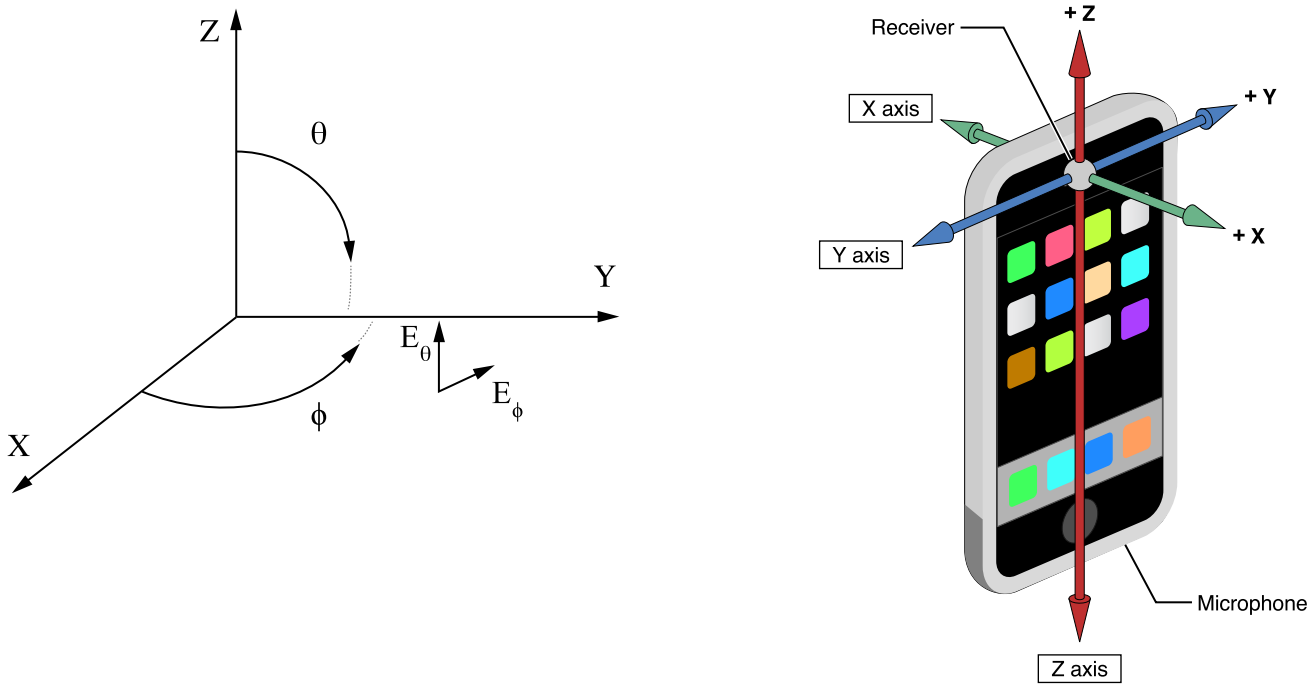
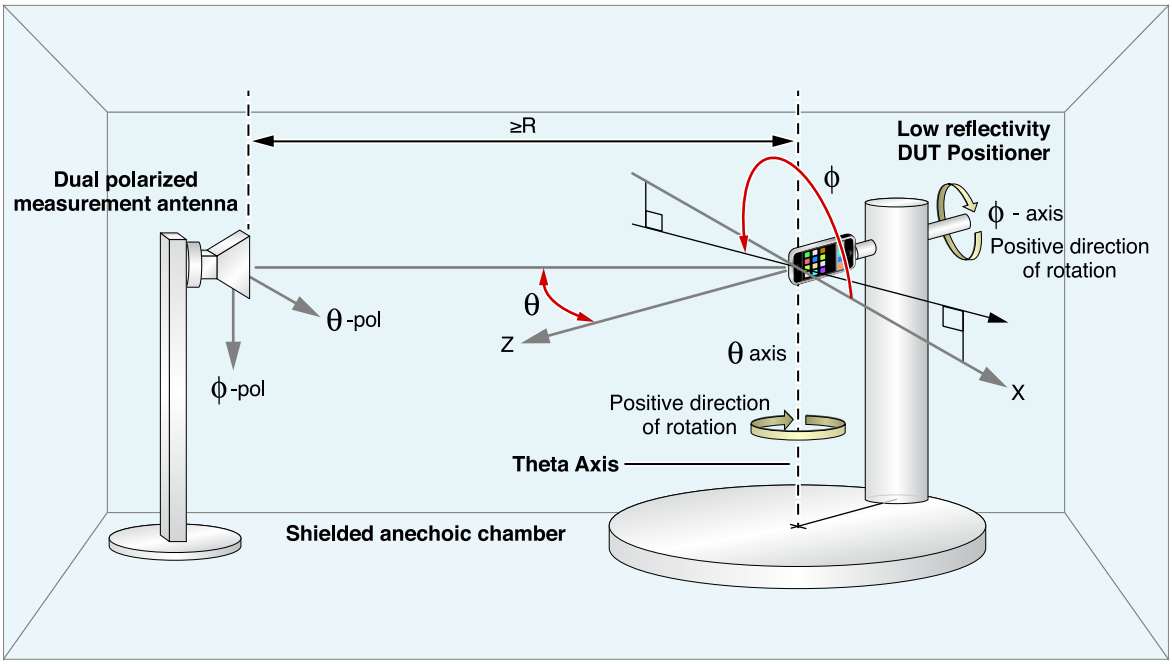


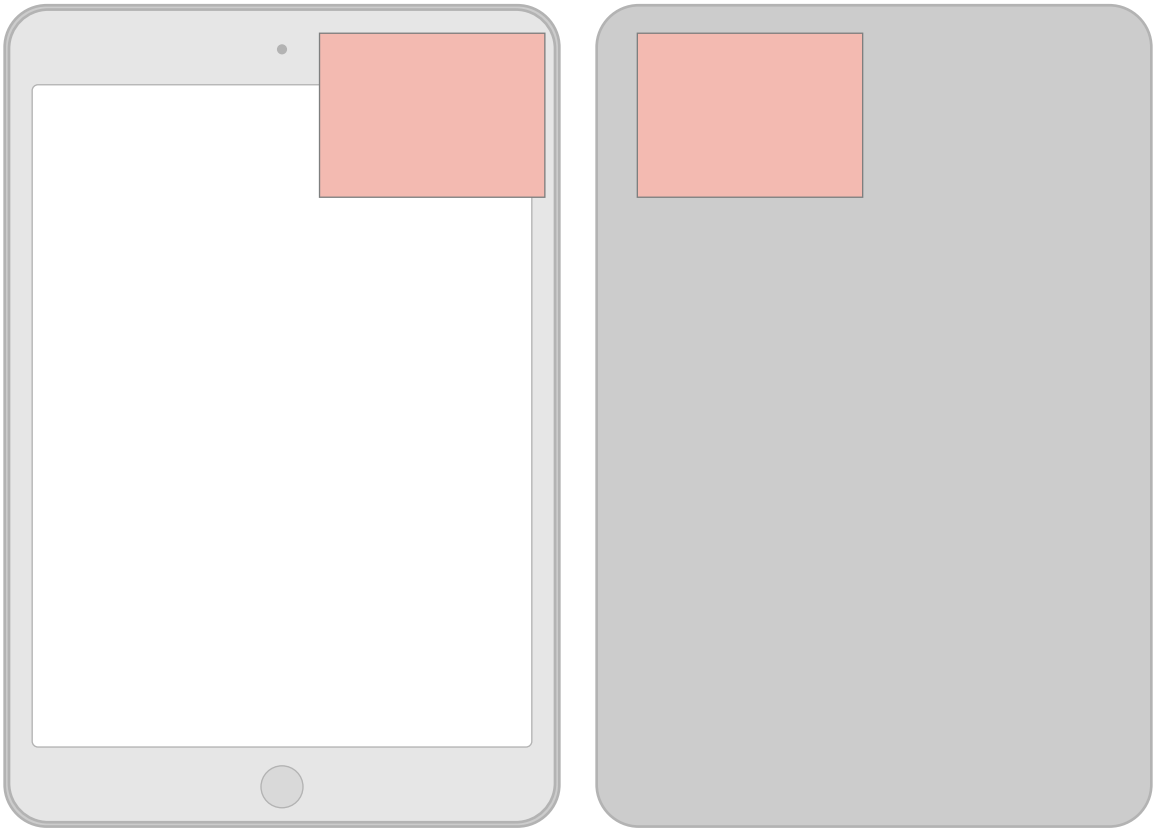
Figure 24-10 Generic Setup for TRP/TIS(Free Space)



24.9.4.3.1 iPad

If the reference Apple device is an iPad, secure 3 layers of Post-It notes or equivalent on the top right front and top left back. The paper must remain in place for all testing, both for reference data collection and during testing of iPad cases. See Figure 24-11 (page 506).

Figure 24-11 iPad with Post-it Notes



24.9.4.3.2 TRP Uncertainty

TRP measurements must be performed under Free Space conditions. Multiple reference Apple devices may be used under the following conditions:

- TRP values of each Apple device must not vary more than 0.5 dB.
- Uncertainty of the same Apple device must be less than 0.3 dB from each measurement.
- Uncertainty of the Apple device and accessory must be less than 0.5 dB.
- If the Apple device is changed during the measurement, the uncertainty must be less than 0.5 dB.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.9.4.3.3 TIS Uncertainty

TIS measurements must be performed under Free Space conditions. Multiple reference Apple devices may be used under the following conditions:

- TIS values of each Apple device must not vary more than 1.0 dB.
- Uncertainty of the same Apple device must be less than 1.0 dB from each measurement.
- Uncertainty of the Apple device and accessory must be less than 1.0dB.
- If the Apple device is changed during the measurement, the uncertainty must be less than 1.0 dB.

24.9.4.3.4 Peak Position

3D TRP/TIS measurements must be taken from each reference Apple device in each available chamber to determine peak position.

24.9.4.3.5 EIRP/EIS Measurement Matrix

Table 24-2 EIRP/EIS Measurement Matrix for case OTA testing procedure

Band	Channel	EIRP with Case	EIS with Case	Intermediate Channel
Band IV	L	Y	N	N
	M	Y	N	N
	H	Y	N	N
Band V	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
Band VIII	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
Band II	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
Band I	L	Y	Y	N

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Band	Channel	EIRP with Case	EIS with Case	Intermediate Channel
	M	Y	Y	N
	H	Y	Y	N
LTE 17	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
LTE 7	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N
TD-LTE 40	L	Y	Y	N
	M	Y	Y	N
	H	Y	Y	N

24.9.4.3.6 TRP/TIS Measurement Matrix

Note: This measurement must be tested using the latest CTIA spec. No intermediate test is required.

- UMTS Band I/II/IV(TRP only)/V/VIII Mid channel TRP/TIS
- LTE 17/7 Mid channel TRP/TIS
- TD-LTE B40 Mid channel TRP/TIS

Table 24-3 TRP/TIS Measurement Matrix for case OTA testing procedure

Band	Channel	TRP	TIS	Intermediate Channel
Band IV	L	N	N	N
	M	Y	N	N
	H	N	N	N
Band V	L	N	N	N

Copyright © 2015 Apple Inc. All Rights Reserved.

Band	Channel	TRP	TIS	Intermediate Channel
	M	Y	Y	N
	H	N	N	N
Band VIII	L	N	N	N
	M	Y	Y	N
	H	N	N	N
Band II	L	N	N	N
	M	Y	Y	N
	H	N	N	N
Band I	L	N	N	N
	M	Y	Y	N
	H	N	N	N
LTE 17	L	N	N	N
	M	Y	Y	N
	H	N	N	N
LTE 7	L	N	N	N
	M	Y	Y	N
	H	N	N	N
TD-LTE 40	L	N	N	N
	M	Y	Y	N
	H	N	N	N

24.9.4.3.7 iOS Configuration

- Cellular Testing:
  - Cellular must be turned on (Airplane Mode off)
  - Wi-Fi must be turned off
  - Bluetooth must be turned off

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

    - Screen must be turned off
- Wi-Fi Testing:
    - Cellular must be turned off (Airplane Mode on)
    - Bluetooth must be turned off
    - Wi-Fi must be turned on
    - Location Services must be turned off
    - Auto-Lock must be set to Never
    - Screen must be turned on
- The Apple device must be fully charged when starting EIRP measurement.
- EIRP measurements must not be performed when the Apple device's battery has less than 25% remaining.
- EIS measurements are not affected by battery level, but Apple highly recommends that EIS measurements be performed on a fully charged battery.
- See Peak Position (page 507) for additional setup information.

24.9.4.3.8 Apple Device Models

Specific Apple device models must be used to test certain OTA bands.

Table 24-4 Required Apple device models for testing of specific OTA bands

Device	Type A	Type B
iPhone 6s Plus	Unlocked A1634 or Unlocked A1687	Unlocked A1634 or Unlocked A1687
iPhone 6s	Unlocked A1633 or Unlocked A1688	Unlocked A1633 or Unlocked A1688
iPhone 6 Plus	Unlocked A1524	Unlocked A1524
iPhone 6	Unlocked A1586	Unlocked A1586
iPhone 5s	Unlocked A1453	Unlocked A1530
iPhone 5c	Unlocked A1456	Unlocked A1529
iPad Pro	Unlocked A1652	Unlocked A1652
iPad mini 4	Unlocked A1550	Unlocked A1550
iPad Air 2	Unlocked A1567	Unlocked A1567
iPad mini 3	Unlocked A1600	Unlocked A1601

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Device	Type A	Type B
iPad Air	Unlocked A1475	Unlocked A1476

- Cellular for LTE B7, TD-LTE B40 must be tested using Type B
- Cellular for UMTS I/II/IV/V/VIII, LTE B17 must be tested using Type A
- Wi-Fi: Either Type A or Type B must be used

24.9.4.4 Setup

Note: If the case has multiple configurations (e.g. 'open' and 'closed', and/or 'off' and 'on' modes), all modes must be tested and passed.

Case:

- The case and Apple device must be mounted using a material with a very low dielectric constant. Refer to *CTIA 3.1 Appendix A.1* for more information on free space mounts that may be used.
- The case and Apple device must be mounted in the same position (with respect to the measuring antenna) as the corresponding reference Apple device free space setup.
- If the case has 'on' and 'off' operating modes, it must be 'off' or not operating during TRP testing.
- If the case has 'on' and 'off' operating modes, it must be 'on' or operating during TIS testing.
- If the case can provide power to the Apple device during a phone call, the Apple device must be charging during TIS testing

Note: The link/communication antenna must be at least 3-5 feet away from the case and Apple device.

24.9.4.5 Procedure

Apple device setup:

1. Turn on Apple device
2. Put Apple device into Airplane Mode
3. Turn off the Apple device
4. Attach case to Apple device

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

5. Take Apple device out of Airplane Mode
6. Wait for Apple device to search for and attach to network
7. Connect call and wait > 1 minute
8. Start testing

24.9.4.5.1 TRP/EIRP for All Bands

- Cellular EIRP measurements must be taken at the Apple device's peak position (see Peak Position (page 507)) in free space.
- Wi-Fi EIRP Short Sweep measurements must be taken within ±15deg the Apple device's peak position in free space.
- See TRP Uncertainty (page 506) for TRP uncertainty requirements
- Cellular EIRP measurements must be taken at Low/Mid/High channels for each available band.
- Wi-Fi peak EIRP measurements must be taken at channels 1,7,13, 36, 60, 100, 120, and 165.
- 3D TRP measurements must be taken only when EIRP does not meet specification

24.9.4.5.2 TIS/EIS for All Bands

- EIS measurements must be taken at the Apple device's peak position (see Peak Position (page 507)) in free space.
- See TIS Uncertainty (page 507) for TIS uncertainty requirements
- EIS measurements must be taken at Low/Mid/High channels for each available band.
- 3D TIS measurements must be taken only when EIS does not meet specification.
- When testing WCDMA, place the Apple device at peak EIS position and decrease the output power of the base station emulator until 1.2% BER is reached.
- When testing LTE, place the Apple device at peak EIS position and decrease the output power of the base station emulator until 5% BLER is reached.

24.9.4.5.3 TRP/EIRP for UMTS Band I/II/IV/V/VIII

EIRP measurements must be taken every 15° in the Theta and Phi axis to derive the TRP value. All equipment settings must be set according to CTIA spec except for the following:

- FSP (Spectrum Analyzer)
  - Resolution BW = 30 kHz
  - Video BW = 10 MHz

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

    - Sweep Time = 100 ms
    - Points per trace = 501
    - Averaging Factor = 1 point.
    - Filter Trace Average Count = 2
    - Filter Selection = Integrated Channel Power
    - Detector Selection = RMS Detector
- Power meter may be used per CTIA spec
- TS-RSP (RF Switch Setting)
    - Pause after switching = 0 ms.

24.9.4.5.4 TRP/EIRP for FD-LTE Band 17/7 (10 MHz/QPSK modulation only)

EIRP measurements must be taken every 15° in the Theta and Phi axis to derive the TRP value. All equipment settings must be set according to CTIA spec except for the following:

- Power meter/Spectrum analyzer may be used per CTIA spec
- UL RB allocation of 12
- UL RB start 0 for low channel, 20 for mid channel, 38 for high channel
- TS-RSP (RF Switch Setting)
    - Pause after switching = 0 ms.

24.9.4.5.5 TRP/EIRP for TD-LTE Band B40 (20 MHz BW,QPSK)

EIRP measurements must be taken every 15° in the Theta and Phi axis to derive the TRP value. All equipment settings must be set according to CTIA spec except for the following:

- Power meter/Spectrum analyzer (FSQ/FSU)
    - Resolution BW = 50 MHz Video BW = 30 MHz
    - Span = 0 Hz
    - Sweep Time = 1.7 ms
    - Points per trace = 501
    - Trace = Average
    - Detector Selection = RMS Detector
- UL RB allocation of 18
- UL RB start 0 for low channel, 41 for mid channel, 82 for high channel

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

24.9.4.5.6 TIS/EIS for UMTS Band I/II/IV/V/VIII (all bands that are supported by DUT)

Note: Expected time for the TIS/EIS test is 3-5 hours/band

All equipment settings must be set according to CTIA spec except for the following:

- Sensitivity Setting
  - Sensitivity Search Algorithm = "Fast Probing" (RSCP Replacement is not allowed)
  - Max/Min Frames = 82
  - Power Settling Time = 0 ms
  - Output Channel Power Settling time = 0 ms
  - Initial Power Setting = -55 ~ -60dBm
  - TPC Setting = "all 1"
  - Probing Search Frames = 5
  - Initial Probe retries = 3
  - Probe Bias = 1
  - Mobile PCL Settle Time = 0 ms
  - Base Settle Time = 0 ms
  - Max Step = 2dB
  - Fine Step = 1dB
  - Minimum Step = 0.5dB
- TS-RSP (RF Switch Setting)
  - Pause after switching = 0 ms

24.9.4.5.7 TIS/EIS for FD-LTE Band 17/7 (10 MHz BW/QPSK modulation only)

Note: Expected time for the TIS/EIS test is 3-5 hours/band

All equipment settings must be set according to CTIA spec except for the following:

- Sensitivity Setting
  - Max/Min Frames = 2000

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Initial Power Setting = -55 ~ -60dBm
- Power Setting = "all 1"
- Max Step = 2dB
- Fine Step = 1dB
- Minimum Step = 0.5dB
- DL RB allocation of 50

- TS-RSP (RF Switch Setting)
  - Pause after switching = 0 ms

24.9.4.5.8 TIS for TD-LTE Band 40 (20 MHz BW/QPSK modulation only)

Note: Expected time for this test is 5 hours/band

All equipment settings must be set according to CTIA spec except for the following:

- Sensitivity Setting
  - Max/Min Frames = 2000
  - Initial Power Setting = -55 ~ -60dBm
  - Power Setting = "all 1"
  - Max Step = 2dB
  - Fine Step = 1dB
  - Minimum Step = 0.5dB
  - DL RB allocation of 50
- TS-RSP (RF Switch Setting)
  - Pause after switching = 0 ms

24.9.4.5.9 TRP for Wi-Fi

EIRP measurements must be taken every 15° in the Theta and Phi axis to derive the TRP value. Use the following test equipment settings for 802.11b and 802.11a:

- Data rate:
  - 802.11b: 11 Mbps
  - 802.11a: 6 Mbps

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Packet size: 500 bytes
- Interval between packets: 19 ms
- Preamble: Long (if applicable)

Connect the Apple device to the test network SSID:

- Under Settings > Wi-Fi, search for test network SSID and manually connect
- Once associated return Apple device to home screen

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

25. Communications

The communications feature includes the following sub-features:

- **Call State Updates**: Allow accessories to receive updates to the state of calls on Apple devices that support making telephony and/or FaceTime Audio/Video calls.
- **Communications Updates**: Allow accessories to receive updates about the communications status of the Apple device such as signal strength, carrier name, etc.
- **Call Controls**: Allow accessories to manage incoming/outgoing calls to/from the Apple device.
- **List Updates**: Allow CarPlay accessories to receive the recents and favorites lists from the Apple device.

> **Note:** Accessories that do not support CarPlay must not support the List Updates sub-feature.
>
> CarPlay accessories must only use these messages to display and control telephony features on displays not controlled by CarPlay such as an instrument cluster, HUD, or other screens.

25.1 Communications Requirements

All accessories that support the Call State Updates feature via iAP2 must send or receive the following iAP2 control session message(s):

StartCallStateUpdates (page 824)

CallStateUpdate (page 825)

StopCallStateUpdates (page 827)

All accessories that support the Communications Updates feature via iAP2 must send or receive the following iAP2 control session message(s):

StartCommunicationsUpdates (page 827)

CommunicationsUpdate (page 828)

StopCommunicationsUpdates (page 830)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

All accessories that support the Call Controls feature via iAP2 must send or receive the following iAP2 control session message(s):

StartCallStateUpdates (page 824)

CallStateUpdate (page 825)

StopCallStateUpdates (page 827)

All accessories that support the Call Controls feature via iAP2 may also send or receive the following iAP2 control session message(s):

InitiateCall (page 831)

AcceptCall (page 832)

EndCall (page 832)

SwapCalls (page 833)

MergeCalls (page 833)

HoldStatusUpdate (page 833)

MuteStatusUpdate (page 834)

SendDTMF (page 834)

All accessories that support the List Updates feature via iAP2 must send or receive the following iAP2 control session message(s):

StartListUpdates (page 835)

ListUpdate (page 836)

StopListUpdates (page 838)

25.2 Communications Usage

Accessories supporting the Communications feature must identify themselves as sending and receiving all required messages associated with the sub-features. If the device does not support any of the Communications messages then it will reject all identification attempts that include these messages.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

25.2.1 Call State Updates

Accessories may use choose to receive CallStateUpdate (page 825) messages to be notified when a new call is started or the status of a call changes on the Apple device. The StartCallStateUpdates (page 824) and StopCallStateUpdates (page 827) messages must be sent correspondingly to start or stop Call State Updates.

- ConferenceGroup will only be sent if IsConferenced is True.
- DisconnectReason will only be sent if Status is Disconnecting or Disconnected.

One initial CallStateUpdate (page 825) message will be sent by the Apple device to the accessory for each call at the time when the StartCallStateUpdates (page 824) message is received. If there are no calls at this time, a CallStateUpdate (page 825) message will be sent with a State of Disconnected. Additional update messages will be sent as the status of calls change on the Apple device. In some cases, you may receive duplicate CallStateUpdate (page 825) messages with the same information. You must use the provided CallUUID to keep track of the status of each call in progress.

25.2.2 Communications Updates

Accessories may choose to receive CommunicationsUpdate (page 828) messages to be notified when the status of the supported communications parameters change on the Apple device. The StartCommunicationsUpdates (page 827) and StopCommunicationsUpdates (page 830) messages must be sent correspondingly to start or stop Communications Updates.

- SignalStrength will always be reported as 0 when the device is in Airplane Mode. This parameter will only be sent if CellularSupported is True.
- RegistrationStatus will only be sent if CellularSupported is True.
- CarrierName will either report the name of the cellular carrier or "Searching...", "No Service", and "No SIM" when appropriate. This parameter will only be sent if CellularSupported is True.
- CellularSupported will report whether the Apple device has a cellular radio or not. A value of true does not guarantee that the device has active cellular service.
- TelephonyEnabled will report whether the Apple device can make telephony calls. This can either be over the cellular radio or through call relaying using WiFi. A value of true does not guarantee that the device has an active cellular/network connection. The call may fail if the device does not have an active cellular/network connection.
- FaceTimeAudioEnabled/FaceTimeVideoEnabled parameters report whether FaceTime Audio/Video are enabled on the Apple device.
- MuteStatus is global for the device. It is not a per-call status.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Some of communications parameters such as InitiateCallAvailable, EndAndAcceptAvailable, HoldAndAcceptAvailable, SwapAvailable, MergeAvailable, and HoldAvailable may differ based on the cellular technology of the carrier (CDMA vs. GSM).

It is also recommended to use the StartPowerUpdates (page 864) and PowerUpdate (page 864) messages from the Power feature (see Power (page 660)) to receive the battery level of the Apple device. It is suggested that the battery level be displayed next to the signal strength of the device.

25.2.3 Call Controls

Accessories may use choose to send various call control messages to manage incoming/outgoing calls to/from the Apple device. In order to use these call controls, the accessory must also support the Call State Updates sub-feature.

In some situations, certain call controls may not be possible. You must verify via Call State Updates that the selected action has actually taken place before updating the accessory UI. If a call control message has been sent to the device but the accessory does not receive a corresponding Call State Update, then the action was not possible at that time.

- MuteStatusUpdate is global for the device.
- SendDTMF is only available for calls with the service type Telephony.

25.2.4 List Updates

Accessories may use choose to receive ListUpdate (page 836) messages to be notified when a new call is started or the status of a call changes on the Apple device. The StartListUpdates (page 835) and StopListUpdates (page 838) messages must be sent correspondingly to start or stop List Updates.

The accessory can register to receive the recents and favorites lists from the Apple device as well as an update whenever these lists change.

When a list update is ready to be sent from the Apple device to the accessory, the device will first send a message with only the Available and Count (if available) parameter(s). This allows the accessory to prepare for the incoming list data by receiving the count (number of entries) in the list before the entries arrive in the messages following. If the Available parameter is false, this means the list is currently unavailable and no list entries will be sent to the accessory.

- RecentsList/RecentsListCount parameters will only be sent if the RecentsListProperties parameter was included in the StartListUpdates (page 835) message and the list is available.
- FavoritesList/FavoritesListCount parameters will only be sent if the FavoritesListProperties parameter was included in the StartListUpdates (page 835) message and the list is available.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- RecentsList -> UnixTimestamp is the timestamp of the call in represented as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970.
- RecentsList -> Duration is the duration of the call in seconds. This parameter is only sent if RecentsList -> Occurrences is 1.
- RecentsList -> Occurrences represents the number of consecutive calls to the same person that have been combined. This parameter will always be 1 unless RecentsListCombine is True.

It is also recommended to identify for the DeviceTimeUpdate (page 841) message which will provide TimeZoneOffsetMinutes and DaylightSavingsOffsetMinutes. These offsets can be used to correctly display the timestamp of the recent call according to the Apple device's time zone.

The accessory may only cache list data while the Apple device has an active iAP session with the accessory. The accessory must clear any cached data when the accessory is disconnected from the device. If the device sends a list update where the Available parameter is False, the accessory must clear any cached list data and display a message that the list is currently unavailable instead of any previously obtained list data.

25.2.5 Support for Apple devices running iOS 8.2 or older

Accessories that support the Communications feature and claim compatibility with iOS devices running iOS 8.2 or older must support legacy definitions of the `CallStateUpdateStatus` (see Table 59-46 (page 826)) and `CallStateUpdateDirection` (see Table 59-47 (page 826)) parameters.

The legacy definitions are `CallStateUpdateStatusLegacy` (see Table 59-50 (page 827)) and `CallStateUpdateDirectionLegacy` (see Table 59-51 (page 827)) respectively.

To determine if the Apple device supports the legacy definitions, the accessory should claim that it supports the SendDTMF message (see SendDTMF (page 834)) during Accessory Identification (see Accessory Identification (page 265)).

- If identification succeeds, the accessory should not use the legacy parameter definitions.
- If identification fails, the accessory should re-identify without the SendDTMF message and use the legacy parameter definitions.

25.3 Test Procedures

25.3.1 iAP2 Tests

Verify that the following iAP2 control session message(s) are sent or received for each feature.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Call State Updates:

- StartCallStateUpdates (page 824)
- CallStateUpdate (page 825)
- StopCallStateUpdates (page 827)

Communications Updates:

- StartCommunicationsUpdates (page 827)
- CommunicationsUpdate (page 828)
- StopCommunicationsUpdates (page 830)

Call Controls:

- StartCallStateUpdates (page 824)
- CallStateUpdate (page 825)
- StopCallStateUpdates (page 827)
- All Call Control messages supported by the accessory.

List Updates:

- StartListUpdates (page 835)
- ListUpdate (page 836)
- StopListUpdates (page 838)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

26. Device Authentication

The Device Authentication feature is used to verify the authenticity and/or identity of an Apple device.

26.1 Device Authentication Requirements

All accessories that support the Device Authentication feature via iAP2 must send or receive the following iAP2 control session message(s):

RequestDeviceAuthenticationCertificate (page 839)

DeviceAuthenticationCertificate (page 839)

RequestDeviceAuthenticationChallengeResponse (page 839)

DeviceAuthenticationResponse (page 840)

DeviceAuthenticationSucceeded (page 840)

DeviceAuthenticationFailed (page 840)

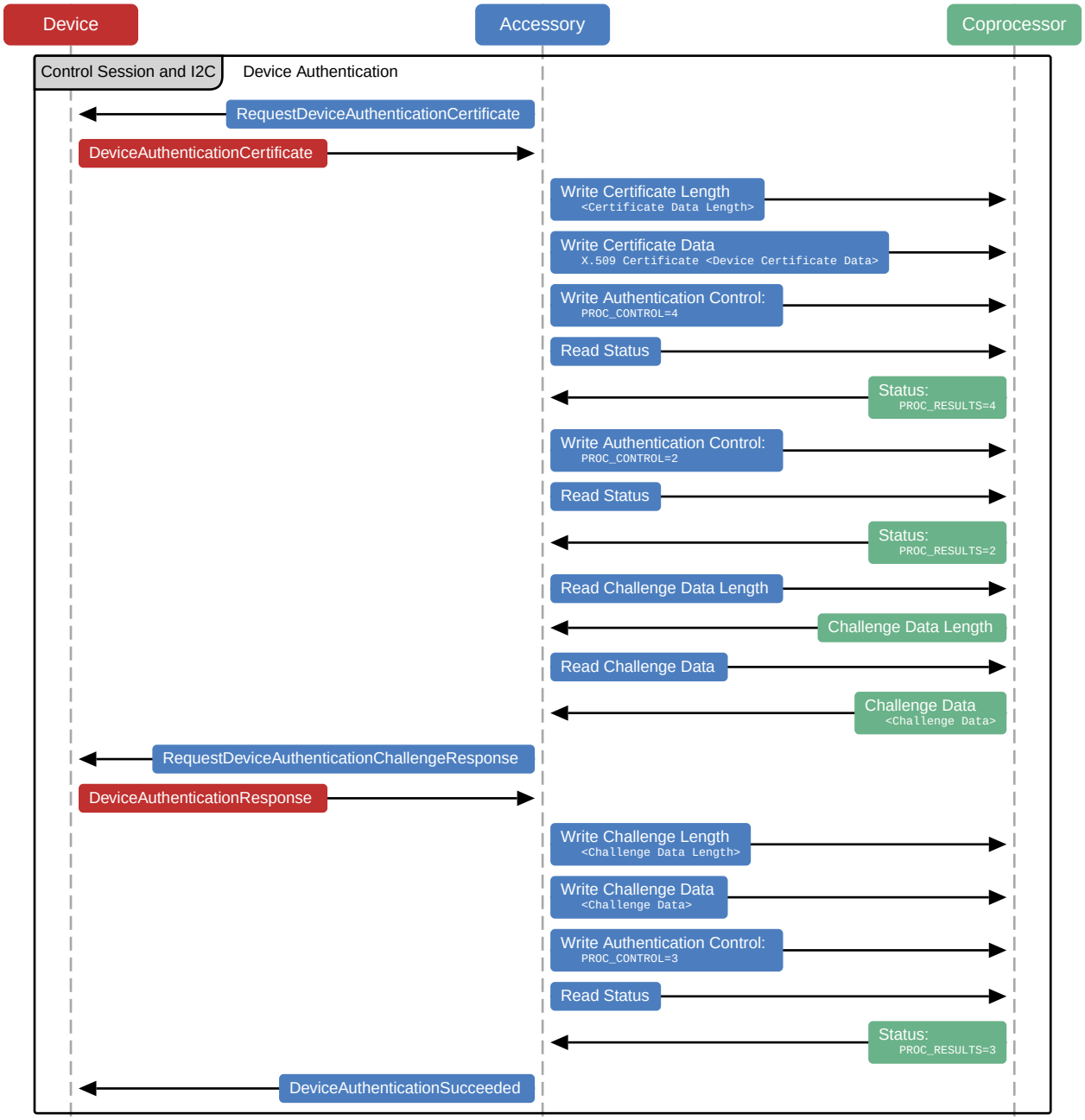
26.2 Device Authentication Usage

The Device Authentication process is the same as that specified in Accessory Authentication (page 261) except that the roles are reversed, and the accessory must terminate the iAP2 connection if the device fails to authenticate. For example, the accessory may continue to communicate with the device but with restricted (non-iAP2) functionality. Accessory Authentication (page 261) and Accessory Identification (page 265) must take place before Device Authentication may occur.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

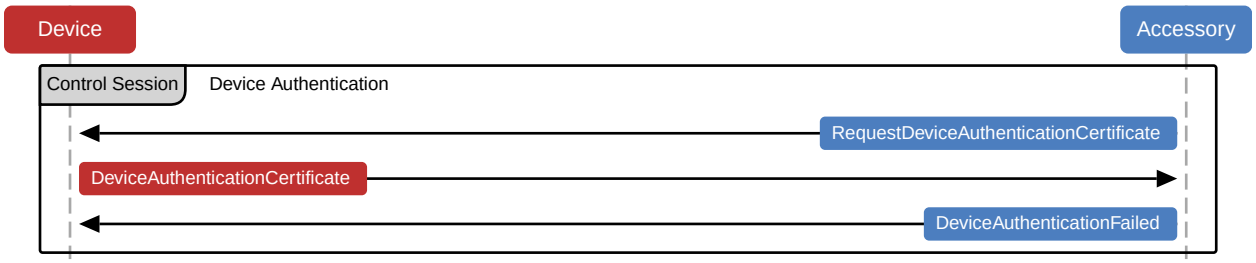
26.3 Device Authentication Examples

26.3.1 Typical Device Authentication

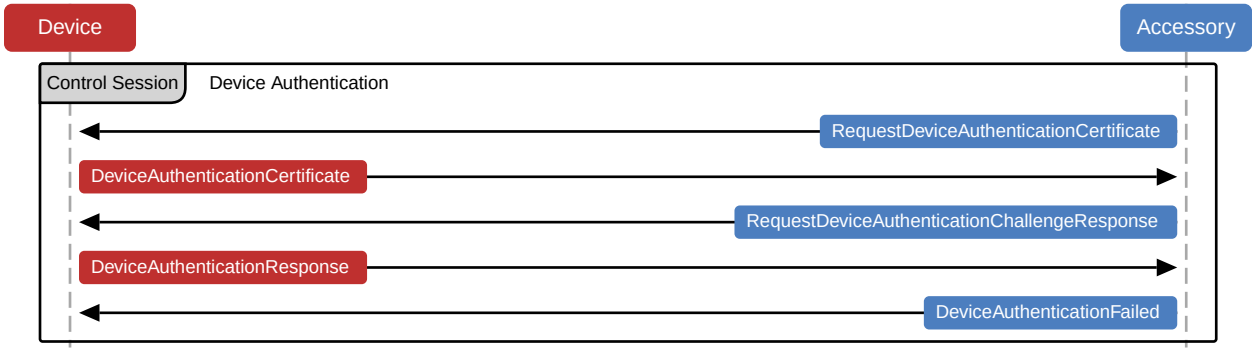


2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

26.3.2 Device Authentication Failure Due To Invalid Certificate



26.3.3 Device Authentication Failure Due To Invalid Response



26.4 Test Procedures

26.4.1 iAP2

Verify that the following iAP2 control session message(s) are sent or received:

- RequestDeviceAuthenticationCertificate (page 839)
- DeviceAuthenticationCertificate (page 839)
- RequestDeviceAuthenticationChallengeResponse (page 839)
- DeviceAuthenticationResponse (page 840)
- DeviceAuthenticationSucceeded (page 840)
- DeviceAuthenticationFailed (page 840)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

27. Device Notifications

An accessory that supports the Device Notifications feature can register for messages from the Apple device that convey various parts of device state. Device information that can be provided includes:

- Human-readable name as set by the user (i.e. "John Doe's iPhone")
- Current UI language
- Current date/time information

27.1 Device Notifications Requirements

All accessories that support the Device Notifications feature via iAP2 may also send or receive the following iAP2 control session message(s):

DeviceInformationUpdate (page 841)

DeviceLanguageUpdate (page 841)

DeviceTimeUpdate (page 841)

DeviceUUIDUpdate (page 842)

All messages related to this feature are optional.

27.2 Device Notifications Usage

The Apple device will automatically send update messages if the accessory claims to receive them during Accessory Identification. After successful identification, the Apple device will send one update message to inform the accessory of initial state, and will send additional messages if that state changes. Accessories must be prepared to receive multiple update messages while connected to the device.

After receiving DeviceTimeUpdate (page 841), the accessory must maintain its own clock. Subsequent notifications will be sent when there is a significant change, e.g., a device time zone or daylight savings time change.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The DeviceUUIDUpdate (page 842) can be used by accessories to determine if they are connected to the same device over multiple transports. The UUID will remain constant as long as there are any iAP accessories connected to the device (over any transport). The UUID will be changed no less than one minute after all iAP accessories disconnect.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

28. Digital Audio

Apple strongly recommends the use of digital audio paths to and from accessories. USB Host Mode audio is the recommended approach; the USB Device Mode audio feature is being placed into a maintenance mode.

When an Apple device is in USB Host Mode, it can interact with accessories that are compliant with either the USB Audio 1.0 or 2.0 specification. Version 2.0 is recommended, even for USB Full Speed accessories. USB High Speed accessories must not implement the USB Audio 1.0 specification.

When an Apple device is in USB Device Mode, it can appear as a USB Audio 1.0 device for the purpose of streaming audio to the accessory. The audio stream contains stereo PCM data and is synchronized using USB 1 ms Start-Of-Frame (SOF) packets. USB Device Mode audio does not support streaming audio from an accessory to the device.

28.1 Digital Audio Requirements

Note: All accessories must comply with TDMA noise requirements. See TDMA Noise (page 65).

28.1.1 Readiness for Audio Streaming

Failure to render all audio output from an Apple device once it has routed outgoing audio to an accessory is grounds for failure to pass self certification. Accessories must not establish an audio transport connection (in either direction) to an Apple device until they are completely ready to provide or receive audio streaming data.

28.1.2 Multiple Audio Connections

All accessories must maintain a maximum of one audio transport connection with a device at any given time. This requirement covers the following audio transport connections:

- USB Device Mode audio
- USB Host Mode audio
- Bluetooth A2DP audio
- Headset Jack audio

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Lightning Audio Module audio
- CarPlay audio

If an accessory is capable of streaming audio to or from an Apple device via more than one audio transport connection, it must shut down the active audio transport connection before starting the next one. The presence or absence of audio on an audio transport connection has no bearing on the connection state. Shutdown and startup actions for each audio transport connection are specified in the following table:

Table 28-1 Audio Transport Connection Actions

Audio Transport Connection	Startup Action	Shutdown Action
USB Device Mode	StartUSBDeviceModeAudio (page 866)	StopUSBDeviceModeAudio (page 867)
USB Host Mode	USB attach	USB detach
Bluetooth A2DP	Bluetooth connect	Bluetooth disconnect
Headset Jack	Plug insertion	Plug removal
Lightning Audio Module	Lightning Audio Module connect	Lightning Audio Module disconnect
CarPlay	CarPlay connect	CarPlay disconnect

If an accessory is capable of establishing a Bluetooth A2DP audio transport connection along with any other audio transport connection to an Apple device, the accessory must satisfy all requirements stated in Bluetooth (page 344).

28.1.3 Audio Input Source Switching

Accessories that can accept audio input from sources other than Apple devices must select the Apple device input source as soon as the accessory establishes an audio transport connection to the device (see Table 28-1 (page 529)), whether as a result of direct user action or otherwise.

Conversely, if the user takes direct action to switch the accessory's audio input source away from the Apple device, the accessory must shut down the active audio transport connection (see Table 28-1 (page 529)) before switching to the alternate source.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

28.1.4 Copy Protection of Digital Audio Output

Any accessory that outputs digital audio obtained from an Apple device must implement copy protection in its output stream; for example, by setting the output's Serial Copy Management System (SCMS) bits to 10.

28.1.5 USB Host Mode Audio

An accessory implementing USB Host Mode Audio must support the following USB features:

- 16-bit linear PCM.
- Support for both 44100 and 48000 Hz sampling rates.
- If the input and output endpoints are both asynchronous, both endpoints must use the same clock source (i.e. be sample locked) and use implicit feedback.
- The maximum packet size descriptor must comply with requirements for USB Audio packet sizes per Section 2.3.1.1 of the USB Audio Data Formats 2.0 specification. Packets must never vary by more than ±1 sample frame per packet.
- The maximum packet size for each endpoint must comply with the calculation method specified by Technical Note TN2274 in the Mac Developer Library online documentation.
- When streaming, accessories must always observe the bandwidth restrictions for the current sample rate and format settings.
- Volume Control Feature Units must use a Status Interrupt Pipe, per Section 3.7.1.2 of the USB Audio 1.0 specification or Section 6 of the USB Audio 2.0 specification.
- A USB Host Mode accessory that wants to give iOS access to its volume control capabilities must implement either a USB Master Volume Control Feature Unit and/or USB Individual Volume Control Feature Units on all the relevant streaming audio endpoints. If iOS wants to set the accessory's overall input/output volume and no master control exists, it will aggregate the individual controls and set them in tandem.
- All USB audio interfaces must include zero-bandwidth alternate settings.

The following USB features are recommended for accessories implementing USB Host Mode Audio, but are not required:

- Synchronous audio endpoints, particularly for input-only or output-only accessories.
- 24-bit linear PCM.
- Apple devices running iOS 5.0 or later support more than two channels of audio input in USB Host Mode. Accessories that use this feature must bundle all audio input channels into a single audio streaming interface.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Apple devices running iOS 6.0 or later support more than two channels of audio output in USB Host Mode. Accessories that use this feature must bundle all audio output channels into a single audio streaming interface.

Accessories should be tested against the latest OS X Audio driver; the application Audio MIDI Setup, in the OS X Applications/Utilities folder, can be used to verify compatibility. Additionally, the USB Prober application, available with Xcode, should be used to inspect and verify the contents of USB descriptors.

28.1.6 USB Device Mode Audio

The accessory must register a USB Device Mode component during Identification to make use of this feature. Additionally, the USB Device Mode component must report the audio sampling rates that the accessory supports. All accessories must at least support the 32000 Hz, 44100 Hz, and 48000 Hz sample rates.

> **Note:** The Apple device USB isochronous audio data endpoint descriptor bmAttributes field erroneously returns the Synchronization Type field (D3:2) as b00 (no synchronization) instead of the correct value, b11 (synchronous). The Apple device supports synchronous data transfers, so the accessory must ignore those erroneous attribute bits. The erroneous b00 value is retained for backwards compatibility with older accessories.

When receiving USB Device Mode audio over a USB streaming interface, the accessory may buffer data to achieve consistent audio playback. Since buffering introduces latency, it is recommended that this latency be kept below 200 ms to ensure timely response to user input, such as a play/pause button.

All accessories that support the USB Device Mode Audio feature via iAP2 must send or receive the following iAP2 control session message(s):

StartUSBDeviceModeAudio (page 866)

USBDeviceModeAudioInformation (page 867)

StopUSBDeviceModeAudio (page 867)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

28.2 Digital Audio Usage

28.2.1 USB Host Mode Audio

The accessory must authenticate and identify itself to the Apple device using iAP2 before the Apple device will enumerate and start using the USB Audio interfaces. When audio streaming is active, the streaming interface will be selected, and when audio streaming is inactive, the zero-bandwidth interface will be selected. Device-powered accessories must use this event to trigger compliance with requirements stated in Power (page 660).

28.2.2 USB Device Mode Audio Usage

The accessory's USB host controller must enumerate the device and select the second configuration, which corresponds to a composite USB Device Mode Audio and iAP2 HID device. Once an iAP2 connection is established and the accessory has successfully identified and authenticated, it must select a nonzero-bandwidth alternate USB audio streaming interface on the Apple device. The accessory must not send USB IN tokens to the audio streaming interface until a USBDeviceModeAudioInformation (page 867) message has been received.

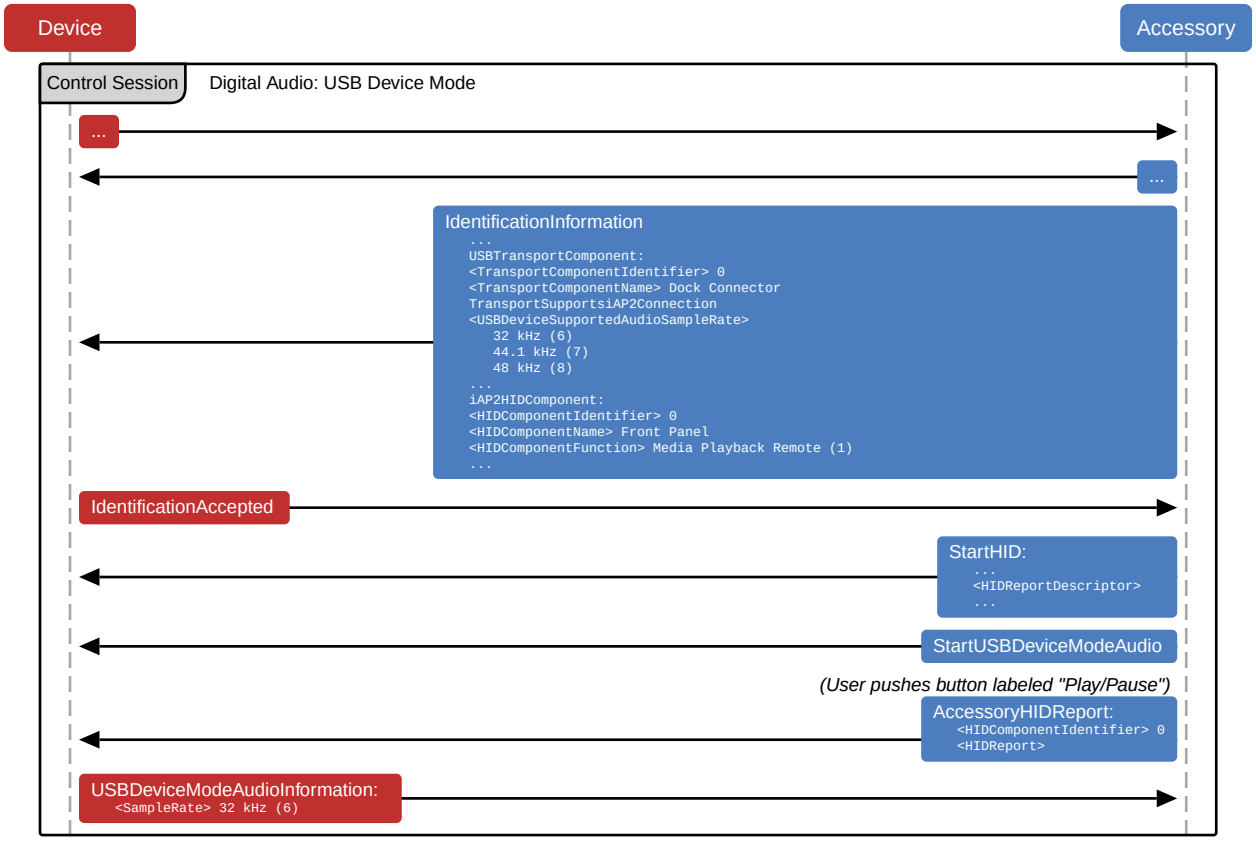
When the accessory is ready for audio streaming output, it must send StartUSBDeviceModeAudio (page 866) to the device. The device will inform the accessory of the stream properties via USBDeviceModeAudioInformation (page 867) messages. Multiple update messages will be sent while streaming is active to inform the accessory of changes in audio stream properties. When the accessory no longer wants to consume audio streaming output, it must send StopUSBDeviceModeAudio (page 867) to the device.

28.3 Digital Audio Examples

28.3.1 Typical USB Device Mode Digital Audio

The following sequence diagram illustrates how to correctly identify for and start USB Device Mode Digital Audio using iAP2.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.



28.4 Test Procedures

These tests must be performed on accessories that support audio playback.

28.4.1 Equipment Required

- Latest revision of ATS software
- ATS hardware
- The following test content:
  - Audiobooks
  - Music Albums

28.4.2 Test Cases

1. Verify that the music playback starts when connecting Apple device to the accessory (you may need to select iPod source on some accessories). Repeat the test with all supported Apple devices.
2. Verify audio routes from Apple device to the accessory.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

   a. Start playing music on Apple device, then connect to the accessory.
   b. Verify that music playback starts when connecting an Apple device with an empty playback engine to the accessory (you may need to select iPod source on some head units).

3. Verify that audio resumes from the last track played when connecting an Apple device in sleep mode.
   a. Put the Apple device in Airplane Mode.
   b. Leave the Apple device idle for 3 minutes (detach power, make sure currently playing track is paused).
   c. Connect the Apple device to the head unit.
4. Verify Podcast/Audiobook/Genius Mixes audio resumes when connected to the accessory.
   a. Play a test track on the Apple device.
   b. Connect the Apple device to the accessory.
5. Verify that there are no issues with the audio playback when playing Podcast/Audiobook/Genius Mixes.
6. Verify the connectivity by detaching and attaching the Apple device 10 times (very quickly) from the accessory. After 10 times, verify that the Apple device is recognized by the accessory and can play music. Also verify the user is able to browse and select tracks.
7. Verify that the accessory supports the sample rates of 32 kHz, 44.1 kHz and 48 kHz for regular music.
8. Verify that there are no issues when switching audio sources on the accessory (e.g., iPod/AM/FM/CD/Bluetooth audio/AUX).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

29. External Accessory Protocol

The iOS External Accessory framework provides accessories with the means to communicate with one or more iOS apps via one or more External Accessory (EA) sessions. These sessions present a simple read/write bytestream interface; it is up to the accessory developer to specify a custom protocol between the app and the accessory. The design and maintenance of communication protocols between accessories and applications are entirely the responsibility of the developers of these products. Apple makes no attempt to secure protocol name space or provide for communication security between accessories and applications.

All accessories may implement this feature by defining and implementing an iAP2 EA session. Additionally, accessories with a USB Host Mode transport component may choose to implement an EA Native Transport (USB Host Mode). This will generally provide higher performance than the iAP2 EA session. Apple does not guarantee reliable delivery and/or a minimum level of transfer speed.

29.1 External Accessory Protocol Requirements

The following requirements apply to accessories that support this feature:

- Accessories must declare at least one `ExternalAccessoryProtocol` parameter in their IdentificationInformation (page 807) message.
- All declared `ExternalAccessoryProtocolIdentifier` and `ExternalAccessoryProtocolName` parameters must be unique in an IdentificationInformation (page 807) message.
- Protocol names must be in reverse-DNS format and must be associated with domains that are registered with the Internet Corporation for Assigned Names and Numbers (ICANN), so that Apple can rely on each protocol having a unique owner. Apple does not require a developer to be the owner of the domain name used, but does require that the developer have permission to use the domain name for the protocol. Protocol names used for demonstration or example purposes in accessory reference designs must not be used in shipping accessories.
- Accessories must choose a match action (Table 59-12 (page 810)) for each supported protocol. Apple recommends that accessories support the App Match feature (see App Match (page 340)) and use 1 unless the protocol in question is meant for infrequently accessed accessory features, such as firmware updates or diagnostics.
- Accessories must demonstrate use of all supported protocols during self certification.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Accessories must also either implement one iAP2 EA session or one EA Native Transport over a USB Host Mode transport component. Accessories only need to implement one iAP2 EA session regardless of the number of EA protocols supported using that session. Conversely, an EA Native Transport (USB Host Mode) can only support a single EA protocol.

Accessories may implement both an iAP2 EA session for some protocols and use EA Native Transports for others, but an EA protocol must not use both an iAP2 EA session and an EA Native Transport.

For more information on developing iOS apps that communicate with accessories, see *External Accessory Programming Topics* in the iOS SDK documentation at http://developer.apple.com/library/ios/featuredarticles/ExternalAccessoryPT/Introduction/Introduction.html.

29.1.1 iAP2 EA Session Requirements

All accessories that support the External Accessory Protocol feature via iAP2 must send or receive the following iAP2 control session message(s):

StartExternalAccessoryProtocolSession (page 843)

StopExternalAccessoryProtocolSession (page 843)

To implement the iAP2 EA session, accessories must also declare the presence of that session during iAP2 link synchronization. Details on the session implementation can be found in External Accessory Session (page 802).

All accessories that support the External Accessory Protocol feature via iAP2 may also send or receive the following iAP2 control session message(s):

StatusExternalAccessoryProtocolSession (page 843)

If the accessory declares support for StatusExternalAccessoryProtocolSession (page 843), the Apple device will allow multiple iAP2 EA sessions per protocol (excluding EA Native Transport).

29.1.2 EA Native Transport (USB Host Mode) Requirements

To implement an EA Native Transport over USB Host Mode, an accessory must declare one additional USB interface in its device descriptor. The interface must have two alternate settings; alternate setting 1 must have one bulk IN and one bulk OUT endpoint, and alternate setting 0 must be a zero-bandwidth setting with no endpoints. Also, the accessory must declare the presence of an EA Native Transport component during identification.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: Accessories must not reuse the same USB Bulk In/Out endpoints for multiple purposes, such as iAP2 and EA native transport. They also must not shut down the iAP2 connection once a EA Native Transport session has been started.

Table 29-1 USB Interface Descriptor (Alternate Setting 1 - Active Session) for External Accessory Native Transport (USB Host Mode)

USB Descriptor	Value	Comments
Interface Number	`0xNN`	Must be different from the iAP2 interface number.
Interface Class	`0xFF`	Vendor-specific interface
Interface SubClass	`0xF0`	MFi Accessory
Interface Protocol	`0x01`	External Accessory Native Transport
Interface String	`com.company.protocol`	Must match an identified `ExternalAccessoryProtocolName`
Number of Endpoints	2	1 Bulk IN and 1 Bulk OUT endpoint descriptor must be specified
Alternate Setting	1	

Table 29-2 USB Interface Descriptor (Alternate Setting 0 - Zero Bandwidth) for External Accessory Native Transport (USB Host Mode)

USB Descriptor	Value	Comments
Interface Number	`0xNN`	Must be the same as the default interface
Interface Class	`0xFF`	Vendor-specific interface
Interface SubClass	`0xF0`	MFi Accessory
Interface Protocol	`0x01`	External Accessory Native Transport
Interface String	`com.company.protocol`	Must match an identified `ExternalAccessoryProtocolName`
Number of Endpoints	0	
Alternate Setting	0	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

29.2 External Accessory Protocol Usage

29.2.1 iAP2 EA Session Usage

iAP2 EA Session protocol initiation is always driven by an app running on the Apple device. When the accessory receives a StartExternalAccessoryProtocolSession (page 843) message, it may start to send and receive traffic on the identified iAP2 EA Session for the specified protocol and session identifier.

Conversely, the accessory must cease sending and receiving traffic for the specified session identifier when a StopExternalAccessoryProtocolSession (page 843) message is received. Device-powered accessories may also be required to transition to a low power state as specified in the Accessory Power feature.

29.2.2 EA Native Transport (USB Host Mode) Usage

If it is using EA Native Transport over USB Host Mode, the accessory's default interface will be selected by the Apple device when an app opens an EA session with the accessory. Once selected, the accessory must continually service both the bulk IN and OUT endpoints to maintain communication with the app. When the app has closed the EA session, the alternate zero-bandwidth interface will be selected to signal the accessory that EA traffic for that protocol has ceased. Device-powered accessories may also be required to transition to a low power state as specified in the Accessory Power feature.

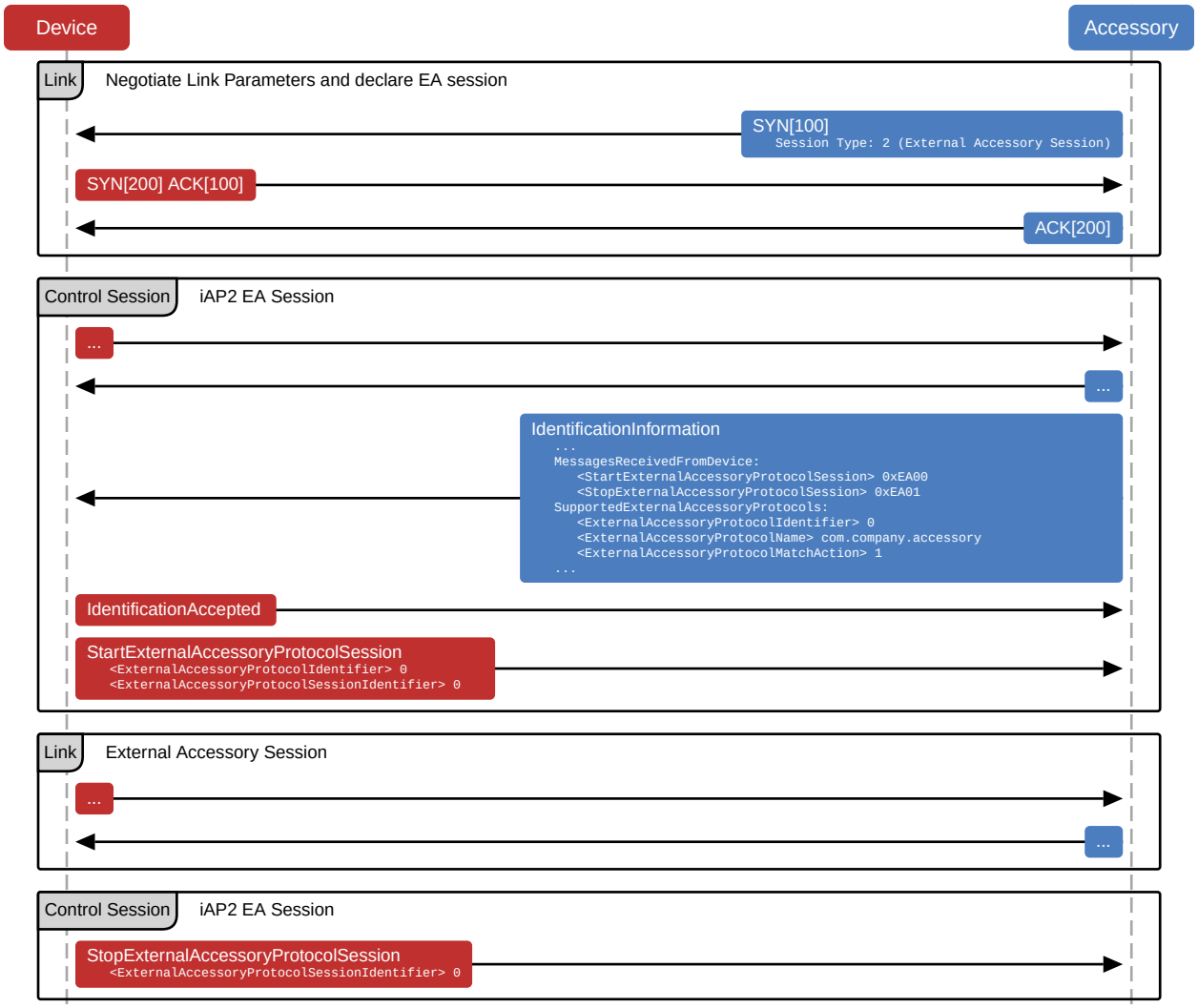
EA Native Transport (USB Host Mode) performance will vary based on many factors including, but not limited to:

- USB Full Speed versus USB High Speed.
- Bulk endpoint packet size.
- Depth of USB packet queues.
- Other concurrent USB traffic on the same bus/controller.

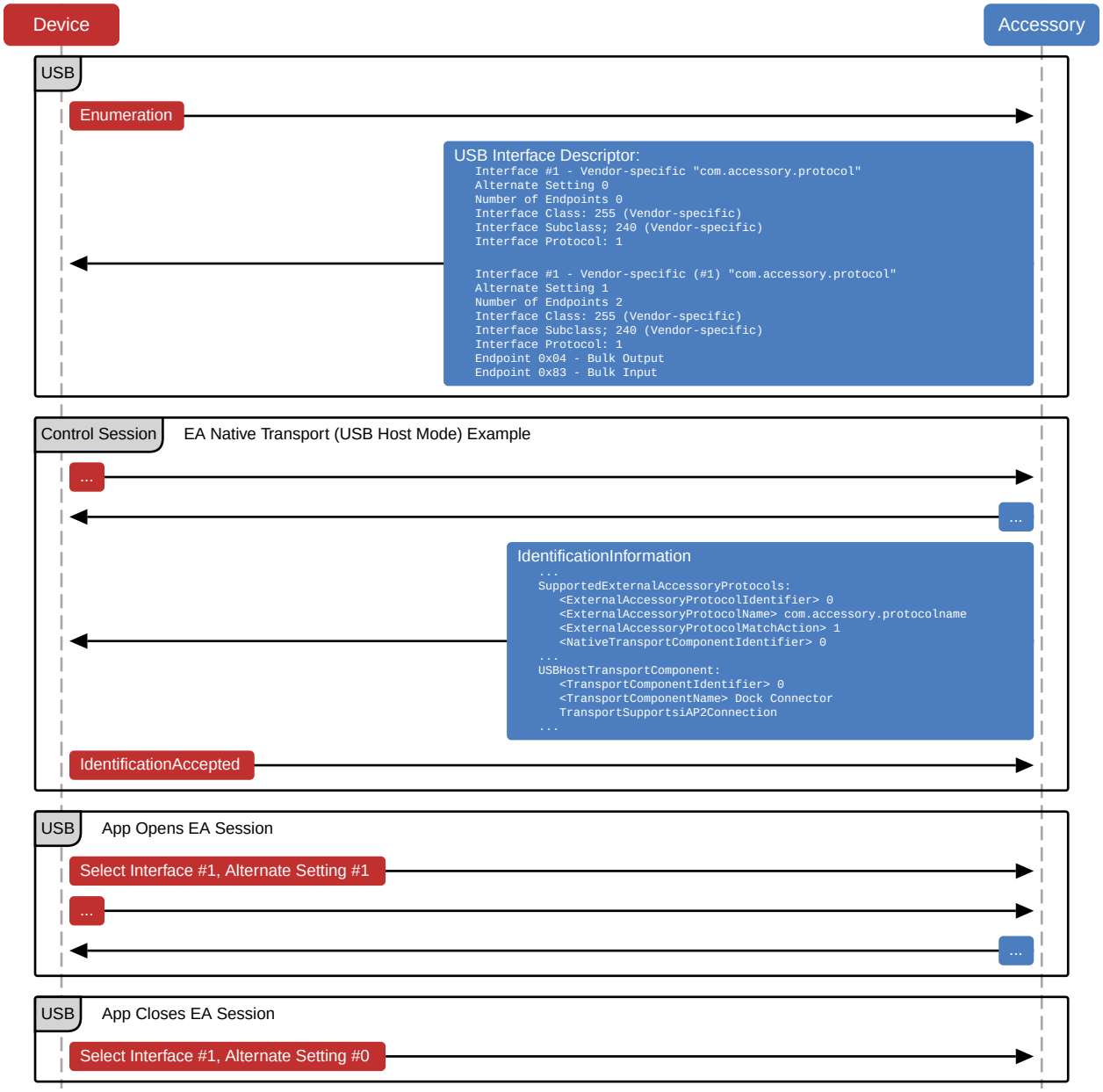
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

29.3 External Accessory Protocol Examples

29.3.1 iAP2 EA Session Example



29.3.2 EA Native Transport (USB Host Mode) Example



29.4 Test Procedures

29.4.1 SDK Apps

These tests must be performed on accessories that communicate with iOS apps.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

29.4.1.1 Equipment Required

- Latest released version of Xcode
- Mac running OS X version 10.10 or greater
- Latest revision of ATS software
- ATS hardware
- BPA100 hardware for Bluetooth capture
- iOS application to be tested
- iOS application must be in either .ipa or .app format

29.4.1.2 Test Cases

1. Verify that there are no errors in ATS when launching and closing app.
2. Connect an Apple device that does not support External Accessory protocol such as the iPod nano. Verify that there are no errors in ATS.
3. Connect Apple device that does not support IDPS (iPod classic). Verify that there are no errors in ATS.

For accessories supporting iAP2:

1. From within ATS, note the value for ExternalAccessoryProtocolName inside IdentificationInformation. The value should be in the form of *com.name.protocol*.
2. On the Mac, control-click the .app file and select "Show Package Contents".
3. Open "info.plist" with Xcode and note the value for "Supported External Accessory Protocols".
4. Verify that the "Supported External Accessory Protocols" value matches the value in step 1.

For accessories supporting iAP1:

1. From within ATS, note the value for EAProtocolToken inside SetFIDTokenValues. The value should be in the form of *com.name.protocol*.
2. On the Mac, control-click the .app file and select "Show Package Contents".
3. Open "info.plist" with Xcode and note the value for "Supported External Accessory Protocols".
4. Verify that the "Supported External Accessory Protocols" value matches the value in step 1.

29.4.2 External Accessory Protocol over iAP2

1. Verify the accessory declares the presence of an External Accessory session in the Link Packets traffic view.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Verify that RequestAppLaunch is only sent in response to direct user action, such as pushing a button on the accessory or connecting the accessory to the Apple device via Lightning or via Bluetooth. (Note that this does not apply to automotive head units.)

29.4.3 External Accessory Protocol over Native

1. Verify StartExternalAccessoryProtocol and StopExternalAccessoryProtocol are not listed as declared messages in IdentificationInformation (unless the accessory also uses External Accessory Protocol over iAP2.
2. In IdentificationInformation, verify that there is at least one SupportedExternalAccessoryProtocol group.
3. Accessory must also declare a NativeTransportComponentIdentifier in the IdentificationInformation > SupportedExternalAccessoryProtocol group (parameter ID 10).
4. Each NativeTransportComponentIdentifier must correspond to a declared USBHostTransportComponent group (and the TransportComponentIdentifier must match).
5. Verify that the External Accessory Protocol interface name in the USB descriptor (in the USB Transfers traffic view) matches the protocol name sent in IdentificationInformation.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

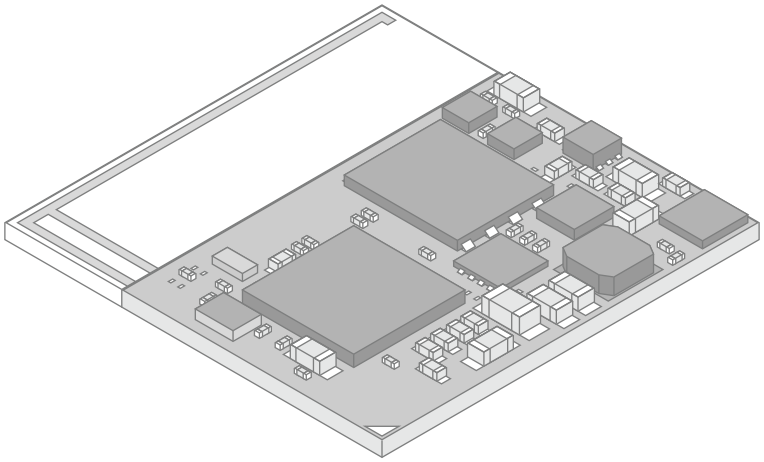
30. Game Controller Module

Accessories may make use of the Game Controller Module to create game controllers for Apple devices and Macintosh computers that work with all apps that make use of the Game Controller framework.

The Game Controller Module supports the following features:

- HID Game Controller (page 591)
- Bluetooth (page 344)
- HID Native Transport (Bluetooth), see HID (Human Interface Device) (page 580)
- USB 2.0
- Apple Lightning Receptacle (page 186)
- App Match (page 340)
- External Accessory Protocol (page 535)

Figure 30-1 Game Controller Module



The Game Controller Module includes the following components:

- Apple Authentication Coprocessor 2.0C (page 66)
- Apple Lightning Receptacle Controller (page 208)
- Battery charger

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

30.1 Overview

Key accessory-facing interfaces of the Game Controller Module are:

- Digital inputs for:
  - Menu Button
  - Hold Switch
  - Bluetooth Pairing Button
- Analog inputs for:
  - Joysticks
  - Face Buttons
  - L1/R1 Shoulder Buttons
  - L2/R2 Shoulder Buttons
  - Directional Pad
- 4x LED drivers
- Battery charger (and thermistor input)
- Factory configuration interface

Additional specifications and certain software for the Game Controller Module are available under license from Qualcomm Technologies International, Ltd. (formerly known as Cambridge Silicon Radio Limited), "QTIL", and can be requested by emailing `requestmfi@qti.qualcomm.com` with the following information:

- Email subject should contain "Software for Game Controller Module"
- Requesting Company's Full Name
- Requesting Company's Address
- Apple MFi License Number (6 digits)
- MFi Primary Contact (name, email address, phone number)
- CSRSupport ID or e-mail addresses for users requiring access

30.2 Firmware Updates

Game Controller Module accessories must be capable of field-deployable firmware updates from Apple devices running iOS and optionally Macintosh computers running OS X. The module integrator must include firmware update functionality within their own branded iOS applications.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The module supports firmware update via Bluetooth from iOS using the External Accessory Protocol.

30.3 Authentication

Accessories that integrate the Game Controller Module do not need to integrate a separate Apple Authentication Coprocessor.

30.4 Bluetooth

Accessories that integrate the Game Controller Module must connect to the Apple device using Bluetooth (see Bluetooth (page 344)).

30.5 Lightning Receptacle

Accessories that integrate the Game Controller Module may integrate a Lightning Receptacle (see Apple Lightning Receptacle (page 186)) to:

- Draw power from Apple branded or MFi certified USB power adapters.
- Draw power from USB Dedicated Charging Ports and USB hosts, such as a Mac.
- Enable USB connection to a Mac.

Accessories that integrate the Game Controller Module do not need to integrate a separate Lightning Receptacle Controller.

30.6 Mechanical

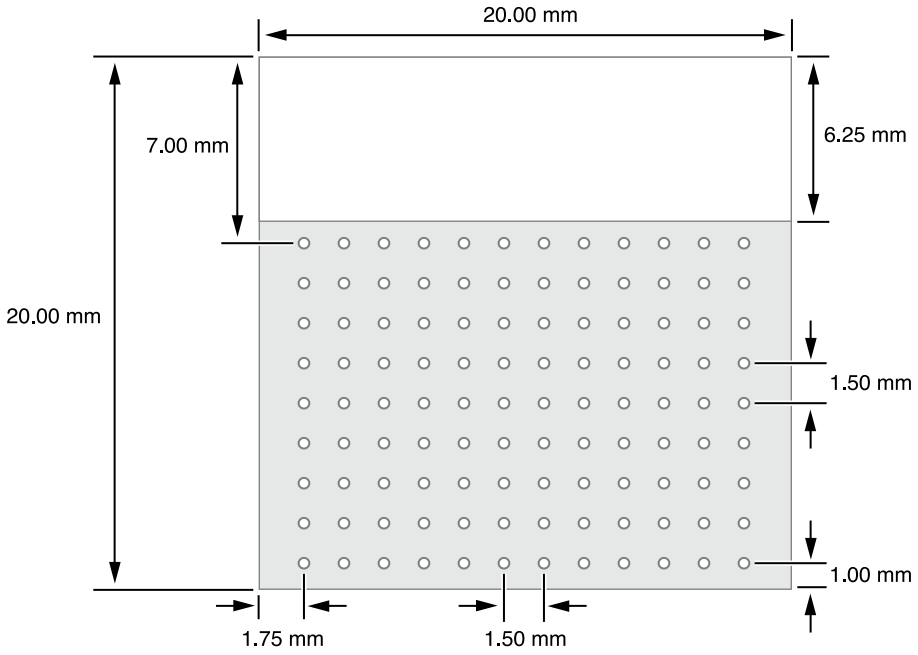
The Game Controller Module has the following mechanical characteristics:

- 20 mm x 20 mm LGA package
- Pad diameter of 0.75 mm
- Pad spacing of 1.5 mm
- Pickup is offset to the CSP surface
- A1 pad has a fiducial mark that may be used as a reference
- Designed to withstand a standard reflow profile with a peak reflow temperature below 250 °C for 20 seconds maximum

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- 0-70 ºC working temperature range

Figure 30-2 Game Controller Module Dimensions



For the best antenna performance, the module must protrude 6.25 mm beyond the edge of the PCB that it is mounted on. The antenna pattern must have a clearance of 2 mm in all directions. The host PCB must have a ground plane below the module which extends along its edge for 15 mm to either side of the module. The antenna should be placed at the front edge of the controller (pointing away from the user) and should not be placed in close proximity to a user's hands during normal use. Avoid metallic objects in the area surrounding the antenna.

30.7 Electrical

The module includes the following:

- Microcontroller
- Bluetooth radio
- Bluetooth antenna
- Apple Authentication Coprocessor 2.0C (see Apple Authentication Coprocessor 2.0C (page 66))
- Apple Lightning Receptacle Controller (see Apple Lightning Receptacle Controller (page 208))
- Power supply

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- 1.0 A or greater battery charger (see Battery (page 547))
- Storage for configuration options and to support firmware update (Flash/EEPROM/etc.)

30.7.1 Battery

The module supports rechargeable lithium-ion and lithium-ion polymer batteries.

At a minimum, the following batteries are supported:

- Lishen LP63450R - 1300 mAh, 3.7 V, 2.0 C, 6.5 mm x 33.6 mm x 49.5 mm
- Lishen LP103450R - 1900 mAh, 3.7 V, 2.0 C, 11 mm x 34 mm x 49.8 mm
- Panasonic UR18650A - 2250 mAh, 3.6 V, 0.5-0.7 C, cylindrical 18650 (18 mm x 65 mm)

30.7.2 I/O

The module supports the following outputs:

- 4x LED drivers

The LED drivers support variable brightness and animation sequences in accordance with game controller specifications in LED Array (page 603).

The module supports the following inputs:

- Digital Switches
  - Menu Button
  - Hold Switch
  - Bluetooth Pairing Button
- Pressure Sensitive Switches
  - Face Buttons
  - L1/R1 Shoulder Buttons
  - Directional Pad
- Position Encoders or Pressure Sensitive Switches
  - L2/R2 Shoulder Buttons

The module supports device reset and clearing Bluetooth pairings using a combination of the menu button and the Bluetooth pairing button, see Bluetooth Button (page 606).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

30.7.3 Pressure Sensitive Switches and Position Encoders

The module supports pressure sensitive switches and position encoders, see Switches and Position Encoders (page 597).

30.7.4 Joysticks

The module supports all two-axis analog controls listed in Joysticks (page 602).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

31. Headsets

Accessories may implement headset components. Apple devices treat accessory headsets differently from accessories with integrated speakers. Headsets are expected to help provide the user with a personal audio experience.

> **Note:** All accessories must comply with TDMA noise requirements. See TDMA Noise (page 65).

31.1 Product Design

The following product design requirements apply to accessory headset components:

- They must have drivers that are positioned on, over, or in a user's ears.
- They must have an integrated microphone, and it must be positioned to primarily record audio from the user's mouth.

31.2 Audio and Data Interfaces

Headsets must establish audio connections to Apple devices via one of the following interfaces:

- Headset jack (see Headset Plug (3.5 mm) (page 555))
- Lightning connector (see Apple Lightning Connector (page 128))
- Bluetooth HFP and A2DP (see Bluetooth (page 344))

If connected via the headset jack, headsets must support the Headset Remote and Mic feature (see Headset Remote and Mic (3.5 mm) (page 559).

If connected via the Lightning connector, headsets must integrate a Lightning Audio Module (see Apple Lightning Audio Module (page 93)).

If connected via Bluetooth, the headsets must support iAP2 and identify themselves as headsets to the Apple device by declaring the presence of a HID Headset Remote component (see Accessory Identification (page 265).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

31.3 Remote Controls

Accessory headsets that connect to an Apple device via the headset jack, Lightning connector, or Bluetooth must implement a 3-button array that mimics the behavior of the Apple Headset Remote. The buttons must be implemented as one of the following:

- Direct electrical connections to the Apple headset remote transmitter chip.
- Direct electrical connections to the Lightning Audio Module S0-S2 inputs as specified in S0-S2 Headset Remote Inputs (page 108).
- Part of a HID Headset Remote component via the Lightning Audio Module.
- Part of a HID Headset Remote component via Bluetooth (HID over iAP2).

If the headset connects to the Apple device via the Lightning connector or Bluetooth, additional remote control buttons/control surfaces may be implemented to control media playback and interact with Apple features like iTunes Radio.

31.4 Native Voice Recognition

If the accessory has native voice recognition capability of its own separate from the Siri feature, it must not be triggered via means like 'always listening' or buttons/switches that toggle between native voice recognition and Siri. The native voice recognition capability must be triggered via one of the following two approaches:

- The accessory may implement a physical button or control that is different/separate from the physical button or control that triggers Siri.
- The accessory may use the same physical button or control, but the user must trigger the native capability via a 'press-and-release' action, as opposed to the 'press-and-hold' action associated with Siri.

31.5 Extension Cables and Adapters

Accessories must not bundle headset jack extension cables or adapters unless:

- The accessory is form-fitting as defined in Form-Fitting Accessories (page 167).
- The accessory design prevents the Apple EarPods with Remote and Mic from being connected.
- The adapter is a double mono (airline) adapter. See Figure 31-1 (page 551).
- The adapter is a 3.5 mm to 6.35 mm (1/4 in) adapter. See Figure 31-2 (page 551).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 31-1 Double Mono Airline Adapter

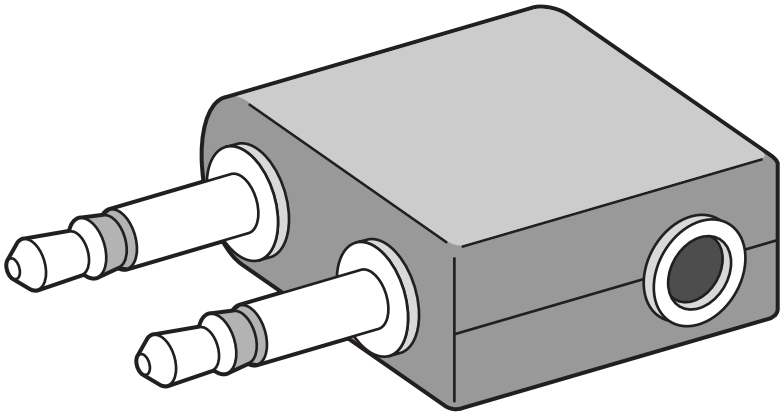
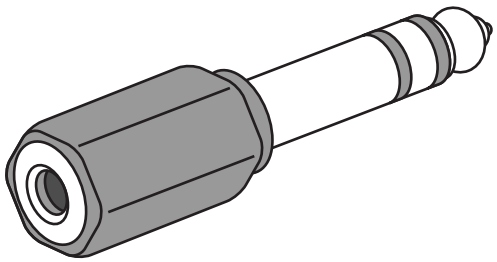


Figure 31-2 3.5 mm to 6.35 mm (1/4 in) Adapter



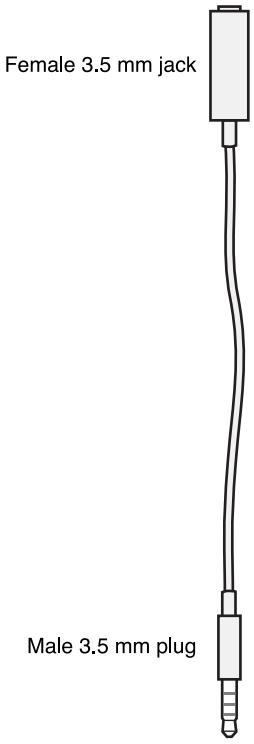
All bundled headset jack extension cables and adapters must:

- Incorporate 3.5 mm plugs and jacks with 4 conductors.
- Carry the 4 signals from the plug to the jack using separate conductors.
- Not support the Headset Remote and Mic feature as specified in Headset Remote and Mic (3.5 mm) (page 559).

Additionally, the headset jack extension cable or adapter should be as short as possible.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 31-3 Example Headset Extension Cable



31.6 Test Procedures

31.6.1 Product Design

1. Verify that the headset drivers are positioned on, over, or in the user's ears.
2. Verify that the headset microphone is positioned to primarily record audio from the user's mouth.

31.6.2 Headset Jack

If the headset connects to the Apple device via the headset jack:

1. Execute Headset Jack Test Procedures (page 558).
2. Execute Headset Remote and Mic Test Procedures (page 577).

31.6.3 Lightning

If the headset connects to the Apple device via the Lightning connector:

1. Verify that it integrates a Lightning Audio Module.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Verify that it integrates a Lightning (C10E) connector.
3. Verify that it implements 3 buttons that mimic the behavior of the Apple Headset Remote.
4. Execute Lightning Audio Module Test Procedures (page 125).

31.6.4 Bluetooth

If the headset connects to the Apple device via Bluetooth:

1. Verify that it supports iAP2.
2. Verify that it implements 3 buttons that mimic the behavior of the Apple Headset Remote.
3. Verify that it identifies a HID Headset Remote component to the Apple device over iAP2.
4. Execute Bluetooth Test Procedures (page 358).

31.6.5 Controls

For all headsets:

1. Verify that there is a button for Volume Up and that it increases the volume on the Apple device.
2. Verify that there is a button for Volume Down and that it decreases the volume on the Apple device.
3. Verify that there is a button that has the same Center button behavior as on the Apple Headset Remote.
    a. Verify Play and Pause works for songs and video tracks when pressing Center button.
    b. Verify Next Track works when pressing Center button twice quickly.
    c. Verify Fast Forward works when pressing then press-and-holding the Center button.
    d. Verify Previous Track works when pressing Center button 3 times quickly.
    e. Verify that pressing Center button once will answer incoming phone call.
    f. Verify that pressing Center button once will end phone call.
    g. Verify that holding down Center button will decline an incoming phone call. Verify you hear 2 low beeps to confirm that you declined the call.
    h. After the Apple device hibernates, launch Siri through pressing and holding either home button or headset Center button. Verify that the Siri double beep prompt is heard in full.
    i. After the Apple device hibernates, send an iMessage to the hibernating device to trigger notification. Confirm notification sound is heard in full.

If the Volume Up and Volume Down buttons are implemented as part of a HID Headset Remote component:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1. Verify that pressing the Volume Up button results in a button press usage immediately followed by a button released usage even if the button is pressed and held. It must not be possible to ramp the Apple device to full volume by pressing and holding the button.
2. Verify that pressing the Volume Down button results in a button press usage immediately followed by a button released usage even if the button is pressed and held. It must not be possible to ramp the Apple device to zero volume (mute) by pressing and holding the button.

If the headsets have additional controls/buttons:

1. Verify that each additional control has only one function associated with it.
2. Verify that each additional control has appropriate iconography/labeling for the associated function.
3. Verify that each associated function only triggers due to direct user action.

31.6.6 Native Voice Recognition

If the headset has its own native voice recognition capability separate from Siri:

1. Verify that it does not 'always listen' and trigger either the native voice recognition or Siri
2. Verify that it does not toggle between native voice recognition and Siri via buttons or switches
3. Verify that it has its own separate physical button or control that activates native voice recognition or triggers the native voice recognition via a 'press-and-release' action, while 'press-and-hold' triggers Siri.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

32. Headset Plug (3.5 mm)

Each Apple device contains a standard stereo headset jack supporting microphone input and control buttons. Most dimensions of the headset jack conform to the JEITA standard RC 5325A, *4-Pole Miniature Concentric Plugs and Jacks*.

Figure 32-1 Headset Plug Dimensional and Flange Details

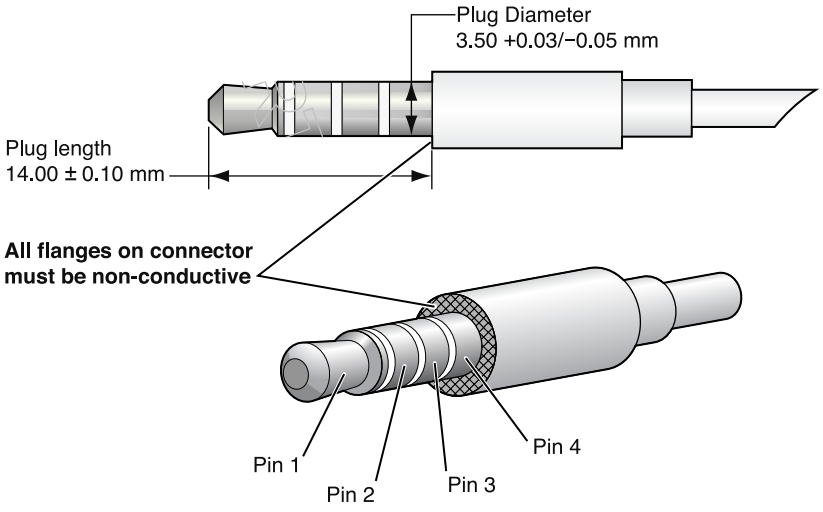
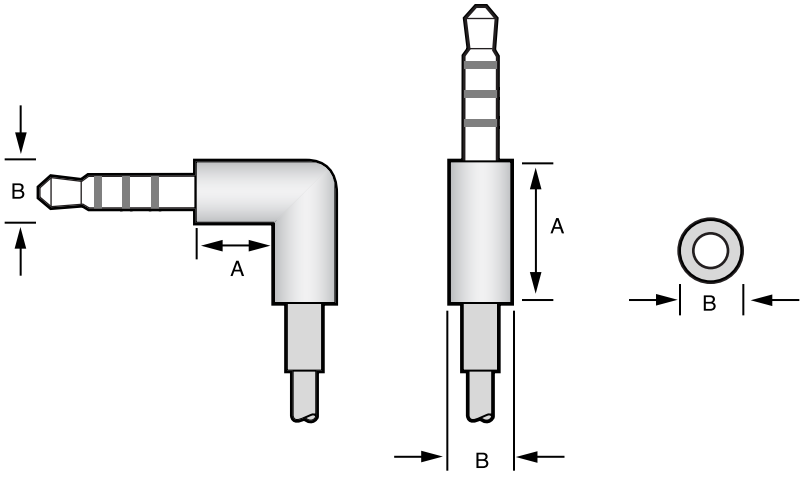


Figure 32-2 Headset Plug Cable Shell Dimensions



The following requirements apply to all accessory connectors intended for insertion into an Apple device's headset jack:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- The connector plug length and diameter dimensions must match those specified in Figure 32-1 (page 555).
- The connector's cable shell diameter must not exceed (B) 5.9 mm for a minimum length of (A) 11 mm from where the flange starts as shown in Figure 32-2 (page 555).
- The connector's cable shell must not have any bends, curves, or protrusions for a minimum length of (A) 11 mm from where the flange starts as shown in Figure 32-2 (page 555).

Additionally, every accessory connector with 4 conductors intended for insertion into an Apple device's headset jack must not have a conductive flange as specified in Figure 32-1 (page 555).

Any accessory headset jack that can accept the Apple EarPods with Remote and Mic and is electrically connected to the headset jack on an Apple device must support full functionality of the Apple Earphones with Remote and Mic.

32.1 Pin Assignments

The following table lists pin assignments for the headset plug, see Figure 32-1 (page 555):

Table 32-1 Headset Plug Pin Assignments

Pin	Description
1	Audio Output (Left)
2	Audio Output (Right)
3	Signal Return
4	Microphone Input

Note: Some Apple devices will apply and then measure a voltage across Pin 3 and Pin 4 in order to determine whether the headset is wired as specified in Table 32-1 (page 556) or not. The measured voltage must be greater than 1.5 V.

Some Apple devices will also apply a reverse bias voltage across Pin 3 and Pin 4 and then measure the response for the same purpose. The measured voltage must not be less than -0.75 V.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

32.2 Example Circuitry

The following figure is an example of circuitry in a headset accessory with an action button.

Figure 32-3 Typical circuitry in a headset accessory

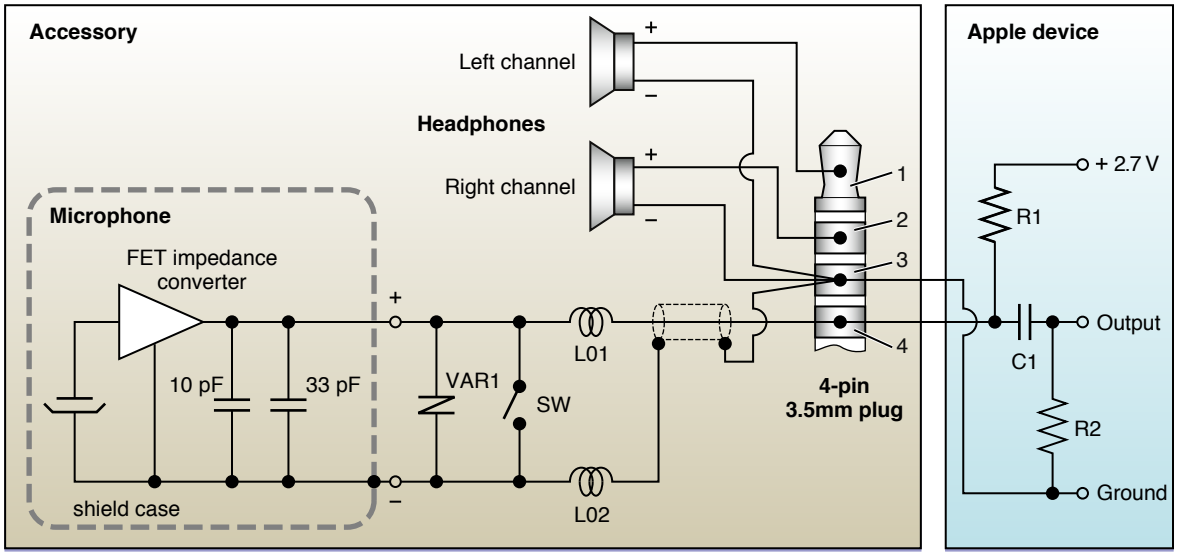


Table 32-2 Recommended component values for typical circuitry in a headset accessory

Component	Value
C1	1 µF
R1	2.21 kΩ ±1%
R2	Codec input impedance ≥ 2 kΩ
L01/L02	600 Ω at 100 MHz
SW	Action Button
VAR1	12 V varistor

Note: The value given for L01 and L02 is typical. These ferrite chokes reduce time division multiple access (TDMA) noise; their exact values depend on the specific accessory design.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Switch SW shorts the microphone signal to ground. The Apple device treats its closure as a headset button press and initiates a context-specific action (for example, answering a phone call). The microphone bias current must be between 210 and 500 µA, measured into a circuit pulled up to 2.7 V through 2.21 kΩ , to ensure button press detection. The recommended microphone sensitivity is -44 dBV with a maximum impedance of 2.2 kΩ at the output of the equivalent circuit shown above, measured under test conditions of Vs = 2.0 V, RL = 2.2 kΩ , Ta = 20 °C, and relative humidity = 65%.

32.3 Test Procedures

1. Verify that every accessory cable and every connector that can physically be inserted into an Apple device's headset/mic jack has non-conductive flanges.
2. For headsets with removable cables, ensure that end of the cable closest to the microphone is inserted into the headset and not the Apple device. Perform subsequent tests on the Apple device end of the cable.
3. Verify that the 3.5 mm shell diameter does not exceed 5.9 mm for a minimum length of 11 mm from where the flange starts.
4. Verify that the 3.5 mm plug length is 14.00 mm (+,- 0.10 mm).
5. Verify that the 3.5 mm plug diameter is 3.50 mm (+ 0.03 / - 0.05 mm).
6. Verify 3.5 mm plug can be fully inserted into the Apple device's headset jack. There must not be physical defects in the 3.5 mm plug preventing insertion.
7. Verify that there are no grounding issues, heavy static, audio interference, or audio interference when headset is attached to Apple device.
    a. Repeatedly plug the headset in and out of the headset jack.
    b. Wiggle the connector.
    c. Touch the input connector to the rim of the headset.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

33. Headset Remote and Mic (3.5 mm)

Apple devices can receive button press information from accessory headsets that incorporate the headset remote and mic feature via the Apple device's 3.5 mm headset jack (see Headset Plug (3.5 mm) (page 555)).

33.1 Headset Remote and Mic Requirements

All accessories implementing the Headset Remote and Mic feature must comply with the following requirements:

- The remote microphone must be located 120-160 mm from the center of a headset driver.
- There must be 6 wires originating from the 3.5 mm audio plug that attaches the Apple device to the headset, corresponding to the following signals:
  - Right Driver
  - Right Return
  - Left Driver
  - Left Return
  - Microphone Bias
  - Microphone Return
- All signals must be run separately to their respective components.
- There must be 3 physical remote buttons that perform the Volume Up, Volume Down, and Center button functions.

Accessory headsets that support the Headset Remote and Mic feature must integrate the following components:

- The Apple-provided transmitter chip from Table 33-1 (page 564).
- A MEMS digital microphone from Table 33-10 (page 577).

Accessory headsets must comply with the following requirements:

- The drivers must have a minimum load impedance of 16 ohms.
- The drivers must have a maximum load capacitance of 150 pF.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Accessory headsets must also implement one of the following configurations with one exception. The microphone may be located on either the left or right side of the headset.

Figure 33-1 Headset Remote and Mic Example A

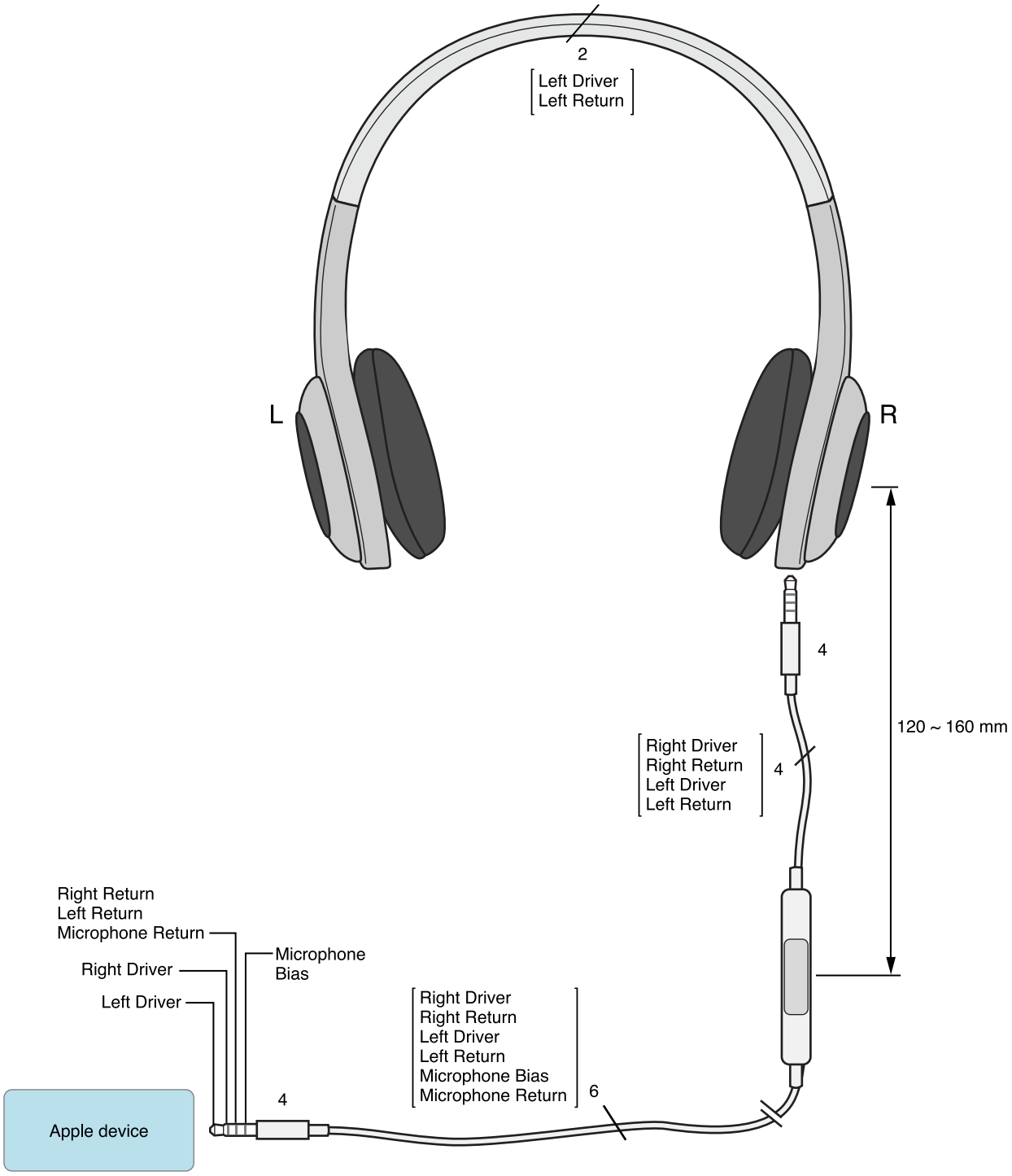


Figure 33-2 Headset Remote and Mic Example B

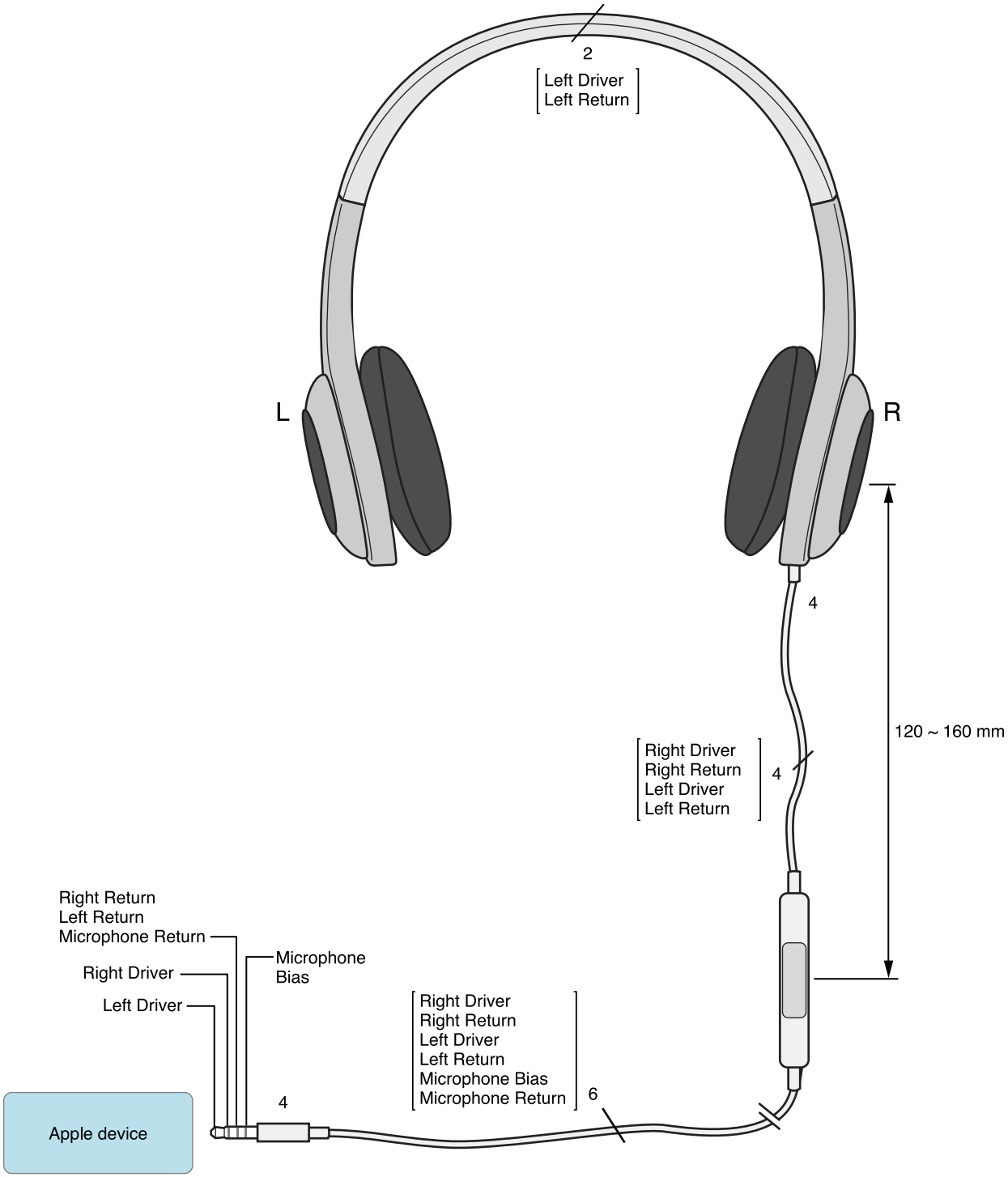


Figure 33-3 Headset Remote and Mic Example C

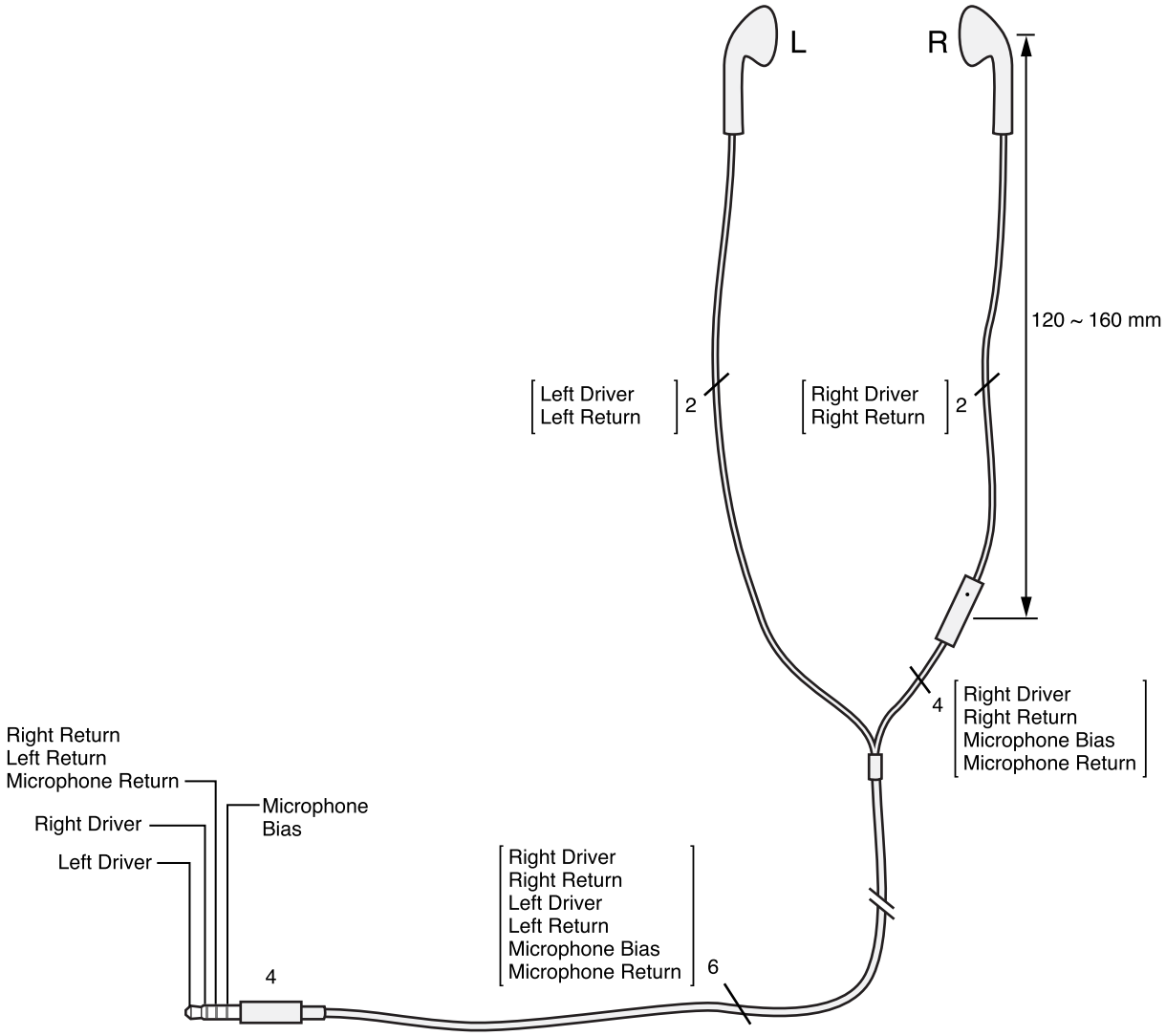
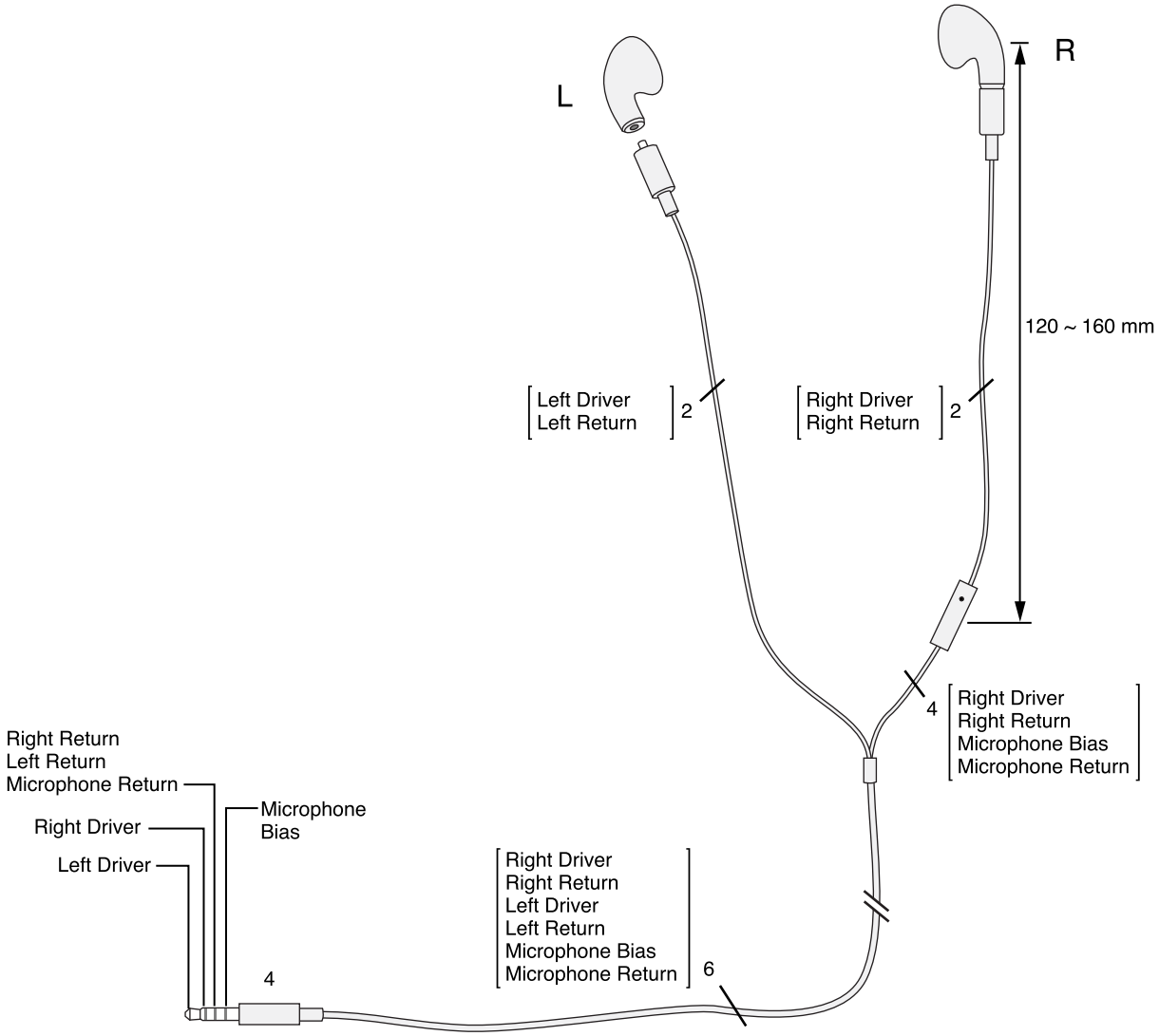


Figure 33-4 Headset Remote and Mic Example D



33.2 Headset Remote Transmitter Chip Usage

The transmitter chip operates together with a controller in the Apple device to enable remote button press detection via their common microphone bias line. The transmitter chip is a MEMS microphone interface and button decoder device located at the microphone and button end of the line, in the headset accessory. The controller in the Apple device provides regulated downstream power (nominally 2.7 V or 2.0 V) to the transmitter chip and MEMS microphone through the microphone bias line, and it decodes the button information from the transmitter chip.

The transmitter sends button-press information over the microphone bias line in either of two modes: button mode or tone mode. If the voltage on the microphone bias line is less than 2.35 V, indicating that the microphone is not in use, the transmitter enters button mode and sends button-press information as discrete voltage levels.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

If the microphone bias voltage is greater than 2.35 V, indicating that the microphone is in use, the transmitter enters tone mode and sends the same button-press information as ultrasonic tone sequences in the range of 99 to 300 kHZ.

Subjective listening tests with the newest Apple device are recommended to determine which chip produces the best user experience.

Table 33-1 Transmitter Chip Part Numbers

Part Number	Usage
MFI353S2429	Noise-occluding headsets. These are headsets that block or cancel outside sound.
MFI353S2430	Standard non-occluding headsets.

33.2.1 Transmitter Chip Pin Assignments and Physical Packaging

Table 33-2 Transmitter Chip Pin Assignments

Number	Name	I/O	Description
A1	TONE	Output	Tone generator output
A2	GND	Power	Audio return
B1	MIC	Input	Microphone bias
B2	REM	Input/Output	Remote switch network
C1	VSHUNT	Input	Shunt regulator supply
C2	MICPWR	Output	Microphone power

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 33-5 Transmitter Chip Package

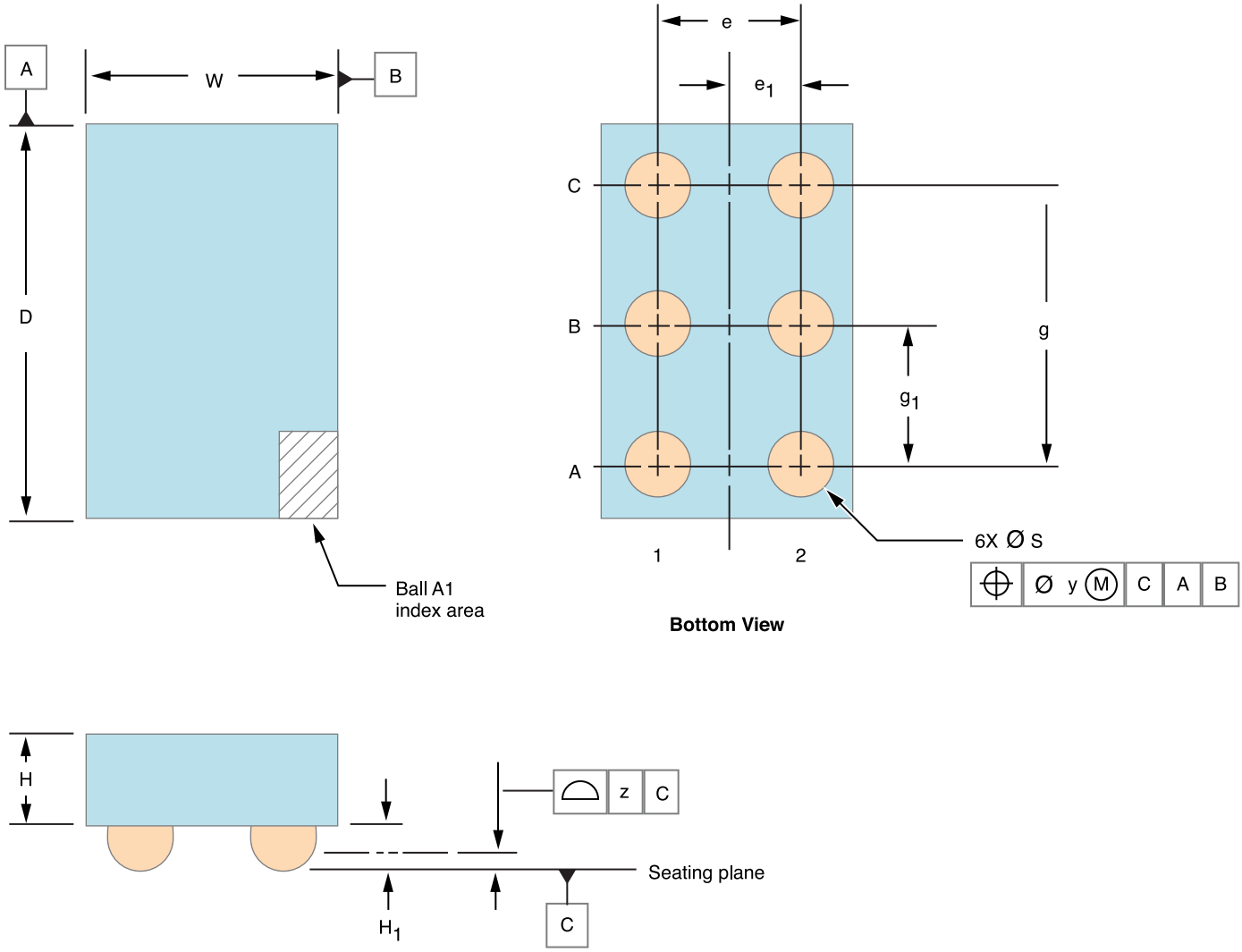


Table 33-3 Transmitter Chip Package Dimensions

Dimension	Value in mm
W	0.95/0.85
D	1.45/1.35
H	0.50 max
H1	0.19/0.15
e	0.50
e1	0.25

Dimension	Value in mm
g	1.00
g1	0.50
s	0.25/0.21
y	0.015
z	0.05

33.2.2 Transmitter Chip Maximum Voltage and Current Ratings

Table 33-4 (page 566) lists the transmitter chip's maximum voltage and current ratings while operating over a free-air temperature range (TA) of -40 to +85 °C. The chip's maximum storage temperature range is -65 to +150 °C. Stresses beyond these ratings may cause permanent damage, and exposure to maximum conditions for extended periods may degrade its reliability. These are stress ratings only; functional operation of the chip at these or any other conditions beyond those specified is not implied.

All voltages are measured with respect to ground. All input and output clamp-current ratings must be observed.

Table 33-4 Transmitter Chip Maximum Voltage and Current Ratings

Symbol	Description	Minimum	Maximum
VSUPPLY	Supply voltage, VSHUNT, MIC	-0.5 V	4.6 V
VI	Input voltage, REM	-0.5 V	4.6 V
V0	Output voltage, MICPWR, TONE	-0.5 V	4.6 V
IIK	Input clamp current, REM (VI < 0)	-20 mA	
I0K	Output clamp current, MICPWR, TONE (VO < 0)	-20 mA	
ISUPPLY, IGND	Continuous current through VSHUNT, MIC, or GND	-50 mA	50 mA

33.2.3 Transmitter Chip Thermal Impedance

The transmitter chip's package thermal impedance, calculated in accordance with Specification JESD51-7, is 123 °C/W.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

33.2.4 Transmitter Chip Moisture Sensitivity

The transmitter chip has a Moisture Sensitivity Level (MSL) of 1, as defined by industry standard JEDEC specifications.

33.2.5 Transmitter Chip Electrical Characteristics

Table 33-5 (page 567), Table 33-6 (page 568), and Table 33-7 (page 569) list the transmitter chip's electrical and timing characteristics under the following conditions:

- Operating temperature = -40 to +85 °C.
- Button mode, VMICBIAS = 1.8 to 2.1 V; MIC is connected to VMICBIAS through a 2.21 kΩ ±1% resistor.
- Tone mode, VMICBIAS = 2.56 to 2.84 V; MIC is connected to VMICBIAS through a 2.21 kΩ ±1% resistor.

The values in the "Typical" column of the tables are measured at 25 °C.

Table 33-5 Transmitter Chip Electrical Characteristics (General)

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
$I_{MICBIAS-B}$	Quiescent Current into MIC+VSHUNT	Button Mode, $V_{MICBIAS}$ = 2.1 V		3 µA	6 µA
$I_{MICBIAS-B}$	Quiescent Current into MIC+VSHUNT	Button Mode, $V_{MICBIAS}$ = 1.5 V		3 µA	6 µA
I_{MIC-T}	Quiescent Current into MIC	Tone Mode		34 µA	46 µA
$I_{VSHUNT-T}$	Quiescent Current into VSHUNT	Tone Mode (see note below)		60 µA	70 µA
I_{MIC-TA}	Active Current into MIC	Tone Mode		35 µA	45 µA
$I_{VSHUNT-TA}$	Active Current into VSHUNT	(see note below)		104 µA	118 µA
V_{TR}	Tone Mode Threshold Voltage	MIC Rising (Microphone enable), V_{MICPWR} = 1.0 V	2.20 V	2.35 V	2.50 V

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
V_{TF}	Tone Mode Threshold Voltage	MIC Falling (Microphone disable), V_{MICPWR} = 400 mV	0.55 V	0.8 V	1 V
V_{MICPWR}	MICPWR Output Voltage	I_{MICPWR} = 120--150 µA	1.51 V	1.56 V	1.61 V
R_{SO}	Shunt Regulator Output Impedance	Freq = 100 Hz	5 Ω	18 Ω	25 Ω
R_{SO}	Shunt Regulator Output Impedance	Freq = 20 Hz	12 Ω	21 Ω	35 Ω
R_{ONA}	Switch A, R_{DSON}	Tone Mode, I_{MICPWR} = 1 mA, $V_{MICBIAS}$ = 2.56V		40 Ω	55 Ω
R_{ONB}	Switch B, R_{DSON}	V_{MIC} = 1.2V, I_{REM} = 1 mA		22 Ω	30.5 Ω

Note: This current is pulled through R_{VSHUNT} between MIC and VSHUNT and is the minimum current to keep VSHUNT regulated at 1.56 V. Excess current through R_{VSHUNT} is available to the load at MICPWR. Excess current not used by the load at MICPWR is internally shunted to GND.

Table 33-6 Transmitter Chip Electrical Characteristics (Tone Mode)

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
$e_{n\text{-}mic100}$	MIC Integrated Noise	100 Hz to 20 kHz		1.5 µVrms	2 µVrms
f_{TONE1}	Button 1 Frequency	R_{REM} = 6.81 kΩ	109 kHz	130 kHz	159 kHz
f_{TONE2}	Button 2 Frequency	R_{REM} = 9.42 kΩ	138 kHz	165 kHz	200 kHz
f_{REL}	Button Release Frequency		81 kHz	97 kHz	117 kHz

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
R_{BT1}	Button 1 Boundary		6.61 kΩ	6.81 kΩ	7.01 kΩ
R_{BT2}	Button 2 Boundary		9.33 kΩ	9.42 kΩ	9.51 kΩ
V_{TA}	Tone Amplitude	R_{TONE} = 1 MΩ	350 mV	550 mV	720 mV
V_{TA}	Tone Amplitude	R_{TONE} = 100 kΩ	300 mV	515 mV	710 mV

Table 33-7 Transmitter Chip Electrical Characteristics (Button Mode)

Symbol	Parameter	Test Conditions	Minimum	Typical	Maximum
t_{ONA}	Switch A Enable Time	Time to turn on Switch A	0.8 ms	1.2 ms	2 ms
t_{OFFB}	Switch B Disable Time	Time to turn off Switch B	0.7 ms	1 ms	2 ms
t_{REG}	Shunt regulator enable time	Time From MIC = 2.3 V to MICPWR = 1.56 V	1 ms	2.5 ms	3.5 ms

33.2.6 Transmitter Chip Theory of Operation

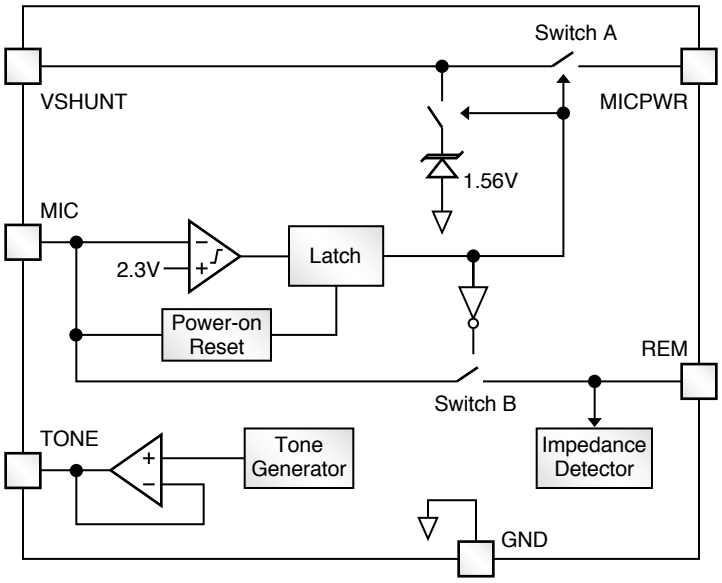
The transmitter chip has three primary functions:

- Provide an interface to a button switch-resistor network.
- Provide power for a local microphone.
- Provide a tone generator for sending discrete frequency tones on the bias line corresponding to button events.

The controller in the Apple device provides regulated downstream power (nominally 2.7 or 2.0 V) to the transmitter chip and microphone through the microphone bias line. Figure 33-6 (page 570) illustrates the functional components of the transmitter chip. In this diagram, a latch drives the configuration of switches A and B. The power-on reset monitors voltage on the MIC pin to ensure that there is a enough power before initiating the turn-on sequence; it shuts the chip down if there is insufficient voltage.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 33-6 Transmitter Chip Block Diagram



Button events are sent from the transmitter to the controller in one of two modes, button mode or tone mode. When a microphone is not present or is not in use, the transmitter is put in button mode by the controller in the Apple device, and button events are detected using discrete voltage levels. These discrete voltage levels are a percentage of a regulated output voltage on the microphone bias line. When a microphone is in use, the controller puts the transmitter into tone mode by placing more than 2.35 V on the microphone bias line, and the transmitter then sends button events using tone sequences of discrete frequencies in the range 99 kHz to 300 kHz.

33.2.7 Transmitter Chip Button Mode

In button mode, the transmitter chip operates as a pass-through element switching a button switch-resistor network onto the bias line. Each switch represents a unique button. When a button is pressed, the DC level on the bias line is changed and detected by the controller. Table 33-8 (page 570) shows the DETECT pin voltages with $V_{MICBIAS}$ = 2.0 V.

Table 33-8 Transmitter Chip DETECT Pin Voltages

Switch Closed	Voltage
S0	0.000 V ±1%
S1	1.510 V ±1%
S2	1.603 V ±1%

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

When the transmitter chip is in button mode (VMIC has never reached 2.35 V), it shorts the MIC and REM pins together and disables all other inputs and outputs. When a button event occurs, the DC voltage on the microphone bias line changes. Table 33-8 (page 570) shows the DC voltage corresponding to a given button press when using the R1 and R4 resistor values listed in Table 33-9 (page 576). This DC level is then detected by the controller in the Apple device. Switch S0 is a unique switch that shorts the V_{MIC} line to ground.

When the V_{MIC} line is shorted to ground, power is removed from the transmitter chip. When power recovers, the transmitter chip enters button mode or tone mode, depending on the voltage detected at the MIC pin.

33.2.8 Transmitter Chip Tone Mode

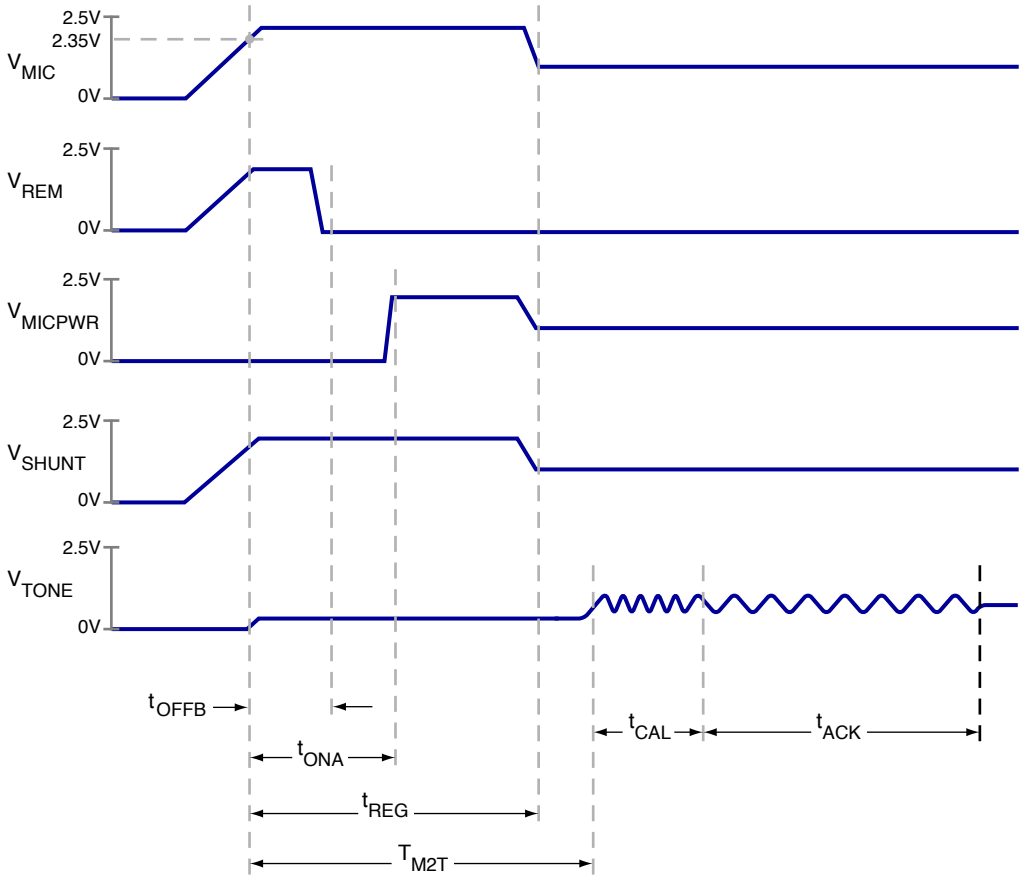
When the transmitter chip detects a voltage greater than 2.35 V at the MIC pin, it enters tone mode. With a microphone biased and in use, the switch-resistor network used for button mode would cause large DC level shifts in the bias voltage. Such shifts would result in unwanted audible clicks or pops or would cause de-biasing of the microphone. To prevent this problem, when the transmitter chip enters tone mode it disconnects the switch-resistor network from the microphone bias line, enables the microphone via the FET switch, and engages the tone generation circuit shown in Figure 33-6 (page 570).

In tone mode the transmitter chip has two functions. First, it turns on the MEMS microphone by forcing a FET switch to ground. Second, it detects button events and places a discrete tone sequence onto the microphone bias line. The tone frequencies in each sequence are unique to each button press. The controller detects the tones on the bias line and determines the corresponding button event.

The transmitter chip's startup timing when it enters tone mode is shown in Figure 33-7 (page 572). Values for the timing parameters are given in Table 33-6 (page 568).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 33-7 Transmitter Chip Startup Timing

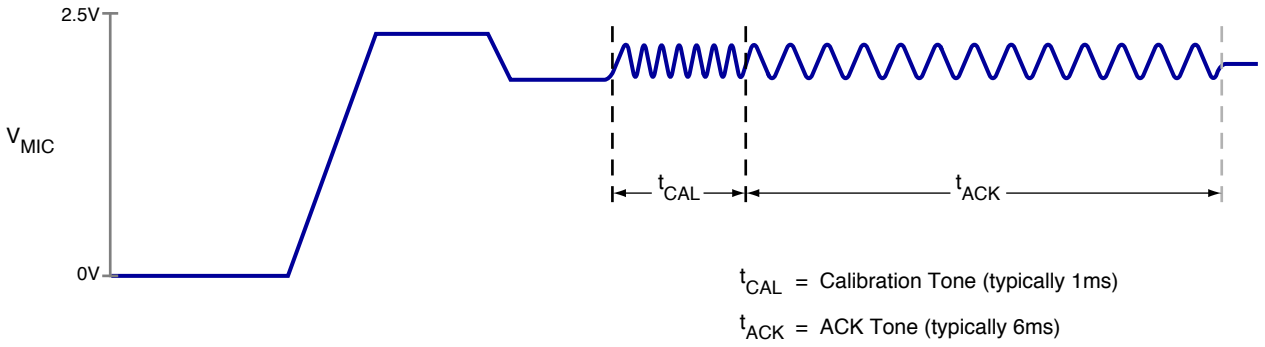


The tone mode startup sequence is as follows:

1. Upon detecting V_{MIC} > 2.35 V, the switch connecting the MIC and REM pins together is opened after time t_{OFFB} (see Figure 33-7 (page 572) and Table 33-6 (page 568)).
2. After a delay of t_{REG} after V_{MIC} > 2.35 V, the SHUNT pin and the MICPWR pins are shorted. The microphone is enabled by turning on the FET switch through the MICPWR pin.
3. Once the noise prevention process has settled, the transmitter chip sends a preset acknowledge (ACK) tone sequence.
4. The controller detects the ACK sequence (see Figure 33-8 (page 573)) and authenticates the presence of the transmitter chip.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

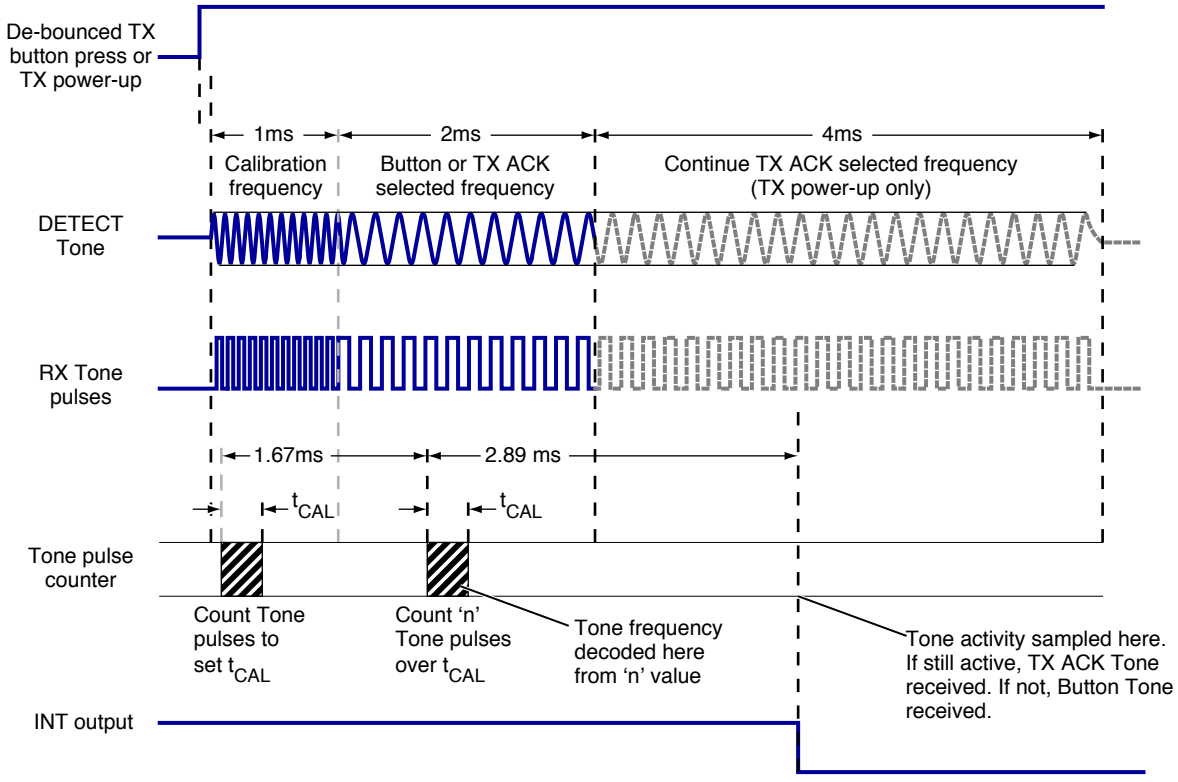
Figure 33-8 Tone Mode ACK Sequence



The tone generation circuit of the transmitter chip internally detects each button press and sends a high frequency tone sequence between 99 kHz and 300 kHz. The high frequency tone sequence is unique to each button. The controller detects the frequency of each tone and translates it into a predetermined button event. A button release has a different frequency than a button press.

For accuracy, the transmitter chip sends two tones for each button press as shown in Figure 33-9 (page 573). The first tone, lasting 1 ms, is a calibration frequency and the second, lasting 2 ms, is the unique frequency for the selected button. The ratio of these two frequencies is calculated and translated into button press information. This provides a very accurate result that is independent of clock frequency variation.

Figure 33-9 Tone Transmit/Decode Method



The transmitter chip remains in tone mode until the MIC pin is pulled below 0.8 V. When power recovers, the transmitter chip enters button mode or tone mode depending on the voltage detected at the MIC pin.

33.3 Button Detection Circuitry Usage

The circuits in the accessory that support these components must be those shown in Figure 33-10 (page 575) and Figure 33-11 (page 575). The nominal values of the components shown in these schematics are given in Table 33-9 (page 576).

These circuits are designed to produce a tone amplitude between the microphone bias line and the microphone return, at the end of a cable 1 meter long, of at least 30 mV peak-to-peak into a 2000-ohm load. If necessary, the value of R3 must be adjusted to achieve this result. Figure 33-11 (page 575) shows how a voltage on the Microphone Power line from the transmitter chip enables the MEMS microphone chip through Q1. It also shows components R7, C4, and R8, which control the microphone's frequency response. The equation that determines the values of these components is given in Button Detection Circuitry Adjustments (page 576).

Figure 33-10 (page 575) and Figure 33-11 (page 575) are two parts of one circuit. The two Microphone Return lines shown in these sub-circuits must be connected at the component locations. Their common return line and the return lines for each of the two drivers must then be routed separately through the cable that goes to the Apple device, being tied together only at the headset connector. This configuration is required to minimize crosstalk between the separate driver channels and the microphone.

Note: With the exception of the MEMS digital microphone listed in Table 33-9 (page 576), symbol U2, components of equal or better specifications may be substituted for the components called out below.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 33-10 Transmitter Circuit

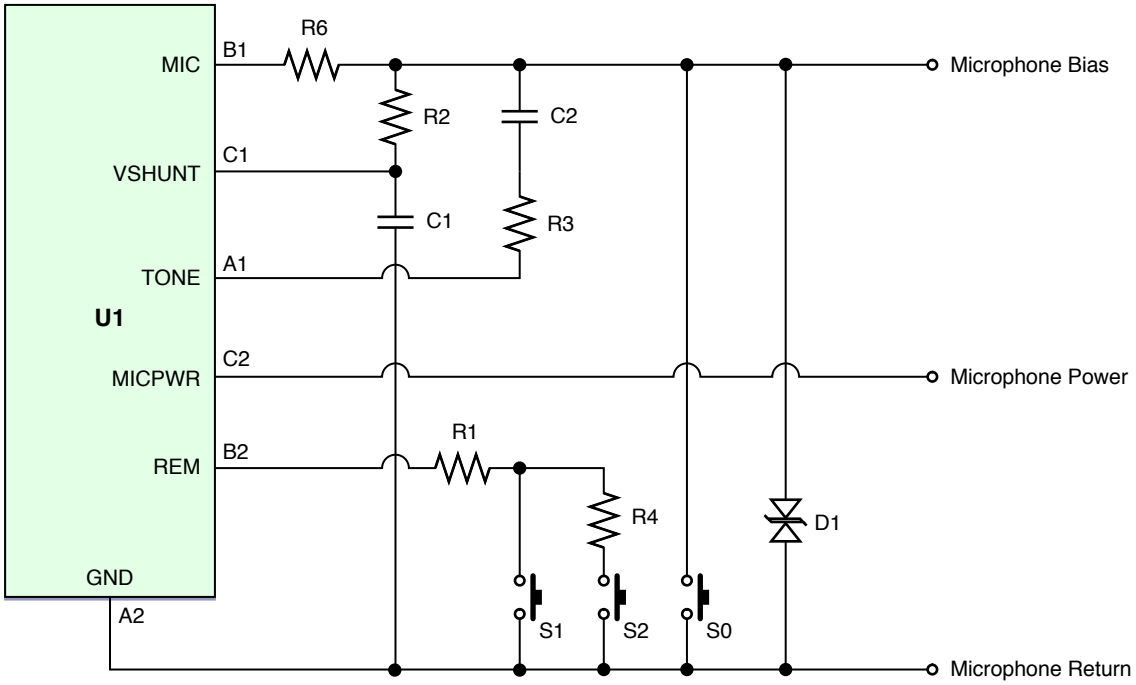


Figure 33-11 Microphone Circuit

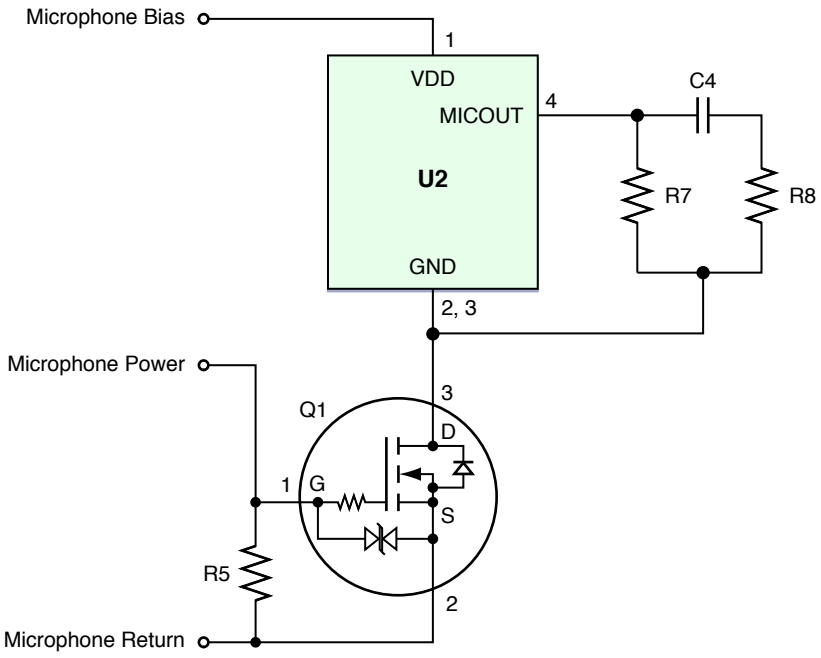


Table 33-9 Transmitter Circuit Components

Symbol	Description	Notes
C1	Capacitor, 0.1 µF ±10%, 6.3 V	
C2	Capacitor, 220 pF ±5%, 25 V	Ceramic
C4	Capacitor, 2.2 µF ±10%, 6.3 V	
D1	ESD protection diode, 5 pF, 6.1 V	ST Micro ESDALC6V1-1BU2; install as close to chip pin B1 as possible
Q1	MOS field-effect transistor	CEDM 7001
R1	Resistor, 6.81 kΩ ±0.5%, 1/20 W	
R2	Resistor, 2 kΩ ±1%, 1/20 W	
R3	Resistor, 1.2 kΩ ±0.5%, 1/20 W	
R4	Resistor, 2.61 kΩ ±0.5%, 1/20 W	
R5	Resistor, 887 kΩ ±1%, 1/20 W	
R6	Resistor, 49.9 Ω +0.2%/-1%, 1/20 W	Must not exceed 50 Ω .
R7	Resistor, 17.4 kΩ ±1%, 1/20 W	
R8	Resistor	see Table 33-10 (page 577)
S0	Dome switch	Center button; must not exceed 20Ω when closed.
S1	Dome switch	Volume down; must not exceed 20Ω when closed.
S2	Dome switch	Volume up; must not exceed 20Ω when closed.
U1	Headset interface transmitter chip	Provided by Apple
U2	MEMS digital microphone	see Table 33-10 (page 577)

33.3.1 Button Detection Circuitry Adjustments

The accessory must use one of the MEMS digital microphone components and its associated R8 resistor value specified in Table 33-10 (page 577).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 33-10 Approved transmitter circuit MEMS digital microphone components

MEM Digital Microphone (U2)	Resistor (R8)
Knowles SPQ2409HE5H-PB	1.2 kΩ ±1%, 1/20 W
Knowles SPY1824LR5H-B	1.6kΩ ±1%, 1/20 W
Goertek S15OB383-002	1.91kΩ ±1%, 1/20 W

The values of some of the components listed in Table 33-9 (page 576) may be adjusted to optimize the performance of the headset accessory, using these formulas:

- **High-pass filter corner frequency in Hertz** ≈ $1/(2\pi \cdot R_8 \cdot C_4)$, where R_8 is the value of resistor R8 in ohms and C_4 is the value of capacitor C4 in Farads. This formula assumes that the value of R7 is greater than the value of R8.
- **System sensitivity at 1 Pascal in Volts** = $(M_0/R_8) \cdot R_2$, where M_0 is the microphone sensitivity in Volts per Pascal, R_8 is the value of resistor R8 in ohms, and R_2 is the value of resistor R2 in ohms in parallel with 1.05 kΩ .
- **Maximum excursion of the microphone in Volts** = $(1/R_7) \cdot R_2$, where R_7 is the value of resistor R7 in ohms, and R_2 is the value of resistor R2 in ohms in parallel with 1.05 kΩ .

Note: If the microphone bias voltage drops below 1.6 V, the transmitter chip will begin to fail and the microphone chip may produce indeterminate outputs.

33.4 Test Procedures

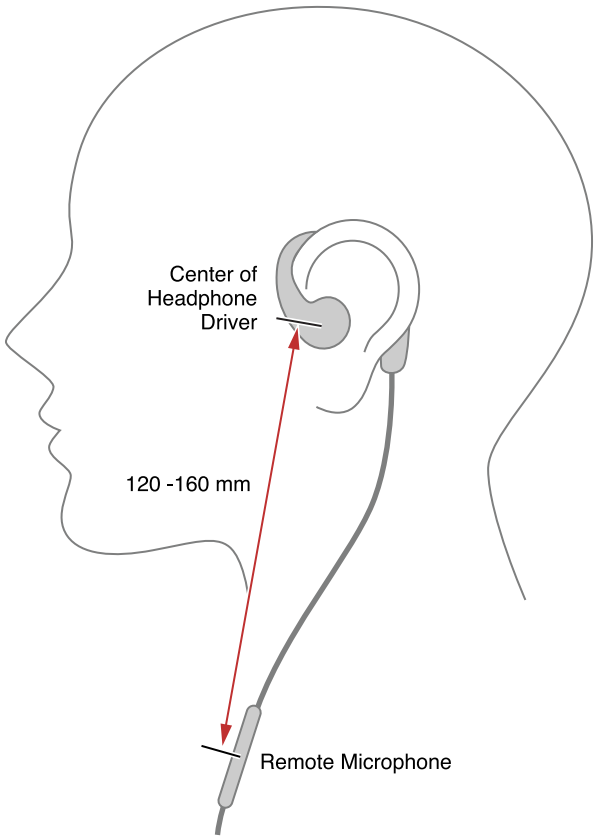
33.4.1 Product Design

1. Verify that the remote microphone is located 120-160 mm from the center of a headset driver.
2. See Test Procedures (page 552).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: If the headset cabe is intended to loop around the ear (see Figure 33-12 (page 578)), measure the distance as if the accessory is being worn.

Figure 33-12 Headsets Looped Around Ear Test Setup



33.4.2 Electrical

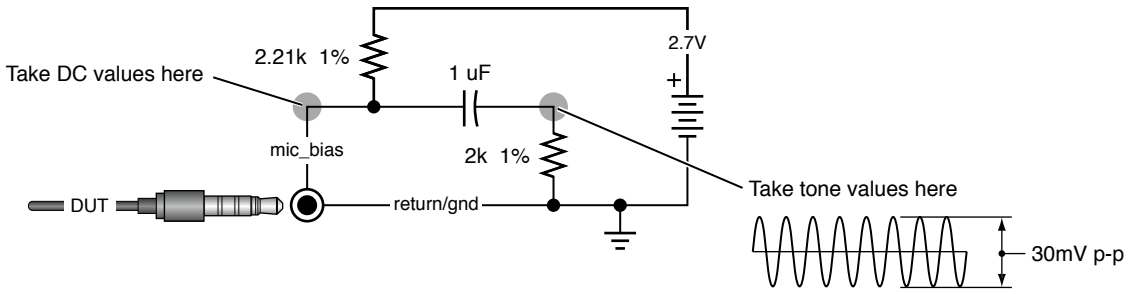
The following items are required:

- A precision power supply
- 4-pin headset jack, 3.5 mm diameter, such as Digikey SJ-43515RS-SMT
- An oscilloscope with measuring capabilities
- A 2.21 kΩ 1% resistor
- A 2 kΩ 1% resistor
- A 1 µF capacitor

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

These items must be connected as shown in Figure 33-13 (page 579)

Figure 33-13 Headset Remote and Mic Test Setup



Testing must demonstrate the following:

- All tones must have a peak-to-peak amplitude of at least 30 mV as measured across the 2 kΩ load.
- Labeled oscilloscope pictures must be generated for all button press tones, button release tones, and acknowledgment tones, showing their amplitudes. The button release frequency of 99 kHz must have an amplitude of at least 30 mV peak-to-peak.
- The mic_bias point static DC voltage must be between 1.85 and 2.0 V. A labeled oscilloscope picture must be generated showing that the voltage at the mic_bias point is stable and within specification within 50 ms after power-up.
- Set the power supply in the test circuit to 2.0 V to place the transmitter in button mode. An oscilloscope picture must be generated to demonstrate DC shifts occurring during button presses with static DC mic_bias voltages as shown in Table 33-11 (page 579).

Table 33-11 Headset Remote and Mic Expected mic_bias Voltages in Button Mode

Switch Closure	Voltage
S0	0.000 V ±1%
S1	1.510 V ±1%
S2	1.603 V ±1%

34. HID (Human Interface Device)

Some Apple devices can accept input from Human Interface Device (HID) accessories, such as external keyboards, and also send HID reports to those accessories. This capability is then made available system-wide for all apps on the device or certain features built into the device system software. If an accessory is designed to provide human input events to a specific third-party app, it should consider implementing the External Accessory Protocol feature instead.

The HID feature can be implemented over:

- the iAP2 protocol
- a HID Native transport (USB Host Mode)
- a HID Native transport (Bluetooth)
- the CarPlay communication protocol
- the Lightning Audio Module serial interface (see Apple Lightning Audio Module (page 93))

34.1 HID Requirements

Accessory connections to an Apple device via USB Host Mode or Bluetooth must implement the HID feature over a HID Native transport.

Accessories supporting the HID feature over iAP2 or over native transports must comply with the following requirements:

- The accessory must identify at least one Table 59-18 (page 813) or Table 59-23 (page 815) or Table 59-26 (page 816).
- Each identified HID component must have a unique `HIDComponentIdentifier`.
- Each identified HID component must declare one intended function.

Accessories supporting the HID feature over the CarPlay communication protocol must comply with the following requirements:

- User events must be represented as Human Interface Device (HID) usage reports and must be transported using the CarPlay communication protocol.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Accessories must not use iAP2 to transport HID events while using CarPlay, with one exception: accessories supporting HID Media Playback Remote may send those commands (except for Voice Control / Siri) over iAP2 (see HID Media Playback Remote (page 631)).
- When supporting CarPlay, the Voice Command / Siri usage must always be transported using the CarPlay communication protocol. To send HID events over CarPlay use the command requestSiri as described in requestSiri (page 482).

All accessories supporting the HID feature must comply with the following requirements:

- Unless otherwise specified, the accessory must be capable of generating and receiving all HID usages declared in the component's HID descriptor.
- Unless otherwise specified, the accessory's declared HID usages must map directly to physical or virtual control surfaces on a 1:1 basis. For example, a button labeled "Play/Pause" must send a Play/Pause HID usage and not "Play" or "Pause" usages.
- Unless otherwise specified, physical or virtual control surfaces that generate accessory HID usage reports must be labeled with appropriate iconography or text that corresponds to the resulting Apple device behavior. For example, a Play/Pause button must be labeled with the text 'Play/Pause' or a Play/Pause icon.
- Unless otherwise specified, one accessory HID usage report must be sent in response to each direct user action on the corresponding physical or virtual control surface. For example, when the user presses a button, one 'button pressed' usage report must be sent, and a separate 'button released' usage report must be sent when the user releases the button.
- Accessories must only send HID reports for changes in physical or virtual control surfaces registered in the HID descriptor.
- Each HID report must contain the correct number of bytes as described in its corresponding HID report descriptor.
- The accessory must not anticipate or assume corresponding state changes in the Apple device after sending HID usages. For further explanation, see Presentation of Apple Device Updates (page 61).
- Accessories must not send a HID report if there has not been any change in the state of the control surface(s). For example, the accessory must never generate a "Play/Pause" event without the user pressing a dedicated "Play/Pause" button.

34.1.1 HID Report Descriptor Requirements

- When padding packets to align within a byte boundary, each Main item tag (Input, Output, or Feature) must be marked constant. Padding bits should be set to 0.
- When defining Variable type Input/Output fields, the Report Count number must match the number of Usages specified.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

34.1.2 HID over iAP2 Requirements

If implementing HID over iAP2, an accessory must send or receive the following iAP2 control session messages:

- StartHID (page 844)
- AccessoryHIDReport (page 845)

An accessory may also send or receive the following iAP2 control session messages:

- DeviceHIDReport (page 845)
- StopHID (page 846)

34.1.3 HID Native Transport (USB Host Mode) Requirements

If implementing HID over USB Host Mode transport, the accessory must comply with the *Device Class Definition for Human Interface Devices* (available from the USB-IF). Additionally, the accessory must declare a USB Host Mode transport component (Table 59-23 (page 815)) during identification.

34.1.4 HID Native Transport (Bluetooth) Requirements

If implementing HID over Bluetooth native transport, the accessory:

- Must support Bluetooth HID 1.1.
- Must declare a Bluetooth HID component (Table 59-26 (page 816)) during identification.
- Must register to receive StartNativeHID (page 846).
- Must support sniff mode (see Sniff Mode for Low Power Consumption (page 344)).
- Should use the following parameters in SDP for sniff subrating:
  - HIDSSRHostMaxLatency - 450 ms (720 slots)
  - HIDSSRHostMinTimeout - 45 ms (72 slots)
- Should use a typical report packet of 22 bytes or less. This is small enough to fit into a DH1 packet with L2CAP and HID header.

If accessory identification (see Accessory Identification (page 265)) is rejected due to the Bluetooth HID component (Table 59-26 (page 816)), the accessory should attempt to identify as a HID over iAP2 accessory (see HID over iAP2 Requirements (page 582)). The accessory must not attempt to simultaneously support both HID over Bluetooth transport and HID over iAP2 over Bluetooth.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

34.2 HID Usage

34.2.1 HID over iAP2 Usage

To use HID over an iAP2 control session, the accessory must send a StartHID (page 844) message to the device. The `HIDComponentIdentifier` parameter must correspond to a previously identified Table 59-18 (page 813). Each Table 59-18 (page 813) must be registered with a separate StartHID (page 844) message. Similarly, to stop usage of a particular HID report descriptor, a separate StopHID (page 846) message must be sent for that component.

In between the StartHID (page 844) and StopHID (page 846) pairs for a particular component, the accessory may send one or more AccessoryHIDReport (page 845) messages to the device. Each AccessoryHIDReport (page 845) message must reference a specific HID component in addition to the HID report itself. Similarly, the Apple device may send one or more DeviceHIDReport (page 845) messages to the accessory with the same parameters.

34.2.2 Native Bluetooth HID Usage

> **Note:** This section is a Developer Preview and is not intended for use in the development of Proposed Products or Licensed Products under a MFi License. Although this content has been reviewed for accuracy, it is not final. Apple is supplying this content to help accessory developers plan for the adoption of the accessory interface features described herein. This information is subject to change, and accessories implemented according to this content must be tested with final operating system software and final documentation before going through self certification.

After receiving the StartNativeHID (page 846) message, an accessory may start HID on its Bluetooth native transport.

34.3 Test Procedures

34.3.1 iAP2

1. Verify that the accessory generates and receives all HID usages declared in the component's HID descriptor.
2. Verify that the accessory does not send a HID report if there has not been any change in the state of the control surfaces (i.e., no polling of HID reports).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Verify that if any accessory has physical or virtual control surfaces that generate accessory HID usages, the controls are labeled with appropriate iconography or text that corresponds to the resulting Apple device behavior (i.e., a Play/Pause button must be labeled with the text Play/Pause or a Play/Pause icon.
4. Verify HID usages map to physical or virtual controls on a 1:1 basis (e.g. Play button only sends Play commands, not Play/Pause).
5. Verify that one accessory HID usage report is sent in response to each direct user action on the corresponding physical or virtual control surface. For example, when the user presses a button, one 'button pressed' usage report must be sent, and a separate 'button released' usage report must be sent when the user releases the button.
6. Verify that the following iAP2 control session message(s) are sent or received:
   - StartHID (page 844)
   - AccessoryHIDReport (page 845)

34.3.2 HID over Native

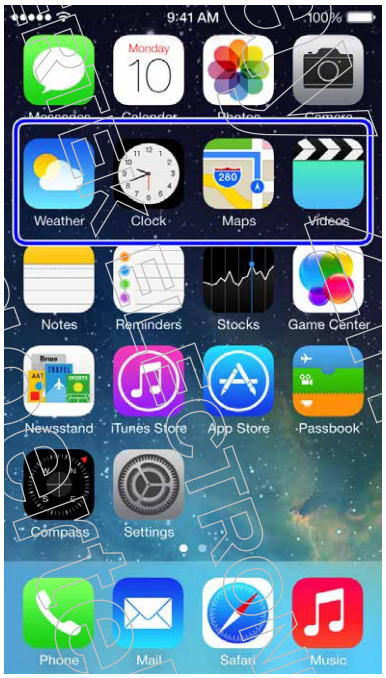
1. Verify that StartHID (page 844), DeviceHIDReport (page 845), AccessoryHIDReport (page 845), and StopHID (page 846) are not included in IdentificationInformation. This do not apply if the accessory also uses HID over iAP2.
2. Verify that the accessory declares a USB Host mode transport component during identification. This can be achieved by verifying USBHostHIDComponent (parameter 23) is listed inside the IdentificationInformation message.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

35. HID Assistive Switch Control

The Assistive Switch Control feature permits users to operate Apple devices using one or more switches. Accessory developers should consider supporting Assistive Switch Control if the accessory is intended for use by users with special needs who can see the display and manipulate simple switch inputs. If the user cannot see the display, consider supporting the VoiceOver feature instead.

Figure 35-1 HID Assistive Switch Control



35.1 HID Assistive Switch Control Requirements

Accessories that implement HID Assistive Switch Control components must support the HID (Human Interface Device) (page 580) feature and comply with all the requirements listed in HID Requirements (page 580).

A maximum of one HID component in an accessory may declare itself to be an Assistive Switch Control, and the accessory must be intended for use by a user with special needs.

Assistive Switch Controls must have at least one button. Two are recommended. Up to 16 are supported.

- Button 1 must be labeled "Select" or its localized equivalent.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- If present, button 2 must be labeled "Next" or its localized equivalent.
- If present, buttons 3-16 must not be labeled.

35.2 HID Assistive Switch Control Usage

If only Button 1 is present, the Apple device will automatically configure the Assistive Switch Control feature to use that button to select an onscreen element and scan through onscreen elements automatically.

If both Button 1 and Button 2 are present, the Apple device will automatically configure the Assistive Switch Control feature to use the first button to select an onscreen element and only scan through onscreen elements when the second button is pressed.

If more than two buttons are present, Button 1 and Button 2 are automatically configured and additional buttons may be mapped to other actions by the user.

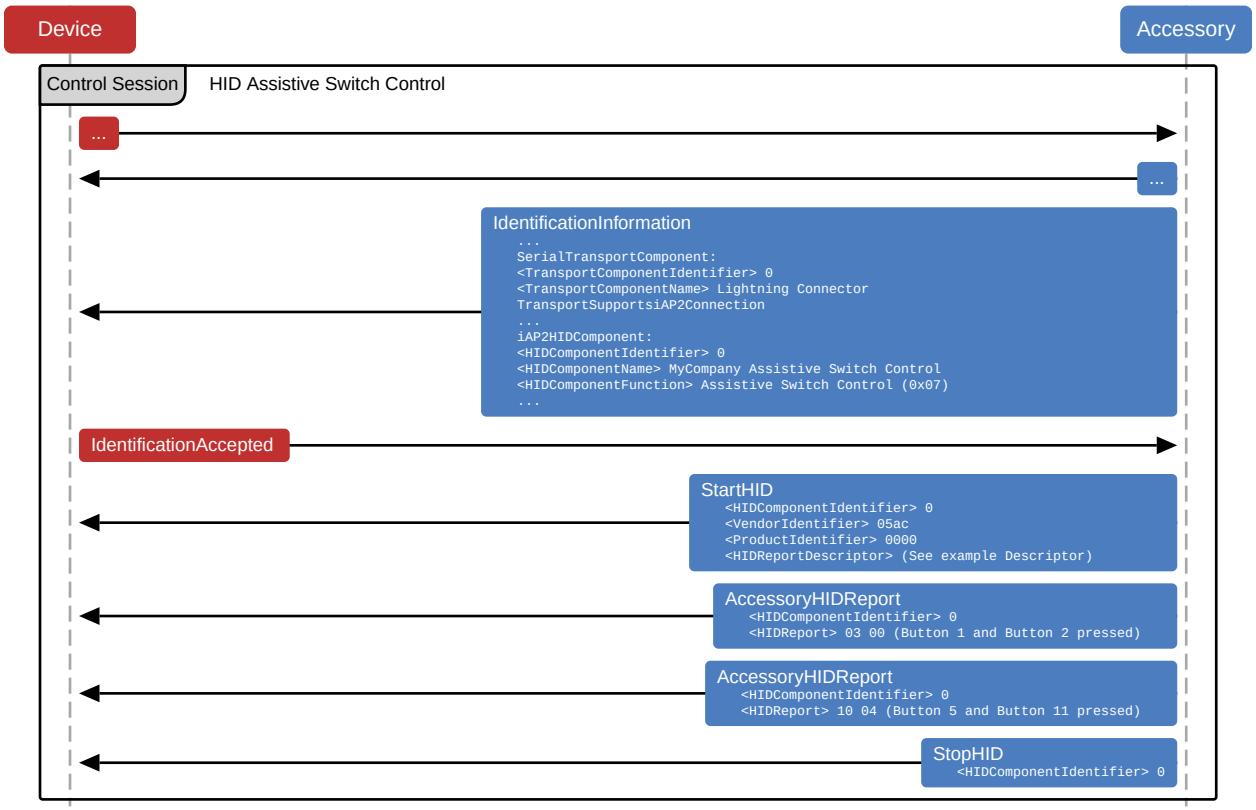
35.3 HID Assistive Switch Control Examples

35.3.1 Assistive Switch Control HID Report Descriptor

```
USAGE PAGE (Button)                 05 09
USAGE (Button 1)                    09 01
COLLECTION (Application)            A1 01
  USAGE (Button 1)                  09 01
  COLLECTION (Physical)             A1 00
    USAGE MINIMUM (Button 1)        19 01
    USAGE MAXIMUM (Button 16)       29 10
    LOGICAL MINIMUM (0)             15 00
    LOGICAL MAXIMUM (1)             25 01
    REPORT COUNT (16)               95 10
    REPORT SIZE (1)                 75 01
    INPUT (Data,Var,Abs)            81 02
  END COLLECTION (Physical)         C0
END COLLECTION (Application)        C0
```

Copyright © 2015 Apple Inc. All Rights Reserved.

35.3.2 Assistive Switch Control Usage Example



35.4 Test Procedures

35.4.1 Tests Cases

featuresHIDAssistiveSwitchControl:TestsCases

1. Verify that a maximum of one HID component in the accessory is declared to be an Assistive Switch Control.
2. Verify that the accessory is intended for use by a user with special needs.
3. Verify that Assistive Switch Controls have at least one button.
4. Verify that Button 1 is labeled "Select".
5. If present, verify that button 2 is labeled "Next".
6. If present, verify that buttons 3-16 are not labeled.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

36. HID AssistiveTouch Pointer

36.1 HID AssistiveTouch Pointer Requirements

Accessories that implement HID AssistiveTouch Pointer components must support the HID (Human Interface Device) (page 580) feature and comply with all the requirements listed in HID Requirements (page 580). Additionally, they must support the AssistiveTouch (page 342) feature.

A maximum of one HID component in an accessory may declare itself to be an AssistiveTouch pointer. That component must be a physical pointing device intended for use by a user with special needs.

The HID report descriptor for an AssistiveTouch pointer must declare support for the HID Generic Desktop Page and the Mouse usage.

The following requirements apply to the generation of HID mouse reports from the accessory:

- All x and y movements must be reported in increments of 1, proportionally scaled to the physical movement of the user. If the accessory is a joystick, for example, then a small movement of the joystick must report a movement delta of 1, but a large movement of the joystick must report a larger movement delta.
- The accessory must send repeated HID pointer movement reports at a constant rate appropriate for the accessory. The accessory must not perform its own scaling of the report rate; the AssistiveTouch feature uses its own speed scaler setting for this purpose. If no movement has taken place, the accessory must send a movement report of 0 in both x and y directions.
- The accessory must generate HID reports for two buttons, one for a touch event and the other for a contextual menu trigger. Both button down and button up reports must be sent individually and must match actual user actions on the accessory. When the user presses on the first button, a button1 'down' report must be sent, and button1 events must not be sent until the user releases the button, after which a button1 'up' report must be sent.
- The accessory must start sending HID reports to the Apple device as soon as the Apple device sends a notification to the accessory indicating that the AssistiveTouch cursor has been enabled.
- The accessory must cease sending HID reports to the Apple device as soon as the Apple device sends a notification to the accessory indicating that the AssistiveTouch cursor has been disabled.
- The accessory must be capable of interleaving pointer movement reports with button up and down reports. The accessory must let the user hold a button down and move the pointer at the same time.
- The accessory must report relative, not absolute, pointer movement.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

See Appendix E.10: Report Descriptor(Mouse) in *Device Class Definition for Human Interface Devices 1.11* (available from the USB-IF) for a sample HID descriptor.

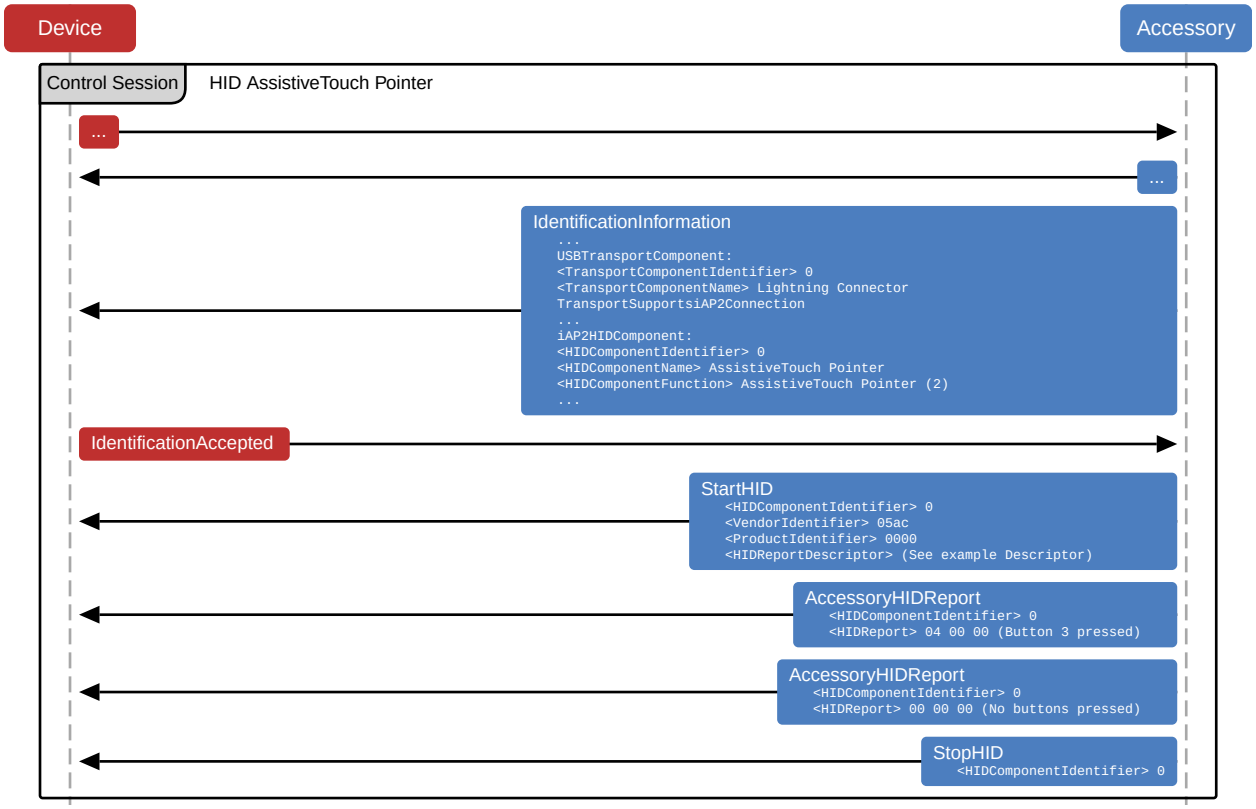
36.2 HID AssistiveTouch Pointer Examples

36.2.1 AssistiveTouch HID Report Descriptor

```
USAGE_PAGE (Generic Desktop)        05 01
USAGE (Mouse)                       09 02
COLLECTION (Application)            A1 01
  USAGE (Pointer)                   09 01
  COLLECTION (Physical)             A1 00
    USAGE_PAGE (Button)             05 09
    USAGE_MINIMUM (Button 1)        19 01
    USAGE_MAXIMUM (Button 3)        29 03
    LOGICAL_MINIMUM (0)             15 00
    LOGICAL_MAXIMUM (1)             25 01
    REPORT_COUNT (3)                95 03
    REPORT_SIZE (1)                 75 01
    INPUT (Data,Var,Abs)            81 02
    REPORT_COUNT (1)                95 01
    REPORT_SIZE (5)                 75 05
    INPUT (Cnst,Ary,Abs)            81 01
    USAGE_PAGE (Generic Desktop)    05 01
    USAGE (X)                       09 30
    USAGE (Y)                       09 31
    LOGICAL_MINIMUM (-127)          15 81
    LOGICAL_MAXIMUM (127)           25 7F
    REPORT_SIZE (8)                 75 08
    REPORT_COUNT (2)                95 02
    INPUT (Data,Var,Rel)            81 06
  END_COLLECTION                    C0
END_COLLECTION                      C0
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

36.2.2 AssistiveTouch Usage Example



36.3 Test Procedures

36.3.1 Test Cases

1. Verify that a maximum of one HID component in the accessory is declared to be an AssistiveTouch Pointer.
2. Verify that the HID report descriptor declares support for the HID Generic Desktop Page and the Mouse usage.
3. Verify that control sensitivity is proportional to user input. (e.g. slow, slight movement of the accessory results in a movement report delta of 1.)
4. Verify that If no movement has taken place, the accessory sends a movement report of 0 in both x and y directions.
5. Verify that the accessory allows the user to hold a button down and move the pointer at the same time.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

37. HID Game Controller

iOS apps may take advantage of accessories that implement one or more HID Game Controller components via the Game Controller framework. Two controller component types are defined in this specification:

- Form-Fitting Gamepad
- Non Form-Fitting Gamepad

Information on how to create iOS apps that make use of HID Game Controller accessories can be found at https://developer.apple.com/library/ios/documentation/ServicesDiscovery/Conceptual/GameControllerPG/Introduction/Introduction.html.

37.1 HID Game Controller Requirements

Accessories that implement HID Game Controller components must support the HID (Human Interface Device) (page 580) feature and comply with all the requirements listed in HID Requirements (page 580).

Form-fitting controllers must physically enclose the Apple device (in landscape orientation) and allow the user's thumbs to reach the Apple device's touchscreen. They must also comply with the requirements set forth in Form-Fitting Accessories (page 167) and Cases (page 487). Specifically, the device's touchscreen must not be occluded in any way.

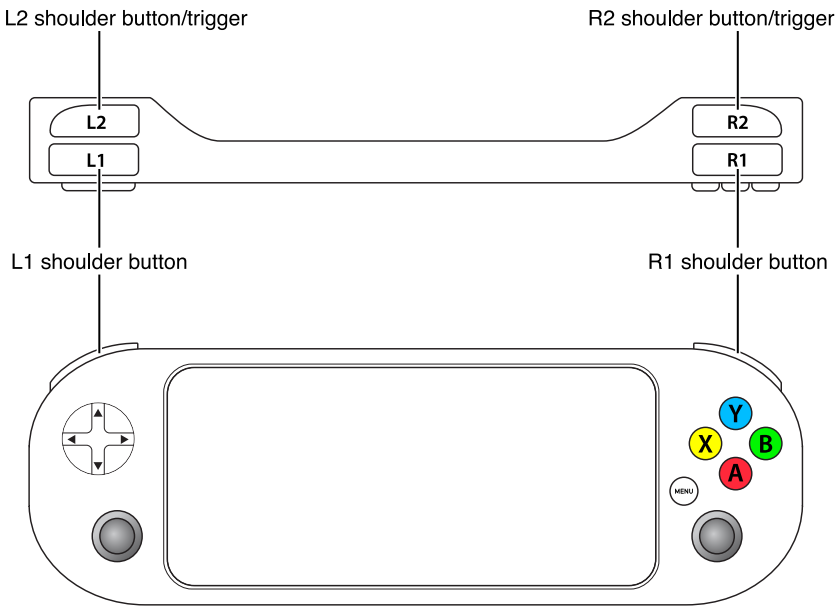
An accessory must implement only one form-fitting controller component.

If an accessory has multiple physical configurations that would allow it to be both a form-fitting gamepad and a non form-fitting gamepad, it must terminate the iAP2 connection, re-connect, and appropriately identify itself to the Apple device whenever it enters a new physical configuration. Such accessories must not always claim to be a non form-fitting gamepad.

> **Note:** Controllers that claim compatibility with Apple TV (4th generation) must integrate a Game Controller Module (page 543).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

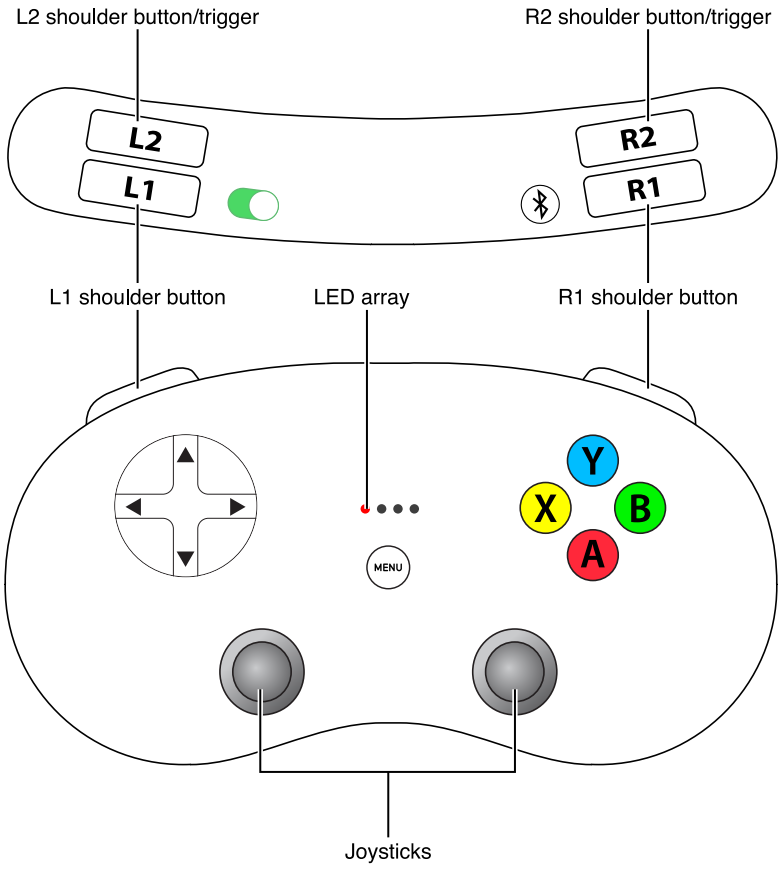
Figure 37-1 Form-Fitting Gamepad Sample



Non form-fitting controllers must not physically enclose the Apple device and must comply with gamepad requirements. One accessory may implement up to 4 physically separate non form-fitting controller components.

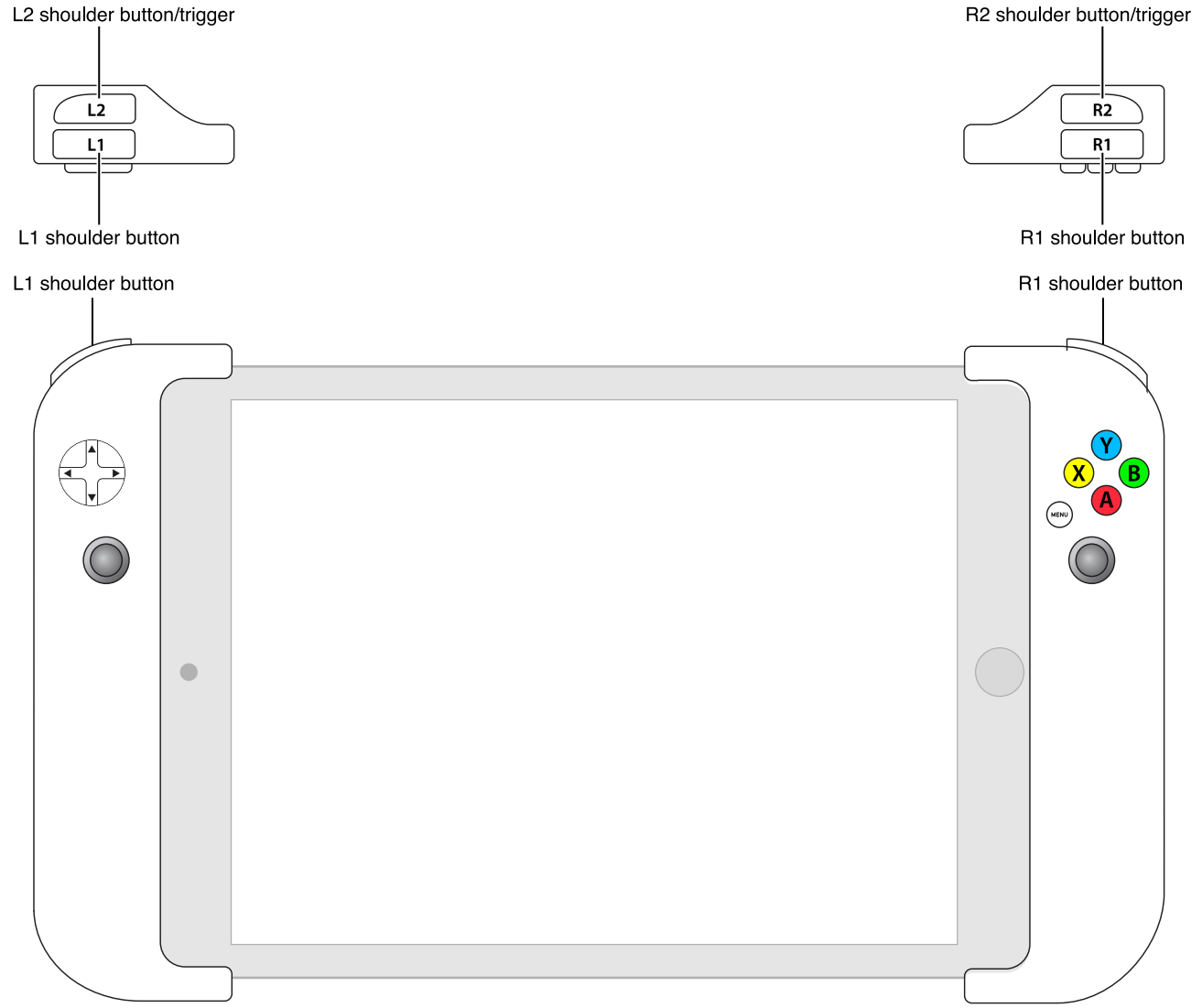
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 37-2 Non Form-Fitting Gamepad Sample



Form-fitting controllers can be made for any Apple device supporting this feature such as the iPad mini (see Figure 37-3 (page 594)).

Figure 37-3 iPad mini Form-Fitting Gamepad Sample



Each HID component that declares itself as a game controller must implement a prescribed configuration of physical control surfaces. All HID reports sent from the controller must occur in response to direct user action, i.e. depressing one of the control surfaces.

Controllers must not implement additional physical control surfaces other than those required by the gamepad definition except for the following:

- Hold switch to disable control surfaces and to power off the controller. See Hold Switch (page 604)
- Charging on/off button or switch for a controller with integrated battery that provides power to the Apple device.
- Button to initiate Bluetooth pairing. See Bluetooth Button (page 606).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Reset button. See Reset Button (page 604).

Controller accessories must be serialized, i.e. the *SerialNumber* parameter in their IdentificationInformation (page 807) message must be unique for a given *Manufacturer* and *ModelIdentifier* pairing.

Controllers must sample the state of all control surfaces at a rate of 120 Hz.

When the state of any control changes, the controller must send a complete HID report containing the state of every control surface to the Apple device. Partial updates are not allowed. Controllers must not send HID reports if control state has not changed.

Battery powered controllers must be able to operate for a minimum of 20 hours of continuous operation (i.e., sending continuous HID reports at the controller sampling rate). 40 hours of continuous operation is recommended.

Controllers should allow users to adopt comfortable postures during normal use, namely grip, reach, and finger placement on input buttons. Children and females with small hands (5th percentile), as well as large males (95th percentile) should be considered during the design process. Apple recommends taking advantage of resources such as AnthroKids (http://ovrt.nist.gov/projects/anthrokids/), the U.S. Army Anthropometry Survey (http://mreed.umtri.umich.edu/mreed/downloads.html#ansur), and the Civilian American and European Surface Anthropometry Resource (http://store.sae.org/caesar/index.htm).

A user must be able to accurately hold a control surface at any position.

All controllers must be capable of field-deployable firmware updates from Apple devices running iOS and/or Macintosh computers running OS X.

Controllers must not implement alternate operating modes that emulate keyboards or any other type of Human Interface Device besides a gamepad.

Controllers should implement support for Macintosh computers running OS X.

37.1.1 Gamepad

To support the Gamepad definition, an accessory must implement the following control surfaces:

- Menu button
- Face button group
- Directional pad
- Top left shoulder button (L1)
- Top right shoulder button (R1)
- Bottom left shoulder button/trigger (L2)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

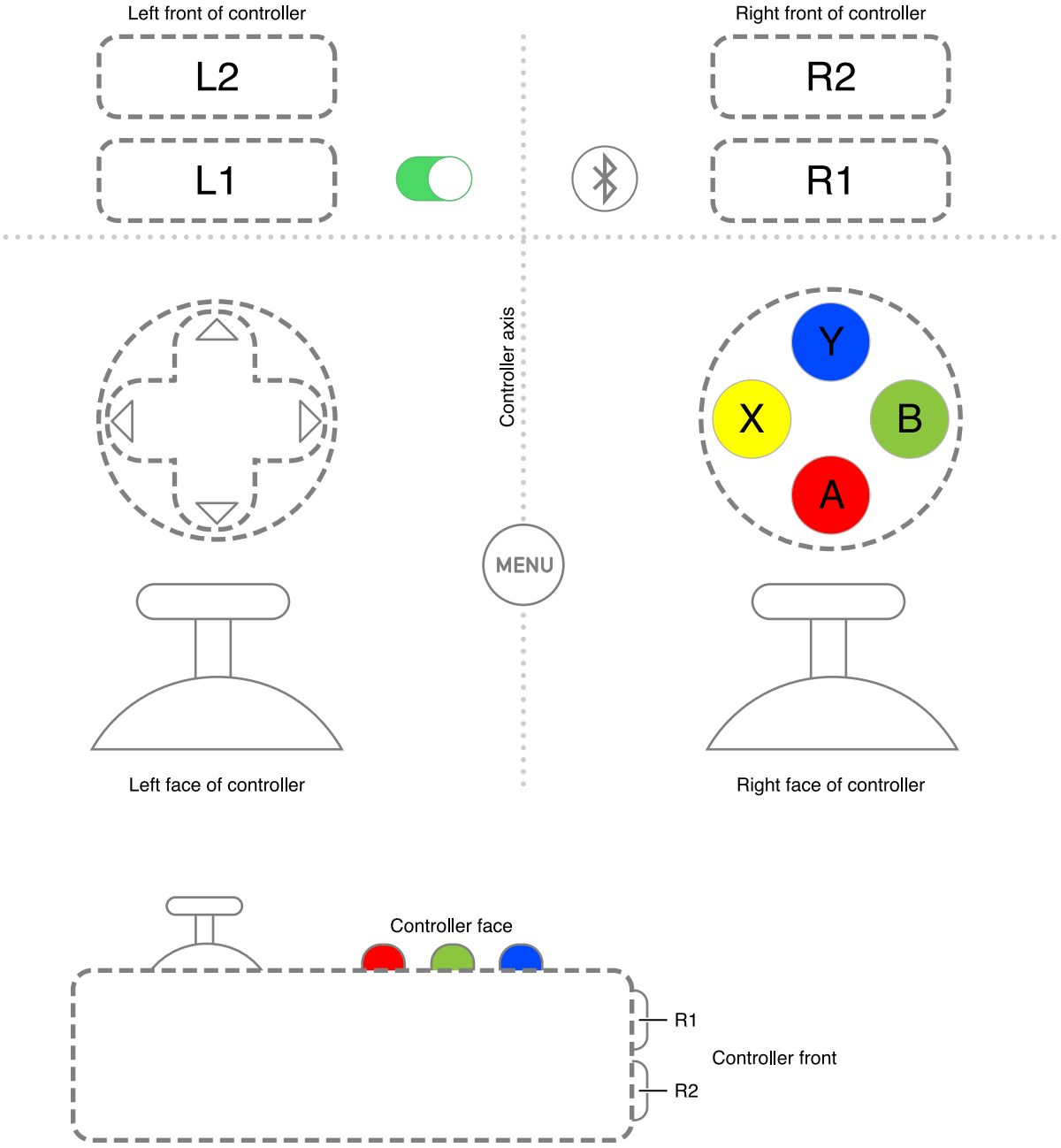
- Bottom right shoulder button/trigger (R2)
- Left joystick
- Right joystick

The layout and labeling of the control surfaces must match that of Figure 37-4 (page 597), except in four cases:

- Form-fitting gamepads may position the menu button on either the left or right side of the controller axis. The right side is recommended.
- Non form-fitting gamepads must position the menu button on the controller axis, with one exception. If there is a mechanical element along the controller axis, placing the menu button on the right side is recommended.
- Gamepad joysticks may be swapped vertically with their corresponding face button group or directional pad.
- Non form-fitting gamepads must also implement a LED array as defined in LED Array (page 603). Form-fitting gamepads may implement the LED array.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 37-4 HID Gamepad Control Layout



37.1.2 Switches and Position Encoders

Game controller switches are either analog (pressure sensitive or position encoder) or digital. They may be combined with buttons or triggers to create controller control surfaces.

Pressure sensitive switches and position encoders must report status using an unsigned *n*-bit integer, where *n* is 8 or more. They must report a value of 0 when not depressed or 2^n-1 when fully depressed to their mechanical limit. They must report values between 0 and 2^n-1 inclusive on a linear scale throughout the control surface's range of motion.

Digital switches must report status using a 1-bit value. They must report 1 when fully depressed to their mechanical limit and 0 otherwise.

Pressure sensitive switches must be sourced from specific contacts at Apple approved suppliers (see the MFi Portal for more details). Those suppliers may provide customized designs, or standardized assemblies.

Fujikura makes standardized parts that can be adapted to a range of designs:

- GCZM-012-00 - D-pad
- GCZM-013-00 - A,B,X,Y buttons
- GCZM-014-00 - Shoulder buttons

Pressure sensitive switches must meet the following mechanical requirements:

Table 37-1 HID Game Controller pressure sensitive switch mechanical requirements

Parameter	Value at 0 Cycles	Value at 1M Cycles
Force to achieve mechanical `click'	70 - 130 gf	50 - 150 gf
Force to reach `on' state	< 130 gf	< 155 gf
Force to reach `fully on' state	< 400 gf	< 480 gf
Switch travel at `on' state	0.8 - 1.4 mm	0.7 - 1.5 mm

Position encoders may be implemented using potentiometers, Hall effect sensors, or other means.

37.1.3 Product Design

37.1.3.1 Surfaces

The contact surfaces of the controller (control surfaces, grips, etc.) should be designed for comfort during extended use.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

37.1.3.2 Weight and Balance

The controller should feel solid and balanced when held. A mass of 200-300 g is recommended.

37.1.3.3 Labels and Colors

All text labels should be rendered in at least 10 point font. Helvetica is recommended. Unless otherwise specified, the color of the labels should be black or white, whichever provides most contrast against the controller enclosure or button.

All markings must be legible after 1 million cycles of use.

To obtain color standards for matching or quality control purposes:

- Go to www.ptli.com/applecolor
- Choose "Order Color Standards"
- Enter user name and password to access and order standards. First time users must choose "Create New Account" and follow the instructions.

37.1.4 Menu Button

The menu button must be circular with a diameter of at least 6 mm and labeled as "MENU" in all capital letters. See Figure 37-5 (page 599) for an example using recommendations from Labels and Colors (page 599).

The menu button must use a digital switch. The actuation force of the menu button should be 1.2-3.5 N.

Bluetooth controllers must use the menu button to power on the controller, see Bluetooth (page 605).

Figure 37-5 HID game controller menu button



37.1.5 Face Button Group

A face button group must contain four circular buttons oriented 90° in a diamond layout. The buttons must be at least 6 mm in diameter, and the outer circle of the layout must be at least 20 mm in diameter.

The button group should be oriented within 5° of the controller axis. All of the button switches must be pressure sensitive.

Buttons must comply with the layout, color, and labeling as illustrated in Figure 37-6 (page 600) or in Figure 37-7 (page 601).

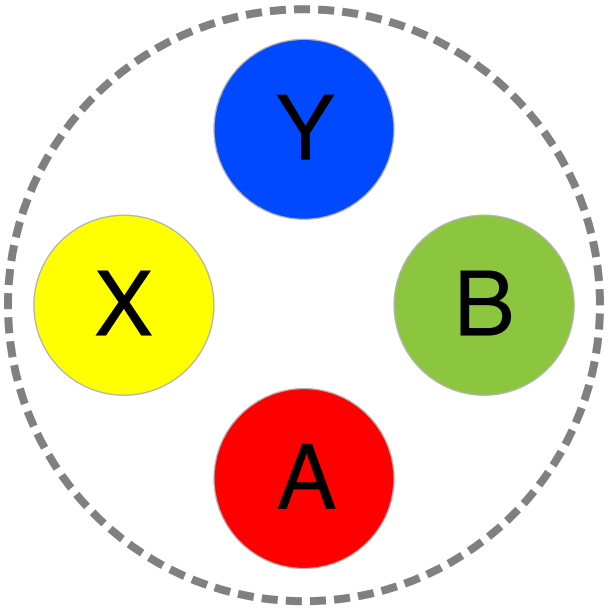
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The following color definitions are provided as reference for the face button group:

Table 37-2 HID Game Controller face button group color definitions

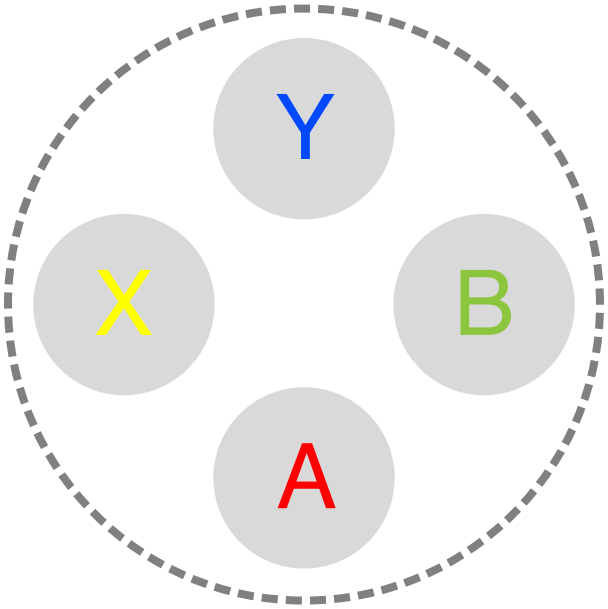
Button	Description	PTLI Color Code
A	Passion Red	912-0371-B
B	California Kiwi 120%	912-0754-A
X	Pollen Yellow Dark	912-0759-A
Y	Blue Fin Tuna 3	912-0753-A

Figure 37-6 HID game controller face button group layout



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 37-7 HID game controller face button group layout (alternate)



37.1.6 Directional Pad

A directional pad must be implemented as one circular button that pivots in all directions. The button must not translate in any direction. The button must be at least 17 mm in diameter.

The directional pad must report its status using 4 pressure-sensitive switches, one for each cardinal direction (up, down, left, and right).

Users must be able to smoothly, quickly, and accurately generate all positions within the minimum unit circle using a directional pad.

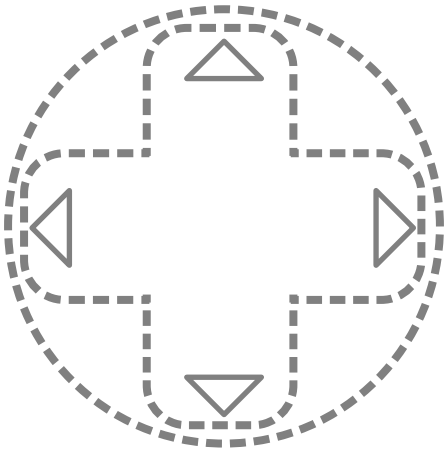
The button must implement a raised cross-shaped region that covers each cardinal direction. Each endpoint of the cross-shaped region must be labeled with a triangle oriented in the corresponding cardinal direction. The cross-shaped region must be proud of the controller's surface, but the remainder of the button may be flush or hidden from direct user contact.

The directional pad mechanical design must guarantee that opposing cardinal directions (i.e. up/down or left/right) cannot be depressed simultaneously.

The directional pad must be oriented within 5° of the controller axis. Directional pads must comply with the layout, color, and labeling as illustrated in Figure 37-8 (page 602).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 37-8 HID gamepad directional pad layout



37.1.7 Shoulder Buttons

Shoulder buttons must be labeled as L1, L2, R1, and R2.

Additionally,

- L1 and R1 on a gamepad must use pressure sensitive switches.
- L2 and R2 on a gamepad must use either pressure sensitive switches or position encoders.

Implementing L2 and R2 as displacement triggers with position encoders is recommended.

Displacement triggers must have a minimum travel of 5 mm.

See Figure 37-4 (page 597) for shoulder button layout.

37.1.8 Joysticks

Joysticks must be implemented using one of the following two-axis analog controls:

- Alps RKJXV1220001
- Alps RKJXY
- Alps RKJXU

The joystick must report a full range of movement between the unit circle and unit square using two signed *n*-bit integers where *n* is 8 or more, one for the X axis and one for the Y axis. The joystick should exhibit a positive centering force about both axes.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The joystick must rotate to at least 15° and up to 40° or translate at a minimum of 3 mm in all directions. The force to reach maximum travel in any direction must be between 0.7N and 1.3N. The breakout force to initiate values greater than (0,0) from rest state must be no greater than 30% of the force at maximum output. The output values of the joystick must be linearly reported with respect to physical displacement or angular displacement of the joystick. Extreme sensitivity or sluggishness in response to small control deflections or force must be avoided.

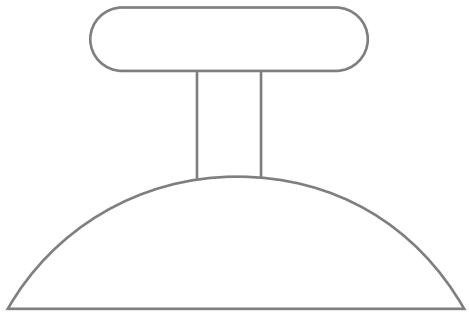
Joysticks must have a circular outline with a diameter of at least 10 mm.

At rest, the joystick must report (0,0) for the X and Y axes respectively. Additionally, each axis must report a full range of values between $-2^{n-1}+1$ and $+2^{n-1}-1$ inclusive on a linear scale, where n is 8 or more. The maximum displacement on any given cardinal direction must be symmetrical to its opposite cardinal direction. The joystick dead zone must be a maximum of 5% of total joystick travel.

Users must be able to smoothly, quickly, and accurately generate all positions within the minimum unit circle using a joystick.

Note: Joysticks must not 'click' or permit the user to depress the stick to generate an input event.

Figure 37-9 HID game controller joystick



37.1.9 LED Array

The LED array must be physically implemented as a linear array of four red LEDs. The LED array must be at least nominally visible while the controller is being used. The LEDs 1 through 4 should be arranged from left to right.

The LED array must be used to indicate controller state according to Table 37-3 (page 604) with only one LED state indicated at a time (i.e. Battery Low temporarily overrides the Connected state). The LED array Connected state is controlled by the Apple device and must map to 4 unsigned 1-bit values (usage IDs 0xFF00-0xFF03) in the accessory's HID report descriptor. LED array state is not reported to the Apple device.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 37-3 HID game controller LED states

Priority	State	Behavior
1	Error	The LED array may be used for temporarily indicating error conditions.
2	Battery Low (<10%)	When powering on with low battery or when battery level drops below 10%, flash LED 1 for 10 seconds at 1 Hz.
3	Battery Full	When charging is complete or charger is connected and battery is full, turn on all 4 LEDs for 10 seconds, then fade them off.
4	Pairable	When pairable, fade LEDs on and off together at 1 Hz. See Bluetooth (page 605).
5	Connecting	When connecting, fade LEDs on and off at 1 Hz each, cycling from LED 1 to 4 and back to 1 (i.e. 1 to 4, 4 to 1).
6	Connected	Turn on LEDs according to the HID report from the currently connected Apple device. If no LEDs are set, turn on LED 1. The LEDs should dim 5 seconds after any change.
7	Charging	When charging, fade LEDs on and off at 1 Hz each, cycling from LED 1 to 4 continuously (i.e. 1 to 4, 1 to 4). Keep LED 1 on if battery level >25%, LED 1-2 on if >50%, and LED 1-3 on if >75%.

Animations should complete before an LED state transition and should not terminate abruptly.

37.1.10 Hold Switch

Controllers may include a hold switch that disables input. When in the hold position, all inputs from control surfaces must not be treated as direct user action. The hold switch should be located between the left shoulder buttons and the controller axis. The hold position should be towards the left shoulder buttons.

The hold position must be indicated by exposing a red surface when in the hold position. A text label of "HOLD" in all capital letters may also be used to indicate the hold position. See Labels and Colors (page 599).

The controller may power off if the hold switch in the hold position, but must still be able to charge.

37.1.11 Reset Button

If the controller has a reset button, it must not be in a location that can be accidentally pressed by the user during normal operation. It should be flush with or recessed from the controller surface. A location on the bottom of the device is recommended.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Controllers with a Bluetooth button must not implement a reset button, see Bluetooth Button (page 606).

37.1.12 Bluetooth

All controllers that utilize the Bluetooth transport must set their Major Device Class to 'Peripheral' and their Minor Device Class to 'Gamepad'.

In addition to the requirements in Bluetooth (page 344) and HID (Human Interface Device) (page 580), the controller:

- Should implement HID Native Transport (Bluetooth) Requirements (page 582).
- May support HID over iAP2.

Bluetooth controllers must use the menu button to turn the controller on according to the following behavior:

- Pressing and holding the menu button for 0.6 seconds must power on the controller from an off state.

After powering on, the controller must:

- Be connectable for 2 minutes.
- Attempt to reconnect to the most recently connected Apple device for 1 minute.
- Visually indicate that is in attempting to connect. See LED Array (page 603).
- Enter a pairable mode for 2 minutes if it has no saved pairings.

A pairable mode can also be initiated using the Bluetooth button, see Bluetooth Button (page 606).

When a controller is in a pairable mode, it must:

- Be discoverable and pairable for 1 minute.
- Visually indicate that it is in a pairable mode. See LED Array (page 603).
- Exit the pairable mode after a successful pairing or if it does not receive any pairing requests.

The controller must be able to remember at least 5 paired Apple devices. When the controller receives a pairing request, if it has already remembered its maximum number of paired Apple devices, it must replace the pairing entry for the least recently connected Apple device.

The controller may implement a mechanism to clear pairing memory using the Bluetooth button, see Bluetooth Button (page 606).

Once a connection is established, the controller must:

- Always be connectable.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Always accept the most recent connection request it receives from previously paired devices.

If an active connection is lost, the controller must:

- Be connectable for 2 minutes.

The controller may power off if all of the above conditions are met and it does not have an active connection with an Apple device.

37.1.13 Bluetooth Button

Controllers that utilize the Bluetooth transport must implement a button on the controller specifically for initiation of Bluetooth pairing. The button must not be in a location that can be accidentally pressed by the user during normal operation. The Bluetooth button should be located between the right shoulder buttons and the controller axis.

The button must be labeled or visually treated with an official Bluetooth icon that makes its purpose apparent to the end user.

Pressing and holding the button for 2 seconds must place the controller into a pairable mode.

Pressing and holding both the menu button and Bluetooth button simultaneously for 5 seconds should clear all saved Bluetooth pairings. This may also function as a hardware reset.

37.1.14 App Match for Controller-Enabled Games

Game controllers must support App Match for an Apple-maintained list of controller-enabled games.

To do so, the controller must identify support for an External Accessory Protocol (see Table 59-11 (page 810)) with the following parameters:

Table 37-4 HID Game Controller App Match Parameters

Parameter	Value
ExternalAccessoryProtocolName	'com.apple.mfi.gamecontroller'
ExternalAccessoryProtocolMatchAction	1

This External Accessory Protocol must not be used for application data, firmware updates, or any other purpose. A separate External Accessory Protocol may be declared for these uses. See External Accessory Protocol (page 535).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

37.2 HID Game Controller Examples

37.2.1 Gamepad Example HID Report Descriptor

This HID report descriptor includes the LED array that is required for non form-fitting controllers.

```
USAGE PAGE (Generic Desktop)        05 01
USAGE (GamePad)                     09 05
COLLECTION (Application)            A1 01
  USAGE (GamePad)                   09 05
  COLLECTION (Logical)              A1 02
    REPORT SIZE (8)                 75 08

    // Directional pad (up, right, down, left)
    REPORT COUNT (4)                95 04
    LOGICAL MINIMUM (0)             15 00
    LOGICAL MAXIMUM (255)           26 FF 00
    PHYSICAL MINIMUM (0)            35 00
    PHYSICAL MAXIMUM (255)          46 FF 00
    USAGE PAGE (Generic Desktop)    05 01
    USAGE (DPadUp)                  09 90
    USAGE (DPadRight)               09 92
    USAGE (DPadDown)                09 91
    USAGE (DPadLeft)                09 93
    INPUT (Data,Var,Abs)            81 02

    // Face button group and left/right shoulder/trigger buttons
    // (A, B, X, Y, L1, R1, L2, R2)
    REPORT COUNT (8)                95 08
    USAGE PAGE (Button)             05 09
    USAGE MINIMUM (Button 1)        19 01
    USAGE MAXIMUM (Button 8)        29 08
    INPUT (Data,Var,Abs)            81 02

    // Menu button
    LOGICAL MINIMUM (0)             15 00
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
LOGICAL MAXIMUM (1)             25 01
PHYSICAL MINIMUM (0)            35 00
PHYSICAL MAXIMUM (1)            45 01
REPORT SIZE (1)                 75 01
REPORT COUNT (1)                95 01
USAGE PAGE (Consumer)           05 0C
USAGE (AC Home)                 0A 23 02
INPUT (Data,Var,Abs)            81 02
REPORT COUNT (7)                95 07
INPUT (Cnst,Var,Abs)            81 03

// LED array
LOGICAL MINIMUM (0)             15 00
LOGICAL MAXIMUM (1)             25 01
PHYSICAL MINIMUM (0)            35 00
PHYSICAL MAXIMUM (1)            45 01
REPORT SIZE (1)                 75 01
REPORT COUNT (4)                95 04
USAGE PAGE (LED)                05 08
USAGE MINIMUM (Reserved 0xFF00) 1A 00 FF
USAGE MAXIMUM (Reserved 0xFF03) 2A 03 FF
OUTPUT (Data,Var,Abs)           91 02
REPORT SIZE (4)                 75 04
REPORT COUNT (1)                95 01
OUTPUT (Cnst,Ary,Abs)           91 01

// Left/right joysticks
LOGICAL MINIMUM (-127)          15 81
LOGICAL MAXIMUM (127)           25 7F
PHYSICAL MINIMUM (-127)         35 81
PHYSICAL MAXIMUM (127)          45 7F
USAGE PAGE (Generic Desktop)    05 01
USAGE (Pointer)                 09 01
COLLECTION (Physical)           A1 00
  REPORT SIZE (8)               75 08
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
      REPORT COUNT (4)              95 04
      USAGE (X)                     09 30 // left axis horizontal
      USAGE (Y)                     09 31 // left axis vertical
      USAGE (Z)                     09 32 // right axis horizontal
      USAGE (Rz)                    09 35 // right axis vertical
      INPUT (Data,Var,Abs)          81 02
    END COLLECTION (Physical)       C0
  END COLLECTION (Logical)          C0
END COLLECTION (Application)        C0
```

37.3 Test Procedures

The test procedures make use of the following tools and resources:

- ATS 3.4 or greater
- Caliper
- Force Gauge
- PTLI Color standards - see Labels and Colors (page 599)

37.3.1 General Requirements

1. Verify that the controller does not implement alternate operating modes that emulate keyboards or any other type of Human Interface Device besides HID Game Controller.
2. Using ATS 3.4 or greater, verify that multiple controller samples have unique serial numbers via inspection of the SerialNumber parameter in the IdentificationInformation (page 807) message.
3. Verify that the controller is capable of having its firmware updated from Apple devices running iOS and/or Macintosh computers running OS X.

> **Note:** When submitting samples to Apple for self-certification, include the necessary files and instructions to verify the firmware update mechanism.

37.3.2 Bluetooth Controllers

1. Verify that the Bluetooth Major and Minor Device Class fields are set as specified in Bluetooth (page 605).
2. See Menu Button (page 612).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. See Bluetooth Button (page 613).
4. Power on the controller and verify that the controller stays on and connectable for at least 2 minutes.
5. Press and hold the Bluetooth pairing button for 2 seconds. Verify that the accessory is discoverable and pairable for at least 30 seconds. Verify that it indicates this visually.
6. Verify that the accessory exits discoverable and pairable mode automatically.
7. Pair the accessory with an Apple device (device "A"). Verify that the accessory is paired.
8. Pair the accessory with 1 more Apple device (device "B").
9. Pair the accessory with as many more Apple devices as it can remember (at least 3 more).
10. Use first Apple device (device "A") to connect to the accessory.
11. Pair the accessory with 1 more Apple device.
12. Verify that the accessory does not connect to device "B" and does connect with device "A" and the other paired devices.
13. Connect the controller to device "A". Initiate a connection from device "C" (or one of the other paired devices). Verify that the controller disconnects from device "A" and connects to device "C".

37.3.3 Form-Fitting Controllers

1. Verify that the form-fitting controllers must meet the same requirements as Cases (page 487).
2. Verify that there is no relative movement between the accessory's Lightning connector and the Apple device when connected.
    a. Attach/connect the Apple device to the accessory.
    b. While holding the accessory, attempt to move the Apple device in various directions.
    c. Verify that the accessory's Lightning connector and the Apple device do not move independent of each other.
3. Verify that controller physically encloses the Apple device in landscape orientation and allows the user's thumbs to reach the Apple device's touchscreen.
4. Verify that the Apple device's touchscreen is not occluded by the accessory in any way.
5. Verify that the controller consists of only one physical form-fitting component.
6. Verify that the controller implements a 4 element LED array or no LED array.

37.3.4 Non Form-Fitting Controllers

1. Verify that controller does not physically enclose the Apple device.
2. Verify that the controller consists of 1 to 4 physically separate components.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Verify that the controller implements a 4 element LED array.

37.3.5 Control Surface Layout and Labeling

1. Verify that the controller does not implement physical control surfaces other than those required by the gamepad definitions (see Table 37-5 (page 611)), with the following exceptions:
   a. Hold switch to disable control surfaces.
   b. Charging on/off button or switch for a controller with integrated battery that provides power to the Apple device.
   c. Bluetooth button for a controller with Bluetooth support.
   d. Reset button.

Table 37-5 Required Game Controller Control Surfaces

Control Surface	Gamepad
Menu button	√
Face button group (A, B, X, Y)	√
Directional pad	√
Top left shoulder button (L1)	√
Top right shoulder button (R1)	√
Bottom left shoulder button/trigger (L2)	√
Bottom right shoulder button/trigger (R2)	√
Left joystick	√
Right joystick	√

2. Verify that the layout and labeling of the control surfaces match as shown in Figure 37-4 (page 597) along with the following exceptions:
   a. Form-fitting controllers may position the menu button on either the left or the right side of the controller axis.
   b. Gamepad joysticks may be swapped vertically with their corresponding face button group or directional pad.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Verify that all text labels are rendered in at least 10 point font.
4. Verify that label text is black or white for all buttons except the menu button (see Menu Button (page 612)) and face button group (see Face Button Group (page 612)).
5. Verify that the color of labels provides contrast against the surface (black text on light surface or white text on dark surface).

37.3.5.1 Menu Button

1. Using a caliper, verify that the menu button is circular with a diameter of at least 6 mm.
2. Verify that the menu button is labeled with the specified icon.
3. Verify that the button can not be accidentally pressed by the user during normal operation.
4. Verify that the layout and labeling of the menu button matches Menu Button (page 599).

In the case of a Bluetooth controller:

1. With the controller powered off, press and hold the button for 0.6 seconds and verify that the controller powers on.

37.3.5.2 Face Button Group

1. Verify that the face button group contains four circular buttons oriented 90° in a diamond layout.
2. Using a caliper, verify that the face button group buttons are at least 6 mm in diameter.
3. Using a caliper, verify that the outer circle of the face button group is at least 20 mm in diameter.
4. Verify that the face button group is oriented within 5° of the controller axis.
5. Verify that the layout and labeling of the face button group matches either Figure 37-6 (page 600) or Figure 37-7 (page 601).
6. Verify that the face buttons match the color requirements in Table 37-2 (page 600).

37.3.5.3 Directional Pad

1. Verify that the directional pad is one circular button that pivots in all directions and does not translate in any direction.
2. Using a caliper, verify that the directional pad is at least 17 mm in diameter.
3. Verify that the directional pad implements a raised cross-shaped region covering each cardinal direction, and that the cross-shaped region is proud of the controller's surface. Verify that each endpoint of the raised region is labeled with a triangle oriented in the corresponding direction.
4. Verify that the directional pad is oriented within 5° of the controller axis.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

5. Verify that the layout and labeling of the directional pad matches Figure 37-8 (page 602).

37.3.5.4 Joystick

1. Using a caliper, verify that the joystick has a circular outline with a diameter of at least 10 mm.
2. Using a caliper, verify that the maximum displacement on any given cardinal direction is symmetrical to its opposite cardinal direction.
3. Verify that the joystick does not "click" or translate in any axes other than up/down and left/right.

37.3.5.5 LED Array

1. Verify that the LED array is implemented as a linear array of four red LEDs. (Use whatever means is available to activate the LEDs. For example, use the Bluetooth pairing button if one is present.)
2. Verify that the LED array is nominally visible while the user is holding the controller.
3. Using ATS 3.4 or greater, verify that the array of four red LEDs state maps to 4 unsigned 1-bit values in the accessory's HID report descriptor.

37.3.5.6 Hold Switch

1. Verify that when in the hold position, no inputs from control surfaces are treated as direct user actions.

37.3.5.7 Reset Button

1. Verify that there is a dedicated button on the controller to reset the controller.
2. Verify that the button can not be accidentally pressed by the user during normal operation.
3. Verity that the button is labeled or otherwise visually treated to makes its purpose apparent.

37.3.5.8 Bluetooth Button

1. Verify that there is a dedicated button on the controller to initiate Bluetooth pairing.
2. Verify that the button can not be accidentally pressed by the user during normal operation.
3. Verity that the button is labeled with the Bluetooth icon or otherwise visually treated to makes its purpose apparent.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

37.3.6 Control Surfaces

37.3.6.1 Menu Button

1. Using a digital force gauge, verify that the menu button actuates when force between 1.2 to 3.5 N is applied.
2. Using ATS 3.4 or greater, verify that the menu button is digital by verifying that the HID usage report size is equal to 1.
3. Using ATS 3.4 or greater, verify that the menu button reports 1 when fully depressed and 0 otherwise.

37.3.6.2 Pressure Sensitive Buttons and Displacement Triggers

For every pressure sensitive button (in the face button group, shoulder buttons, & directional pad) as well as any displacement triggers:

1. Using ATS 3.4 or greater, verify that the control surface report a value of 0 when not depressed or 2^n-1 when fully depressed to their mechanical limit. (n = the Report Size value for the HID report descriptor's definition of the switch.)
2. Using ATS 3.4 or greater, verify that button down vs. button up reports are received separately.
3. Using ATS 3.4 or greater, verify that HID reports are only sent when button state / control state changes (not sent repeatedly if value is not changing).
4. For all pressure sensitive buttons:
    a. Using a digital force gauge, verify that the button achieves a mechanical `click' when force between 70 to 130 gf is applied.
    b. Using a digital force gauge, verify that the button reaches its `on' state (a value that is not 0) when < 130 gf is applied.
    c. Using a digital force gauge, verify that the button reaches its `fully on' state (2^n-1) when < 400 gf is applied.
    d. Using calipers, measure the travel of the button to reach `on' state and verify that it is between 0.8 - 1.4 mm.
5. For all displacement triggers:
    a. Using calipers, measure the travel of the trigger from `on' to `fully on' and verify that it is at least 5 mm.

Additionally, for the directional pad:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1. Using ATS 3.4 or greater, verify that the directional pad reports its status using 4 pressure-sensitive switches, one for each cardinal direction (up, down, left, and right).
2. Using ATS 3.4 or greater, verify that opposing cardinal directions commands (i.e. up/down or left/right) are not sent while the buttons are depressed simultaneously.

37.3.6.3 Joysticks

For each joystick:

1. Using a digital force gauge, verify that the joystick rotates to 30°±10° or translates at a minimum of 1.5 mm in all directions when a linear, horizontal force between 0.7 N and 1.3 N is applied.
2. Using ATS 3.4 or greater, verify that the joystick is implemented as a two-axis analog control using two signed *n*-bit integers where *n* is 8 or more, one for the X axis and one for the Y axis.
3. Using ATS 3.4 or greater, verify that the joystick reports values of *0* for each axis when at rest.
4. Verify that the joystick reports a full range of movement between the unit circle and unit square.
    a. Extend the joystick as far as possible in both directions of each cardinal direction (up, down, left, and right) and verify that the values $-2^{n-1}+1$ and $+2^{n-1}-1$ are appropriately sent.
    b. Extend the joystick as far as possible in both directions of each intracardinal direction (up-left, up-right, down-right, down-left) and verify that the values $\leq \left(\frac{\sqrt{2}}{2}\right)\left(-2^{n-1}+1\right)$ and $\geq \left(\frac{\sqrt{2}}{2}\right)\left(+2^{n-1}-1\right)$ are appropriately sent.
5. Verify that the joystick dead zone is a maximum of 5% of total joystick travel.
    a. Using a caliper, measure the total joystick distance of travel in one direction from center.
    b. Calculate 5% of the measured distance from above and verify that non-zero values are sent when the joystick is moved this distance from center in the 4 cardinal and 4 intercardinal directions.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

38. HID Headset Remote

Accessories that support the HID Headset Remote feature may interact with Apple devices in a manner that builds on top of the behaviors enabled by the Headset Remote and Mic feature.

38.1 HID Headset Remote Requirements

All accessories that implement a HID Headset Remote component must be a headset (see Headsets (page 549)) that connects to the Apple device via either the Lightning connector or Bluetooth. Also, they must implement only one headset remote component.

Headset remote components must support the HID (Human Interface Device) (page 580) feature and comply with all the requirements listed in HID Requirements (page 580).

Headset remotes that send a Volume Down or Volume Up HID button press usage must immediately follow that usage with the corresponding button release usage even if the user has not released the button yet. The result is that repeated volume ramping must not occur when the volume up or down buttons are pressed and held, matching the behavior of Apple headset remotes.

The HID report descriptor for a headset remote must declare support for the HID Consumer and/or Telephony pages and only send usages from the following tables:

Table 38-1 HID Consumer Page controls for use by headset remotes

Usage ID	Usage Name	Apple Function
0x00B5	Scan Next Track	Transport Right
0x00B6	Scan Previous Track	Transport Left
0x00B9	Random Play	Shuffle
0x00BC	Repeat	Repeat
0x00E2	Mute	Mute
0x00E9	Volume Increment	Volume Up
0x00EA	Volume Decrement	Volume Down

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Usage ID	Usage Name	Apple Function
0x025B	Promote	iTunes Radio "Play More Like This"
0x025C	Demote	iTunes Radio "Never Play This Song"
0x0262	Add to Cart	iTunes Radio "Add to iTunes Wish List"

Table 38-2 HID Telephony Page controls for use by headset remotes

Usage ID	Usage Name	Apple Function
0x0021	Flash	Center

38.2 HID Headset Remote Examples

38.2.1 Headset Remote Example HID Report Descriptor (Telephony)

The following sample HID descriptor demonstrates how to implement the same controls found on the Apple Headset Remote that can be used with all apps, including telephony.

```
USAGE_PAGE (Consumer Devices) 05 0C
USAGE (Consumer Control)      09 01
COLLECTION (Application)      A1 01
  LOGICAL_MINIMUM (0)         15 00
  LOGICAL_MAXIMUM (1)         25 01
  REPORT_SIZE (1)             75 01
  REPORT_COUNT (2)            95 02
  USAGE (Volume Increment)    09 E9    // Volume Up
  USAGE (Volume Decrement)    09 EA    // Volume Down
  INPUT (Data,Var,Abs)        81 02
  USAGE_PAGE (Telephony)      05 0B
  REPORT_COUNT (1)            95 01
  USAGE (Flash)               09 21    // Center
  INPUT (Data,Var,Abs)        81 02
  REPORT_SIZE (5)             75 05
  REPORT_COUNT (1)            95 01
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
  INPUT (Cnst, Var, Abs)      81 03
END_COLLECTION                C0
```

Each report is one byte, and each bit corresponds to one of the functions. For instance, the following sample reports communicate that the referenced button has been pressed:

- Volume Up is `0x01`
- Volume Down is `0x02`
- Center is `0x04`

38.2.2 Headset Remote Example HID Report Descriptor (Media Playback)

The following sample HID descriptor demonstrates how to implement media playback controls, including iTunes Radio buttons.

```
USAGE PAGE (Consumer Devices) 05 0C
USAGE (Consumer Control)      09 01
COLLECTION (Application)      A1 01
  LOGICAL MINIMUM (0)         15 00
  LOGICAL MAXIMUM (1)         25 01
  REPORT SIZE (1)             75 01
  REPORT COUNT (8)            95 08
  USAGE (Scan Next Track)     09 B5    // Next Track
  USAGE (Scan Previous Track) 09 B6    // Previous Track
  USAGE (Mute)                09 E2    // Mute
  USAGE (Shuffle)             09 B9    // Shuffle
  USAGE (Repeat)              09 BC    // Repeat
  USAGE (Promote)             0A 5B 02 // iTunes Radio 'Play More Like This'
  USAGE (Demote)              0A 5C 02 // iTunes Radio 'Never Play This Song'
  USAGE (Add to Cart)         0A 62 02 // iTunes Radio 'Add to Wish List'
  INPUT (Data,Var,Abs)        81 02
END COLLECTION                C0
```

Each report is one byte, and each bit corresponds to one of the functions. For instance, the following sample reports communicate that the referenced button has been pressed:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Next Track is `0x01`
- Previous Track is `0x02`
- Mute is `0x04`

38.2.3 Headset Remote Example HID Report Descriptor (Telephony and Media Playback)

The following sample HID descriptor demonstrates how to implement all possible media playback controls along with the same controls found on the Apple Headset Remote.

```
USAGE_PAGE (Consumer Devices) 05 0C
USAGE (Consumer Control)      09 01
COLLECTION (Application)      A1 01
  LOGICAL_MINIMUM (0)         15 00
  LOGICAL_MAXIMUM (1)         25 01
  REPORT_SIZE (1)             75 01
  REPORT_COUNT (10)           95 0A
  USAGE (Scan Next Track)     09 B5
  USAGE (Scan Previous Track) 09 B6
  USAGE (Mute)                09 E2    // Mute
  USAGE (Shuffle)             09 B9    // Shuffle
  USAGE (Repeat)              09 BC    // Repeat
  USAGE (Promote)             0A 5B 02 // iTunes Radio 'Play More Like This'
  USAGE (Demote)              0A 5C 02 // iTunes Radio 'Never Play This Song'
  USAGE (Add to Cart)         0A 62 02 // iTunes Radio 'Add to Wish List'
  USAGE (Volume Increment)    09 E9    // Volume Up
  USAGE (Volume Decrement)    09 EA    // Volume Down
  INPUT (Data,Var,Abs)        81 02
  USAGE_PAGE (Telephony)      05 0B
  REPORT_COUNT (1)            95 01
  USAGE (Flash)               09 21    // Center
  INPUT (Data,Var,Abs)        81 02
  REPORT_SIZE (5)             75 05
  REPORT_COUNT (1)            95 01
  INPUT (Cnst, Var, Abs)      81 03
END COLLECTION                C0
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Each report is two bytes. The bits are assigned top-to-bottom (from scan next track to select). For instance, the following sample reports communicate that the referenced button has been pressed:

- Next Track is `0x0100`
- Previous Track is `0x0200`
- Mute usage is `0x0400`
- Add To Wish List is `0x8000`
- Volume Up is `0x0001`
- Volume Down is `0x0002`
- Center is `0x0004`

Jay Zhang
WEIFANG GOERTEK ELECTRONICS CO.,LTD.
Jay.Zhang@goertek.com

39. HID Keyboard

iOS apps may accept user input from accessories that implement HID Keyboard components in place of the onscreen keyboard.

39.1 HID Keyboard Requirements

Accessories that implement HID Keyboard components must support the HID (Human Interface Device) (page 580) feature and comply with all the requirements listed in HID Requirements (page 580).

> **Note:** Accessory keyboards must not identify themselves as Apple-branded keyboards, e.g. use the Apple Vendor ID and/or Product IDs.

Accessory keyboard keys that exhibit any of the following behaviors are explicitly prohibited:

- Send anything other than a 'key pressed' or 'key released' for the key that has physically been pressed
- Emulate combinations of other keys (e.g. macros like Command-C for Copy)
- Emulate timed user actions such as 'press-and-hold'
- Send different HID usages depending on the state of another control surface

Each HID component that declares itself as a keyboard must correspond to a physical keyboard on the accessory that the user uses for tasks that might have otherwise been performed using the Apple device's onscreen keyboard. All HID usages sent from the keyboard accessory must occur in response to direct user action, i.e. pressing a key on the keyboard.

Conversely, any accessory that has physical or virtual user controls that appear to be a general purpose keyboard must implement a keyboard HID component and send corresponding HID usages. Mapping those controls to other iAP2 messages is grounds for failure to pass self certification.

Keyboards may integrate an LED to indicate the status of Caps Lock. Apple devices do not support any other keyboard status LEDs. Keyboards must not incorporate any other keyboard status LEDs, with one exception. Bluetooth keyboards may include a connection status LED.

Mechanical key layout must be based on the ISO/IEC 9995-2, ANSI-INCITS 154-1988, or JIS X 6002-1980 standards.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

If the HID keyboard component is implemented using the HID Native Transport (USB Host Mode), its USB HID descriptor must set the *bCountryCode* field to the appropriate country code as defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor*.

If the HID keyboard component is implemented using iAP2, the *LocalizedKeyboardCountryCode parameter* in the StartHID (page 844) message must be set to the appropriate country code as defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor*.

The HID report descriptor for a keyboard must declare support for the HID Keyboard/Keypad Page. For efficiency, the HID report descriptor may declare that it supports alphanumeric HID usages in the Keyboard/Keypad page which are not sent.

Accessory keyboards must implement individual keys that emit the following HID Keyboard/Keypad page usages:

Table 39-1 Required HID Keyboard/Keypad Page controls for use by keyboard components

Usage ID	Usage Name	Apple Function
0x0004	a and A	a and A
0x0005	b and B	b and B
0x0006	c and C	c and C
0x0007	d and D	d and D
0x0008	e and E	e and E
0x0009	f and F	f and F
0x000A	g and G	g and G
0x000B	h and H	h and H
0x000C	i and I	i and I
0x000D	j and J	j and J
0x000E	k and K	k and K
0x000F	l and L	l and L
0x0010	m and M	m and M
0x0011	n and N	n and N

Jay Zhang
Shenzhen Meiligoe Electronics Co.,LTD.
Meiligoe@gmail.com

Usage ID	Usage Name	Apple Function
0x0012	o and O	o and O
0x0013	p and P	p and P
0x0014	q and Q	q and Q
0x0015	r and R	r and R
0x0016	s and S	s and S
0x0017	t and T	t and T
0x0018	u and U	u and U
0x0019	v and V	v and V
0x001A	w and W	w and W
0x001B	x and X	x and X
0x001C	y and Y	y and Y
0x001D	z and Z	z and Z
0x001E	1 and !	1 and !
0x001F	2 and @	2 and @
0x0020	3 and #	3 and #
0x0021	4 and $	4 and $
0x0022	5 and %	5 and %
0x0023	6 and ^	6 and ^
0x0024	7 and &	7 and &
0x0025	8 and *	8 and *
0x0026	9 and (	9 and (
0x0027	0 and)	0 and)
0x0028	Return/Enter	Return
0x002A	Delete/Backspace	Delete

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Usage ID	Usage Name	Apple Function
0x002B	Tab	Tab
0x002C	Spacebar	Spacebar
0x002D	- and _	- and _
0x002E	= and +	= and +
0x002F	[and {	[and {
0x0030	] and }	] and }
0x0031	\and |	\and |
0x0033	; and :	; and :
0x0034	' and "	' and "
0x0035	Grave Accent and Tilde	` and ~
0x0036	, and <	, and <
0x0037	. and >	. and >
0x0038	/ and ?	/ and ?
0x0039	CapsLock	Caps Lock
0x004F	RightArrow	Right Arrow
0x0050	LeftArrow	Left Arrow
0x0051	DownArrow	Down Arrow
0x0052	UpArrow	Up Arrow
0x00E1	LeftShift	Left Shift
0x00E2	LeftAlt	Left Option and Alt
0x00E3	LeftGUI	Left Command and ⌘
0x00E5	RightShift	Right Shift
0x00E6	RightAlt	Right Option and Alt
0x00E7	RightGUI	Right Command and ⌘

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

JIS keyboards must also implement additional keys found on the Apple Wireless Keyboard (Japanese). Non-JIS keyboards must not implement these keys.

Table 39-2 Required HID Keyboard/Keypad Page controls for use by JIS keyboard components

Usage ID	Usage Name	Apple Function
0x0032	Non-US # and ~	see Apple Wireless Keyboard (Japanese)
0x0087	Keyboard International1	see Apple Wireless Keyboard (Japanese)
0x0089	Keyboard International3	see Apple Wireless Keyboard (Japanese)
0x0090	LANG1	Switch to Previous Language
0x0091	LANG2	Switch to Next Language

Accessory keyboards may implement individual keys that emit the following HID Keyboard/Keypad page usages:

Table 39-3 Optional HID Keyboard/Keypad Page controls for use by keyboard components

Usage ID	Usage Name	Apple Function
0x0029	Escape	Escape
0x00E0	LeftControl	Left Control
0x00E4	RightControl	Right Control
0x004A	Home	Home
0x004D	End	End
0x0054	Keypad /	Keypad /
0x0055	Keypad *	Keypad *
0x0056	Keypad -	Keypad -
0x0057	Keypad +	Keypad +
0x0058	Keypad Enter	Keypad Enter
0x0059	Keypad 1 and End	Keypad 1
0x005A	Keypad 2 and Down Arrow	Keypad 2

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Usage ID	Usage Name	Apple Function
0x005B	Keypad 3 and PageDn	Keypad 3
0x005C	Keypad 4 and Left Arrow	Keypad 4
0x005D	Keypad 5	Keypad 5
0x005E	Keypad 6 and Right Arrow	Keypad 6
0x005F	Keypad 7 and Home	Keypad 7
0x0060	Keypad 8 and Up Arrow	Keypad 8
0x0061	Keypad 9 and PageUp	Keypad 9
0x0062	Keypad 0 and Insert	Keypad 0
0x0063	Keypad . and Delete	Keypad .
0x0067	Keypad =	Keypad =

Accessory keyboards may implement individual keys that emit the following HID Consumer page usages:

Table 39-4 HID Consumer Page controls for use by keyboard components

Usage ID	Usage Name	Apple Function
0x0030	Power	Lock
0x0040	Menu	Home
0x006F	Display Brightness Increment	Brighter
0x0070	Display Brightness Decrement	Dimmer
0x00B5	Scan Next Track	Transport Right
0x00B6	Scan Previous Track	Transport Left
0x00CD	Play/Pause	Play/Pause
0x00CF	Voice Command	Siri
0x00E2	Mute	Mute
0x00E9	Volume Increment	Louder

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Usage ID	Usage Name	Apple Function
0x00EA	Volume Decrement	Softer
0x01AE	AL Keyboard Layout	Toggle Onscreen Keyboard
0x029D	AC Keyboard Layout Select	Toggle Language Switch/Emoji Keyboard
0x0221	AC Search	Spotlight
0x025B	Promote	iTunes Radio "Play More Like This"
0x025C	Demote	iTunes Radio "Never Play This Song"
0x0262	Add to Cart	iTunes Radio "Add to iTunes Wish List"

Note: Some HID Usages, such as Siri, may only be allowed from authenticated accessories, see Accessory Authentication (page 261).

39.2 HID Keyboard Examples

39.2.1 Keyboard Example HID Report Descriptor

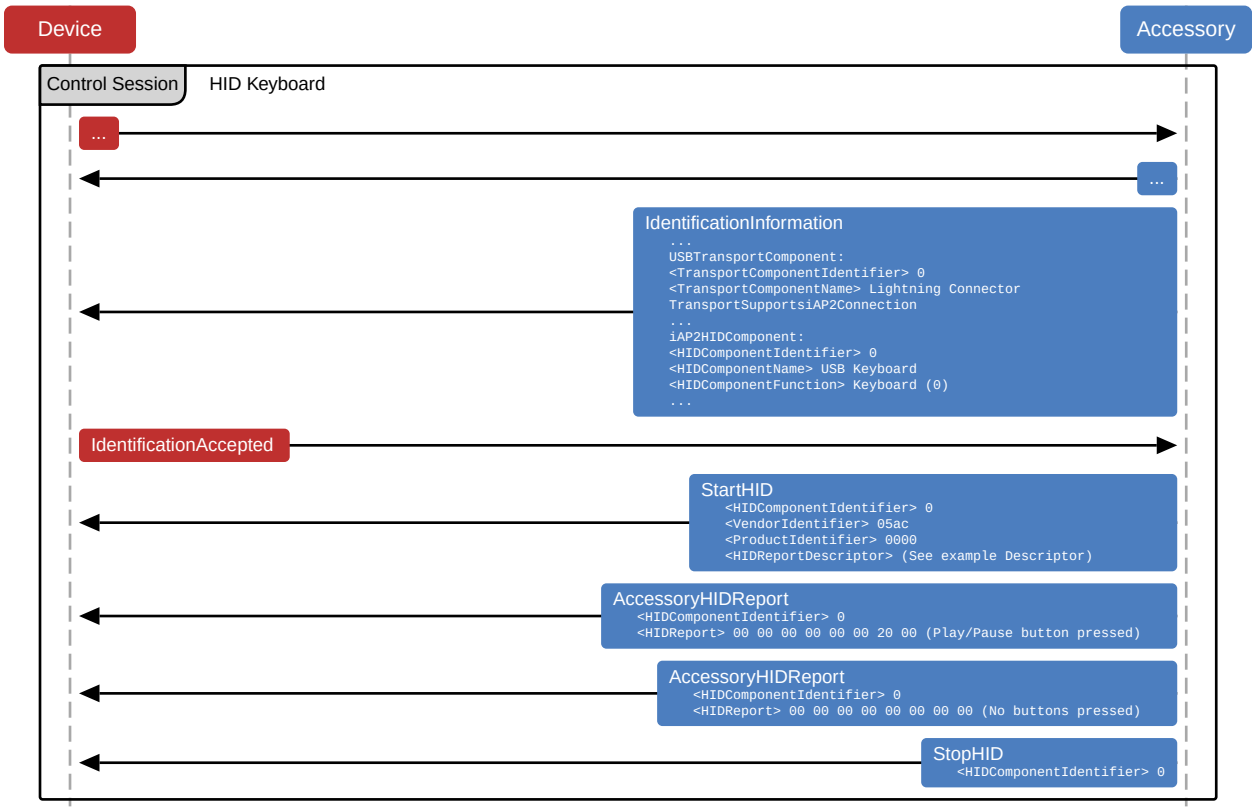
```
USAGE PAGE (Generic Desktop)            05 01
USAGE (Keyboard)                        09 06
COLLECTION (Application)                A1 01
  USAGE PAGE (LEDs)                     05 08
  LOGICAL MINIMUM (0)                   15 00
  LOGICAL MAXIMUM (1)                   25 01
  USAGE (Caps Lock)                     09 02
  REPORT SIZE (1)                       75 01
  REPORT COUNT (1)                      95 01
  OUTPUT (Data,Var,Abs)                 91 02
  REPORT SIZE (7)                       75 07
  REPORT COUNT (1)                      95 01
  OUTPUT (Cnst,Var,Abs)                 91 03
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
  USAGE PAGE (Keyboard)                   05 07
  USAGE MINIMUM (Keyboard Left Control)   19 E0
  USAGE MAXIMUM (Keyboard Right GUI)      29 E7
  REPORT SIZE (1)                         75 01
  REPORT COUNT (8)                        95 08
  INPUT (Data,Var,Abs)                    81 02
  LOGICAL MINIMUM (0)                     15 00
  LOGICAL MAXIMUM (255)                   26 FF 00
  USAGE MINIMUM (0)                       19 00
  USAGE MAXIMUM (255)                     2A FF 00
  REPORT SIZE (8)                         75 08
  REPORT COUNT (5)                        95 05
  INPUT (Data,Ary,Abs)                    81 00
  USAGE PAGE (Consumer Devices)           05 0C
  LOGICAL MINIMUM (0)                     15 00
  LOGICAL MAXIMUM (1)                     25 01
  USAGE (Menu)                            09 40
  USAGE (AC Search)                       0A 21 02
  USAGE (AL Keyboard Layout)              0A AE 01
  USAGE (Scan Previous Track)             09 B6
  USAGE (Play/Pause)                      09 CD
  USAGE (Scan Next Track)                 09 B5
  USAGE (Mute)                            09 E2
  USAGE (Volume Down)                     09 EA
  USAGE (Volume Up)                       09 E9
  USAGE (Power)                           09 30
  REPORT SIZE (1)                         75 01
  REPORT COUNT (10)                       95 0A
  INPUT (Data,Var,Abs)                    81 02
  REPORT SIZE (6)                         75 06
  REPORT COUNT (1)                        95 01
  INPUT (Cnst,Var,Abs)                    81 03
END COLLECTION                            C0
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

39.2.2 Keyboard Usage Example



39.3 Test Procedures

39.3.1 General Requirements

39.3.1.1 Product Design

Verify that the accessory:

1. Implements a physical keyboard that the user uses for tasks that might have otherwise been performed using the Apple device's onscreen keyboard.
2. Implements individual keys that emit the HID Keyboard/Keypad page usages listed in Table 39-1 (page 622).
3. Implements individual keys that emit the LANG1 and LANG2 HID Keyboard/Keypad page usages listed in Table 39-2 (page 625) if the accessory is a JIS keyboard.
4. Does not incorporate any other keyboard status LEDs other than Caps Lock.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

5. Has a mechanical key layout based on the ISO/IEC 9995-2, ANSI-INCITS 154-1988, or JIS X 6002-1980 standards.

39.3.1.2 HID Component

Verify that the accessory:

1. Supports the HID (Human Interface Device) (page 580) feature.
2. Complies with the requirements listed in HID Requirements (page 580).
3. Sets the *bCountryCode* field to the appropriate country code as defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor* if the HID component is implemented over a HID Native Transport (USB Host Mode).
4. Sets the *LocalizedKeyboardCountryCode* parameter in the StartHID (page 844) message to the appropriate country code as defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor* if the HID component is implemented over iAP2.

39.3.1.3 Direct User Action

Verify that the accessory:

1. Only sends HID usages in response to direct user action, i.e. pressing a key on the keyboard.
2. Does not send anything other than 'key pressed' or 'key released' for the key that has physically been pressed.
3. Does not emulate combinations of other keys (e.g. macros like Command-C for Paste).
4. Does not emulate timed user actions such as 'press-and-hold'.
5. Does not send different HID usages depending on the state of another control surface.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

40. HID Media Playback Remote

Accessories that implement HID Media Playback Remote components must support the HID (Human Interface Device) (page 580) feature and comply with all the requirements listed in HID Requirements (page 580).

40.1 HID Media Playback Remote Requirements

Each HID component that declares itself as a media playback remote must correspond to a physical set of media playback control surfaces. All HID usages sent from the media playback remote accessory must occur in response to direct user action, i.e. pressing one of the control surfaces. There are only three exceptions:

- If a media playback remote wishes to start playback immediately upon connection with the Apple device, it may send the `Play` HID usage.
- If a media playback remote wishes to start playback immediately upon changing audio sources to the Apple device, it may send the `Play` HID usage.
- If a media playback remote wishes to stop playback immediately upon changing audio sources from the Apple device, it may send the `Pause` HID usage.

> **Note:** Use of the `Play/Pause` HID usage for the exceptions listed above is not allowed.

Conversely, any accessory that has physical or virtual user controls that map to the media playback remote HID component must implement a media playback remote HID component and send corresponding HID usages. Mapping those controls to other iAP2 messages is grounds for failure to pass self certification.

The HID report descriptor for a media playback control must declare support for the HID Consumer Page and only send usages from the following table:

Table 40-1 HID Consumer Page controls for use by media playback remote components

Usage ID	Usage Name	Apple Function
0x00B0	Play	Play
0x00B1	Pause	Pause

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Usage ID	Usage Name	Apple Function
0x00B5	Scan Next Track	Transport Right
0x00B6	Scan Previous Track	Transport Left
0x00B9	Random Play	Shuffle
0x00BC	Repeat	Repeat
0x00BE	Tracking Normal	Reset audiobook playback speed to default
0x00CA	Tracking Increment	Increase audiobook playback speed
0x00CB	Tracking Decrement	Decrease audiobook playback speed
0x00CD	Play/Pause	Play/Pause
0x00CF	Voice Command	Siri
0x00E2	Mute	Mute
0x00E9	Volume Increment	Louder
0x00EA	Volume Decrement	Softer
0x025B	Promote	iTunes Radio "Play More Like This"
0x025C	Demote	iTunes Radio "Never Play This Song"
0x0262	Add to Cart	iTunes Radio "Add to iTunes Wish List"

If a user presses and holds the accessory control surface corresponding to the Scan Next Track or Scan Previous Track HID usage, Apple devices may scrub forwards or backwards within the current playing media item. Accessories must not present a separate 'Fast-Forward' or 'Reverse' control surface to the user for the same feature.

40.2 HID Media Playback Remote Examples

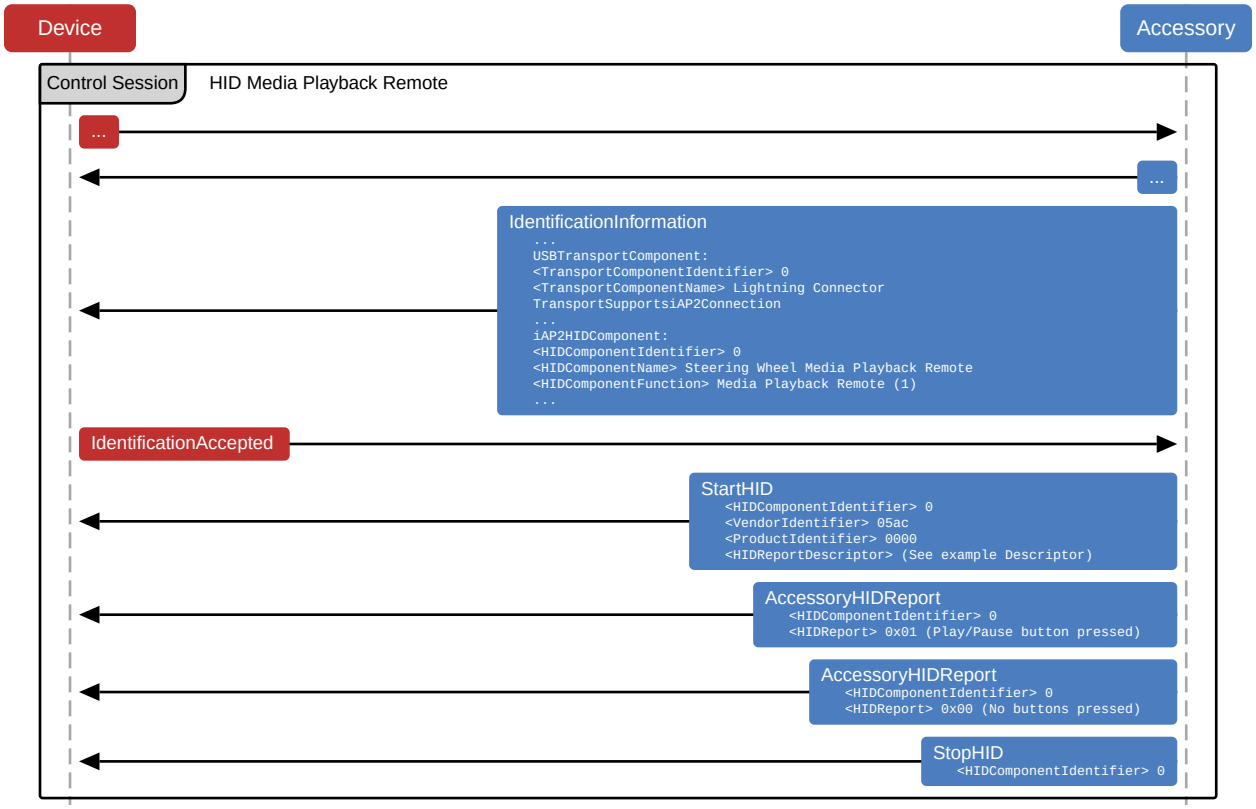
40.2.1 Media Playback Remote Example HID Report Descriptor

```
USAGE PAGE (Consumer Devices) 05 0C
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
USAGE (Consumer Control)       09 01
COLLECTION (Application)       A1 01
  LOGICAL MINIMUM (0)          15 00
  LOGICAL MAXIMUM (1)          25 01
  REPORT SIZE (1)              75 01
  REPORT COUNT (5)             95 05
  USAGE (Play/Pause)           09 CD
  USAGE (Scan Next Track)      09 B5
  USAGE (Scan Previous Track) 09 B6
  USAGE (Volume Up)            09 E9
  USAGE (Volume Down)          09 EA
  INPUT (Data,Var,Abs)         81 02
  REPORT SIZE (3)              75 03
  REPORT COUNT (1)             95 01
  INPUT (Cnst,Var,Abs)         81 03
END COLLECTION                 C0
```

40.2.2 Media Playback Remote Example Usage



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

40.3 Test Procedures

These tests must be performed on accessories that support audio playback and/or video playback.

40.3.1 Equipment Required

- Latest revision of ATS software
- ATS hardware
- The following test content:
  - Audiobooks
  - Music Albums
  - Video - Movies
  - Video - Music Video
  - Video - Podcasts
  - Video - TV Shows

All tests should be performed on each applicable content type.

40.3.2 Tests Cases

1. Verify Play and Pause work on the accessory as expected.
2. Verify Next Track and Previous Track work on the accessory as expected.
3. Verify correct Previous Track behavior when playing song for lesser than 2 seconds and greater than 2 seconds.
   a. For playback lesser than 2 seconds, playback should return to the track previous to current selection.
   b. For playback greater than 2 seconds, playback should return to the begging of current track.
4. When scrubbing forward to the end of the track, verify that the next track begins playing at normal speed.
5. When scrubbing backward to the beginning of the track, verify that the track starts to play at regular speed when the beginning of the track is reached.
6. With shuffle enabled, verify that pressing the Next Track button on head unit shuffles the next track (tracks will play at non-alphabetical or non-numerical order since they are shuffled).
   a. Create alphabetical or numerical playlist in iTunes and sync it to your Apple device.
   b. Connect the Apple device to a head unit.
   c. On the head unit, set shuffle tracks to "on".

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

7. With shuffle enabled, verify that the Album order is random, but that their tracks respect the order of the track numbers. On the head unit, set Shuffle all albums to "on".
8. With shuffle enabled, verify that pressing the Next Track button on the head unit shuffles albums.
    a. Navigate to Artists.
    b. Play all tracks from an Album, then set shuffle Album.
9. Verify Shuffle behavior in each audio / video category.
10. Verify the track metadata on the head unit display matches the track title of the Now Playing track on the Apple device when audio is set to Shuffle (mixed playlist, Podcasts, Audiobooks).
11. Verify "Repeat One" and "Repeat" All behaviors.
12. Verify "Repeat All" behavior with the first track in the playlist and last track in the playlist.
13. Verify Audiobook resumes where it was interrupted.
14. Verify that a track with special characters in song titles (such as Emoji) play and display as expected. Do note that special characters may be displayed as a substitute character such as an open square.
15. Verify elapsed and remaining track time updates properly on accessory.
16. Verify chapter info if the accessory supports it.
17. Verify if you can access the Playlists, Artists and Albums categories and select an audio track to play.
18. Verify a newly created playlist (on the Apple device) is accessible on the accessory.
19. Verify that there are no issues scrolling through long playlists (about 2000 songs) on the accessory.
20. Verify elapsed time shows correctly in long Audiobooks if te accessory supports it (i.e., hours, minutes, seconds).
21. Verify that when switching from a playlist of 3 tracks to a playlist of 4 tracks that the correct track plays and that the metadata is correctly displayed.
    a. Attach Apple device to head unit.
    b. Select a playlist containing 3 tracks.
    c. Start playback.
    d. On the Apple device, change to a playlist containing 4 tracks and select the last track in the playlist.
    e. Verify that the correct track plays and that the metadata is correctly displayed on the head unit.
22. Verify that when switching from a playlist of 4 tracks to a playlist of 3 tracks the correct track plays and that the metadata is correctly displayed.
    a. Attach an Apple device to a head unit.
    b. Select a playlist containing 4 tracks.
    c. Start playback.
    d. On the Apple device, change to a playlist containing 3 tracks and select the last track in the playlist.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

e. Verify that the correct track plays and that the metadata is correctly displayed on the head unit.

40.3.3 iAP2

1. Verify that each HID component that declares itself as a media playback remote must correspond to a physical set of media playback control surfaces. There are three exceptions:
   - If a media playback remote wishes to start playback immediately upon connection with the Apple device, it may send the Play HID usage.
   - If a media playback remote wishes to start playback immediately upon changing audio sources to the Apple device, it may send the Play HID usage.
   - If a media playback remote wishes to stop playback immediately upon changing audio sources from the Apple device, it may send the Pause HID usage.

   Use of Play/Pause for these purposes is not allowed.
2. Verify that if the accessory sends an AccessoryHIDReport (page 845) to start playback directly after initial connect, that the accessory sends Play.
3. Press and verify the functionality of all buttons corresponding to HID commands (Play, Pause, Play/Pause, Scan Next Track, Scan Previous Track, etc.)
4. Verify that a HID media playback remote does not contain prohibited buttons.
5. Verify that the media being played on the Apple device matches the Now Playing metadata on the accessory.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

41. Location Information

The location feature provides compatible accessories with a means to provide external Global Navigation Satellite System (GNSS) information and other sensor data (like speed, gyro) in the form of National Marine Electronics Association (NMEA) sentences to Apple devices. Apple devices that support this feature can take advantage of this additional information to augment their built-in location services. For example, some external Location-enabled accessories can provide more accurate or more frequent position updates. Additionally, Apple devices can conserve power by using location information from a self-powered external accessory instead of using on-board positioning technologies.

41.1 Location Information Requirements

All accessories that support the Location feature via iAP2 must send or receive the following iAP2 control session message(s):

StartLocationInformation (page 846)

LocationInformation (page 847)

StopLocationInformation (page 848)

Accessories must provide location information to the Apple device as follows:

- Accessories with GNSS suitable for use in navigation must implement Global Navigation Satellite System (GNSS) Mode (page 638).
- Accessories without GNSS suitable for use in navigation must implement Sensors Mode (page 639).

Table 41-1 Supported NMEA Sentences and Maximum/Recommended Rates

Sentence Type	Maximum Rate	Recommended Rate
GPGGA	4 sentences per second	1 sentence per second
GPRMC	4 sentences per second	1 sentence per second
GPGSV	12 sentences per second	3-4 sentences per second
GPHDT	4 sentences per second	2 sentences per second

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Sentence Type	Maximum Rate	Recommended Rate
PASCD	50 sentences per second	1-4 sentences per second
PAGCD	50 sentences per second	1-4 sentences per second
PAACD	50 sentences per second	1-4 sentences per second

Accessories must provide location information to the Apple device in the form of NMEA comma-separated sentences at a minimum 1 Hz update rate.

All NMEA sentences provided to the Apple device must be NULL terminated.

> **Note:** NMEA sentences supplied to the Apple device must be live data. Accessories that simulate data are not permitted.

Three custom sentences in NMEA format are available: PASCD, PAGCD, and PAACD. For these sentences, the recommended sensor update rate is 10-50 Hz. Data may be accumulated in a buffer for up to 1 second.

41.1.1 Global Navigation Satellite System (GNSS) Mode

Accessories that implement GNSS Mode must meet the requirements in this section.

Accessories must provide:

- GPGGA
- GPRMC
- PASCD

Accessories may additionally provide:

- GPGSV
- GPHDT

In this mode, it is recommended that accessories make use of inertial sensor and vehicle sensor technology and that the GPGGA and GPRMC information provided is derived from GNSS coupled with data from those sensors (e.g. wheel speed and gyro).

In this mode, the minimum update rate is 1 Hz.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Latitude data from the accessory must fall within the range of -90 to +90 degrees, and longitude data from the accessory must fall within the range of -180 to +180 degrees.

NMEA Sentence Fields (page 639) lists the fields of supported NMEA sentences.

41.1.2 Sensors Mode

Accessories that implement Sensors Mode must meet the requirements in this section.

Accessories must provide:

- PASCD
- PAGCD (at least yaw rate)

Accessories may additionally provide:

- GPHDT
- PAACD

In this mode, sensor data provided by the accessory enables the Apple device's GNSS to provide position and velocity data.

In this mode, the minimum sensor update rate is 10 Hz.

For PAGCD & PAACD, the vehicle coordinate frame is defined as follows:

- X-axis pointing in the forward along-track direction of the vehicle.
- Z-axis pointing down (positive yaw rate is clockwise).
- Y-axis defined by a right-handed convention; positive pitch angle means the vehicle is driving up-hill.

NMEA Sentence Fields (page 639) lists the fields of supported NMEA sentences.

41.1.3 NMEA Sentence Fields

This section lists the fields of supported NMEA sentences. Commas must be used to delimit fields when constructing NMEA sentences. See Location Information Examples (page 646).

41.1.3.1 Standard NMEA Sentences

Fields listed as Required must be populated when the sentence is transmitted; other fields may be left empty.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 41-2 GPGGA Sentence Fields

Field	Required	Description
1	yes	Sentence ID: $GPGGA
2	yes	UTC time of position (hhmmss.ss)
3	yes	Latitude of position (ddmm.mmmmmm)
4	yes	Latitude of position (N or S)
5	yes	Longitude of position (dddmm.mmmmmm)
6	yes	Longitude of position (E or W)
7	yes	Fix Quality indicator (0=Invalid, 1=Valid GPS Fix, 2=Differential GPS Fix, 6=Dead Reckoned Location)
8	no	Number of satellites in view (xx)
9	no	Horizontal dilution of precision (x.x)
10	yes	Antenna altitude above mean sea level (x.x)
11	yes	Units of antenna altitude, must be meters (M)
12	no	Geoidal separation (x.x), recommended
13	yes	Units of geoidal separation, must be meters (M)
14	no	Age of differential GPS data in seconds (x.x)
15	no	Differential reference station ID (xxxx)
16	yes	Checksum, two characters: the bitwise exclusive OR of all characters in the sentence between $ and *. All letters in the checksum must be uppercase.

Note: If the GPGGA Fix Quality indicator is 'invalid', the sentence will be discarded by the Apple device.

Table 41-3 GPRMC Sentence Fields

Field	Required	Description
1	yes	Sentence ID: $GPRMC

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Field	Required	Description
2	yes	UTC time of position (hhmmss.ss)
3	yes	Data status (A=OK, V=Navigation receiver warning, X=specific for a special case of CarPlay)
4	yes	Latitude of position (ddmm.mmmmmm)
5	yes	Latitude of position (N or S)
6	yes	Longitude of position (dddmm.mmmmmm)
7	yes	Longitude of position (E or W)
8	yes	Ground speed in knots
9	yes	Track made good in degrees True (range 0.0000-359.9999)
10	yes	UTC date (DDMMYY)
11	no	Magnetic variation in degrees
12	no	Magnetic variation (E or W)
13	no	Mode indicator (A=Autonomous, D=Differential, E=Approximation, N=Invalid Data)
14	yes	Checksum, two characters: the bitwise exclusive OR of all characters in the sentence between $ and *. All letters in the checksum must be uppercase.

Table 41-4 GPGSV Sentence Fields

Field	Required	Description
1	yes	Sentence ID: $GPGSV
2	yes	Total number of sentences for this epoch of data
3	yes	Sentence Number
4	yes	Number of satellites in view
5	no	Satellite number
6	no	Elevation (degrees)
7	no	Azimuth (degrees)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Field	Required	Description
8	no	Carrier to noise density ratio (dB-Hz)
9	no	Satellite number
10	no	Elevation (degrees)
11	no	Azimuth (degrees)
12	no	Carrier to noise density ratio (dB-Hz)
13	no	Satellite number
14	no	Elevation (degrees)
15	no	Azimuth (degrees)
16	no	Carrier to noise density ratio (dB-Hz)
17	no	Satellite number
18	no	Elevation (degrees)
19	no	Azimuth (degrees)
20	no	Carrier to noise density ratio (dB-Hz)
21	yes	Checksum, two characters: the bitwise exclusive OR of all characters in the sentence between $ and *. All letters in the checksum must be uppercase.

Table 41-5 GPHDT Sentence Fields

Field	Required	Description
1	yes	Sentence ID: $GPHDT
2	yes	Heading: the best estimate of the clockwise angle in degrees between True North (i.e. Geodetic North) and the forward along-track direction of the vehicle (range 0.0000-359.9999)
3	yes	Degrees true (always T)
4	yes	Checksum, two characters: the bitwise exclusive OR of all characters in the sentence between $ and *. All letters in the checksum must be uppercase.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: It is possible for the GPHDT angle to differ from the vehicle direction of travel as indicated in the GPRMC message. For example, if the vehicle is moving in reverse, these angles are different by 180 degrees.

41.1.3.2 Custom NMEA Sentences

All fields are required for sentences in this section unless otherwise specified.

Table 41-6 PASCD Sentence Fields

Field	Name	Type	Description
1	$PASCD	char	sentence ID for vehicle speed data
2	timestamp	float	Reference time in seconds with a resolution of 3 decimal places (for example, 12.345 seconds); timestamps increase monotonically but can roll over (for example, 0.000-86399.999 for daily rollover)
3	sensorType	char	C = combined left and right wheel speed sensors. Other types not supported.
4	transmissionState	char	U = unknown
			P = park
			R = reverse; this value should be used if reverse was engaged at any time within the interval of this data
			D = driving forward
			N = neutral
5	slipDetect	uint	1 = a wheel speed slippage was detected within the interval of this data sentence; 0 = no slip was detected
6	sampleCount	uint	the number of sensor samples included in this sentence (1-50)
7	timeOffset_i	float	the time offset in seconds of the ith sensor sample set from the reference timestamp, with a resolution of two decimal places (for example, 0.12 seconds)
8	speed_i	float	vehicle speed in meters/second at timeOfffset_i, with a minimum resolution of 3 decimal places (for example, 10.123 m/s)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Field	Name	Type	Description
9	Checksum	hex	Two characters: the bitwise exclusive OR of all characters in the sentence between $ and *. All letters in the checksum must be uppercase.

Table 41-7 PAGCD Sentence Fields

Field	Name	Type	Description
1	$PAGCD	char	sentence ID for vehicle gyro data
2	timestamp	float	Reference time in seconds with a resolution of 3 decimal places (for example, 12.345 seconds); timestamps increase monotonically but can roll over (for example, 0.000-86399.999 for daily rollover)
3	sampleCount	uint	the number of sensor samples included in this sentence (1-50)
4	timeOffset_i	float	the time offset in seconds of the ith sensor sample set from the reference timestamp, with a resolution of two decimal places (for example, 0.12 seconds)
5	xAxisSample_i	float	ith x-axis sensor (degrees/second) with a minimum resolution of 4 decimal places (for example, 33.1234 deg/s); this field can be left empty if not available
6	yAxisSample_i	float	ith y-axis sensor (degrees/second) with a minimum resolution of 4 decimal places (for example, 33.1234 deg/s); this field can be left empty if not available
7	zAxisSample_i	float	ith z-axis sensor (degrees/second) with a minimum resolution of 4 decimal places (for example, 33.1234 deg/s); this field is required
8	Checksum	hex	Two characters: the bitwise exclusive OR of all characters in the sentence between $ and *. All letters in the checksum must be uppercase.

Table 41-8 PAACD Sentence Fields

Field	Name	Type	Description
1	$PAACD	char	sentence ID for vehicle accelerometer data

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Field	Name	Type	Description
2	timestamp	float	Reference time in seconds with a resolution of 3 decimal places (for example, 12.345 seconds); timestamps increase monotonically but can roll over (for example, 0.000-86399.999 for daily rollover)
3	gValue	float	reference gravity magnitude in meters per second per second with a resolution of 5 decimal places (for example, 9.81001 m/s2)
4	sampleCount	uint	the number of sensor samples included in this sentence (1-50)
5	timeOffset_i	float	the time offset in seconds of the ith sensor sample set from the reference timestamp, with a resolution of two decimal places (for example, 0.12 seconds)
6	xAxisSample_i	float	ith x-axis sensor gravity reading with a minimum resolution of 4 decimal places (for example, 0.1234 g); this field can be left empty if not available
7	yAxisSample_i	float	ith y-axis sensor gravity reading with a minimum resolution of 4 decimal places (for example, 0.1234 g); this field can be left empty if not available
8	zAxisSample_i	float	ith z-axis sensor gravity reading with a minimum resolution of 4 decimal places (for example, 0.1234 g); this field can be left empty if not available
9	Checksum	hex	Two characters: the bitwise exclusive OR of all characters in the sentence between $ and *. All letters in the checksum must be uppercase.

41.2 Location Information Usage

The accessory must enumerate what types of location information it can send during Accessory Identification. For more information, see Table 59-22 (page 814).

Once the accessory has successfully identified itself, the device will be aware that an external source of location data is available. When the device is ready to accept location information, it will send a StartLocationInformation (page 846) message to the accessory detailing what types of location data it will accept.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

> **Note:** The StartLocationInformation (page 846) message used to have a MinimumIntervalInMilliseconds parameter. That parameter is deprecated. Accessories must ignore it if it is present in received StartLocationInformation (page 846) messages.

Once StartLocationInformation (page 846) has been received, the accessory must send LocationInformation (page 847) messages containing one or more of the specified NMEA sentences. Bundling multiple sentences into one LocationInformation (page 847) message is recommended. The sentences must be sent to the device according to the rate specifications in Table 41-1 (page 637).

Once the device no longer needs external location data, it will send a StopLocationInformation (page 848) message to the accessory, and the accessory must cease sending LocationInformation (page 847) messages. The accessory must be prepared to start and stop providing location data to the device at any time.

41.3 Location Information Examples

41.3.1 GPGGA Sentence Example

```
$GPGGA,225833.0,3719.951324,N,12201.808796,W,1,09,0.6,69.8,M,-27.0,M,,*5F
```

41.3.2 GPRMC Sentence Example

```
$GPRMC,225833.0,A,3719.951324,N,12201.808796,W,0.0,26.1,060813,0.0,E,A*2F
```

41.3.3 GPGSV Sentence Example

```
$GPGSV,3,1,11,32,80,095,21,20,61,310,36,31,42,046,15,01,39,226,35*76
$GPGSV,3,2,11,23,38,285,32,16,17,147,15,11,17,210,34,13,12,268,33*74
$GPGSV,3,3,11,48,45,198,,51,44,156,,46,39,143,*4C
```

41.3.4 GPHDT Sentence Example

```
$GPHDT,0.0000,T*05
$GPHDT,359.9999,T*0A
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

41.3.5 PASCD Sentence Examples

10 Hz sensor data sent at 1 Hz while accelerating forward from 0 to 10 m/s (1 sentence, 139 characters long):

```
$PASCD,1000.001,C,D,0,10,0.00,0.123,0.10,1.123,0.20,2.123,0.30,3.123,0.40,
4.123,0.50,5.123,0.60,6.123,0.70,7.123,0.80,8.123,0.90,9.123*41
```

10 Hz sensor data sent at 1 Hz while accelerating in reverse from 0 to 10 m/s (1 sentence, 139 characters long):

```
$PASCD,1000.001,C,R,0,10,0.00,0.123,0.10,1.123,0.20,2.123,0.30,3.123,0.40,
4.123,0.50,5.123,0.60,6.123,0.70,7.123,0.80,8.123,0.90,9.123*57
```

10 Hz sensor data sent at 1 Hz while accelerating in reverse from 0 to 10 m/s with wheel slippage detected (1 sentence, 139 characters long):

```
$PASCD,1000.001,C,R,1,10,0.00,0.123,0.10,1.123,0.20,2.123,0.30,3.123,0.40,
4.123,0.50,5.123,0.60,6.123,0.70,7.123,0.80,8.123,0.90,9.123*56
```

50 Hz sensor data sent at 2 Hz while accelerating forward from 0 to 10 m/s (2 sentences, 304 characters each):

```
$PASCD,1000.001,C,D,0,25,0.00,0.123,0.02,0.323,0.04,0.523,0.06,0.723,0.08,
0.923,0.10,1.123,0.12,1.323,0.14,1.523,0.16,1.723,0.18,1.923,0.20,2.123,0.22,
2.323,0.24,2.523,0.26,2.723,0.28,2.923,0.30,3.123,0.32,3.323,0.34,3.523,0.36,
3.723,0.38,3.923,0.40,4.123,0.42,4.323,0.44,4.523,0.46,4.723,0.48,4.923*77

$PASCD,1000.501,C,D,0,25,0.00,5.123,0.02,5.323,0.04,5.523,0.06,5.723,0.08,
5.923,0.10,6.123,0.12,6.323,0.14,6.523,0.16,6.723,0.18,6.923,0.20,7.123,0.22,
7.323,0.24,7.523,0.26,7.723,0.28,7.923,0.30,8.123,0.32,8.323,0.34,8.523,0.36,
8.723,0.38,8.923,0.40,9.123,0.42,9.323,0.44,9.523,0.46,9.723,0.48,9.923*73
```

41.3.6 PAGCD Sentence Examples

10 Hz yaw-rate only (single axis) sensor data sent at 1 Hz, while turning at 12.1234 deg/s (1 sentence, 174 characters long):

```
$PAGCD,20000.001,10,
0.00,,,12.1234,0.10,,,12.1234,0.20,,,12.1234,0.30,,,
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

```
12.1234,0.40,,,12.1234,0.50,,,12.1234,0.60,,,12.1234,0.70,,,12.1234,0.80,,,
12.1234,0.90,,,12.1234*7C
```

50 Hz yaw-rate only (single axis) sensor data sent at 1 Hz, while turning at 12.1234 deg/s (2 sentences, 399 characters each):

```
$PAGCD,20000.002,25,0.00,,,12.1234,0.02,,,12.1234,0.04,,,12.1234,0.06,,,
12.1234,0.08,,,12.1234,0.10,,,12.1234,0.12,,,12.1234,0.14,,,12.1234,0.16,,,
12.1234,0.18,,,12.1234,0.20,,,12.1234,0.22,,,12.1234,0.24,,,12.1234,0.26,,,
12.1234,0.28,,,12.1234,0.30,,,12.1234,0.32,,,12.1234,0.34,,,12.1234,0.36,,,
12.1234,0.38,,,12.1234,0.40,,,12.1234,0.42,,,12.1234,0.44,,,12.1234,0.46,,,
12.1234,0.48,,,12.1234*43

$PAGCD,20000.502,25,0.00,,,12.1234,0.02,,,12.1234,0.04,,,
12.1234,0.06,,,12.1234,0.08,,,12.1234,0.10,,,12.1234,0.12,,,12.1234,0.14,,,
12.1234,0.16,,,12.1234,0.18,,,12.1234,0.20,,,12.1234,0.22,,,12.1234,0.24,,,
12.1234,0.26,,,12.1234,0.28,,,12.1234,0.30,,,12.1234,0.32,,,12.1234,0.34,,,
12.1234,0.36,,,12.1234,0.38,,,12.1234,0.40,,,12.1234,0.42,,,12.1234,0.44,,,
12.1234,0.46,,,12.1234,0.48,,,12.1234*46
```

41.3.7 PAACD Sentence Example

10 Hz x-axis and z-axis only sensor data sent at 1 Hz, (1 sentence, 236 characters long):

```
$PAACD,40000.001,9.80665,10,0.00,0.0340,,1.0000,0.10,-0.0106,,
1.0000,0.20,0.0283,,1.0000,0.30,0.0298,,1.0000,0.40,0.0412,,
1.0000,0.50,-0.0302,,1.0000,0.60,-0.0165,,1.0000,0.70,0.0268,,
1.0000,0.80,-0.0222,,1.0000,0.90,0.0054,,1.0000*7B
```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

42. Media Library Access

The media library feature allows accessories to download the metadata contents of an Apple device's media libraries (not the media items themselves) and request playback of media items. The feature is divided into three subfeatures: information, updates, and playback. Media library information informs the accessory about available media libraries on the device. Media library updates provide an accessory with an updated view of the contents of a particular media library. Media library playback allows the accessory to request playback of one or more items from the media library.

42.1 Media Library Access Requirements

Accessories that use this feature must have sufficient computational and storage resources to store and continuously update their mirror of the Apple device's media libraries. Accessories may choose not to receive certain types of media item metadata to minimize storage and computational requirements. If an accessory runs out of storage capacity while receiving media library updates it must not use this feature until sufficient storage capacity has been freed to resume receiving media library updates.

Additionally, accessories must allow the user to access and request playback of items from multiple media libraries, such as the device's main music library and the library that corresponds to iTunes Radio.

42.1.1 Media Library Information Requirements

All accessories that support the Media Library Information feature via iAP2 must send or receive the following iAP2 control session message(s):

StartMediaLibraryInformation (page 848)

MediaLibraryInformation (page 849)

StopMediaLibraryInformation (page 849)

42.1.2 Media Library Updates Requirements

All accessories making use of Media Library Updates must support Media Library Information.

The accessory must implement and declare an iAP2 file transfer session as specified in File Transfer Session (page 795).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Additionally, the accessory must not request a full database update using the StartMediaLibraryUpdates (page 850) message more than once during an iAP2 connection. The accessory must be capable of processing incremental media library updates from the device and persist its media library mirror across connections to minimize the occurrence of full updates.

All accessories that support the Media Library Updates feature via iAP2 must send or receive the following iAP2 control session message(s):

StartMediaLibraryUpdates (page 850)

MediaLibraryUpdate (page 852)

StopMediaLibraryUpdates (page 856)

42.1.3 Media Library Playback Requirements

All accessories making use of Media Library Playback must support Media Library Updates.

All accessories that support the Media Library Playback feature via iAP2 may also send or receive the following iAP2 control session message(s):

PlayMediaLibraryCurrentSelection (page 856)

PlayMediaLibraryItems (page 856)

PlayMediaLibraryCollection (page 857)

PlayMediaLibrarySpecial (page 858)

PlayMediaLibraryCurrentSelection (page 856) must only be sent by the accessory in response to a direct user action, such as pressing a button on the accessory labeled 'iPod' or 'iPhone'. Similarly, PlayMediaLibraryItems (page 856), PlayMediaLibraryCollection (page 857), and PlayMediaLibrarySpecial (page 858) must only be sent by the accessory in response to a direct user action such as selecting a specific group of media items, or a media collection, and pressing a 'Play' or 'Select' button on the accessory.

All accessories must not assume that the playback queue is persistent on accessory detach.

Note: Accessories must not send Media Library Playback messages in response to any direct user actions that map to Media Remote Control HID usages (see HID Media Playback Remote Requirements (page 631)). For example, if an accessory has a 'Next Track' button, the accessory must generate a 'Next Track' HID usage when that button is pressed, not a PlayMediaLibraryItems (page 856) message.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Autonomous generation of these messages without being preceded by direct user action is grounds for failure to pass self certification. If an accessory wishes to resume media playback upon initial connection to the Apple device, it must send a `Play` HID usage as specified in HID (Human Interface Device) (page 580).

Note that `Play` must not be sent upon connection when the accessory and the Apple device both support the CarPlay feature. See Session Establishment (page 394).

42.2 Media Library Access Usage

42.2.1 Media Library Information Usage

Accessories must send StartMediaLibraryInformation (page 848) to start receiving MediaLibraryInformation (page 849) messages. Each MediaLibraryInformation (page 849) message contains information about all available media libraries on the Apple device. The `MediaLibraryName` for each library may be used to populate the accessory display, and the accessory must use the assigned `MediaLibraryUniqueIdentifier` if it wants to receive updates or request playback for that media library.

The accessory must send StopMediaLibraryInformation (page 849) when it stops displaying or making use of media library information. For example, if an accessory supports multiple audio sources, including an Apple device, the accessory must stop receiving media library information from the Apple device if the user takes action to switch to another audio source.

42.2.2 Media Library Updates Usage

To make use of media library updates, accessories must send a StartMediaLibraryUpdates (page 850) message enumerating the media item and/or media playlist properties that they are interested in for a particular media library. MediaLibraryUpdate (page 852) messages will be sent by the device when the contents of the active media library change relative to the last database version sent to the accessory. Updates will be sent only for properties for which the accessory specifically requested updates.

Accessories must be prepared to receive duplicate MediaLibraryUpdate (page 852) messages. In such situations, the accessory does not need to apply the duplicate update. Additionally, non-actionable update messages, such as a message that instructs the accessory to delete a media item that is not present in the accessory's media library, must be ignored by the accessory. Other actions taken by the user, other devices owned by the user, or other accessories may result in those situations; the accessory must not treat them as error states.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Accessories that register for media item and/or media playlist property updates via the StartMediaLibraryUpdates (page 850) message must process media item and/or media playlist deletions when present in received MediaLibraryUpdate (page 852) messages. Such deletions must be processed immediately and reflected in any user-visible displays. Accessory selection of a deleted media item or media playlist for playback after notice of the deletion has been received is grounds for failure to pass self certification.

The accessory must send StopMediaLibraryUpdates (page 856) when it stops displaying or making use of media library updates. For example, if an accessory supports multiple audio sources, including an Apple device, the accessory must stop receiving media library updates from the Apple device if the user takes action to switch to another audio source.

Regardless of whether the accessory sends StopMediaLibraryUpdates (page 856) or disconnects from the Apple device, the accessory must store the last received `MediaLibraryRevision` parameter value. When the accessory next sends StartMediaLibraryUpdates (page 850) to the same Apple device it must set `LastKnownMediaLibraryRevision` to that value to avoid having to retrieve a complete media library update. If the accessory has limited storage capacity it may choose to only retain a media library snapshot from one Apple device at a time.

Accessories may receive a percent completion of the current set of media library updates from the Apple device. This may be useful for informing the user of progress when there are a large number of updates from previous revision. For more information, see the `MediaLibraryUpdateProgress` parameter in StartMediaLibraryUpdates (page 850) and MediaLibraryUpdate (page 852).

If an accessory does not recognize the media library type (see Table 59-107 (page 849)), the media library must be ignored.

Accessories may receive media library updates indicating whether the Apple device's onscreen UI is not presenting remotely stored media items (such as content from iTunes in the Cloud) in certain situations like Airplane Mode. Additionally, accessories may choose to be informed whether a particular media item is resident on the Apple device or not. For more information, see the `MediaItemPropertyIsResidentOnDevice` and `MediaLibraryIsHidingRemoteItems` parameters in StartMediaLibraryUpdates (page 850), MediaLibraryUpdate (page 852), Table 59-113 (page 853), and Table 59-110 (page 850). Accessories must not assume that any particular media item is resident on the Apple device until a MediaLibraryUpdate message has been received from the device explictily indicating that it is resident. If a media library item is resident on the device, the accessory may choose to indicate that to the user it may be played in situations where the Apple device may be unable to play other non-resident media library items, such as when the device has poor network connectivity.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

42.2.3 Media Library Playback Usage

Accessories may make use of Media Library Playback to either play a media library item collection (such as a playlist) or even an arbitrary set of media items that has been assembled by the accessory. Every media item in that set must be obtained from Media Library Updates.

To resume playback of a library's current selection, the accessory must send PlayMediaLibraryCurrentSelection (page 856).

To start playback of specific media items or a media library collection, the accessory must send PlayMediaLibraryItems (page 856) or PlayMediaLibraryCollection (page 857), respectively. The library's current selection will be replaced with the new items or collection contents.

Music, Podcast, Audiobooks and iTunesU media items should not be mixed in the list of items to play using PlayMediaLibraryItems (page 856).

The number of items that can be specified for playback in the PlayMediaLibraryItems (page 856) message is bounded by the maximum possible size of an iAP2 message. Accessories should keep the list of media items to a reasonable length.

> **Note:** If an invalid or no longer available item is referred to in either a PlayMediaLibraryItems (page 856) or PlayMediaLibraryCollection (page 857) message, then the device will ignore those items and subsequently proceed to add the remaining, valid, items to the current selection.

To start playback of all songs in the media library, the accessory should use the PlayMediaLibrarySpecial (page 858) message using the `AllSongs` parameter. The library's current selection will be replaced.

Additionally, accessories that request playback of media items must not anticipate or assume corresponding state changes by the Apple device. For further explanation, see Presentation of Apple Device Updates (page 61).

42.3 Test Procedures

42.3.1 Media Library Info

1. Verify that the following iAP2 control session message(s) are sent or received:
   - StartMediaLibraryInformation
   - MediaLibraryInformation
   - StopMediaLibraryInformation

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Verify that the accessory sends a StopMediaLibraryInformation message when switching to a non-Apple device source.

42.3.2 Media Library Updates

1. Verify that the accessory supports Media Library Information.
2. Verify that the following iAP2 control session message(s) are sent or received:
   - StartMediaLibraryUpdates (page 850)
   - MediaLibraryUpdate (page 852)
   - StopMediaLibraryUpdates (page 856)
3. Verify that the StartMediaLibraryUpdates message is sent only once during an iAP2 connection.
4. Verify that deleting media on the Apple device results in:
   - No errors on the accessory.
   - The accessory immediate reflects these changes via its user interface.
5. Verify that the accessory sends a StopMediaLibraryUpdates message when switching to a non-Apple device source.
6. Verify the value of the LastKnownMediaLibraryRevision parameter via the following steps:
   a. Set the accessory to a Apple device source.
   b. Switch playback to a non-Apple device source.
   c. Note the value of the MediaLibraryRevision parameter of the last sent MediaLibraryUpdate message prior to switching sources.
   d. Switch back to an Apple device source.
7. Verify that the LastKnownMediaLibraryRevision parameter of the StartMediaLibraryUpdate message has been set to the same value.

42.3.3 Media Library Playback

1. Verify that the accessory supports Media Library Updates.
2. Verify that the following iAP2 control session message(s) are sent or received:
   - PlayMediaLibraryCurrentSelection (page 856)
   - PlayMediaLibraryItems (page 856)
   - PlayMediaLibraryCollection (page 857)
   - PlayMediaLibrarySpecial (page 858)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Verify that the accessory allows the user to access and request playback of items from multiple media libraries, such as the device's main music library and the library that corresponds to iTunes Radio.
4. Verify that the PlayMediaLibraryCurrentSelection message is only be sent by the accessory in response to a direct user action, such as pressing a button on the accessory labeled 'iPod' or 'iPhone'.
5. Verify that the PlayMediaLibraryItems, PlayMediaLibraryCollection, and PlayMediaLibrarySpecial message are only sent by the accessory in response to a direct user action such as selecting a specific group of media items, or a media collection, and pressing a 'Play' or 'Select' button on the accessory.
6. Verify that the accessory does not send the Media Library Playback messages in response to any direct user actions that map to Media Remote Control HID usages. For example, if an accessory has a 'Next Track' button, the accessory must generate a 'Next Track' HID usage when that button is pressed, not a PlayMediaLibraryItems (page 856) message.
7. Verify that the accessory does not send Media Library Playback messages without direct user action.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

43. Musical Instrument Digital Interface (MIDI)

Starting with iOS 4.2.1, some Apple devices provide support for USB Host Mode MIDI. Compatible accessories can interface directly with iOS apps that make use of the Core MIDI framework.

To support MIDI, accessories must follow the requirements in *Universal Serial Bus Definition for MIDI Devices*, Release 1.0, available at www.usb.org. Developers should test their accessory designs against the latest OS X MIDI driver; the application Audio MIDI Setup, in the OS X Applications/Utilities folder, can be used to verify accessory compatibility.

Every accessory that supports the USB Host Mode MIDI must implement a MIDI Streaming IN endpoint.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

44. Now Playing Updates

The Now Playing feature enables an accessory to display information about the current "Now Playing" media source and media item on an Apple device. Media sources include both the built-in Music and Video apps on Apple devices and certain third-party iOS apps that support the generation of Now Playing metadata (see *MPNowPlayingInfoCenter* in the iOS SDK documentation). Accessories must be prepared for the Now Playing media source and media item to change at any time, whether the accessory requested the change or not.

44.1 Now Playing Updates Requirements

All accessories that support the Now Playing feature via iAP2 must send or receive the following iAP2 control session message(s):

StartNowPlayingUpdates (page 858)

NowPlayingUpdate (page 860)

StopNowPlayingUpdates (page 863)

All accessories that support this feature must demonstrate correct handling of Now Playing information from the following Apple-developed iOS apps during self certification:

- Music
- Videos
- Podcasts
- iBooks
- iTunes U

If the accessory will register for album artwork associated with the now playing media item, the accessory must implement and declare a file transfer session as specified in File Transfer Session (page 795). All album artwork is transmitted to the accessory in JPEG format.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

44.2 Now Playing Updates Usage

A StartNowPlayingUpdates (page 858) message from the accessory to the device starts the generation of NowPlayingUpdate (page 860) messages from the device. The accessory must declare what types of Now Playing information changes are desired in StartNowPlayingUpdates (page 858). There are two major types of Now Playing information changes: changes in the now playing media item, and changes in the state of the media playback engine. In StartNowPlayingUpdates (page 858), the accessory must specify exactly which media item attributes and playback engine state changes it is interested in to receive those updates.

The very first NowPlayingUpdate (page 860) message will be sent immediately after the Now Playing feature is started, and will contain a complete list of media item attributes and playback engine states. All subsequent messages will be sent only when one or more attributes or states change, and the message will only contain the new values. Accessories must assume the last reported attribute or state value remains constant until it is overwritten by a NowPlayingUpdate (page 860) message.

Accessories that interact with iTunes Radio must handle all associated media item attributes (see Table 59-126 (page 861)).

On average, NowPlayingUpdate (page 860) messages will be sent by the Apple device once a second. However, this interval can vary and must not be relied on to drive accessory behavior such as display updates.

The accessory must send a StopNowPlayingUpdates (page 863) message to the device once it no longer needs these updates.

44.3 Test Procedures

44.3.1 Now Playing Updates

1. Verify that the following iAP2 control session message(s) are sent or received:
   - StartNowPlayingUpdates (page 858)
   - NowPlayingUpdate (page 860)
   - StopNowPlayingUpdates (page 863)
2. Verify that the accessory correctly handles Now Playing information from the following Apple-developed iOS apps during self certification:
   - Music
   - Videos
   - Podcasts
   - iBooks

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- iTunes U

3. If the accessory registers for album artwork, verify that the accessory implements and declares a file transfer session.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

45. Power

The Power feature enables accessories to:

- Report their power characteristics.
- Receive Apple device power status.
- Provide power to an Apple device.
- Draw power from an Apple device.

Accessories are defined as:

- Self-powered, if capable of providing their own power (e.g., from a battery or directly from an external power source).
- Device-powered, if capable of drawing power from the Apple device.
- Power-providing, if capable of providing power to the Apple device.

45.1 Power Requirements

All accessories must support the Power feature.

Accessories that provide power to or draw power from an Apple device must connect to the Apple device either through an integrated Lightning connector or a cable as described in Cables with USB Connectors (page 63) or Cables with Non-USB Connectors (page 63).

All accessories that do not have the potential for data communication with the Apple device must use a USB Device Mode connector configuration and provide power to the Apple device. See Connector Pad/Pin Configuration (page 139) for more details on the USB Device Mode connector configuration. See Providing Power to the Apple Device (page 661)

Additionally, certain accessory interface features require the accessory to provide power to the Apple device.

Otherwise, accessories may provide power to the Apple device (Providing Power to the Apple Device (page 661)), draw power from the Apple device (Drawing Power From the Apple Device (page 667)), do both (in certain situations), or do neither (Neither Providing Power nor Drawing Power (page 661)).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Apple strongly recommends providing power to the Apple device whenever possible for the best user experience.

45.1.1 Neither Providing Power nor Drawing Power

Accessories that do not provide power to or draw power from an Apple device (as in the case of a wireless accessory) must declare this during Accessory Identification, see Accessory Identification of Power Capabilities (page 267).

These accessories may implement the device power status feature.

All accessories that support the Device Power Status feature via iAP2 may also send or receive the following iAP2 control session message(s):

- StartPowerUpdates (page 864)
- PowerUpdate (page 864)
- StopPowerUpdates (page 866)

45.1.2 Providing Power to the Apple Device

Accessories that provide power to an Apple device must implement an accessory power source.

Accessory power sources must provide power at all times unless direct user action (button press, power switch, menu item selection, detach of external power supply, etc.) is taken to put the accessory into an 'off' state. Failure to provide power at all times may result in the accessory being unable to charge an Apple device whose battery level is too low for the device to boot and is grounds for failure to pass self certification.

Accessory power sources must establish an iAP2 connection over the Lightning connector and declare their capability to provide power (see IdentificationInformation (page 807) the `PowerProvidingCapability`, `USBHostTransportComponent`, `USBDeviceTransportComponent`, and `SerialTransportComponent` parameters).

All accessories that support the Accessory Power Source feature via iAP2 must send or receive the following iAP2 control session message(s):

- StartPowerUpdates (page 864)
- PowerUpdate (page 864)
- StopPowerUpdates (page 866)
- PowerSourceUpdate (page 866)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

45.1.2.1 External Power Source

If the accessory can accept power from an external power source and provide all or a portion of that power to the Apple device, it must identify the power source's capability and report accordingly to the Apple device.

> **Note:** Accessories must not manipulate an Apple device into drawing more power from the external power source than the device would normally draw when directly connected to the external power source.

Accessories that draw power from an external power source may inform the Apple device that power is not available or available at a reduced level (such as from an internal battery) via a PowerSourceUpdate (page 866) message when the user unplugs the accessory from the external power source (such as a wall socket). Power to the Apple device must be restored (and a corresponding PowerSourceUpdate (page 866) message must be sent) when the user re-connects the external power source.

See Integrated USB Receptacles (page 63) and User Supplied Cables and Power Supplies (page 64) for additional requirements specific to external USB power supplies/cables.

45.1.2.2 Power Adapter Converter Switching Frequencies

To be compatible with the frequency-hopping touch sensors in Apple devices, every AC or DC power adapter design must conform to the following requirements for its converter switching frequencies:

- To avoid interference with audio output, the switching frequency must always be greater than the audio band (that is, more than 22 kHz) for all loads greater than 5 mA.
- The switching frequency must always be above 60 kHz, and preferably above 450 kHz, for all loads greater than 20 mA.

45.1.2.3 Power Adapter Noise Reduction Using a YCAP AC Capacitor

AC adapter control switching frequencies are much higher than power line frequencies. They or their harmonics can easily interfere with the touch sensor modulation frequencies in an Apple device. It is strongly recommended that any AC adapter design for an Apple device include a YCAP AC capacitor (up to 1000 pF) between the primary and secondary sections of the adapter's transformer to reduce common-mode noise at these higher switching frequencies.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

45.1.2.4 Power Adapter Impedance Stability

The diodes used in an AC adapter's full-wave bridge rectifier can be a major source of abrupt changes in the adapter's series impedance. To reduce unwanted touch sensor output oscillations, the AC adapter circuit should be designed such that its series impedance does not change abruptly.

If the AC adapter bridge diodes have large inherent reverse capacitance (greater than 100 pF, as many large power diodes do), then the net impedance change due to diode switching may be acceptably small; it will not adversely affect the touch sensor output. In more compact IC designs, however, the chip area of each diode may be reduced in size and its reverse capacitance may become correspondingly smaller.

To stabilize the impedance of bridge diodes with unacceptably low reverse capacitance, follow the example shown in Figure 45-1 (page 663). In this example, capacitors C1, C2, C3, and C4 have been placed in parallel with diodes D1, D2, D3, and D4 to stabilize the bridge impedance. Their values are larger than the inherent reverse capacitances of the diodes.

Resistors R1, R2, R3, and R4 are optional; if included, they can block noise at very high frequencies, which can help with EMI compatibility. The suggested values of R1, R2, R3, and R4 shown were chosen to have trivial levels of impedance relative to the impedances of C1, C2, C3, and C4 at power line frequencies.

Figure 45-1 Typical AC adapter diode bridge circuit

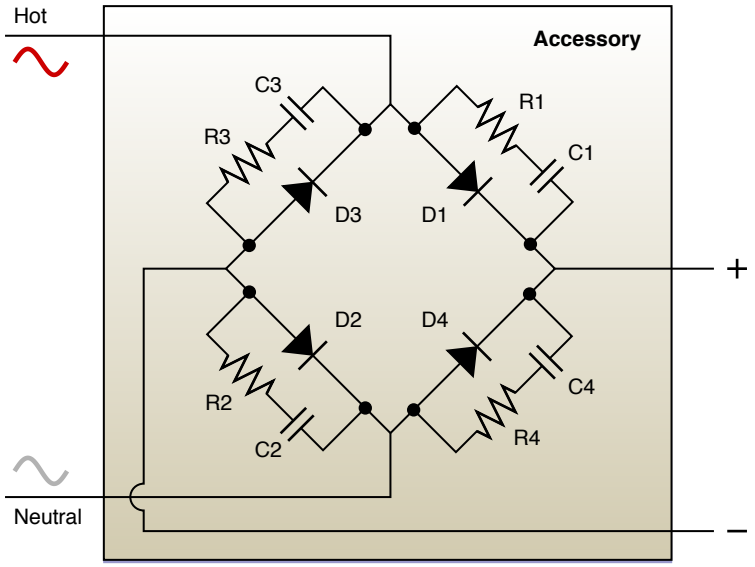


Table 45-1 Typical component values for an AC adapter diode bridge circuit

Component	Value
C1, C2, C3, C4	47 pF
R1, R2, R3, R4	2 kΩ

Copyright © 2015 Apple Inc. All Rights Reserved.

45.1.2.5 Fuse Protection

A fuse should be present at the input of the power supply accessory circuitry to protect it under any fault condition.

45.1.2.6 Short Circuit Response

The output of the power supply accessory should drop or fold back if its output is shorted to the secondary common, and no damage should result. A short circuit is defined as less than 10 milliohms resistance.

45.1.2.7 Lightning Connectors on Power-Providing Accessories

All Lightning connectors on a power-providing accessory must meet the following requirements under load:

- The Lightning connector must supply 2100 mA at 4.55 V to claim compatibility with iPad and iPad mini. 2400 mA at 5.0 V is recommended.
- The Lightning connector must supply 1000 mA at 4.70 V to claim compatibility with iPhone and iPod. 2100 mA at 5.0 V is recommended.

45.1.2.8 Non-Lightning Connectors on Power-Providing Accessories

All power-providing accessory connectors (such as a USB-A receptacle) designed for use with a separate cable that terminates in a Lightning connector must meet the following requirements, regardless of whether the cable is included with the accessory or provided by the end user:

- The connector must supply 2100 mA at 4.97 V to claim compatibility with iPad and iPad mini. 2400 mA at 5.2 V is recommended.
- The connector must supply 1000 mA at 4.90 V to claim compatibility with iPhone and iPod. 2100 mA at 5.2 V is recommended.
- The connector must meet or exceed all applicable USB-IF specifications, both mechanical and electrical (or only electrical if the connector is not a standard USB connector).

45.1.2.9 Cables Terminating in Lightning Connectors

All accessory cables that terminate in a Lightning connector must meet the following requirements:

- The cables must not share or split power between the Lightning connector and any other connector.
- The cables must meet or exceed all applicable USB-IF specifications.
- The cables must have DC resistances below the limits shown in Table 45-2 (page 665).
- The cables must exhibit no more inductance than a simple ferrite bead on every circuit.
- Every cable shield must be connected to at least one of the ground pads on the Lightning connector.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- All captive cables must short the shields to the accessory's ground path.

Table 45-2 USB cable maximum DC resistances

Specification	Maximum DC resistance
Round-trip V_{BUS} with shield shorted to GND at each end	200 mΩ ($R_{GND||SHIELD}$)
V_{BUS} conductor alone	160 mΩ (R_{VBUS})
Ground conductor alone	140 mΩ (R_{GND})

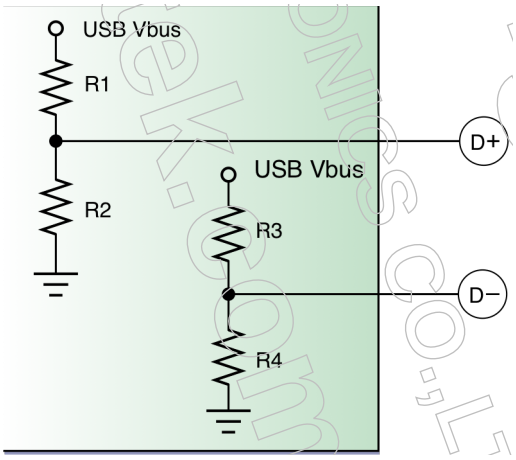
45.1.2.10 Connectors That Do Not Implement iAP2

All accessory connectors that are capable of providing power to the Apple device, but do not implement an iAP2 connection, must either connect the USB D+ and USB D- pins to resistor networks as shown in Figure 45-2 (page 665) or, if the accessory is a USB-compliant embedded host (see USB Embedded Host Implementation (page 706), send an Apple-specific USB vendor request to communicate how much power is available to the Apple device.

If resistors are used, these resistors must be present on the D+ and D- pins at the time the Apple device is connected without requiring user action such as moving a switch or pressing a button.

Additionally, all such connectors must supply between 4.9 V to 5.25 V on the VBUS (Device Power) pin when there is no current being drawn.

Figure 45-2 USB D+/D- resistor networks for power-providing accessory connectors that do not implement iAP2



Note: All resistors used to implement the networks specified in Figure 45-2 (page 665) must have a tolerance of 1% or better.

Table 45-3 USB D+/D- resistor values for power-providing accessory connectors that do not implement iAP2

Current	R1	R2	R3	R4
2400 mA	43.2 kΩ	49.9 kΩ	43.2 kΩ	49.9 kΩ
2100 mA	43.2 kΩ	49.9 kΩ	75.0 kΩ	49.9 kΩ
1000 mA	75.0 kΩ	49.9 kΩ	43.2 kΩ	49.9 kΩ

If a USB vendor request is used, the accessory must enumerate and identify the presence of an Apple device (see Enumeration (page 706)), then send the vendor request. The vendor request must be sent every time the Apple device is enumerated by the accessory.

Table 45-4 USB Vendor Request for accessory USB Embedded Host connector that does not implement iAP2 to communicate available power

Field	Value	Comments
`bmRequestType`	`0x40`	Device-to-host request, vendor-defined type, device is recipient
`bRequest`	`0x40`	Vendor-defined USB get enabled capabilities request
`wValue`	see comments	Charging current available, expressed as an offset from 500 mA. Must be 500 (1000 mA charging current available), 1600 (2100 mA charging current available), or 1900 (2400 mA charging current available)
`wIndex`	see comments	Must be the same as wValue
`wLength`	`0`	0 bytes expected

Copyright © 2015 Apple Inc. All Rights Reserved.

45.1.2.11 Multiple Connectors

> **Note:** Every Apple device-compatible connector on an accessory that uses D+/D- resistors must have its own set of resistors. The accessory must be capable of supplying the total current required when all connectors are in use, regardless of whether the USB-A receptacles are compatible with Apple devices or not.

If the accessory has standard USB-A receptacles supplying 500 mA in addition to Apple device-compatible USB-A receptacles, then the following labeling requirements apply:

- The standard USB-A receptacles supplying 500 mA must be labeled using the USB icon.
- USB-A receptacles capable of identifying themselves to an Apple device as supplying 1000 mA must be labeled, singly or in groups, with the text 'iPhone/iPod'.
- USB-A receptacles capable of identifying themselves to an Apple device as supplying 2100 mA or 2400 mA must be labeled, singly or in groups, with the text 'iPad'.

If the accessory has multiple Lightning connectors with different device compatibilities, then the iPad-compatible connectors must be labeled with the text 'iPad' unless it is physically impossible to connect an iPad to the iPhone/iPod compatible connectors.

45.1.2.12 Overcurrent Protection

For safety reasons, an accessory providing power to an Apple device must detect any nontransient current drain of more than 2.9 A. The accessory must immediately cut off its power supply, after which it may perform a power-up reinitialization. This over-current detection and shut-off circuitry must reset itself without mechanical intervention.

45.1.3 Drawing Power From the Apple Device

Any accessory that draws power from the Apple device at any time is considered to be a device-powered accessory, even if it contains an accessory power source that can provide power to the Apple device. All device-powered accessories must implement both Low Power and Intermittent High Power operational modes.

Pad 5 (Accessory Power) on the Lightning connector can supply up to 100 mA at 2.9-3.4 V ±5% (2.71 V to 3.57 V).

The current drawn by the Apple Authentication Coprocessor does not count towards the current limits. The Authentication Coprocessor must be put into a sleep state if it is not being used for accessory authentication.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

All device-powered accessories must establish an iAP2 connection over the Lightning connector and declare their maximum power draw during Accessory Identification (see IdentificationInformation (page 807) `MaximumCurrentDrawnFromDevice`, `USBHostTransportComponent`, `USBDeviceTransportComponent`, and `SerialTransportComponent` parameters).

All accessories that support the Device Powered Accessory feature via iAP2 must send or receive the following iAP2 control session message(s):

StartPowerUpdates (page 864)

PowerUpdate (page 864)

StopPowerUpdates (page 866)

45.1.3.1 Low Power Mode Requirements

The following table specifies how much current an accessory in Low Power Mode may draw from an Apple device, depending on its iAP2 transport component:

Table 45-5 Maximum allowable Low Power Mode current draw

iAP2 Transport	Maximum Current Draw
None	0 mA
Serial	5 mA
USB Device Mode	0 mA
USB Host Mode (Full Speed)	10 mA
USB Host Mode (High Speed)	30 mA
Bluetooth	0 mA

45.1.3.2 Intermittent High Power Mode Requirements

When an accessory is in Intermittent High Power mode it may draw additional current from the Apple device. An accessory in Intermittent High Power mode must not ever draw more than 100 mA from the Apple device.

Accessories must maintain their transport connection to the Apple device while transitioning from Low Power Mode to Intermittent High Power Mode or vice versa. For example, USB Host Mode accessories must not simulate a USB disconnect and restart Accessory Identification/Authentication.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

45.1.4 Multiple Power States

Accessories whose power capabilities can change (i.e. going between providing/drawing/neither) must inform the Apple device of such changes using the PowerUpdate (page 864). Such accessories must not dynamically disconnect/reconnect to the Apple device when the power capabilities of the accessory changes.

45.2 Power Usage

Accessories must send a StartPowerUpdates (page 864) message with at least one parameter if they will draw power from or provide power to the Apple device and must react to any received PowerUpdate (page 864) messages.

The Apple device may send PowerUpdate (page 864) messages at any time.

Accessories may send a StopPowerUpdates (page 866) message only if they are no longer providing power to or drawing power from the Apple device.

45.2.1 Providing Power to the Apple Device

If the accessory can provide power to the Apple device from its own power source, it must send the IdentificationInformation (page 807) message with a `PowerProvidingCapability` parameter value of `Advanced`.

The accessory must send a PowerSourceUpdate (page 866) message to the device:

- Within 1 second after Accessory Identification (page 265) has completed successfully. The first PowerSourceUpdate (page 866) message sent after identification must contain the `AvailableCurrentForDevice` and `DeviceBatteryShouldChargeIfPowerIsPresent` parameters.
- Whenever the accessory power source state changes, for example:
  - When power is not available or available at a reduced level (e.g., when the user unplugs the accessory from the wall socket).
  - When power is once again available or the level has increased (e.g., when the user plugs the accessory back into a wall socket).

The `DeviceBatteryShouldChargeIfPowerIsPresent` parameter in PowerSourceUpdate (page 866) messages must be `true` for the Apple device's UI to indicate to the user that it is charging its internal battery from the accessory power source.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The device may send PowerUpdate (page 864) messages to the accessory in response to PowerSourceUpdate (page 866) messages indicating how much current the device intends to draw from the accessory power source. If the accessory power source is providing more current than the device is drawing then the accessory may use the extra power for its own purposes.

45.2.2 Drawing Power from the Apple Device

All device-powered accessories must identify their maximum possible current draw in Intermittent High Power Mode via the `MaximumCurrentDrawnFromDevice` parameter in IdentificationInformation (page 807). Also, they must start operation in Low Power Mode.

Accessories supporting Intermittent High Power Mode must include the `AccessoryPowerMode` parameter when sending the StartPowerUpdates (page 864) message.

45.2.2.1 Entering Intermittent High Power Mode

An accessory may enter Intermittent High Power Mode (maximum 100 mA current draw) if it receives a PowerUpdate (page 864) message from the device with a corresponding `AccessoryPowerMode` parameter. Accessories that provide Location updates to an Apple device may be permitted to enter Intermittent High Power Mode when the Apple device requests those updates. Similarly, accessories that implement an iAP2 EA Session may enter Intermittent High Power Mode when the Apple device opens an EA session with the accessory.

When an Apple device is in USB Host Mode, an attached accessory in Low Power mode may also enter Intermittent High Power mode upon receipt of any of the following USB events:

Table 45-6 USB Events that permit Intermittent High Power Mode

USB Device Type	Event
Audio	The Apple device selects a nonzero bandwidth interface setting
EA Native Transport	The Apple device selects a nonzero bandwidth interface setting
MIDI	The Apple device starts polling a MIDI Streaming IN endpoint

If an accessory receives permission to enter Intermittent High Power Mode via multiple mechanisms, the highest power limit overall applies.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

45.2.2.2 Exiting Intermittent High Power Mode

An accessory must exit Intermittent High Power Mode and re-enter Low Power Mode within 1 second if it receives a PowerUpdate (page 864) message from the device with a corresponding `AccessoryPowerMode` parameter.

Similarly, if an Apple device is in USB Host Mode, an attached accessory in Intermittent High Power Mode must re-enter Low Power Mode within 1 second after all of the USB events that permitted Intermittent High Power Mode are negated by one or more of the following USB events:

Table 45-7 USB Events that exit Intermittent High Power Mode

USB Device Type	Event
Audio	The Apple device selects a zero bandwidth interface setting
EA Native Transport	The Apple device selects a zero bandwidth interface setting
MIDI	The Apple device stops polling a MIDI Streaming IN endpoint

45.3 Test Procedures

45.3.1 Accessory Power Sources

45.3.1.1 Output Voltage Regulation

Test conditions: At an ambient temperature of 25 °C, the load on the power supply accessory's output must be increased in steps to produce the following output currents: 0, 250, 500, 700, 1000, 1500, 2100, and 2400 mA. The accessory should be tested at all currents up to and including the maximum current it supports. If the accessory is a car charger, the test must be repeated with input voltages of 10, 12, and 14 V DC. The output voltage must be noted for each combination of input voltage and output current.

Pass/Fail criteria: For every combination of input voltage and output current, the output measured at the dock connector of the power supplies listed in Table 45-8 (page 671) (for Lightning accessories) and Table 45-9 (page 672) (for non-Lightning accessories) must remain between the minimum and maximum voltages shown.

Table 45-8 Power Supply Voltage Outputs for Lightning Accessories

Supply rating	Minimum output	Maximum output
1 A	4.70 V	5.25 V

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Supply rating	Minimum output	Maximum output
2.1 A delivering up to 1.0 A	4.70 V	5.25 V
2.1 A delivering 1.0 to 2.1 A	4.55 V	5.25 V
2.4 A delivering 1.0 to 2.4 A	4.55 V	5.25 V

Table 45-9 Power Supply Voltage Outputs for non-Lightning Accessories

Supply rating	Minimum output	Maximum output
1 A	4.90 V	5.25 V
2.1 A delivering up to 1.0 A	4.90 V	5.25 V
2.1 A delivering 1.0 to 2.1 A	4.97 V	5.25 V
2.4 A delivering 1.0 to 2.4 A	4.97 V	5.25 V

45.3.1.2 Ripple and Noise

Test conditions: Ripple voltage and noise must be measured at each output pin of the Lightning connector while the output is decoupled by a high-frequency 0.2 uF capacitor. The measurement bandwidth must be 0 to 20 MHz.

Pass/Fail criteria: Periodic and Random Deviation (PARD) on each output pin must be less than 100 mV while the power supply accessory is loaded to produce the following output currents: 0, 250, 500, 700, 1000, 1500, 2100, and 2400 mA. The accessory should be tested at all currents up to and including the maximum current it supports.

45.3.1.3 Dynamic Load Response

Test conditions:

- The power supply accessory must be subjected to 500 mA load increases and decreases.
- Power supplies that provide at least 1 A must also be subjected to load increases and decreases between 80 mA and 1 A.
- Power supplies that provide at least 2.1 A must be subjected to increases and decreases from 0 to 500 mA, 500 to 1000 mA, 1000 to 1500 mA, and 2100 mA.
- Power supplies that provide at least 2.4 A must be subjected to increases and decreases from 0 to 500 mA, 500 to 1000 mA, 1000 to 1500 mA, and 2100 to 2400 mA.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- In all tests the load slew rate must not be less than 100 mA/usec.

The frequency of change must be set to give the most readable deviation and settling times. If the accessory is a car charger, the input voltage must be 12 V DC. No load capacitors may be used to stabilize the output.

Pass/Fail criteria: With a decoupling capacitor of 10 uF and an ambient temperature of 25 °C, the power supply accessory output, as measured at the dock connector, must not undershoot below the minimum output voltage or overshoot above the maximum output voltage for each type of power supply listed in Table 45-8 (page 671) and Table 45-9 (page 672)

45.3.1.4 Overload Protection

Overloads applied to the power supply output, at a rate faster than 100 mA/µsec, should cause the output to disconnect before they cause damage to the accessory or the Apple device, and the output should remain disconnected until the overload is removed. This design goal for overload protection also applies to any single fault condition. It must not be possible for a power supply to provide more than 2.5 A rms in any overload situation.

45.3.1.5 Overvoltage Protection

Taking into account the delay time of the overvoltage protection circuit, no single-point fault should be able to cause a sustained overvoltage condition on the power supply output. The power supply should provide a latch-mode overvoltage protection circuit that resets itself within 30 seconds. Optionally, a power off/on cycle could restore normal operation. An overvoltage fault can be simulated by opening the feedback loop that regulates the output voltage. A voltage greater than 6.3 V must not be possible during a single-point failure.

45.3.1.6 Car Charger Tests

45.3.1.6.1 Input Voltage Surge

Test conditions: While the charging accessory is operating at both maximum load and minimum load, the line voltage must be switched 5 times to the surge voltage in a 50% duty cycle, as follows:

1. From 12 V DC to 40 V DC; hold for 16 msec.
2. Back to 12 V DC; hold for 16 msec.
3. Repeat 5 times.

Pass/Fail criteria: The accessory's power supply must survive the repeated applications of the input voltage surge with no component damage. Voltages and logic signals must remain within specification limits during and after the line transients (making it a "transparent surge"). Loss of function or performance must not occur. Permanent damage must not occur. A fuse opening is a failure.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

45.3.1.6.2 Dynamic Line Response

Test conditions: The car charger must be subjected to instantaneous ±2 V DC input variations. The frequency of change must be set to give the most readable deviation and settling times. Output loads must be chosen to produce worst-case conditions. No load capacitors may be used to stabilize the output.

Pass/Fail criteria: The car charger output, as measured at the dock connector, must not undershoot below 4.75 V or overshoot above 5.25 V.

45.3.1.6.3 Turn-On and Turn-Off Characteristics

Test conditions: 12 V DC input voltage must be applied to and removed from the car charger input, with 5 µF capacitive loading on the output.

Pass/Fail criteria: The following measurements must all be within the limits stated:

1. **Turn-on delay:** The car charger output must settle to a value between 4.75 and 5.25 V within 4 seconds after the DC input is applied.
2. **Output rise time:** The time between 10% and 90% of the nominal output voltage must be ≤ 20 ms.
3. **Overshoot:** The output voltage overshoot upon application or removal of the input voltage must be less than 10% of the nominal voltage.
4. **Voltage ramp:** There must be a smooth and continuous ramp of the DC output voltage from 10% to 90% of its final value within a range of 4.75 to 5.25 V, while the output is loaded as specified above under "Test conditions." Voltages of opposite polarity must not be present on the output at any time during turn-on or turn-off.

45.3.1.7 USB Receptacle Load Test

This test is applicable to accessories that provide power to an Apple device via a USB receptacle.

Equipment Required

- Programmable DC load
- USB break out board
- Digital multimeter

Load Test

1. Set the programmable DC load to the appropriate load.
   - For accessories that claim compatibility with iPhone/iPod, set the DC load to 1.0 A.
   - For accessories that claim compatibility with iPad, set the DC load to 2.1 A.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Connect the USB break out board to the accessory's USB-A receptacle.
3. Configure the digital multimeter to measure voltage, and connect it to the USB break out board at VCC and GND.
4. Connect the DC load to the USB break out board at VCC and GND.
5. Read the voltage from the digital multimeter.
    - For accessories that claim compatibility with iPhone/iPod, the voltage must be higher than 4.90 V.
    - For accessories that claim compatibility with iPad, the voltage must be higher than 4.97 V.

45.3.2 Power Rules over iAP2

1. Verify that the StartPowerUpdates (page 864) message includes the parameter AccessoryPowerMode if and only if the accessory draws power from the Apple device.
2. Verify that all Apple device powered accessories identify their MaximumCurrentDrawnFromDevice in IdentificationInformation.
3. If MaximumCurrentDrawnFromDevice is set to 0 mA, verify that the accessory does not draw power. Within ATS, select Hardware and look up the value displayed by the Power to Accessory Current field. Watch this field for several minutes while the accessory is functioning. This value must not exceed 0 mA.
4. Verify that the accessory never draws more power than it claims via MaximumCurrentDrawnFromDevice. Within ATS, select Hardware and look up the value displayed by the Power to Accessory Current field. Watch this field for several minutes while the accessory is functioning. This value must not exceed the value claimed.
5. Verify that an Apple device powered accessory never draws over 100 mA. Within ATS, select Hardware and look up the value displayed by the Power to Accessory Current field. Watch this field for several minutes while the accessory is functioning. This value must not exceed 100 mA.
6. Verify that an Apple device powered accessory starts operation in Low Power Mode. Within ATS, select Hardware and look up the value displayed by the Power to Accessory Current field. Watch this field for the first several seconds.
7. Verify that accessories that have a power source (claim PowerProvidingCapability: Advanced) send a PowerSourceUpdate (page 866) message containing the parameters AvailableCurrentForDevice and DeviceBatteryShouldChargeIfPowerIsPresent.

45.3.3 Accessories That Draw Power

45.3.3.1 USB Speed Verification

For USB Host Mode accessories which implement intermittent High Power Mode, verify the USB speed.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1. Connect the accessory and the Apple device to ATS.
2. Before beginning Authentication, verify that ATS displays the USB speed (USB Full Speed or USB High Speed).

45.3.3.2 Low Power Mode Startup

If the accessory uses Intermittent High Power Mode, verify that it begins in Low Power Mode and that the current draw remains within acceptable bounds.

1. Connect the accessory and the Apple device to ATS.
2. If necessary, turn on the accessory.
3. Before beginning the functionality that causes the accessory to enter Intermittent High Power Mode, view the hardware summary in ATS and verify that the power to accessory current remains within specification requirements. Note that the current drawn by the Apple Authentication Coprocessor does not count towards the current limits. Current spikes during the authentication process during the first 1 second of connection are to be ignored:
    - USB Host Mode, full speed (10 mA max)
    - USB Host Mode, high speed (30 mA max)
    - Device Mode (0 mA max)
    - Serial Mode (5 mA max)

45.3.3.3 Max Current Drawn Identification

If the accessory uses Intermittent High Power Mode, verify that it identifies its maximum possible current draw in Intermittent High Power Mode via the MaximumCurrentDrawnFromDevice parameter.

1. Connect the accessory and the Apple device to ATS.
2. If necessary, turn on the accessory.
3. Begin the functionality that causes the accessory to enter Intermittent High Power Mode.
4. View the iAP Control Session in ATS and verify that the accessory sent MaximumCurrentDrawnFromDevice.
5. Ensure that max current draw does not exceed 100 mA

If the accessory uses the USB Host Mode transport and is a MIDI device, verify that the accessory does not draw more than 100 mA current from the Apple device when the Apple device starts polling a MIDI Streaming IN endpoint.

If the accessory uses the USB Host Mode transport and is a MIDI device, verify that the accessory returns to Low Power Mode with 1 second after the Apple device stops polling a MIDI Streaming IN endpoint.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

If the accessory uses the USB Host Mode transport and supports an EA protocol session (Over iAP2 or Native), verify that the accessory does not draw more than 100 mA current from the Apple device when the EA protocol session is active.

If the accessory uses the USB Host Mode transport and is an Audio device, verify that the accessory does not draw more than 100 mA current from the Apple device when the Apple device selects a nonzero bandwidth interface setting.

If the accessory uses the USB Host Mode transport and is an audio device, verify that the accessory returns to Low Power Mode with 1 second after the Apple device selects a zero bandwidth interface setting.

Verify that, if drawing power from the Apple device, the accessory does not charge its battery.

If the accessory implements an iAP2 connection that both draws power from a wall socket and draws current from the Apple device, verify that a PowerSourceUpdate (page 866) message is sent when the unplugged accessory is plugged back into power.

1. Connect the accessory and the Apple device to ATS.
2. If necessary, turn on the accessory.
3. Unplug the accessory from its power source.
4. Reconnect the accessory to its power source. In ATS, verify that the iAP2 PowerSourceUpdate (page 866) message is sent from the accessory.

If an accessory is self-powered, draws power from an external power supply, and implements an iAP2 connection that supplies power to the Apple device only when connected to that external power supply, then verify that an appropriate PowerSourceUpdate (page 866) message is sent both when connected and disconnected from the external power supply.

1. Confirm the self-powered accessory is not connected to an external power source.
2. Connect the accessory and the Apple device to ATS.
3. Confirm accessory sends a PowerSourceUpdate message with parameters: AvailableCurrentForDevice = 0; DeviceBatteryShouldChargeIfPowerIsPresent = YES.
4. Connect the accessory to an external power source.
5. Confirm accessory sends a PowerSourceUpdate message with parameters: AvailableCurrentForDevice = <See USB Receptacle Load Test (page 674)>; DeviceBatteryShouldChargeIfPowerIsPresent = YES.
6. Disconnect the external power source.
7. Confirm accessory sends a PowerSourceUpdate message with parameters: AvailableCurrentForDevice = 0; DeviceBatteryShouldChargeIfPowerIsPresent = YES.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 45-10 Maximum allowable Low Power Mode current draw

iAP2 Transport	Maximum Current Draw
None	0 mA
Serial	5 mA
USB Device Mode	0 mA
USB Host Mode (Full Speed)	10 mA
USB Host Mode (High Speed)	30 mA
Bluetooth	0 mA

45.3.4 Battery Pack Power

45.3.4.1 Full Charge Voltage Threshold

Verify that when the battery pack is full its charging voltages are within spec.

1. Power on accessory (if necessary).
2. Charge the battery pack completely.
3. Run USB VBUS load test. Verify that USB VBUS voltage stays within spec under load.
    - For 30 pin/Lightning accessories providing 1.0 A charging, if the VBUS drops below 4.70 V, the test is considered a failure.
    - For 30 pin/Lightning accessories providing 2.1 A charging, if the VBUS drops below 4.50 V, the test is considered a failure.
    - For USB receptacle accessories providing 1.0 A charging, if the VBUS drops below 4.90 V, the test is considered a failure.
    - For USB receptacle accessories providing 2.1 A charging, if the VBUS drops below 4.97 V, the test is considered a failure.

45.3.4.2 Low Charge Voltage Threshold

Verify that when the battery pack is nearly fully depleted that the charging voltages are within spec.

1. Turn accessory power on (if necessary).
2. Deplete the battery pack to its lowest charge.
3. Run USB VBUS load test. Verify that USB VBUS voltage stays within spec under load.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- For 30 pin/Lightning accessories providing 1.0 A charging, if the VBUS drops below 4.70 V, the test is considered a failure.
- For 30 pin/Lightning accessories providing 2.1 A charging, if the VBUS drops below 4.50 V, the test is considered a failure.
- For USB receptacle accessories providing 1.0 A charging, if the VBUS drops below 4.90 V, the test is considered a failure.

45.3.5 Charging

Verify that the charging voltages are within specification if accessory power can be turned off. (Either 0 V or 4.75 V to 5.25 V.)

- Plug in accessory to wall outlet (if necessary).
- Turn off accessory power (if necessary).
- Attach ATS to the accessory.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

46. Serial

The Serial transport may make sense for low-cost accessories that do not need to transmit audio, or for device-powered accessories which emphasize low power draw over high data transmission rates. Otherwise, the USB Host Mode transport is recommended.

Apple devices support serial communication based on the RS-232 serial specification with the exception of signaling levels. For Apple devices, a mark is 2.500 V through 3.57 V and a space is 0 V through 0.8 V.

Table 46-1 Serial transport mark and space levels for Apple devices

Description	Conditions	MIN	MAX	Units
Accessory TX Voltage High		2.500	3.57	V
Accessory TX Voltage Low		0.000	0.800	V
Accessory TX Current	Accessory TX Voltage = 0 V or 3.0 V		±30	µA
Device TX Voltage High	Accessory TX Voltage High = 100 µA	2.500	3.57	V
Device TX Voltage Low	Accessory TX Voltage Low = -100 µA	0.000	0.500	V

Accessories must communicate over the serial transport at 57600 bps and maintain their baud rates within ±2% of the chosen rate over the entire temperature range of the accessory. Accessories must not change baud rate once a Serial communication has started.

The accessory must not rely on the Apple device to hold the Device TX level to the mark state when the Serial transport is idle. When idle, Device TX line must be pulled up to Accessory Power through a 100 kΩ resistor.

Accessories that use an open-collector or open-drain UART driver on Accessory TX must include a pull-up resistor to Accessory Power. The pull-up resistor value must be chosen to meet the baud rate requirements described above. The maximum mark voltage must be measured with the Apple device not connected, and mark signals must be sent only after Accessory Power goes high.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

All serial communications use 8 data bits, no parity bits, and one stop bit (8-N-1). Serial hardware flow controls (RTS/CTS and DTR/DSR) are not used and will be ignored by the Apple device. In addition, the accessory must not use software flow control (XON/XOFF). The accessory must not use bit averaging to produce a mean bit rate not directly achievable from its system clock. All bits transmitted by the accessory must have the same nominal duration.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

47. Siri

47.1 Enabling Custom Siri Commands

Every accessory that works with Siri must support HFP Command AT+XAPL (page 370). The Apple device will use the information sent by this command to enable and disable custom commands related to Siri.

To receive Siri status events, the accessory must send the `AT+XAPL` command after making a successful HFP Service Level Connection (SLC) to the Apple device. The accessory must send an `AT+XAPL` command first, before sending any of the additional Siri-specific commands described below.

47.2 Obtaining Siri Availability Information

After establishing an HFP profile connection, an accessory can determine if Siri is available and enabled on an Apple device. It can also receive notifications of changes in Siri status. If Siri is disabled, Voice Control will be activated instead.

47.2.1 Obtaining Status Information at Connection

The accessory must send the following command after making a successful HFP profile (SLC) connection and sending an `AT+XAPL` command.

47.2.1.1 HFP Command AT+APLSIRI?

Description: AT command to retrieve Siri status information.

Initiator: accessory

Format: `AT+APLSIRI?`

Response: `+APLSIRI:`*value*

Defined Values:

- 0 = Siri is not available on this platform.
- 1 = Siri is available and enabled.
- 2 = Siri is available but not enabled.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Example: `+APLSIRI:1` (Siri is available and enabled)

47.2.2 Receiving Siri Availability Updates from the Apple Device

After initialization has been completed, the Apple device will send the accessory the following notification if there is a change in Siri status. This notification will be provided only if the accessory has requested Siri status (by sending `AT+APLSIRI?`) at least once after connection, and if the Apple device has reported that Siri is available and enabled.

47.2.2.1 HFP Command +APLSIRI

Description: Unsolicited event indicating a change in Siri status.

Initiator: Apple device

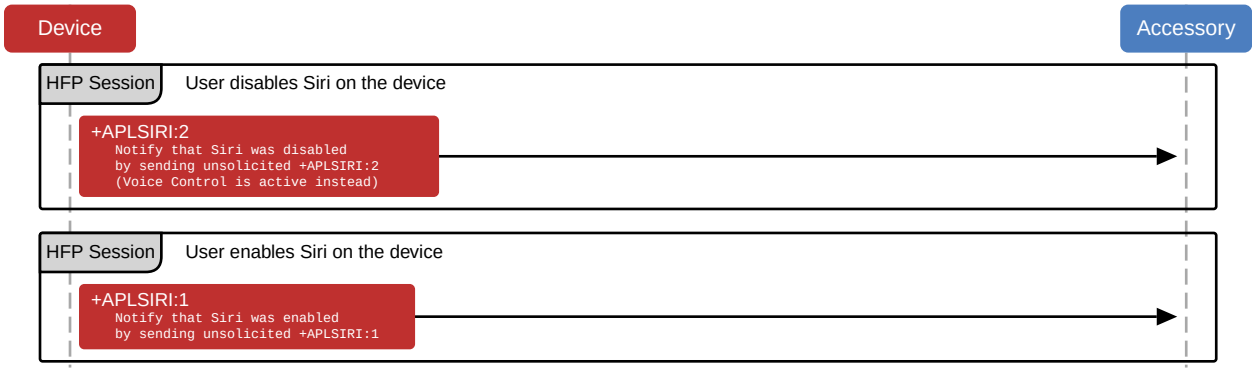
Format: `+APLSIRI:`*value*

Defined Values:

- 1 = Siri is available and enabled.
- 2 = Siri is available but not enabled.

Example: `+APLSIRI:2` (Siri is available but not enabled)

Figure 47-1 Siri is Disabled/Enabled from the Apple Device's Settings



47.3 Initiating a Siri Session

Once support for Siri is established on both the accessory and the Apple device, a Siri session can be started from either one.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

47.3.1 Initiating a Session from the Accessory

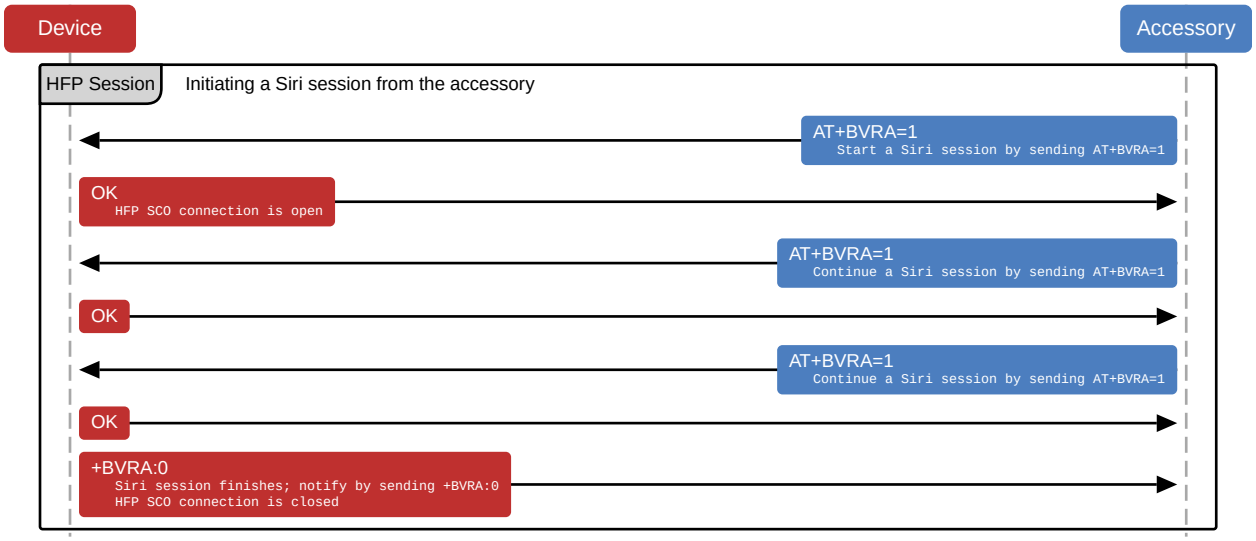
To initiate a Siri session, the accessory must use the voice recognition command `AT+BVRA` defined in the Bluetooth *Hands-Free Profile* specification. For further details, see the Bluetooth *Hands-Free Profile* 1.6 profile specification, section 4.25. The HFP profile must be connected and SLC must exist.

The accessory must use the following command sequence:

- The accessory sends an `AT+BVRA=1` command to the Apple device.
- The Apple device sends an `OK` response.
- The Apple device launches a Siri session and creates a Synchronous Connection (SCO) for the audio.
- If the Siri session is not finished, the accessory must send `AT+BVRA=1` to continue the conversation. This may need to happen multiple times.
- When the Siri session is finished, the Apple device sends a `+BVRA:0` result code to the accessory.
- The Apple device disconnects the SCO connection.

While a Siri session is active, the accessory must let the user continue the conversation and ask follow up questions within the current context. In order to do so, the accessory must be able to send an `AT+BVRA=1` command to the Apple device even after Siri has been already activated and before `+BVRA:0` is received. Figure 47-2 (page 684) shows an overview of the interaction when Siri is triggered from the accessory, the running session was continued twice and once Siri was finished, the device dismissed the session.

Figure 47-2 Initiating a Siri Session from the Accessory



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

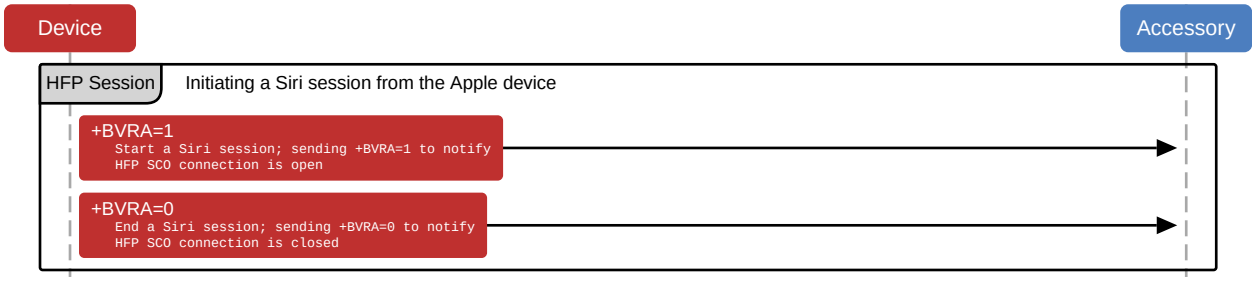
47.3.2 Initiating a Session from the Apple Device

If the accessory supports voice recognition commands, the Apple device sends a +BVRA event to indicate the start of a Siri session. The accessory must enable support for voice recognition and indicate it in its feature response as described in the Bluetooth *Hands-Free Profile* 1.6 specification, section 4.34.1, "Bluetooth Defined AT Capabilities." Specifically, the HFP profile must be connected, SLC must exist, and voice recognition activation (bit 3) must be enabled in the AT+BRSF command. The Apple device will not use virtual call functionality for the Siri session if voice recognition activation is supported by the accessory.

The accessory must expect the following command sequence:

- The Apple device sends a +BVRA:1 event to the accessory.
- The Apple device launches a Siri session and creates a SCO connection for the audio.
- When the Siri session is finished, the Apple device sends a +BVRA:0 result code to the accessory.
- The Apple device disconnects the SCO connection.

Figure 47-3 Initiating a Siri Session from the Apple Device

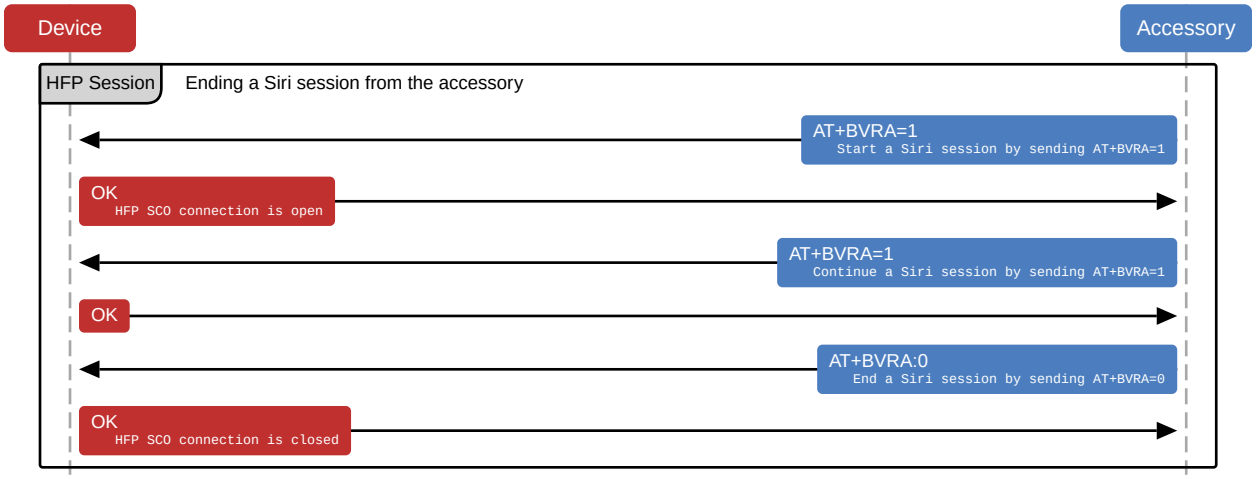


47.3.3 Ending a Session from the Accessory

Once a Siri session is running the accessory must be capable of ending the session by sending an AT+BVRA=0 command to the Apple device. Figure 47-4 (page 686) shows an example of ending a running Siri session from the accessory. The accessory must only end an active session as a direct result of a user action.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 47-4 Ending a Siri Session from the Accessory



47.4 Siri Eyes Free Mode

Siri Eyes Free mode is a feature to control Siri responses that include display information and can be enabled or disabled as needed. In Eyes Free mode, the user experience is tailored towards a driving scenario and interactions with Siri are done primarily via voice to minimize the need for the user to look at a screen. Siri Eyes Free mode is supported only for Bluetooth-enabled vehicle entertainment systems and must not be used by any other accessories. Siri Eyes Free must not be triggered via a voice command.

The Apple device will listen for the HFP AT command `AT+APLEFM` to enable or disable Eyes Free mode.

This command is used by the Apple device to modify Siri responses that contain visual information or require user interaction. Suitable audio feedback and voice commands will be available to the user based on the Siri use case that was initiated.

Eyes Free mode is disabled by default. Once the accessory has enabled Eyes Free mode, it remains enabled for all subsequent Siri sessions initiated from the accessory until the accessory disables it or the Bluetooth connection is disconnected.

47.4.1 HFP Command AT+APLEFM

Description: An accessory sends this command to notify an Apple device of the preferred state of Eyes Free mode.

Initiator: accessory

Format: `AT+APLEFM=`*value*

Response: `OK`

Defined Values:

- 0x00 = Disable Eyes Free mode.
- 0x01 = Enable Eyes Free mode.
- 0x02-0xFF = reserved

Example: `AT+APLEFM=1`

47.5 Improving Voice Recognition

The microphone audio that the accessory sends to the Apple device during a Siri session must be suitable for voice recognition. Audio requirements for optimal voice recognition may differ from requirements for optimal human perception (such as during a cellular phone call).

Filtering of the audio signal to remove echoes or feedback noise is acceptable.

To provide the best possible audio quality as Siri input, the accessory must observe the following recommendations:

- **Echo cancellation and noise suppression (EC/NR)**: Directional microphones and linear beamforming with microphone arrays giving improved SNR are recommended. Linear echo cancellation for reducing unwanted audio sources (such as audio output from the system) without having any other effect on the speech signal are also recommended. However, single channel noise reduction methods (such as spectrum subtraction) must not be applied, as they will be detrimental to the speech recognition accuracy. Similarly, automatic gain control, residual echo suppression and attempts to blank out non-speech periods in the waveform must not be applied.
- **Signal gain**: When adjusting signal levels, the accessory must avoid artifacts, dropouts, and clipping in all circumstances. Automatic Gain Control is not recommended. If the accessory adjusts signal gain, the gain must be held constant across each spoken utterance. The nominal level measured at the uplink output of the accessory must be A-weighted -30 dB ±2 dB root-mean-square (RMS), expressed in units relative to full-scale (dBFS(A)). Alternatively, the nominal level may be 13 dB ±2 dB SLR if using the ITU measurement procedure.
- **Signal-to-noise ratio (SNR)**: An average SNR greater than 20 dB is recommended. Below 20 dB, recognition rates will be impacted.
- **Reverberation**: Maintaining RT60 time at less than 200 msec is recommended.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

47.5.1 Wide Band Speech Support

An accessory using Siri must support 16 kHz wide band speech audio for better audio quality and voice recognition performance. See the Bluetooth *Hands-Free Profile* 1.6 specification for details about wide band speech audio. Narrow band audio signal (8 kHz) is supported but not recommended.

47.6 Optimizing the Siri Experience

For best results in using Siri, the accessory design must follow these guidelines:

- The start of a Siri session must not be accompanied by local beeps or verbal indications (such as an announcement of "...voice dialing...") from the accessory. When a Siri session becomes active, the Apple device sends two beeps indicating that Siri is ready to receive instructions. Adding extra audible notifications only inserts delays in the system.
- The accessory must wait for the Apple device to end each Siri session. The accessory must not send an `AT+BVRA=0` command unless it is prompted to do so by user interaction.
- The Apple device expects that the accessory is capable of rendering audio as soon as the SCO connection is active. This is necessary to make sure that the user always hears the Siri introductory beeps with minimum delay. The delay should be within 200 ms.

47.7 Common Siri Applications

Siri can send messages, find points of interests, place phone calls, and much more. As Siri capabilities are constantly growing, additional use cases may become available after the initial integration. In Eyes Free mode, some of these use cases may not be accessible as the user experience is tailored towards a driving scenario.

47.7.1 Initialization Procedure After Connection is Established

Figure 47-5 (page 689) outlines the sequence the accessory has to trigger to be able to use Siri on an Apple device. After establishing an HFP profile connection, the accessory must first enable the custom Siri commands by sending `AT+XAPL` and provide the features it supports. After a confirmation is received from the device, the accessory must determine Siri's availability with `AT+APLSIRI?`.

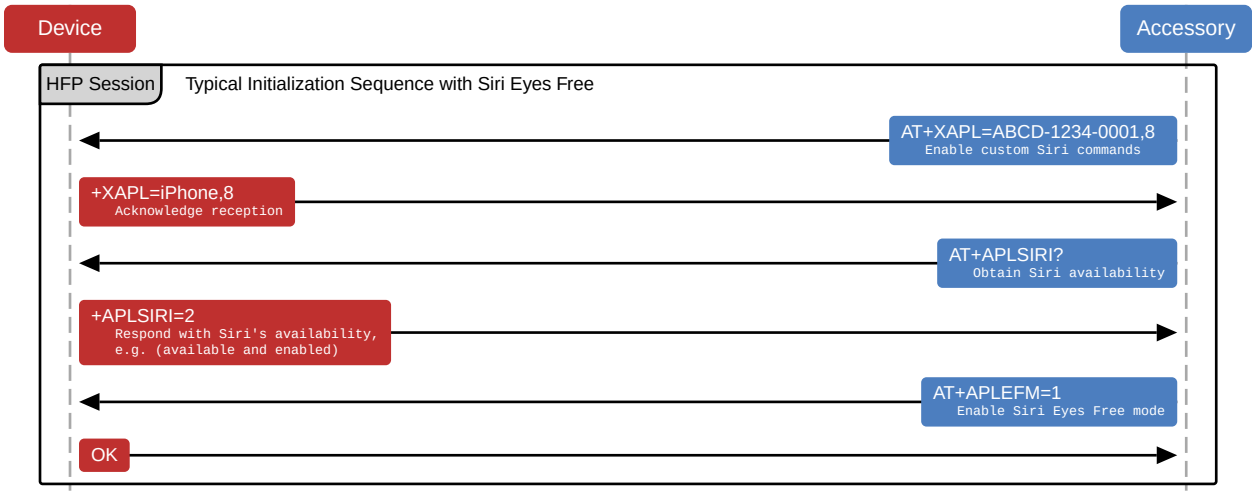
Vehicles with Bluetooth-enabled infotainment systems can also enable Siri Eyes Free Mode during initialization. This is detailed in Figure 47-6 (page 689).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 47-5 Siri Initialization Procedure



Figure 47-6 Siri Initialization Procedure with Siri Eyes Free



47.7.2 Phone Dialing Using Siri

Upon the user's request Siri can initiate an outgoing phone call. The Apple device will initiate HFP call signaling to establish a phone call as described in Bluetooth (page 344). The accessory must be able to transition to Hands-Free dialing at any time during or after a Siri session when signaled by the Apple device.

47.7.3 Audio Routing and Media Playback Using Siri

Siri can control the media playback on an Apple device, and if Siri determines that the user wants to play or pause music it will either start, pause or resume media playback. The Apple device will send a notification to the accessory indicating a change in playback state and any associated track information. The accessory must respond to those notifications, start or stop the music playback as requested, as well as update the correct playback state (e.g. shuffle, repeat).

The accessory must not force a change in the playback state after a Siri session is ended. If music was playing before Siri was started, it must continue playing, if it was paused, it must remain paused.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

After Siri starts music playback the accessory must set its current audio route to match the audio source, depending on how audio is being received from the Apple device (via Bluetooth or by a wired connection).

The available media playback notifications depend on the audio route being used:

- For a Bluetooth audio route, use the approach described in Notifications (page 351).
- For a wired audio route, use the iPod Accessory Protocol, specifically the iAP2 Now Playing Updates (page 657) feature, iAP1 Display Remote Lingo, or iAP1 Extended Interface Mode.

47.7.4 Turn-By-Turn Directions Using Siri

Siri can initiate active route guidance that will play turn-by-turn directions. In case the Apple device is the active source and is already playing music, turn-by-turn directions will be mixed in as part of the audio stream. In case the Apple device is not playing music, the accessory must be able to mix in turn-by-turn directions with the active audio source.

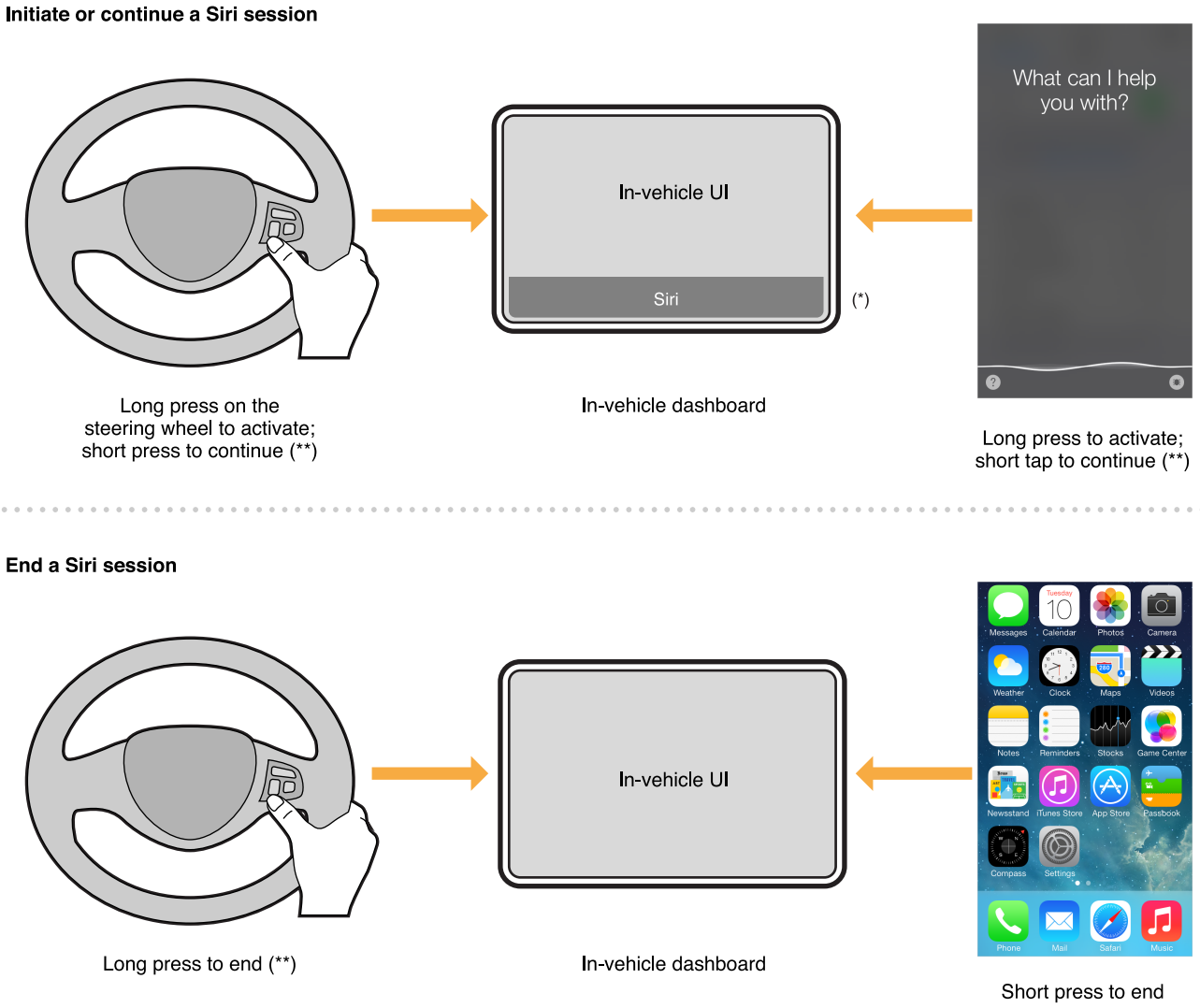
The Apple device will notify the accessory to play turn-by-turn directions only over Bluetooth. Detailed information on how to distinguish between music playback and turn-by-turn notifications is available in Notifications (page 351).

47.8 User Interaction with Siri Eyes Free in a Vehicle

A vehicle that uses Siri Eyes Free mode must integrate the Siri experience with the existing in-vehicle entertainment system and controls. The vehicle must provide a convenient interface to initiate, continue, and end a Siri session. Once a Siri session is running, the vehicle must display a visual cue that voice recognition is in process. Figure 47-7 (page 691) outlines how a Siri interaction must be designed.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 47-7 Siri Eyes Free User Interaction



(*) If the accessory wishes to indicate that Siri is active, it must do one of the following:

- Display the word 'Siri' (as capitalized) with no additional text or icon.
- Use generic text or icon that does not resemble the Siri microphone icon and does not infringe Apple trademarks.

(**) If the vehicle is equipped with steering wheel controls, a dedicated button or a long-press on a button on the vehicle's steering wheel is required to start, continue and end a Siri session. If no steering wheel controls are available, a soft button must be available within the in-vehicle user interface to start, continue or end a Siri session.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

When a vehicle enables Siri Eyes Free mode, the Apple device will not display any onscreen Siri content. If the device was locked at the time the Siri session was activated from the vehicle, it will remain locked and the screen will not turn on. If the user unlocks or manually activates the device while in an Eyes Free Session there will be a notification that the device is in an active Siri session but there will be no visual Siri content displayed.

47.9 Enabling/Disabling Siri from the Apple Device

The user has the ability to disable or enable Siri from the Settings menu on the Apple device. When Siri is disabled, Voice Control becomes the recognition engine on the device and will be triggered by default. The accessory may choose to either launch Voice Control with no further changes, in the same way Siri is launched (shown in Figure 47-8 (page 692)) or display a warning message and not send an activation command to the Apple device (Figure 47-9 (page 692)).

Figure 47-8 Siri is Deactivated - Launching Voice Control

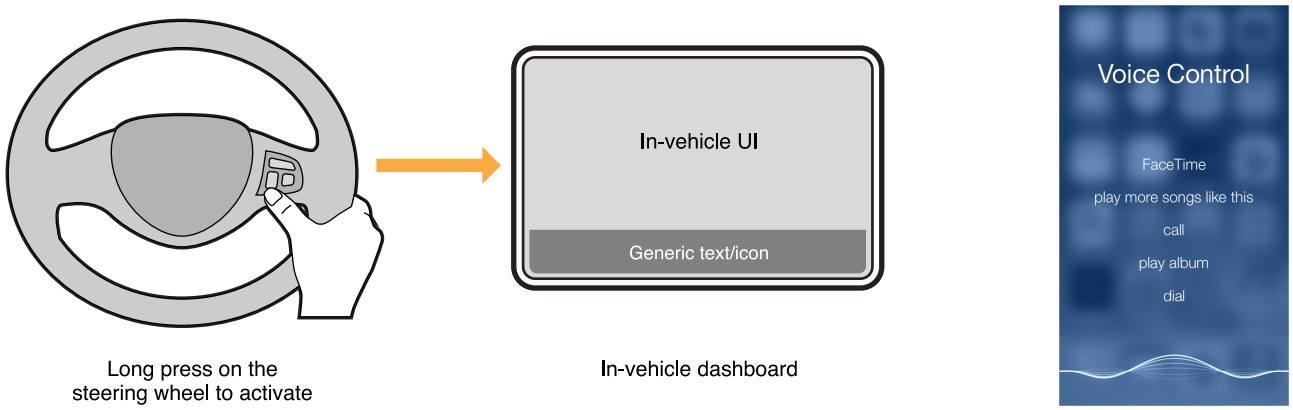
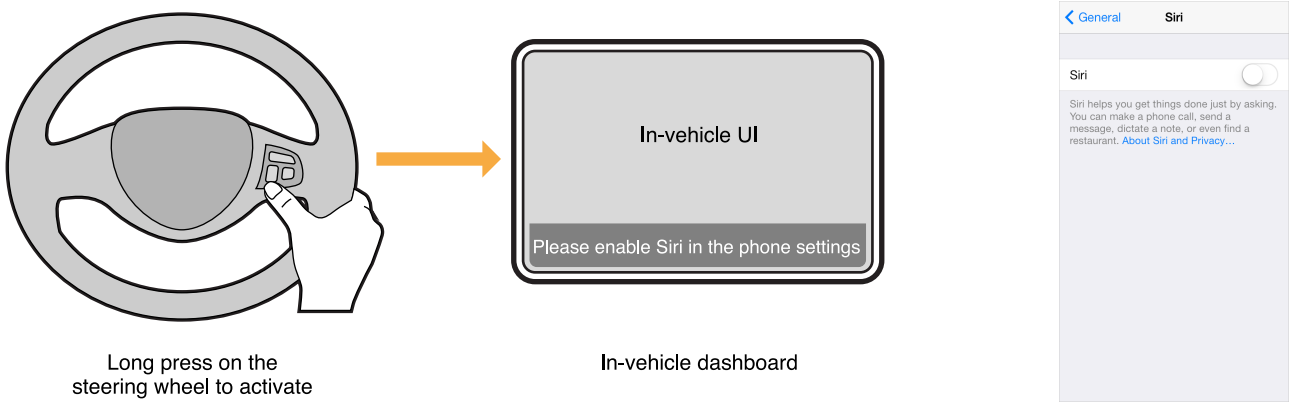


Figure 47-9 Siri is Deactivated - Displaying a Warning Message



47.10 Test Procedures

47.10.1 Siri Eyes Free

The following test procedures are applicable to accessories that interact with Siri Eyes Free.

For the following spoken tests, the speaker should ideally be a native speaker of North American English. If the tester's native language is not English, set Siri to your native language and translate the phrases to be spoken below into your native language.

47.10.1.1 General

1. Pair and establish a Bluetooth Handsfree Profile (HFP) connection between the iPhone and the head unit. Activate Siri from the vehicle steering wheel button (e.g. by press hold):
    a. Observe that the iPhone screen remains inactive after a Siri session has started (a purple status bar will be visible on the Apple device if the screen is activated manually).
    b. Ensure that Siri opening chime is heard completely through the vehicle speakers.
    c. Observe a visual notification in the in-car User Interface (UI) that a Siri session is active (textual notification, on-screen UI, etc.).
2. Activate Siri from the vehicle steering wheel button and say "Send a message to Peter. How are you?". While still saying the message press the vehicle steering wheel button to cancel Siri (e.g. by press hold):
    a. Ensure that the Siri closing chime is heard completely through the vehicle speakers.
    b. Ensure the iPhone screen remains inactive (if manually activated, the purple bar on the phone will disappear).
    c. Verify that the in-car UI for Siri interaction dismisses and the head unit resumes the state before Siri's interaction.
3. Activate Siri from the vehicle steering wheel button and say "How is the weather in San Francisco?". Wait for Siri to respond with the weather forecast. Once the weather forecast is complete, resume Siri from the vehicle steering wheel button and say "What about New York?":
    a. Observe that the purple bar is still active on the phone.
    b. Listen for the Siri opening chime and closing chime.
    c. The vehicle UI should be displaying an on-going Siri session.
    d. Verify that Siri responds with the weather forecast for New York.
4. In case the vehicle UI offers on-screen controls to activate/cancel/resume Siri, repeat steps (1) to (3) for all on-screen controls.
5. Activate Siri from the steering wheel button and say "What's the time". Listen to the current time and do not interact with Siri or the iPhone. After 5 seconds have expired:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

a. Observe that the purple bar on the phone is no longer visible.

b. The in-car UI for Siri interaction should be dismissed.

c. The head unit should resume the state before Siri's interaction.

6. Listen to FM radio from the car speakers (no A2DP streaming active). Press and hold phone's home button to activate Siri from the phone:

   a. Ensure that the Siri closing chime is heard completely through the vehicle speakers.

   b. Observe a visual notification in the in-car UI that a Siri session is active (textual notification, on-screen UI, etc.).

   c. Observe Siri's interaction on the phone's screen and ask "What's the time?"

   d. After Siri has responded, lock the phone again to dismiss the Siri session by pressing the phone's sleep/wake button.

7. On the phone go to Settings and turn off Siri. Activate Siri from the head unit. Observe one of the following depending on the actual implementation (a) Voice Control starts instead of Siri (b) The head unit displays a warning that Siri Eyes Free is not available.

8. On the phone go to Settings and turn Siri back on. Verify that Siri can be activated/cancelled from the head unit and from the Home button on the phone.

9. Turn Bluetooth off from the Settings on the phone. Verify that Siri cannot be started.

10. Turn Bluetooth back on from the Settings on the phone. Verify that Bluetooth HFP profile reconnects and that Siri can be activated/cancelled from the head unit and from the Home button on the phone.

11. Make sure there is no accessory battery status level indicator icon displayed on the phone's status bar.

47.10.1.2 Siri Dialog

1. Activate Siri from the vehicle's steering wheel button and say "Send a text message to **insert contact name**". When Siri prompts for "what would you like it to say", dictate a short message. After Siri has read back your dictated message, say "Review it". After Siri has read back the message again, say "Review it" again. Repeat this cycle ~5 times to ensure that the head unit is able to handle a long interaction with Siri. At the end say "Send it" and verify that the message is sent. Verify that all opening and closing beeps are audible and the message is sent. After the Siri session is closed, the audio playback should go back to the state it was before Siri was started, i.e. if it was paused remains paused, if it was playing remains playing.

2. Start Siri from the vehicle's steering wheel button and ask for directions. Follow up through the dialog until the navigation is started. Verify that the Siri session is closed and that the audio playback goes back to the state it was before Siri was started, e.g. if it was paused remains paused, if it was playing remains playing.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Start Siri from the vehicle's steering wheel button and say "Search the web for polar bears". Verify that Eyes Free mode is on and that this use case is blocked by Siri. Note: In some implementations the vehicle has to be in motion before Eyes Free is activated by the car kit.
4. Start Siri from the vehicle's steering wheel button and say "What is the current time in Munich?". After Siri has answered but before ~5 seconds have elapsed, resume Siri (e.g. by a short press on the steering wheel button) and verify that Siri is initiated again. Say "What about San Francisco?". Repeat (with a different city) and verify that this can continue indefinitely as long as you short press on the steering wheel button within 5 seconds of the last response.

47.10.1.3 Bluetooth HFP A2DP Music

1. Establish a Bluetooth A2DP connection and switch to Bluetooth audio source on the head unit. Activate Siri and ask "Next track". Verify that track advances and that audio is playing through vehicle speakers. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
2. Activate Siri and say "Pause the music". Verify that audio remains paused after Siri has been dismissed. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
3. Pause music playback on the head unit (via AVRCP command). Activate Siri and ask "What time is it?". Verify that the music playback remains paused after the Siri session has been dismissed. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
4. Switch to FM radio on the head unit. Activate Siri and ask "Play me a song". Verify that head unit is able to automatically switch to BT audio and iPhone music starts playing. Verify that the beginning of the selected track is heard, e.g. there is no skipping of audio packets. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
5. Activate Siri and ask "Shuffle all songs". Verify that head unit correctly updates NowPlaying track information. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
6. Activate Siri and ask to play a specific artist or title. Verify that the Siri session is dismissed after the music starts. Make sure the correct metadata is displayed on the screen. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

47.10.1.4 Call

1. Activate Siri and call a contact with more than one phone number (home and mobile). Wait for Siri's response to ask for which phone number to call. Answer with "home". Verify that call transition is handled correctly by the head unit and any Siri UI displayed on the vehicle screen is dismissed.
2. While iPhone music is playing, activate Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback resumes after the call has been answered and terminated on the far end. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. While iPhone music is playing, start Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback resumes after the call has been answered and terminated on the near end (i.e. on the head unit). Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
4. While in a Siri session, receive an incoming call on the head unit. Verify that head unit handles call-signaling correctly and transitions to phone UI once the call has been accepted. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

47.10.1.5 Bluetooth + Wired iAP

1. Connect Apple device to head unit via 30-pin or Lightning connector (iPhone 4s / iPhone 5 respectively). Switch to iPod music and verify that audio is playing. Activate Siri and say "Next track". Verify that track advances and head unit displays track metadata correctly. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
2. From the head unit UI, select a playlist with a single song and start playing it. Start Siri from the vehicle steering wheel and say "Play ..... **make sure to select a song to play that is (a) not in the same album as the single-track playlist and (b) not song track index 0 of its album**". Verify that the new song starts playing and that the head unit displays the track metadata for this new song correctly. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
3. Turn Shuffle off on the head unit UI. Then start Siri and say "Shuffle all songs". Verify that the shuffle indicator on the head unit UI is updated and the correct track metadata for the new now playing song is displayed correctly Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
4. Switch to FM radio on the head unit. Activate Siri and say "Play me a song". Verify that head unit is able to automatically switch to iPOD audio source and that audio starts playing through the speakers. Verify that there is no skipping of audio at the beginning of the selected track. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
5. Pause music playback on the head unit (via iAP commands). Activate Siri and ask "What time is it?". Verify that music playback remains paused after Siri session has been dismissed. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
6. While iPhone music is playing, start Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback resumes after the call has been answered and terminated on the far end. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.
7. While iPhone music is playing, start Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback resumes after the call has been answered and terminated on the near end (i.e. on the head unit). Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

8. Pause music playback on the head unit (via iAP commands). Start Siri and say "Call **insert contact name to call**". Verify that call transition is handled correctly by the head unit. Verify that iPhone music playback remains paused after the call has been answered and terminated on the far end. Verify that the Siri in-car UI is dismissed and the head unit goes back to its initial state.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

48. USB

Apple devices support both USB Device Mode (see USB Device Mode (page 706)) and USB Host Mode (see USB Host Mode (page 712)) transports. USB Host Mode is recommended. Most, if not all, future development of accessory interface features will occur on the USB Host Mode transport.

48.1 USB Host Mode Passthroughs

Accessories must not facilitate the connection of arbitrary USB peripherals to an Apple device in USB Host Mode.

48.2 USB Signal Integrity

USB Host Mode accessories and USB Device Mode accessories must pass the USB signal integrity test procedures (see Signal Integrity (page 698)).

Accessories should use tunable USB PHYs to maximize their chances of passing the tests.

Accessories that support USB High-Speed must:

- Support test packet mode.
- Detect the test Vendor ID (0x1A0A) and Product ID (0x0104) and trigger test packet generation.

48.3 Test Procedures

Test procedures for both USB Device Mode and USB Host Mode accessories are contained in this section.

48.3.1 Signal Integrity

The purpose of this signal integrity (SI) procedure is to provide clear information on logic levels, jitter budget, rise time, and fall time for USB signals transmitted by embedded hosts at the near end test point (USB-A receptacle).

This test procedure must be performed using the configuration of a typical user. It must not be performed:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- On an incomplete system.
- On development hardware.
- Using custom connectors.
- Using short cables.
- Using a boosted signal.

This test procedure addresses the following:

- High-Speed USB signal logic levels, rise time, fall time, and jitter budget requirements
- Full-Speed USB signal logic levels, rise time, fall time, and jitter budget requirements
- Tools required for testing
- Setup, test procedures and complete system test bed definitions

These tests must be run by product developers or 3rd party test labs. The full test report must be submitted in order to obtain certifications.

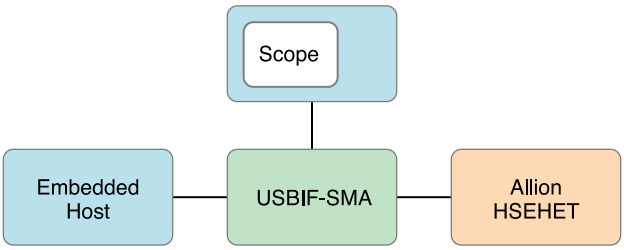
48.3.2 High-Speed USB

The following are required for validating USB host High-Speed (HS) signal integrity:

- Allion HSEHET board
- USBIF-SMA board (USBIF-SMA USB2.0 test fixture: USB-TF-HS-DEP-V21)
- USB Compliance scope with TDSUSB2 application

A proper interface to the USBIF-SMA board must be used to take eye diagrams.

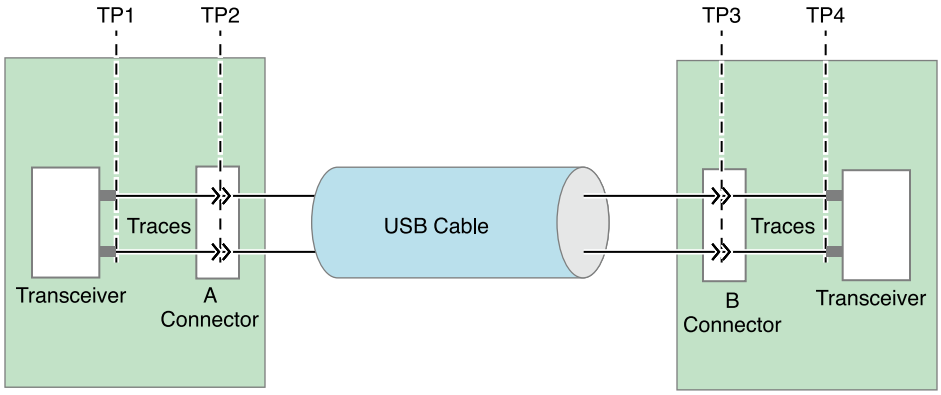
Figure 48-1 High-Speed Test Environment



All measurements must be done at the USB-A receptacle at test point TP2 near the host as shown in Figure 48-2 (page 700). See the USB 2.0 specification.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 48-2 USB Test Points



The transmit waveform from the USB host/hub at TP2 must meet the requirements in Figure 48-3 (page 700) and Table 48-1 (page 701) (see the USB 2.0 specification). Table 48-2 (page 701) is a summary of pass/fail criteria for a given host under test.

Figure 48-3 Eye Diagram Template at TP2

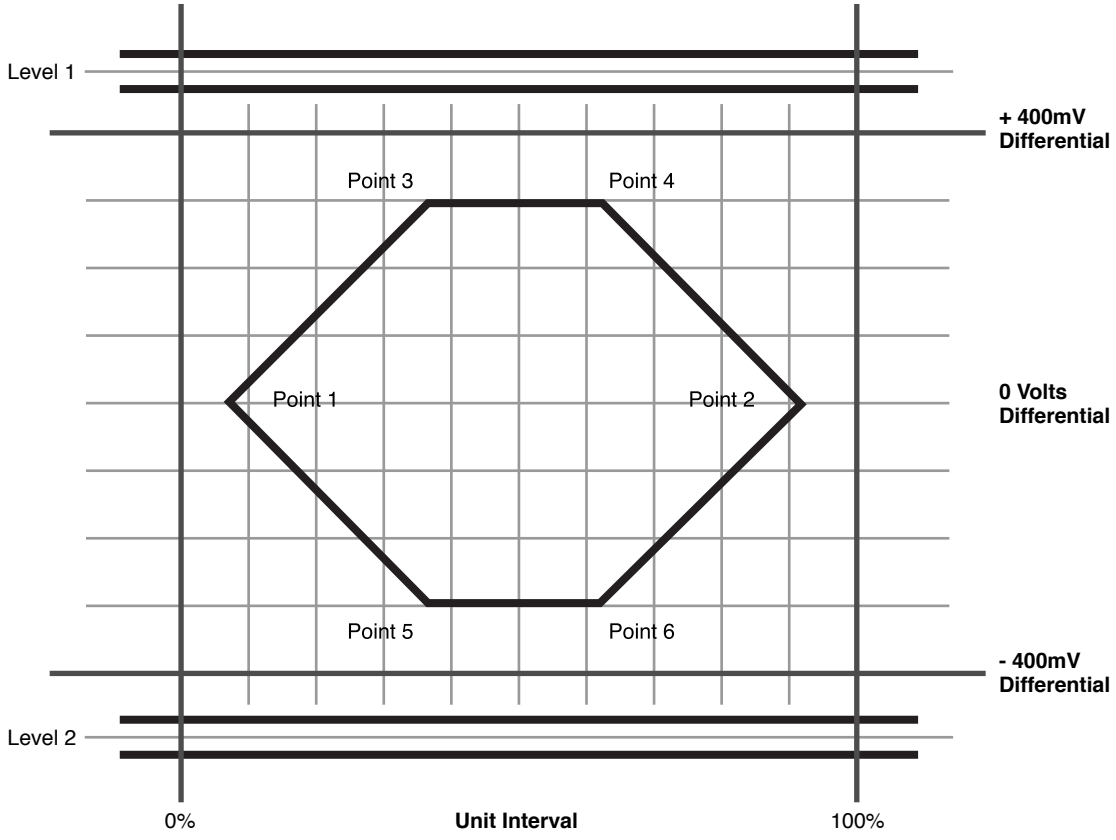


Table 48-1 Eye Diagram Template Specifications

	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition -475 in all others	N/A
Point 1	0 V	7.5% UI
Point 2	0 V	92.5% UI
Point 3	300 mV	37.5% UI
Point 4	300 mV	62.5% UI
Point 5	-300 mV	37.5% UI
Point 6	-300 mV	62.5% UI

Table 48-2 Pass/Fail Criteria for Host Under Test

Req. #	Requirement	Value
SI_HS_01	USBDP/DM Mask Levels	±300 mV
SI_HS_02	Rise Time	500 ps
SI_HS_03	Fall Time	500 ps
SI_HS_04	Jitter	50 ps
SI_HS_05	Signal Rate	479.76 Mbps - 480.240 Mbps

The host must continuously transmit a test packet (see Table 48-3 (page 701)) within inter-packet gap limits of 88 bits and < 125 usec.

Table 48-3 Test Packet Layout

NRZI Symbol (Fields)	NRZ Bit Strings	Number of NRZ Bits
{KJ * 15}, KK (SYNC)	{00000000 * 3}, 00000001	32
KKJKJKKK (DATA0 PID)	11000011	8
JKJKJKJK * 9	00000000 * 9	72
JJKKJJKK * 8	01010101 * 8	64

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

NRZI Symbol (Fields)	NRZ Bit Strings	Number of NRZ Bits
JJJJKKKK * 8	01110111 * 8	64
JJJJJJJJKKKKKKKK * 8	0, {111111S * 15}, 111111	97
JJJJJJJK * 8	S, 111111S, {0111111S * 7}	55
{JKKKKKKK * 10}, JK	00111111, {S0111111 * 9}, S0	72
JJJKKKJJKKKKJKK (CRC16)	0110110101110011	16
JJJJJJJJ (EOP)	01111111	8

48.3.2.1 Test Procedure (HS Signal Quality EL_2, EL_3, EL_6, EL_7)

Perform the following steps:

1. Connect Host to USBIF-SMA board, set switch in "Init" mode.
2. Connect Scope differential probes to USBIF-SMA board with proper orientation.
3. Set the SW2 on Allion HSEHET board to position "4" Test Packet PID (0x0104).
4. Connect the Allion HSEHET board to the USBIF-SMA board. Wait 15 sec.
5. Set the switch on USBIF-SMA board to apply terminations.
6. Run the TDSUSB application on scope with near end setting.
7. Capture waveform and analyze with TDSUSB application.
8. Save the HS eye diagram, waveforms and USBIF report.

Accessories supporting USB Role Switch must be tested in both USB Host Mode and USB Device Mode. The accessory should transmit test packets for 5 minutes, role switch, and transmit test packets for 5 minutes again. An indicator such as an LED can be helpful to indicate this change.

48.3.2.2 Naming Report

High-Speed accessory file name format:

```
HostName_Model#_FirmwareVersion_HSEHET_Date_PASS/FAIL_MFi_Report.html
```

Accessories supporting USB Role Switch must submit separate reports corresponding to USB Device Mode and USB Host Mode.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

48.3.2.3 Pass/Fail Criteria

The report generated by TDSUSB will indicate failure if there is a mask, timing or jitter failure. Submit the report as is without any modifications.

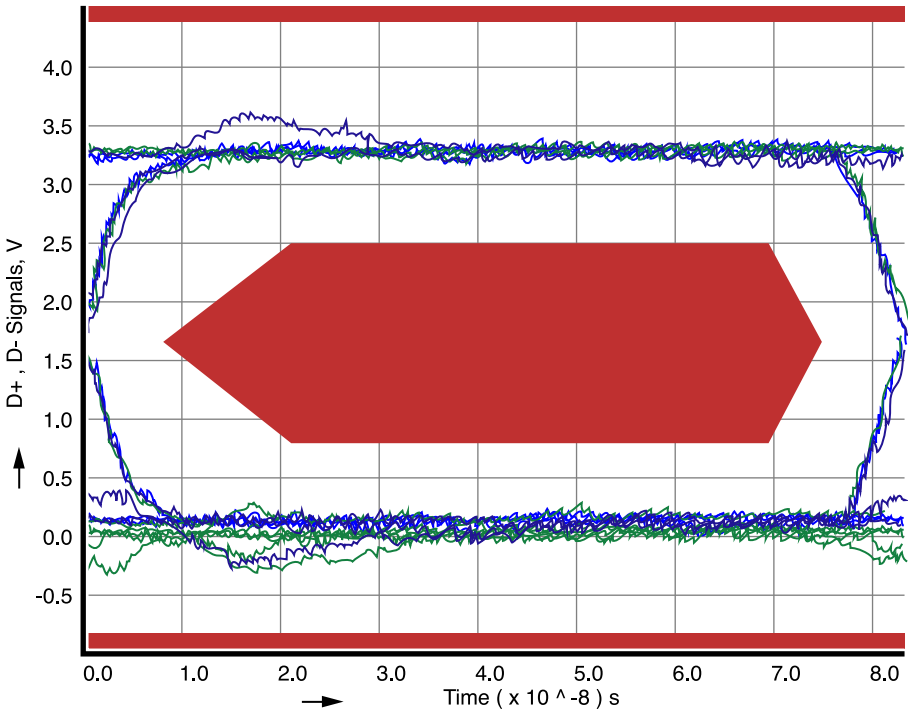
48.3.3 Full-Speed USB

The following are required for validating USB host Full-Speed (FS) signal integrity:

- FS Adjacent Trigger device (a FS thumb drive or HS thumb drive)
- Three single ended active probes
- TDSUSBF signal quality (SQiDD) board
- TDSUSB2 running on a compliance scope
- The USB host under test must enumerate the device and send SOF packets every 1 msec. See Start of Frame Packet (SOF) (page 704).

Testing must be done at TP2 at the point where host provides a USB-A receptacle. The host must qualify the FS USB TDSUSB compliance mask. See Figure 48-4 (page 703) and Table 48-4 (page 704).

Figure 48-4 TP2 Eye Diagram Mask



Copyright © 2015 Apple Inc. All Rights Reserved.

Table 48-4 Full-Speed Signal Quality Requirements

Req. #	Requirement	Value
SI_FS_01	Mask	0.8 V - 2.5 V
SI_FS_02	Rise Time	4 ns - 20 ns (10/90)
SI_FS_03	Fall Time	4 ns - 20 ns (10/90)
SI_FS_04	Undershoot	-1.0 V
SI_FS_05	Overshoot	4.6 V
SI_FS_06	Source Jitter (next transition)	-3.5 ns to 3.5 ns
SI_FS_07	Signal Rate	12Mb/s ±0.25%

The FS eye diagram must be taken at TP2 while the FS transmitter is sending SOF packets to the device. See Start of Frame Packet (SOF) (page 704). The device must fully enumerate with the host for SOF packets to be present.

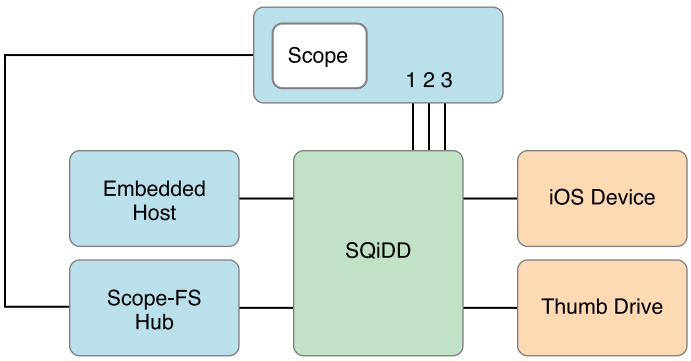
48.3.3.1 Start of Frame Packet (SOF)

Once the USB device is fully enumerated by the USB host, the host will continuously send SOF packets every 1 msec ±500ns. The SOF packet consists of an 11-bit frame number.

Sync	PID	Frame #	CRC5	EOP
	8 bits	11 bits	5 bits	

48.3.3.2 Test Environment

Figure 48-5 Full-Speed Test Environment



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The USB host must be connected to the device and an FS adjacent trigger device must be connected to the scope USB bus. Channel connections are as follows:

- Channel 1: Apple device USBDP
- Channel 2: Apple device USBDM
- Channel 3: FS adjacent trigger device USBDP

The TDSUSB2 application running on compliance scope is configured for above channel settings and full speed testing.

48.3.3.3 Test Procedure

Perform the following steps:

1. Connect accessory to SQ board, switch in "init" Position.
2. Connect Scope Ch1 and Ch2 with single ended probes to USBDP and USBDM with proper grounding.
3. Ensure the device gets enumerated and observe SOF packets every 1 msec on scope.
4. Connect a thumb drive via FS hub to scope.
5. Connect Ch3 to USBDP of adjacent trigger (use inrush current section of SQiDD board).
6. Run the TDSUSB application with FS settings.
7. Save the TDSUSB report including eye diagram.

48.3.3.4 Naming Report

Full-Speed accessory file name format:

```
HostName_Model#_FirmwareVersion_FSSQ_Date_PASS/FAIL_MFi_Report.html
```

48.3.3.5 Pass/Fail Criteria

The report generated by TDSUSB will indicate failure if there is a mask, timing or jitter failure. Submit the report as is without any modifications.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

49. USB Device Mode

When this transport is active, the accessory is a USB host and the Apple device is a USB device. USB Device Mode is not recommended for most accessories as the only features it supports are iAP2, Digital Audio (2 channel output only), and Power (to device only).

49.1 USB Embedded Host Implementation

All accessories that connect to an Apple device in USB Device Mode must comply with the requirements set forth for USB Embedded Hosts in the USB 2.0 specifications.

If the accessory integrates a USB-A receptacle that in turn connects to the Apple device via a detachable cable, a valid Test ID issued by a USB-IF authorized lab (see http://www.usb.org/developers/compliance/labs/) must be submitted to Apple as part of the self-certification audit process.

49.2 Power

All accessories that connect to an Apple device in USB Device Mode must provide power to the device as specified in Power (page 660).

49.3 Enumeration

The initialization and configuration of an attached USB device is specified in the USB 2.0 specification. This specification does not cover this topic in detail, but instead provides information specific to Apple devices. To distinguish an Apple device in USB Device Mode, the accessory must check the device descriptor of attached USB devices for the following fields:

- Vendor ID = 0x05AC
- Product ID = 0x12*nn*

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: Accessories must not attempt to distinguish between different types of Apple devices based on the least significant byte of the Product ID. Additionally, accessories must not use any other aspect of the device descriptor, such as number of configurations, interfaces, of endpoints, to determine whether an Apple device is connected. Use of additional parameters will cause the accessory to fail self certification immediately.

49.4 Connecting Multiple Apple Devices to a Mac

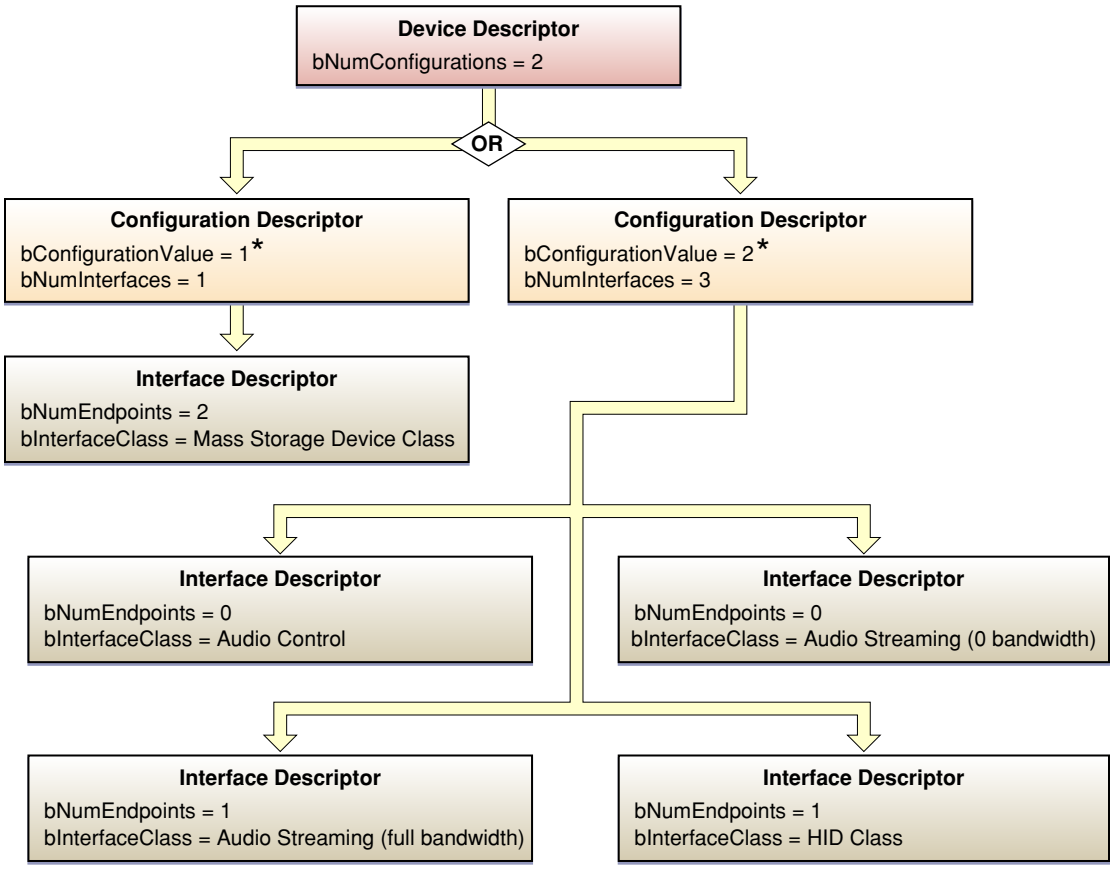
All accessories capable of connecting more than 42 Apple devices in USB Device Mode to a Mac for charge/sync purposes must connect to the Mac via Thunderbolt, not USB 2.0/3.0.

49.5 iAP2 Configuration

The iUI configuration enables iAP2 over the USB Device Mode transport and USB Device Mode Audio. For iAP2, a Human Interface Device (HID) interface is exposed to the accessory and uses two endpoints for communication: the control endpoint (endpoint number 0) is used for OUT data, while the HID interrupt endpoint is used for IN data.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 49-1 USB Device Mode Interface Descriptor



Note: The Apple device USB isochronous audio data endpoint descriptor bmAttributes field erroneously returns the Synchronization Type field (D3:2) as b00 (no synchronization) instead of the correct value, b11 (synchronous). Apple devices support synchronous data transfers, so accessories must override these attribute bits. The erroneous b00 value is retained for backwards compatibility with older Apple device accessories.

The Apple device HID interface utilizes several vendor-specific HID reports, some of which are used to transport data from the accessory (output reports) and some of which are used to transport data to the accessory (input reports). To send data to an Apple device, the accessory must choose one or more appropriately sized HID reports in which to embed the iAP2 link packet and send them to the Apple device HID interface using USB SetReport commands. The Apple device reassembles the iAP2 link packet and processes it. The process is repeated in reverse when the Apple device sends responses or iAP2 link packets to the accessory. In this case, the data is sent on an interrupt endpoint associated with the HID interface.

The different HID report sizes, endpoint requirements, and particulars are all described in the USB descriptors that accompany the interface.

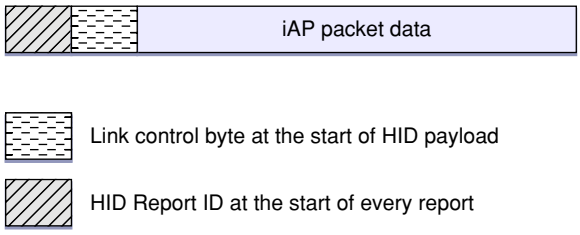
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: Accessories must always request and parse the HID report descriptor each time an Apple device is connected or the accessory resets USB, because the HID Report ID and size descriptions may change.

49.6 HID Interface

As mentioned earlier, the HID interface breaks iAP packets up into a stream of vendor-specific HID reports and transports them across USB in either direction. To help manage this, it breaks this stream up into logical sets of reports, where a set of reports encompasses one or more complete iAP packets. For instance, a set could be a single HID report containing one iAP packet or a set of seven HID reports containing a total of three iAP packets. A vendor-specific HID report, as defined by the USB specification, consists of a Report ID followed by a payload of data that is specific to the vendor and its usage. The payload of this HID report is a link control byte (LCB), followed by iAP packet data.

Figure 49-2 USB Device Mode Interface HID Report



The HID Report ID indicates the type of report and implies the size of the report. Every report of a given type is the same size. The Apple device specifies several different report types. The USB host must analyze the HID report descriptor of the Apple device at runtime to determine which Report ID corresponds to the most appropriate report type for each transfer. Note that the HID report descriptor may change in future Apple devices.

Usage of a HID Report ID is defined by the USB specification and is not specific to Apple devices, in contrast to the LCB and the rest of the payload.

The link control byte provides a mechanism for grouping sets of reports and is used by the HID interface to manage the data flow.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

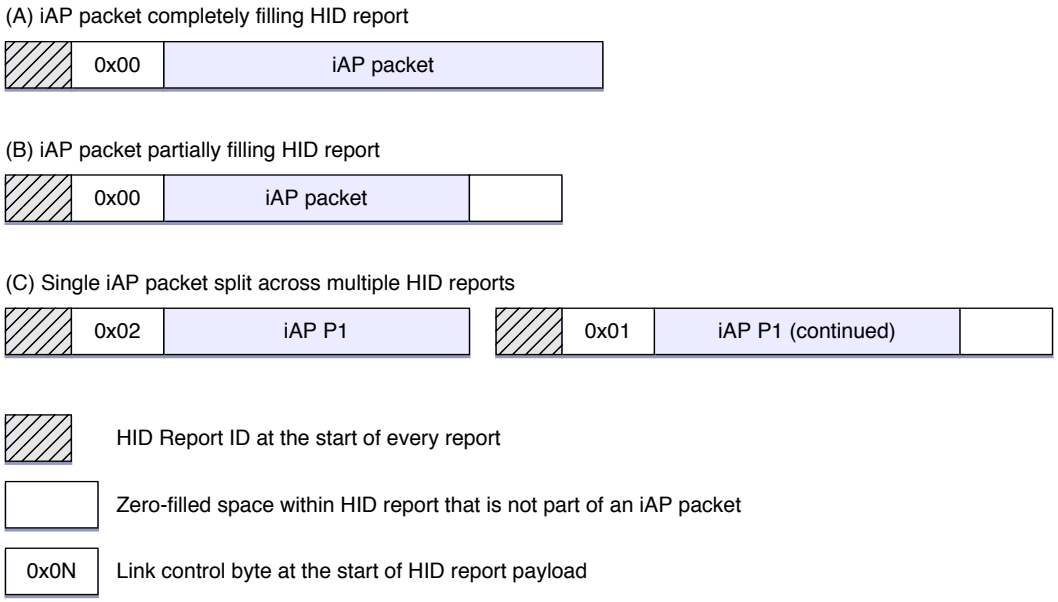
Table 49-1 Link control byte usage

Bit	Name	Usage
Bit 0	Continuation	0 indicates that this HID report is the first in a set of one or more reports. This also implies that any previous sets are completed. Any incomplete iAP packets received prior to the arrival of this report are flushed and lost. 1 indicates that this report is not the start of a set, but is a continuing part of a set.
Bit 1	More to Follow	0 indicates that this report is the last in a set. Any following reports must be part of another set. 1 indicates that the current report set is not yet complete and there is at least one more report expected.
Bits 2-7	Reserved	Set to 0.

In general, iAP packets can be packed into HID reports in any manner, given the following limitations:

- All unused space within any HID report must be set to 0x00.
- If there is more than one iAP packet in the same HID report, there must be no unused space between them.
- If an iAP packet is split across multiple HID reports, all component reports must be in the same logical set of reports.

Figure 49-3 USB Device Mode Interface Report Packing



49.7 Test Procedures

49.7.1 iAP2 Audio Tests

Verify that the following iAP2 control session message(s) are sent or received:

- StartUSBDeviceModeAudio (page 866)
- USBDeviceModeAudioInformation (page 867)
- StopUSBDeviceModeAudio (page 867)

50. USB Host Mode

USB Host Mode is recommended for all new accessory designs other than simple chargers and charge/sync cables. When this transport is active, the Apple device is a USB host.

Apple devices are capable of functioning as either a USB 2.0 High Speed or Full Speed host. Unless otherwise overridden in this specification, the accessory must comply with the USB 2.0 specification, plus any applicable device class-specific USB specifications that are available at http://www.usb.org/developers/docs/usb20_docs/.

The following additional USB implementation requirements apply:

- All USB descriptors (particularly the endpoint descriptors and the bMaxPower field of the configuration descriptors) must accurately represent the accessory's capabilities.
- Every USB device descriptor must declare a unique Vendor ID (VID) assigned by the USB-IF and a unique Product ID (PID) assigned by the accessory developer. The USB-IF Vendor ID must be assigned to the MFi licensee whose brand name appears on the accessory or its packaging. Re-using the Vendor ID of a silicon/component/reference design provider or contract manufacturer is specifically prohibited.
- The USB device descriptor must set the `Device Class`, `Device Subclass`, and `Device Protocol` fields to 0x00 to indicate that each interface descriptor will specify its own class code. If the accessory is using Interface Association Descriptors, it must use the Multi-Interface Function Device Class Codes and instead set the `Device Class`, `Device Subclass`, and `Device Protocol` fields to 0xEF, 0x02, and 0x01 respectively.
- All USB Device, Configuration, and Interface descriptors must be accompanied by human-readable String descriptors. Among these String descriptors, the following must match IdentificationInformation (page 807) message parameter values that the accessory passes to the Apple device:
  - USB Manufacturer String Descriptor must match the `Manufacturer` parameter value.
  - USB Product String Descriptor must match the `Name` parameter value.
  - USB Serial Number String Descriptor must match the `SerialNumber` parameter value.
- There is no USB 5 V VBUS supply from an Apple device; only Accessory Power is available.
- Upon detecting USB Suspend (the USB bus has been idle for 3.0 ms) the accessory must immediately enter Low Power Mode and remain there until it receives any non-idle signaling.
- The accessory must be capable of handling simultaneous bulk IN and bulk OUT transfers.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

50.1 iAP2 Interface Descriptor

Accessories that establish an iAP2 connection to an Apple device in USB Host Mode must declare a vendor-specific interface with one bulk IN endpoint and one bulk OUT endpoint. The accessory can determine that the Apple device has successfully entered USB Host Mode by detecting that it has been fully enumerated by the Apple device.

Table 50-1 USB Host Mode iAP2 interface descriptor

USB Descriptor	Value	Comments
Interface Class	0xFF	Vendor-specific interface
Interface Subclass	0xF0	MFi accessory
Interface Protocol	0x00	
Interface String	'iAP Interface'	
Number of Endpoints	2	1 bulk IN and 1 bulk OUT endpoint descriptor must be specified

50.2 iAP2 Data Transfers

In USB Host Mode, the Apple device sends iAP2 data to the accessory using the bulk OUT endpoint. The accessory must return a USB ACK packet if the write operation is successful or return a USB NAK packet if it is not ready to accept data. If the accessory repeatedly returns a USB NAK packet for more than 1 second, the write operation will time out and information from the Apple device may be lost.

> **Note:** Any iAP2 or EA native USB data being sent by the accessory which is an exact multiple of the USB endpoint's maximum payload size must be followed by a USB zero length packet (ZLP).

50.3 iAP2 Performance Optimization

Accessories should send the largest iAP2 link packets possible. This permits more data to be sent to the Apple device before a response is required. Sending small packets and waiting for the Apple device to respond introduces unnecessary delays and lowers data throughput.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

51. USB Role Switch

Some Apple devices can enter USB Host Mode from USB Device Mode after receiving a specific USB Vendor Request. Accessories that are normally USB hosts when not connected to an Apple device may choose to implement USB Role Switch and take advantage of features that are only available to USB Host Mode accessories while using a standard USB to Lightning cable.

51.1 USB Role Switch Requirements

The following requirements apply to accessories that support the USB role switch feature:

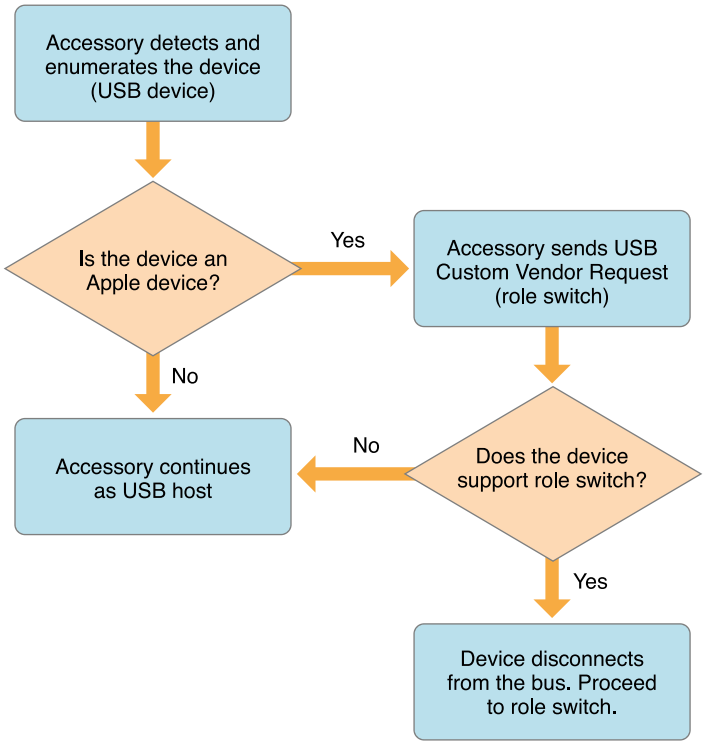
- The accessory must have a USB-A receptacle that is capable of functioning in both USB Host and USB Device roles.
- The receptacle must be labeled in accordance with the requirements specified in the Power feature. See Multiple Connectors (page 667) for details.
- The USB-A receptacle must provide power compatible with Apple devices as specified in the Power feature.
- The accessory must pass the USB test procedures (see Test Procedures (page 698)) in both USB Host Mode and USB Device Mode. Accessories should send test packets for 5 minutes in USB Device Mode, Role Switch, and send test packets for 5 minutes in USB Host Mode.

51.2 USB Role Switch Usage

The process of initiating a USB role switch using a custom vendor request is outlined in Figure 51-1 (page 715).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 51-1 USB Role Switch Custom Vendor Request (Apple device is USB Device, Accessory is USB Host)



The sequence of events that takes place when an accessory is connected to an Apple device is as follows. For Steps 1-5, the accessory is in USB Host Mode and the Apple device is in USB Device Mode. For Steps 6-7, these roles are reversed.

1. The accessory (USB host) must detect and enumerate the Apple device (USB device).
2. The accessory may send the following USB Custom Vendor Requests to get enabled capabilities.

Table 51-1 USB Vendor Request for Apple Device to Get Supported Capabilities

Field	Value	Comments
`bmRequestType`	`0xc0`	Device-to-host request, vendor-defined type, device is recipient
`bRequest`	`0x53`	Vendor-defined USB get enabled capabilities request
`wValue`	`0x00`	Reserved
`wIndex`	`0x00`	Reserved
`wLength`	`4`	4 bytes expected

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The data payload returned from the Apple device will be 4 bytes: a wValue (2 byte little endian bitfield, see Table 51-2 (page 716)) followed by wIndex (2 byte little endian value). Note that wValue is encoded as a bitfield, so all Reserved values must be ignored.

Table 51-2 USB Vendor Request for Apple Device to Host Mode Switch: wValues (bitfield)

Value	Description
`0x00`	Default value
`0x01`	Accessory supports Apple CarPlay (see CarPlay (page 381))
All other values	Reserved

3. The accessory must send the following USB Custom Vendor Request, which asks for a USB role switch:

Table 51-3 USB Vendor Request for Apple Device to Host Mode Switch

Field	Value	Comments
`bmRequestType`	`0x40`	Host-to-device request, vendor-defined type, device is recipient
`bRequest`	`0x51`	Vendor-defined USB role switch request
`wValue`	`0xnn`	See Table 51-2 (page 716)
`wIndex`	`0x00`	Reserved
`wLength`	`0`	No data transfer

4. If the Apple device supports this role switch request, it disconnects itself from the bus. If it does not, it issues a `STALL` packet. See *On-The-Go Supplement, Revision 2.0, version 1.1a*, Section 5.2.1, Step B. If the accessory receives a `STALL` packet, it must assume that the Apple device does not support this feature and proceed accordingly. For example, the accessory may establish an iAP2 connection using USB Device Mode instead.
5. The accessory must detect the bus disconnect and turn on D+ pull-up. See *On-The-Go Supplement, Revision 2.0, version 1.1a*, Section 5.2.1, Step C.
6. The Apple device will assert a bus reset signal to start using the bus. See *On-The-Go Supplement, Revision 2.0, version 1.1a*, Section 5.2.1, Step D.
7. The accessory (USB device) must wait at least 1000 ms for the Apple device (USB host) to start enumeration of the accessory. See *On-The-Go Supplement, Revision 2.0, version 1.1a*, Section 7.1.5.
8. If 1000 ms passes without any traffic, the accessory must transition back to USB Host Mode.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

> **Note:** In previous versions of this specification, a USB Vendor Request with the `bRequest` field set to `0x50` was defined. That Vendor Request is now deprecated and must not be used in new accessory designs.

The following two events take place when the cable connecting the Apple device to the accessory's USB-A receptacle is unplugged from either or both ends:

- The Apple device detects loss of VBUS and switches back to USB Device Mode.
- The accessory detects a minimum of 200 ms of bus inactivity and transitions back to USB Host Mode. See *On-The-Go Supplement, Revision 2.0, version 1.1a* , Section 7.1.6.

If the accessory claims the capability to re-charge an Apple device, it must do so at the required power levels both before and after USB Role Switch has occurred.

If the accessory does not claim the capability to re-charge an Apple device, it must still provide 500 mA to the Apple device whenever the controller is a USB Host and the Apple device is in USB Device Mode. After the Apple device has started USB enumeration of the accessory, the accessory must:

- Stop providing 500 mA to the Apple device.
- Provide a USB configuration descriptor that declares no need for power (bMaxPower = 0) to the Apple device.
- Set the PowerProvidingCapability parameter in its iAP2 AccessoryIdentificationInformation message to 'None'.
- Set the MaximumCurrentDrawnFromDevice parameter in its iAP2 AccessoryIdentificationInformation message to 0 mA.

51.3 Test Procedures

51.3.1 USB Role Switch

Verify that "USB Role Switch (Apple device to USB Host)" is listed under the USB Transfers section of ATS.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

52. Vehicle Status

Accessories may use the Vehicle Status feature to send sensor data and other useful information from a vehicle to an Apple device.

The following types of information can be communicated:

- Engine type
- Estimated range remaining
- Outside temperature
- Low range indicator warning status

52.1 Vehicle Status Requirements

Accessories that implement the Vehicle Status feature must provide live information from a vehicle. Simulation of vehicle status information from other sources is not allowed.

All accessories that support the VehicleStatus feature via iAP2 must send or receive the following iAP2 control session message(s):

StartVehicleStatusUpdates (page 868)

VehicleStatusUpdate (page 868)

StopVehicleStatusUpdates (page 868)

52.2 Vehicle Status Usage

Accessories that implement this feature must identify one Table 59-19 (page 813) and one Table 59-21 (page 813).

If the Apple device can make use of some/all of the information, it will send a StartVehicleStatusUpdates (page 868) message to the accessory informing the accessory which information types send updates for in a VehicleStatusUpdate (page 868) message.

The accessory must only send vehicle status updates to the Apple device:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Between StartVehicleStatusUpdates (page 868) and StopVehicleStatusUpdates (page 868) messages.
- Once after receiving the StartVehicleStatusUpdates (i.e. providing an initial value).
- When the value has changed. The range and temperature values should change by at least 0.5 miles or 0.5 °C respectively before an update is sent.
- At a maximum rate of 1 Hz.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

53. VoiceOver

VoiceOver is the screen reader feature built into both Apple's iOS and OS X operating systems. Users who are blind or have low vision can interact with and control Apple devices and Macs simply by moving their finger over the touchscreen or touch-sensitive trackpad. For more information on VoiceOver, see http://www.apple.com/accessibility/voiceover/. In addition, when VoiceOver is paired with certain accessories users are able to use VoiceOver even if they are unable to touch the display at all.

Most VoiceOver accessories interface with one or more switch-type inputs. Accessory developers should consider supporting VoiceOver if the accessory is intended for users who cannot see the display or touch the screen. If the user can see the display, consider supporting the AssistiveTouch or Assistive Switch Control features.

53.1 VoiceOver Requirements

All accessories supporting the VoiceOver feature must be targeted at users with special needs.

All accessories that support the VoiceOver feature via iAP2 must send or receive the following iAP2 control session message(s):

- StartVoiceOver (page 869)
- RequestVoiceOverMoveCursor (page 870)
- RequestVoiceOverActivateCursor (page 870)
- StartVoiceOverUpdates (page 872)
- VoiceOverUpdate (page 872)

All accessories that support the VoiceOver feature via iAP2 may also send or receive the following iAP2 control session message(s):

- StopVoiceOver (page 869)
- RequestVoiceOverScrollPage (page 870)
- RequestVoiceOverSpeakText (page 871)
- RequestVoiceOverPauseText (page 871)
- RequestVoiceOverResumeText (page 872)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

RequestVoiceOverConfiguration (page 873)

StopVoiceOverUpdates (page 873)

StartVoiceOverCursorUpdates (page 874)

VoiceOverCursorUpdate (page 874)

StopVoiceOverCursorUpdates (page 875)

53.2 VoiceOver Usage

Accessories supporting VoiceOver must identify themselves as using one or more VoiceOver messages and have that configuration accepted by the Apple device. Apple devices that do not support VoiceOver accessories will reject accessory identifications that include VoiceOver messages.

The VoiceOver feature is started and stopped using the StartVoiceOver (page 869) and StopVoiceOver (page 869) messages. All compatible accessories must send the StartVoiceOver (page 869) message. Also, all VoiceOver accessories must receive and process VoiceOverUpdate (page 872) messages to be notified when the feature is enabled or disabled asynchronously. The StartVoiceOverUpdates (page 872) message will signal the device to start generating and sending VoiceOverUpdate (page 872) messages to the accessory.

It is possible (and not uncommon) for more than one VoiceOver-compatible accessory to be connected to a device at the same time. All of those accessories, or their users, may independently start or stop the VoiceOver feature at any time, and a VoiceOver accessory must handle these use cases correctly.

Several additional VoiceOver messages deal with manipulation and selection of user interface elements. Basic VoiceOver accessories will typically only send the RequestVoiceOverMoveCursor (page 870) and RequestVoiceOverActivateCursor (page 870) messages.

Some accessories may choose to also receive VoiceOverCursorUpdate (page 874) messages; this message provides additional information to the accessory about the currently selected user interface element if the app provides it.

The accessory may send RequestVoiceOverConfiguration (page 873) to adjust the volume and/or speaking rate after the feature is started. Additionally, accessories can request that the speech synthesizer be used to render certain useful phrases specific to the accessory at key moments. The speech synthesizer volume is persistent and will be saved and restored by the Apple device when VoiceOver is stopped and started.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

53.3 Test Procedures

53.3.1 iAP2 Tests

1. Verify that the accessory is targeted at users with special needs only.
2. Verify that the following iAP2 control session message(s) are sent or received:
   - StartVoiceOver (page 869)
   - RequestVoiceOverMoveCursor (page 870)
   - RequestVoiceOverActivateCursor (page 870)
   - StartVoiceOverUpdates (page 872)
   - VoiceOverUpdate (page 872)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54. Wi-Fi

54.1 Overview

All accessories that use Wi-Fi as a transport should test using the procedures in Test Procedures for the Consumers (page 723).

54.2 Test Procedures for the Consumers

54.2.1 Association Tests

54.2.1.1 Association and Authentication Test

1. Configure AP for various radio modes with different security and various channels to test for Association/Authentication for all Combinations.
2. Device should be able to Associate with all combinations of radios modes, Security Types and channels.

54.2.1.2 Hidden Networks

1. Configure AP to have hidden network. Attempt to associate.
2. Device should be able to associate and get IP connectivity with the hidden network.

54.2.1.3 Unicast Key Rotation

1. Configure AP to use PSK (test both WPA-PSK and WPA2-PSK). Configure key session timeout to a smaller value. Associate the client with the network.
2. Device should be able to associate and get IP connectivity with the network.
3. Stay connected with the network until the session times out.
4. Device should be able to establish the session again with the network automatically.

54.2.1.4 Group Key Renewal

1. Configure AP to use PSK (test both WPA-PSK and WPA2-PSK). Configure group key renewal timeout to a smaller value. Associate the client with the network.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Device should be able to associate and get IP connectivity with the network.
3. Stay connected with the network until the group key renewal timer expires.
4. Device should stay connected to the AP and renew group keys.

54.2.1.5 MAC Filtering

1. Configure AP with MAC Filter enabled and add Device MAC address to the filter. Attempt to associate to the network.
2. Device should be able to associate and get IP connectivity with the network.

54.2.1.6 DHCP Expiry

1. Configure AP to have lower DHCP lease time. Associate the Apple device with the network.
2. Device should be able to associate and get IP connectivity with the network.
3. Stay connected and keep the Apple device awake (unlocked) until the DHCP lease expires.
4. Device should be able to stay connected with the network and should able to renew DHCP configuration and defend the IP address.

54.2.1.7 Static IP Configuration

1. Setup the AP and associate client with the network with Static IP config of the same subnet.
2. Device should be able to associate and route traffic using the static IP.

54.2.1.8 Guest Mode Association

1. Configure AP for a Network and if available have Guest Mode enabled. Associate the client to the Guest Network.
2. Device should be able to associate and get IP connectivity with the Guest network.

54.2.1.9 Unicast Key Rotation for Guest Mode

1. Configure AP to use PSK (test both WPA-PSK and WPA2-PSK) on Guest Network. Configure key session timeout to a smaller value. Associate the client with the network.
2. Device should be able to associate and get IP connectivity with the network.
3. Stay connected with the Guest network until the session times out.
4. Device should be able to establish the session again with the network automatically.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54.2.1.10 Group Key Renewal for Guest Mode

1. Configure AP to use PSK (test both WPA-PSK and WPA2-PSK) on Guest Network. Configure group key renewal timeout to a smaller value. Associate the client with the network.
2. Device should be able to associate and get IP connectivity with the network.
3. Stay connected with the Guest network until the group key renewal timer expires.
4. Device should stay connected to the AP and renew group keys.

54.2.2 Performance Tests

54.2.2.1 Basic throughput test with iperf with different traffic types

1. Configure an AP and Associate client with it. Run a iperf server/client on a wired host and start single iperf traffic stream to/from client for 30s. Try each type of QoS traffic stream.
2. Client should deliver good uplink and downlink throughput number for TCP and UDP for each traffic type.

54.2.2.2 Run WMM tests on client

1. Configure an AP and Associate client with it. Run a iperf server/client on a wired host and run multiple simultaneous uplink traffic streams of different traffic types and see that WMM works properly.
2. Client should give higher priority to VO and VI traffic over BK and BE traffic.

54.2.2.3 Ensure that WMM is enable when you have 11n+ Mode enabled

1. Check the WMM Mode by default on the AP.
2. If the default radio mode is 11n+, it should have WMM enabled by default.
3. Disable WMM mode and Associate the client with the AP.
4. When you disable WMM mode, it should disable 11n+ mode (HT Rates) on the AP.
5. Device should be able to associate with non-HT data rates.

54.2.2.4 FaceTime Video Testing

1. Have client associated to the AP and configure two clients to have FaceTime account.
2. Both clients should be able to associate with the AP.
3. Start Face time video call between two clients.
4. FaceTime video call should have good quality and video and audio should be in sync.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54.2.2.5 AirPlay Mirroring

1. Have the AP setup and connect an AppleTV to that network and connect AppleTV to a TV.
2. Connect the test client to the same network.
3. The client and AppleTV are both connected to the same network.
4. Start AirPlay with the AppleTV and use mirroring on the client. Play a video on the client.
5. You should be able to discover AppleTV in AirPlay discovery.
6. You should be able to see your client screen on the TV.
7. You should be seeing the video from client on the TV screen with maintaining good quality.
8. There should be no visible or audible blips while mirroring animated content (e.g scrubbing in Safari, playing iTunes Radio) for 2 minutes, minimum.

54.2.2.6 AirDrop

1. Setup an AP and connect two clients to the network. Have AirDrop enabled on both Apple devices for everyone.
2. Device should be able to associate and get IP connectivity with the network.
3. Select a video file (50 MB) and choose to send it via AirDrop to the other client.
4. Client1 should be able to discover Client2 in AirDrop discovery and the file should be transferred with acceptable throughput.
5. Reduce the signal strength (RSSI) for Device1 to the AP and Device1 to Device2 by moving Device1 further away from both the AP and Device2.
6. Client1 should be able to discover Client2 in AirDrop discovery and the file should be transferred with acceptable throughput.

54.2.3 Other Scenarios

54.2.3.1 Dual Band Roaming

1. Configure a dual band AP with the same network name (SSID) on both bands. Associate the client to the network.
2. The client should associate first to the 2.4 GHz network and then roam to 5 GHz network.
3. It should stay connected to the 5 GHz BSSID.
4. Move away from the AP so that signal weakens. At certain threshold (at the edge of the 5 GHz coverage), it should roam to 2.4 GHz BSSID.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54.2.3.2 WDS Bridging

1. Have two APs setup such that AP2 is connected with AP1 in wireless bridge mode. Have a wired/wireless Host connected to AP1.
2. Place AP2 further away from AP1 such that their coverage do not overlap completely.
3. Have the Client connect to the network such a way that it is connected to AP2.
4. Client is connected to AP2.
5. Send traffic to the Host connected to AP1.
6. Client is able to pass traffic to the Host connected on AP1.
7. In Sniffer Capture you should be seeing the data frames having all 4 MAC addresses (Source, destination, Transmitter, Receiver).

54.2.3.3 Longevity

1. Configure AP and associate the client to the network.
2. Device should be able to associate and get IP connectivity with the network.
3. Lock the screen on the client and stay idle for 30 mins or more. And run any kind of traffic after idle cycle.
4. You should stay connected to the network and should be able to do data transfer (download a file) after waking up.

54.2.4 Secondary Tests

54.2.4.1 Default DTIM value

1. Keep the default DTIM value on the AP configuration. Have sniffer running on the AP channel for the test.
2. Device should be able to associate and get IP connectivity with the network.
3. Lock the screen on the Apple device and let it go to power save.
4. Device should wake up on right intervals and we do not drop packets.

54.2.4.2 Minimum configurable DTIM value

1. Keep the minimum configurable DTIM value on the AP configuration. Have sniffer running on the AP channel for the test.
2. Device should be able to associate and get IP connectivity with the network.
3. Lock the screen on the Apple device and let it go to power save.
4. Device should wake up on right intervals and we do not drop packets.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54.2.4.3 Different Beacon Interval values with Different DTIM values

1. Try different Beacon Interval settings along with the DTIM settings mentioned in above tests.
2. Device should be able to associate and get IP connectivity with the network.
3. Lock the screen on the Apple device and let it go to power save.
4. Device should wake up on right intervals and we do not drop packets.

54.3 Test Procedures for the Enterprises

54.3.1 Association Tests

54.3.1.1 Association and Authentication test

1. Configure AP for various radio modes with different security and various channels to test for Association/Authentication for all Combinations.
2. Device should be able to Associate with all combinations of radios modes, Security Types and channels.

54.3.1.2 Hidden Networks

1. Configure AP to have hidden network. Attempt to associate.
2. Device should be able to associate and get IP connectivity with the hidden network.

54.3.1.3 Unicast Key Rotation

1. Configure AP to use PSK (test both WPA-PSK and WPA2-PSK). Configure key session timeout to a smaller value. Associate the client with the network.
2. Device should be able to associate and get IP connectivity with the network.
3. Stay connected with the network until the session times out.
4. Device should be able to establish the session again with the network automatically.

54.3.1.4 Group Key Renewal

1. Configure AP to use PSK (test both WPA-PSK and WPA2-PSK). Configure group key renewal timeout to a smaller value. Associate the client with the network.
2. Device should be able to associate and get IP connectivity with the network.
3. Stay connected with the network until the group key renewal timer expires.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

4. Device should stay connected to the AP and renew group keys.

54.3.1.5 MAC Filtering

1. Configure AP with MAC Filter enabled and add Device MAC address to the filter. Attempt to associate to the network.
2. Device should be able to associate and get IP connectivity with the network.

54.3.1.6 DHCP expiry

1. Configure AP to have lower DHCP lease time. Associate the Apple device with the network.
2. Device should be able to associate and get IP connectivity with the network.
3. Stay connected until the DHCP lease expires.
4. Device should be able to stay connected with the network and should able to renew DHCP configuration.

54.3.1.7 Static IP Configuration

1. Setup the AP and associate client with the network with Static IP config of the same subnet.
2. Device should be able to associate and route traffic using the static IP.

54.3.1.8 DHCP Passthrough/Relay

1. Configure AP to Passthrough/Relay DHCP. Associate the client with the network.
2. Device should be able to stay connected with the network and should able to get DHCP IP.

54.3.1.9 IPv6

1. Configure DHCP server with IPv6. Associate the Apple device with the network.
2. Device should be able to associate and get IPv6 connectivity with the network.

54.3.1.10 Guest Mode Association

1. Configure AP for a Network and if available have Guest Mode enabled. Associate the client to the Guest Network.
2. Device should be able to associate and get IP connectivity with the Guest network.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54.3.2 Performance Tests

54.3.2.1 Basic throughput test with iperf with different traffic types

1. Configure an AP and Associate client with it. Run a iperf server/client on a wired host and start single iperf traffic stream to/from client for 30s. Try each type of QoS traffic stream.
2. Client should deliver good uplink and downlink throughput number for TCP and UDP for each traffic type.

54.3.2.2 Run WMM tests on client

1. Configure an AP and Associate client with it. Run a iperf server/client on a wired host and run multiple simultaneous uplink traffic streams of different traffic types and see that WMM works properly.
2. Client should give higher priority to VO and VI traffic over BK and BE traffic.

54.3.2.3 Ensure that WMM is enable when you have 11n+ Mode enabled

1. Check the WMM Mode by default on the AP.
2. If the default radio mode is 11n+, it should have WMM enabled by default.
3. Disable WMM mode and Associate the client with the AP.
4. When you disable WMM mode, it should disable 11n+ mode (HT Rates) on the AP.
5. Device should be able to associate with non-HT data rates.

54.3.2.4 FaceTime Video Testing

1. Have client associated to the AP and configure two clients to have FaceTime account.
2. Both clients should be able to associate with the AP.
3. Start Face time video call between two clients.
4. FaceTime video call should have good quality and video and audio should be in sync.

54.3.2.5 AirPlay Mirroring

1. Have the AP setup and connect an AppleTV to that network and connect AppleTV to a TV.
2. Connect the test client to the same network.
3. The client and AppleTV are both connected to the same network.
4. Start AirPlay with the AppleTV and use mirroring on the client. Play a video on the client.
5. You should be able to discover AppleTV in AirPlay discovery.
6. You should be able to see your client screen on the TV.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

7. You should be seeing the video from client on the TV screen with maintaining good quality.
8. There should be no visible or audible blips while mirroring animated content (e.g scrubbing in Safari, playing iTunes Radio) for 2 minutes, minimum.

54.3.2.6 AirDrop

1. Setup an AP and connect two clients to the network. Have AirDrop enabled on both Apple devices for everyone.
2. Device should be able to associate and get IP connectivity with the network.
3. Select a video file (50 MB) and choose to send it via AirDrop to the other client.
4. Client1 should be able to discover Client2 in AirDrop discovery and the file should be transferred with acceptable throughput.
5. Reduce the signal strength (RSSI) for Device1 to the AP and Device1 to Device2 by moving Device1 further away from both the AP and Device2.
6. Client1 should be able to discover Client2 in AirDrop discovery and the file should be transferred with acceptable throughput.

54.3.3 Functional Tests

54.3.3.1 Roaming between APs with FaceTime audio/video

1. Configure 2 APs with same SSID and security settings and associate client to first AP.
2. Start FaceTime between the client and a wired host.
3. Roam away from AP1 to AP2.
4. FaceTime should not get disconnected while roaming between APs.

54.3.3.2 Bonjour Gateway Validation

1. Configure Bonjour gateway on 2 WLANs with different VLANs.
2. Connect Apple TV to first WLAN and client to second WLAN.
3. Verify if client is able see Apple TV and do AirPlay to Apple TV.
4. Device should be able to see Apple TV and do AirPlay.

54.3.3.3 AirPrint

1. Configure AP and associate client and a AirPrint capable printer.
2. Print a document from client to printer through share sheet.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Device should be able to see the printer and do AirPrint.

54.3.3.4 iTunes Syncing

1. Connect the client to iTunes (Windows and Mac).
2. Enable Wi-Fi syncing for the client.
3. Configure AP and associate client and verify if you able to sync through Settings -> General -> iTunes Wi-Fi Sync manually.
4. Device should be able to do iTunes Wi-Fi Sync.

54.3.3.5 AirPlay Variants

1. Configure AP and associate Apple TV and client to the AP.
2. AirPlay audio/video and mirroring from client to Apple TV.
3. Device should be able to see Apple TV and do audio/video and mirroring.

54.3.3.6 Popular Operational Ratesets

1. Configure AP with commonly used ratesets for each wireless band.
2. Attempt to associate to the network.
3. Device should be associate and get IP connectivity.

54.3.3.7 CSA processing

1. Configure AP in DFS channel and associate the client. Start unbuffered video traffic.
2. Trigger CSA through instrumentation or radar simulation through spectrum analyzer.
3. Verify if the client honors CSA and follows AP to new channel.
4. Device should follow AP to new channel and continue traffic.

54.3.3.8 GTK/PTK rotation (and session timeouts) in Mixed mode

1. Configure AP with 802.1X security and WPA/WPA2 mixed mode with TKIP/CCMP encryption.
2. Configure combinations of GTK/PTK rotations on AP and associate client.
3. Verify clients connectivity after key rotation.
4. Device should stay connected after key rotation and no deauth/disassoc should be seen during key rotation.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54.3.3.9 GTK/PTK rotation (and session timeouts) on Guest network

1. Configure AP with Guest network with same/different security as main network.
2. Configure combinations of GTK/PTK rotations on AP and associate client.
3. Verify clients connectivity after key rotation.
4. Device should stay connected after key rotation and no deauth/disassoc should be seen during key rotation.

54.3.3.10 Sticky Key Caching

1. Configure 2 APs 802.1X security with SKC, if supported.
2. Associate client to AP1, roam to AP2 and roam back to AP1.
3. Device should be able to associate first time and should send PMKID when it roams back to AP1.
4. Device should fast roam without going through full authentication when it roams back.

54.3.3.11 11r/11w/11v/11u

1. Configure AP with 11r/11w/11v/11u, if supported.
2. Attempt to associate client.
3. Device should be able to associate and get IP connectivity.

54.3.3.12 Multiple VAPs, BSSID Reuse

1. Configure AP with max number of VAPs (both 2.4 GHz and 5 GHz enabled) with first and last VAP in different security.
2. Associate client to first and last VAP and check for connectivity.
3. Device should be able to associate and get IP connectivity.

54.3.3.13 Mesh Implementations

1. Configure Mesh network if implemented and attempt to associate client.
2. Device should be able to associate and get IP connectivity.

54.3.3.14 Remote AP

1. Configure AP in Remote AP mode on controller as supported and attempt to associate client.
2. Device should be able to associate and get IP connectivity.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54.3.3.15 FaceTime Video Call Test During Sleep

1. Configure AP and associate client.
2. Put Apple device in sleep mode with no background traffic.
3. Make a FaceTime video call to the client.
4. Device should receive FaceTime Audio Mobile termination call when the it is in sleep.

54.3.3.16 Dynamic Channel Switch (and FaceTime)

1. Configure AP with Dynamic Channel Switching as implemented.
2. Associate client and start FaceTime. Trigger DCS and verify if client follows APs channel.
3. Device should be able to follow AP to new channel and continue FaceTime.

54.3.3.17 Client Blacklist Re-connectivity

1. Configure AP to blacklist client.
2. Attempt to associate client.
3. Remove blacklist and attempt to associate client.
4. Device should be able to connect after removing from blacklist.

54.3.3.18 Load Balancing

1. Configure load balancing on the AP to max (Status code 17 in assoc response).
2. Attempt to associate client.
3. Device should get assoc response with status code 17 and it should associate to next nearest AP.

54.3.3.19 Band Steering

1. Configure AP with different Band steering mode supported.
2. Attempt to associate client.
3. Device should be able to choose proper band/mode configured in AP.

54.3.3.20 DTIM Interval

1. Configure AP with min/max DTIM interval.
2. Associate client and verify it stays connected for longer duration (30 min plus).
3. Device should stay connected without any deauth/disassoc.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

54.3.3.21 Beacon Interval

1. Configure AP with min/max beacon interval.
2. Associate client and verify it stays connected for longer duration (30 min plus).
3. Device should stay connected without any deauth/disassoc.

54.3.3.22 DTIM and Beacon Interval

1. Configure AP with valid combination of DTIM and Beacon interval .
2. Associate client and verify it stays connected for longer duration (30 min plus).
3. Device should stay connected without any deauth/disassoc.

54.3.3.23 WMM Parameters

1. Configure AP with different WMM parameters and associate client.
2. Send different traffic type to client.
3. Device should be able to associate and it honors WMM parameters when the different traffic is sent.

54.3.3.24 80 MHz Control Channels

1. Configure AP with different control channels supported for 80 MHz.
2. Attempt to associate client.
3. Device should be able to associate and get IP connectivity.

54.3.3.25 Longevity

1. Configure AP and associate the client to the network.
2. Lock the screen on the client and stay idle for 30 mins or more.
3. Device should be able to associate and get IP connectivity with the network.
4. Device should stay connected to the network and should be able to do data transfer (download a file) after waking up.

54.3.4 Additional Tests

54.3.4.1 Idle Client Roaming

1. Configure 2 APs with same SSID and security settings and associate client to first AP.
2. Leave client idle without any traffic.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

3. Roam away from AP1 to AP2.
4. Verify client BSSID through controller.
5. Device should be able to roam to second AP when its idle.

54.3.4.2 Roaming between Wi-Fi and cellular with FaceTime audio/video

1. ConfigureAP and associate client to the network.
2. Start FaceTime between the client and a wired host.
3. Roam away from AP (RSSI < -85dBm).
4. Roam back to AP network (RSSI> -75dBm).
5. FaceTime should not get disconnected while roaming between Wi-Fi and cellular.

54.3.4.3 Dynamic VLANs

1. Configure AP with 802.1X security and Dynamic VLAN assignment on RADIUS server.
2. Attempt to associate client.
3. Device should be able to associate and get IP in the VLAN configured for the it.

54.3.4.4 Inter-Controller Roams

1. Configure 2 controllers and 2 different APs with same Wi-Fi settings.
2. Attempt to roam client from one controller AP to another.
3. Device should be able to roam successfully and continue traffic.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

55. Wi-Fi Accessory Configuration

55.1 Introduction

Apple's Wi-Fi Accessory Configuration feature is designed to allow consumers to easily set up their wireless accessories with the network credentials already stored on their iOS or OS X device.

With a Wi-Fi Accessory Configuration-enabled accessory, users can:

- Wirelessly discover their configurable accessory.
- Give their accessory a descriptive name.
- Pass network credentials to the accessory so it may join an infrastructure network.
- Discover applications compatible with the accessory.

55.2 Hardware Requirements

Wi-Fi Accessory Configuration requires:

- A device running iOS 7.0 or greater or OS X 10.9 with AirPort Utility 6.3.1 or greater to act as the configuring device.
- A Wi-Fi Accessory Configuration enabled accessory with Wi-Fi.

55.3 Network Requirements

Wi-Fi Accessory Configuration requires a wireless TCP/IP network connection between the configuring device and accessory. The Wi-Fi Accessory Configuration enabled accessory must be able to act as both a software Access Point (AP) and as a station (STA) device.

Target wireless networks require an access point compatible with any of:

- 802.11b/g
- 802.11n
- 802.11ac

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

All Wi-Fi Accessory Configuration accessories must incorporate the following features and requirements.

55.4 Wi-Fi required Accessory features

Wi-Fi Accessory Configuration accessories must include the following:

- An 802.11b/g, 802.11n, or 802.11ac radio module.
- A status indicator to indicate when the accessory is in Wi-Fi Accessory Configuration mode.
  - It is strongly recommended that this status indicator only be used for indicating when the accessory is in Wi-Fi Accessory Configuration mode, and not be used for any other purpose.
- The ability for the user to manually enter Wi-Fi Accessory Configuration mode which optionally performs a full factory setting reset of the accessory.
  - If a full factory reset is not performed, it is required that sensitive user information be erased. Examples of information to erase includes, but is not limited to:
    1. Passwords and authentication tokens (e.g. music service provider credentials, etc.).
    2. Medical data.
    3. Financial records.
    4. Personally identifiable information.

Additionally, it is strongly recommended that the accessory not use proprietary wireless technologies in the product that overlap with the international Wi-Fi spectrum.

55.5 Wi-Fi Network requirements

All Wi-Fi Accessory Configuration accessories must support a wireless network connection and Bonjour, Apple's service discovery protocol. Accessories must also support changing the Bonjour name to a user defined value. The default name of all accessories must be unique out of the box.

When in Software AP mode, the accessory must have a DHCP server as well as an HTTP server that supports persistent connections, i.e. multiple HTTP requests sent through the came TCP connection. When in STA mode, the accessory must support link-local IPv4 addressing as specified in RFC 3927.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

55.6 Wi-Fi Implementation requirements

All Wi-Fi Accessory Configuration accessories must meet all applicable requirements in this specification and must incorporate the Apple Authentication Coprocessor (see Apple Authentication Coprocessor 2.0C (page 66)).

55.7 Wi-Fi Accessory behavioral requirements

All Wi-Fi Accessory Configuration accessories must meet the following behavioral requirements:

- When an accessory is powered on and is unconfigured, it must automatically enter Wi-Fi Accessory Configuration mode
- When an accessory has been in Wi-Fi Accessory Configuration mode for more than 30 minutes, it must exit Wi-Fi Accessory Configuration mode
- The accessory may offer a mechanism for exiting Wi-Fi Accessory Configuration mode early on user action.
- When entering Wi-Fi Accessory Configuration mode, the software Access Point must use a unique SSID
- If an accessory's network credentials have been reset to factory setting defaults by the user, the accessory must fall back to the prescribed behaviors for an unconfigured accessory
- If an accessory has been configured, but loses its network connection, it must NOT enter Wi-Fi Accessory Configuration mode automatically
- It is recommended that when the accessory exits Wi-Fi Accessory Configuration mode (either after the 30 minute timeout or due to explicit user action), an alternate network setup mechanism be offered to the user, in order to support older OS X and iOS releases, or non-OS X and non-iOS products, that do not support Wi-Fi Accessory Configuration.

Note that accessories that implement WAC functionality will not show up on iOS or OS X as regular Wireless Access Points. This means that while in WAC mode, these accessories will not be able to offer Wireless Access Point-based features to iOS and OS X devices. These features may include:

- Direct AirPlay streaming.
- Web Interface-based features such as Firmware Upgrade.

If the accessory wishes to offer support to iOS and OS X for these features while not connected to an infrastructure network, it should provide a mechanism for exiting WAC mode and entering a Wireless Access Point mode.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

55.8 Wi-Fi Certification requirements

Wi-Fi Accessory Configuration accessories must complete the following certification requirements:

- Wi-Fi Alliance "Wi-Fi certified" program
- MFi program certification

NOTE: Pre-existing MFi program certification on an accessory is not sufficient for Wi-Fi Accessory Configuration accessory certification.

55.9 Wi-Fi Accessory Configuration Setup Experience

The configuration process allows a configuring device, such as an iPhone, to send configuration information and network credentials to the accessory. This may include joining a Wi-Fi network, specifying a friendly name for the accessory, etc. The general flow of operation is:

1. Configuring device discovers accessories

   Wi-Fi scans are conducted to find unconfigured accessories broadcasting the Apple information element (IE) in the accessory's software access point Wi-Fi beacon frames.

2. Configuring device joins accessory's temporary software access point network
3. Configuring device searches for accessory via Bonjour

   Configuring device browses for `_mfi-config._tcp` and matches the accessory by its Device ID. The Device ID in the Bonjour TXT record is the same one advertised as part of the Apple Device IE.

4. Configuring device resolves via Bonjour and connects via TCP to accessory

   This performs normal Bonjour PTR->SRV->A/AAAA resolving then connects via TCP.

5. Configuring device authenticates accessory

   For MFi-certified accessories, this performs MFi-SAP using the Apple Authentication Coprocessor. See Apple Authentication Coprocessor 2.0C (page 66).

   Note: After this step, all subsequent control requests and responses are encrypted.

6. Configuring device builds config TLVs, encrypts it, and sends it in an HTTP request to accessory at `/config`

   The TLV contains the information needed by the accessory to configure itself. See Table 55-1 (page 741).

7. Accessory receives, validates, and saves config request

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The accessory must not immediately apply the new configuration. Applying it at this point may interrupt its connection to the controller. The accessory must send its response, disable the sending side of its socket to signal it is done sending, then wait until it receives a FIN from the Configuring device (i.e. socket receive returns 0).

8. Configuring device receives accessory response and waits for the accessory to close its connection

   The controller must disable the sending side of its socket to signal it has received the response then waits until it receives a FIN from the accessory (i.e. socket receive returns 0).

9. Accessory waits for controller to disconnect then deregisters its Bonjour service and increments its config seed ID

10. Accessory applies configuration and joins new Wi-Fi network

    The accessory must wait until it has fully joined the new Wi-Fi network before continuing to the next step.

11. Accessory re-registers with Bonjour with the new config seed

    It's important to only advertise the new configuration seed after the configuration has completed and the accessory has joined the new network. Otherwise, the controller may find a stale instance of the service and prematurely assume success.

12. Configuring device re-joins its original Wi-Fi network

13. Configuring device searches for accessory via Bonjour

    The controller browses for `_mfi-config._tcp` and matches the accessory by its Device ID (same one saved off from the earlier step).

14. Configuring device sends an HTTP request to `/configured` on the accessory to indicate configuration is complete

15. Configuring device reports success (or failure) to user

16. Configuring device prompts user to discover an application for the configured accessory if available

Table 55-1 Configuration TLVs

Name	ID	Type	Description
bundleSeedID	0x01	String	Unique 10 character string assigned by Apple to an app via the Provisioning Portal (e.g. 24D4XFAF43).
firmwareRevision	0x02	String	Firmware revision of the accessory.
hardwareRevision	0x03	String	Hardware revision of the accessory.
language	0x04	String	BCP-47 language to configure the accessory for. See http://www.iana.org/assignments/language-subtag-registry.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	Description
manufacturer	0x05	String	Manufacturer of the accessory (e.g. Apple).
mfiProtocol	0x06	String	Reverse-DNS string describing supported MFi accessory protocols (e.g. com.acme.gadget) for accompanying applications. Note: there may be more than one of this item if multiple protocols are supported.
model	0x07	String	Model name of the accessory (e.g. Accessory1,1).
name	0x08	String	Name that accessory should use to advertise itself to the user.
playPassword	0x09	String	Password used to start an AirPlay stream to the accessory.
serialNumber	0x0A	String	Serial number of the accessory.
wifiPSK	0x0B	Data	Wi-Fi PSK for joining a WPA-protected Wi-Fi network. If it's between 8 and 63 bytes each being 32-126 decimal, inclusive then it's a pre-hashed password. Otherwise, it's expected to be a pre-hashed, 256-bit pre-shared key.
wifiSSID	0x0C	String	Wi-Fi SSID (network name) for the accessory to join. This should be UTF-8.

55.10 Bonjour

The Bonjour service type for Wi-Fi Accessory Configuration is `_mfi-config._tcp`. The name of the Bonjour service is the user-visible name of the accessory (e.g. "Basement Thermostat"). The name may contain any Unicode character and is encoded using UTF-8. It has a maximum length of 63 bytes (which may be fewer than 63 characters as a single Unicode character may require multiple bytes). Additional data needed for discovery-time metadata is advertised via a TXT record. This contains fields for feature detection, versions, etc.

Table 55-2 mfi config tcp TXT record keys

Key	Description
deviceid	Globally unique ID for the accessory (e.g. the primary MAC address, such as 00:11:22:33:44:55).
features	Feature flag bits (e.g. 0x3 for bits 0 and 1). See Table 55-3 (page 743).
flags	Status flags (e.g. 0x04 for bit 3). See Table 55-4 (page 743).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Key	Description
protovers	Protocol version string <major>.<minor> (e.g. 1.0). Missing means 1.0.
seed	Configuration seed number. This is 0-255 and updates each time the software configuration changes.
srcvers	Source version number. Populated directly by the source code. Valid version numbers are 1.14, 1.20 and 1.22.

Table 55-3 MFi Configuration Feature Flags

Value	Bit	Description
0x00000001	0	App associated with this accessory.
0x00000004	2	Accessory supports TLV-based configuration.

Table 55-4 MFi Configuration Status Flag

Value	Bit	Description
0x01	0	Problem has been detected.
0x02	1	Accessory is not configured.

55.11 Apple Device Information Element (IE)

55.11.1 General Usage

This IE should be included in the following 802.11 management frames:

- Probe response frames.
- Beacon frames, if applicable.

55.11.2 Structure

This section defines a vendor-specific 802.11 IE using the OUI 00-A0-40 (registered to Apple Inc.). The payload portion of the IE is composed of sub-IEs defined by this document.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 55-5 Apple Device IE overall structure

Name	Size	Value	Description
Element ID	1	0xDD	Vendor-specific element ID as specified in *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11 - 2007* .
Length	1	Variable	Number of bytes in IE (excludes element ID and length bytes).
OUI	3	0x00 0xA0 0x40	Apple Inc. OUI reserved for this IE.
Sub-type	1	0x00	Sub-type of the 00-A0-40 Apple Inc. OUI.
Elements:	Variable	Variable	Sub IE elements defined by this spec.

Table 55-6 Apple Device IE element structure

Name	Size	Description
Element ID	1	Vendor specific element ID as specified in *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11 - 2007* .
Length	1	Number of bytes in the element payload (excludes element ID and length bytes).
Payload	Variable	Payload defined by the element ID.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

55.11.3 Payload

Table 55-7 Apple Device IE elements

Element ID	Name	Format	Description
0x00	Flags	n:bits	Flags about the device: b0-b7, b8-b15, etc. See Table 55-8 (page 746). Each flag is a bit. Bit numbering starts from the leftmost bit of the first byte and uses the minimum number of bytes needed to encode the bits. For example: If only bit 1 is set, it would be 0x40. If bit 1 (0x40) and bit 7 (0x01) are set, it would be 0x41. If bit 1 (0x40), bit 7 (0x01), and bit 10 (0x0020) are set, it would be 0x41, 0x20. If only bit 10 (0x0020), it would be 0x00, 0x20.
0x01	Name	UTF-8	Friendly name of the device. This should only be provided if the user configured a custom name or the firmware of the device has reason to believe it can provide a name that's better than the default name the client software will provide for it based on the model. Due to localization issues, it often better to only provide this element if the user has configured a name.
0x02	Manufacturer	UTF-8	Machine-parsable manufacturer of the device (e.g. "Apple").
0x03	Model	UTF-8	Machine-parsable model of the device (e.g. "Device1,1").
0x04	OUI	3 bytes	OUI of the device including this IE.
0x05	dWDS	2 bytes	<1:DWDS Role><1:DWDS Flags>.
0x06	Bluetooth MAC	6 bytes	MAC address of the Bluetooth radio, if applicable.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Element ID	Name	Format	Description
0x07	Device ID	6 bytes	Globally unique ID of the device. This should be the primary MAC address of the device. If the device has multiple MAC addresses, one must be chosen as the primary MAC address such that it never changes (e.g. doesn't depend on the network interface currently active). The main purpose of this element is to allow devices to discover the device via Wi-Fi scans and then later associate it with an IP-based discovery method, such as Bonjour (where the Device ID is expected to be reported via the TXT record).
0x08-0xFF	TBD	TBD	Reserved.
0xDD	Vendor-specific	n bytes	Same format as a normal vendor-specific IE element.

Table 55-8 Flags

Value	Bit	Description
0x80	0	Supports AirPlay.
0x40	1	Device is unconfigured.
0x20	2	Supports MFi Configuration V1.
0x10	3	Supports Wake on Wireless (WoW).
0x08	4	Device has interference robustness enabled.
0x04	5	Device detected remote PPPoE server.
0x02	6	Supports WPS.
0x01	7	WPS is active on the device.
0x0080	8	Supports AirPrint.
0x0040	9	Reserved.
0x0020	10	Supports CarPlay over Wireless, see CarPlay over Wireless (page 395).
0x0010	11	Provides Internet access (e.g. 3G/4G).
0x0008	12	Reserved.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Bit	Description
0x0004	13	Reserved.
0x0002	14	Supports 2.4 GHz Wi-Fi networks.
0x0001	15	Supports 5 GHz Wi-Fi networks.
0x000080	16	Reserved.
0x000040	17	Supports HomeKit Accessory Protocol

55.12 Accessory Compliance Test Plan

The purpose of this test plan is to specify the testing that vendors must perform to verify that their products conform to the Wi-Fi Accessory Configuration specification provided by Apple Inc. Wi-Fi Accessory Association Verification Tests (page 749) describes tests that must be run and successfully completed for all Wi-Fi Accessory Configuration enabled products.

Table 55-9 Terms And Definitions

Term	Definition
IE	Information Element.
AP	Access Point.
AU	AirPort Utility.
STA	Non-AP 802.11 station.
BSSID	Basic Service Set Identifier.
SSID	Service Set Identifier.
MAC	Media Access Control.
BSS	Basic Service Set.
ESS	Extended Service Set.
DS	Distribution System.
DUT	Device Under Test.
URL	Uniform Resource Locator.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Term	Definition
PHY	Physical Layer (802.11a, b, g, n).
OOB	Out of the Box.

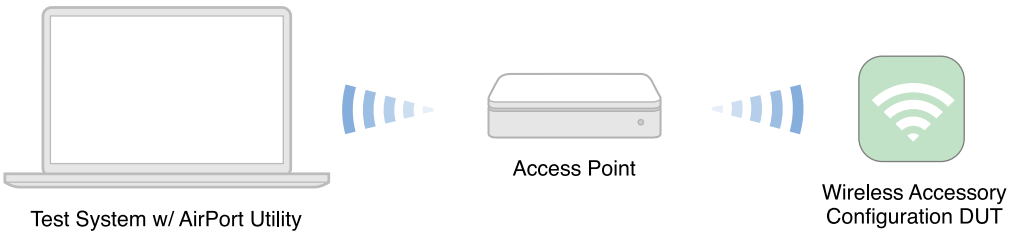
Wi-Fi Accessory Configuration enabled DUTs will be tested on any and all wireless network interfaces that allow for Wi-Fi Accessory Configuration.

The following are minimum requirements to complete the test suites that are included in this test plan.

- MacBook Pro running Mac OS X 10.9 or greater
- One of the following:
  - Apple AirPort Extreme 802.11n (5th Generation) Base Station running 7.6.1 or greater
  - Apple AirPort Express 802.11n (3rd Generation) Base Station running 7.6.2 or greater
- Apple AirPort Utility 6.3.1 or greater

The following test beds are designed and specified for each test case in this test plan. The minimum requirement for every test bed is an OS X computer running the latest version of AirPort Utility (Test System) and a simultaneous dual band Apple Base Station running the latest firmware (Access Point).

Figure 55-1 Test bed



The following test cases must be completed successfully to ensure compliance with the Wi-Fi Accessory Configuration specification and interoperability with Apple Base Station product lines. While it is possible to implement Wi-Fi Accessory Configuration without an Apple Base Station, all tests will be done with an Apple Base Station (AirPort Extreme or Time Capsule) product because certain features of the Base Station are required to complete all of the testing.

Network connectivity is a requirement for Wi-Fi Accessory Configuration. Each DUT must, at minimum, support 802.11g. All supported PHYs must be tested.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

55.12.1 Wi-Fi Accessory Association Verification Tests

Wi-Fi Accessory Configuration enabled DUTs have two options of how to be configured based on whether there is an interface that allows for the SSID and security modes to be configured. Only run the test that applies to your DUT. The following tests assume the accessory has had no configuration done to the DUT and is still in its factory default state.

55.12.1.1 Wi-Fi Accessory Configuration Mode Automatic Shutoff

For the Test Environment, please refer to Figure 55-1 (page 748)

1. Power on the DUT.
2. Wait for DUT to complete booting.
3. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
4. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
5. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.
6. Do not interact with the DUT for at least 30 minutes.
7. The DUT should come out of WAC mode and stop beaconing the Apple Device IE indicating it is unconfigured.
8. The product mode indicator must show that the accessory is not in WAC mode or else FAIL.
9. Power down the DUT.
10. Power on the DUT.
11. Wait for DUT to complete booting.
12. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
13. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
14. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.

55.12.1.2 802.11b/g Association Verification

For the Test Environment, please refer to Figure 55-1 (page 748).

If the DUT supports 802.11b/g the following test must be run.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

1. Power on the Base Station and connect to it via the Apple AirPort Utility. In the AirPort Utility select "Manual Setup".
2. On the bar across the top of the AirPort Utility, select "Wireless"
3. Set Base Station to Network Mode "Create a wireless network".
4. Under the Section that says "Wireless Network Name:" change it to "Soundwave24€".
5. Under the section that says "Wireless Security:" change it to "None"
6. Click "Wireless Options".
7. Check the box that says "5 GHz Network Name" and the box should be come editable. Change the name to "Soundwave5".
8. Then under the "Radio Mode:" pull down menu select, 802.11a - 802.11b/g.
9. Then click save on the AirPort Utility. The AirPort Utility may prompt for various errors and all of them can be ignored. Be sure to click "Update" and wait for the Base Station to reboot.
10. Power on the DUT. Wait for DUT to complete booting. Note the MAC address of the DUT.
11. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
12. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
13. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.
14. Select the DUT in the AirPort Utility.
15. Set the DUT's Wi-Fi Network to "Soundwave24€".
16. Give the DUT the Accessory Name "WAC DUT".
17. Click next in the AirPort Utility.
18. Verify that a setup complete message is displayed.
19. If any error messages are displayed, FAIL.
20. Wait for DUT to complete booting and indicate that it has joined a network.
21. If DUT does not join the network, FAIL.
22. Using the AirPort Utility verify the DUT has joined the network.
23. In the AirPort Utility, select the Base Station. Verify that the DUT is present as a wireless client with the Accessory Name "WAC DUT".
24. If the DUT is not present or if the MAC address is not present, FAIL.
25. Manually put the DUT into Wi-Fi Accessory Configuration mode according to the accessory's instructions.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

26. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
27. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
28. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.

55.12.1.3 802.11n (5 GHz) Association Verification

For the Test Environment, please refer to Figure 55-1 (page 748).

If the DUT supports 802.11n (5 GHz) the following test must be run.

1. Power on the Base Station and connect to it via the Apple AirPort Utility. In the AirPort Utility select "Manual Setup".
2. On the bar across the top of the AirPort Utility, select "Wireless"
3. Set Base Station to Network Mode "Create a wireless network".
4. Under the Section that says "Wireless Network Name:" change it to "Soundwave24€".
5. Under the section that says "Wireless Security:" change it to "None"
6. Click "Wireless Options".
7. Check the box that says "5 GHz Network Name" and the box should be come editable. Change the name to "Soundwave5".
8. Then under the "Radio Mode:" pull down menu select, 802.11a/n - 802.11b/g/n.
9. Then click save on the AirPort Utility. The AirPort Utility may prompt for various errors and all of them can be ignored. Be sure to click "Update" and wait for the Base Station to reboot.
10. Power on the DUT. Wait for DUT to complete booting. Note the MAC address of the DUT.
11. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
12. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
13. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.
14. Select the DUT in the AirPort Utility.
15. Set the DUT's Wi-Fi Network to "Soundwave5".
16. Give the DUT the Accessory Name "WAC DUT".
17. Click next in the AirPort Utility.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

18. Verify that a setup complete message is displayed.
19. If any error messages are displayed, FAIL.
20. Wait for DUT to complete booting and indicate that it has joined a network.
21. If DUT does not join the network, FAIL.
22. Using the AirPort Utility verify the DUT has joined the network.
23. In the AirPort Utility, select the Base Station. Verify that the DUT is present as a wireless client with the Accessory Name "WAC DUT".
24. If the DUT is not present or if the MAC address is not present, FAIL.

55.12.1.4 802.11n (2.4 GHz) Association Verification

For the Test Environment, please refer to Figure 55-1 (page 748).

If the DUT supports 802.11 (2.4 GHz) the following test must be run.

1. Power on the Base Station and connect to it via the Apple AirPort Utility. In the AirPort Utility select "Manual Setup".
2. On the bar across the top of the AirPort Utility, select "Wireless"
3. Set Base Station to Network Mode "Create a wireless network".
4. Under the Section that says "Wireless Network Name:" change it to "Soundwave24€".
5. Under the section that says "Wireless Security:" change it to "None"
6. Click "Wireless Options".
7. Check the box that says "5 GHz Network Name" and the box should be come editable. Change the name to "Soundwave5".
8. Then under the "Radio Mode:" pull down menu select, 802.11a/n - 802.11b/g/n.
9. Then click save on the AirPort Utility. The AirPort Utility may prompt for various errors and all of them can be ignored. Be sure to click "Update" and wait for the Base Station to reboot.
10. Power on the DUT. Wait for DUT to complete booting. Note the MAC address of the DUT.
11. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
12. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
13. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.
14. Select the DUT in the AirPort Utility.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

15. Set the DUT's Wi-Fi Network to "Soundwave24€".
16. Give the DUT the Accessory Name "WAC DUT".
17. Click next in the AirPort Utility.
18. Verify that a setup complete message is displayed.
19. If any error messages are displayed, FAIL.
20. Wait for DUT to complete booting and indicate that it has joined a network.
21. If DUT does not join the network, FAIL.
22. Using the AirPort Utility verify the DUT has joined the network.
23. In the AirPort Utility, select the Base Station. Verify that the DUT is present as a wireless client with the Accessory Name "WAC DUT".
24. If the DUT is not present or if the MAC address is not present, FAIL.

55.12.1.5 802.11 non-broadcast SSID Association Verification

For the Test Environment, please refer to Figure 55-1 (page 748).

1. Power on the Base Station and connect to it via the Apple AirPort Utility. In the AirPort Utility select "Manual Setup".
2. On the bar across the top of the AirPort Utility, select "Wireless"
3. Set Base Station to Network Mode "Create a wireless network".
4. Under the Section that says "Wireless Network Name:" change it to "Soundwave24€".
5. Under the section that says "Wireless Security:" change it to "None"
6. Click "Wireless Options".
7. Check the box that says "5 GHz Network Name" and the box should be come editable. Change the name to "Soundwave5".
8. Check the box "Create hidden network".
9. Under the "Radio Mode:" pull down menu select, 802.11a - 802.11b/g.
10. Then click save on the AirPort Utility. The AirPort Utility may prompt for various errors and all of them can be ignored. Be sure to click "Update" and wait for the Base Station to reboot.
11. Join the hidden network "Soundwave24€" from the Test System.
12. Power on the DUT. Wait for DUT to complete booting. Note the MAC address of the DUT.
13. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

14. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
15. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.
16. Select the DUT in the AirPort Utility.
17. Set the DUT's Wi-Fi Network to "Soundwave24€".
18. Give the DUT the Accessory Name "WAC DUT".
19. Click next in the AirPort Utility.
20. Verify that a setup complete message is displayed.
21. If any error messages are displayed, FAIL.
22. Wait for DUT to complete booting and indicate that it has joined a network.
23. If DUT does not join the network, FAIL.
24. Using the AirPort Utility verify the DUT has joined the network.
25. In the AirPort Utility, select the Base Station. Verify that the DUT is present as a wireless client with the Accessory Name "WAC DUT".
26. If the DUT is not present or if the MAC address is not present, FAIL.

55.12.2 2.4 GHz vs 5 GHz Beaconing Tests

For the Test Environment, please refer to Figure 55-1 (page 748).

If the DUT supports 802.11 2.4 GHz but not 5 GHz the following test must be run.

1. Power on the Base Station and connect to it via the Apple AirPort Utility. In the AirPort Utility select "Manual Setup".
2. On the bar across the top of the AirPort Utility, select "Wireless"
3. Set Base Station to Network Mode "Create a wireless network".
4. Under the Section that says "Wireless Network Name:" change it to "Soundwave24€".
5. Under the section that says "Wireless Security:" change it to "None"
6. Click "Wireless Options".
7. Check the box that says "5 GHz Network Name" and the box should be come editable. Change the name to "Soundwave5".
8. Then under the "Radio Mode:" pull down menu select, 802.11a - 802.11b/g.
9. Then click save on the AirPort Utility. The AirPort Utility may prompt for various errors and all of them can be ignored. Be sure to click "Update" and wait for the Base Station to reboot.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

10. Power on the DUT. Wait for DUT to complete booting. Note the MAC address of the DUT.
11. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
12. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
13. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.
14. Select the DUT in the AirPort Utility.
15. Open the list to set the DUT's Wi-Fi Network.
16. Verify that "Soundwave24€" is displayed in the list.
17. If not displayed, FAIL.
18. Verify that "Soundwave5" is NOT displayed in the list.
19. If displayed, FAIL.

55.12.3 Security Mode Verification Tests for WPA2 Personal

All Wi-Fi Accessory Configuration enabled DUTs must support no security (None) and WPA2 Personal at minimum. Other security modes are supported by the Apple Base Station and other 3rd party AP products however only those two will be tested.

For the Test Environment, please refer to Figure 55-1 (page 748).

1. Power on the Base Station and connect to it via the Apple AirPort Utility. In the AirPort Utility select "Manual Setup".
2. On the bar across the top of the AirPort Utility, select "Wireless"
3. Set Base Station to Network Mode "Create a wireless network".
4. Under the Section that says "Wireless Network Name:" change it to "Soundwave24€".
5. Under the section that says "Wireless Security:" change it to "WPA/WPA2 Personal"
6. Click "Wireless Options".
7. Check the box that says "5 GHz Network Name" and the box should be come editable. Change the name to "Soundwave5".
8. Then under the "Radio Mode:" pull down menu select, 802.11a/n - 802.11b/g/n.
9. Then click save on the AirPort Utility. The AirPort Utility may prompt for various errors and all of them can be ignored. Be sure to click "Update" and wait for the Base Station to reboot.
10. Power on the DUT. Wait for DUT to complete booting. Note the MAC address of the DUT.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

11. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
12. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
13. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.
14. Select the DUT in the AirPort Utility.
15. Set the DUT's Wi-Fi Network to "Soundwave24€".
16. Give the DUT the Accessory Name "WAC DUT".
17. Click next in the AirPort Utility.
18. Verify that a setup complete message is displayed.
19. If any error messages are displayed, FAIL.
20. Wait for DUT to complete booting and indicate that it has joined a network.
21. If DUT does not join the network, FAIL.
22. Using the AirPort Utility verify the DUT has joined the network.
23. In the AirPort Utility, select the Base Station. Verify that the DUT is present as a wireless client with the Accessory Name "WAC DUT".
24. If the DUT is not present or if the MAC address is not present, FAIL.

55.12.4 IP Connectivity Tests

IP connectivity is required for all Wi-Fi Accessory Configuration enabled accessories, with a minimum of IPv4 and IPv6 with a DHCP client to be implemented. Both stacks will need to be verified. It is additionally required that Link Local addressing and configuration is allowed on both IPv4 and IPv6 stacks.

55.12.4.1 IPv4 DHCP

For the Test Environment, please refer to Figure 55-1 (page 748).

1. Power on the Base Station and connect to it via the Apple AirPort Utility. In the AirPort Utility select "Manual Setup".
2. On the bar across the top of the AirPort Utility, select "Wireless"
3. Set Base Station to Network Mode "Create a wireless network".
4. Under the Section that says "Wireless Network Name:" change it to "Soundwave24€".
5. Under the section that says "Wireless Security:" change it to "None"

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

6. Click "Wireless Options".
7. Check the box that says "5 GHz Network Name" and the box should be come editable. Change the name to "Soundwave5".
8. Then under the "Radio Mode:" pull down menu select, 802.11a/n - 802.11b/g/n.
9. Then click save on the AirPort Utility. The AirPort Utility may prompt for various errors and all of them can be ignored. Be sure to click "Update" and wait for the Base Station to reboot.
10. Power on the DUT. Wait for DUT to complete booting. Note the MAC address of the DUT.
11. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
12. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
13. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.
14. Select the DUT in the AirPort Utility.
15. Set the DUT's Wi-Fi Network to "Soundwave24€".
16. Give the DUT the Accessory Name "WAC DUT".
17. Click next in the AirPort Utility.
18. Verify that a setup complete message is displayed.
19. If any error messages are displayed, FAIL.
20. Wait for DUT to complete booting and indicate that it has joined a network.
21. If DUT does not join the network, FAIL.
22. Using the AirPort Utility verify the DUT has joined the network.
23. In the AirPort Utility, select the Base Station. Verify that the DUT is present as a wireless client with the Accessory Name "WAC DUT". Take note of the DUT's IP Address.
24. If the DUT is not present or if the MAC address is not present, FAIL.
25. On the Test Machine connected to the Base Station ping the IP address that was just noted.
26. If all the pings are successful, then PASS, else FAIL.
27. Reboot the DUT and wait for it to come back up.
28. Once connected to the network, open the AirPort, select the Base Station. Verify that the DUT is present as a wireless client with the Accessory Name "WAC DUT". Take note of the DUT's IP Address.
29. If the DUT is not present or if the MAC address is not present, FAIL.
30. On the Test Machine connected to the Base Station ping the IP address that was just noted.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

31. If all the pings are successful, then PASS, else FAIL.

55.12.4.2 IPv4 Link Local

For the Test Environment, please refer to Figure 55-1 (page 748).

1. Power on the Base Station and connect to it via the Apple AirPort Utility. In the AirPort Utility select "Manual Setup".
2. On the bar across the top of the AirPort Utility, select "Network".
3. Under the section that says "Router Mode:" change it to "Off (Bridge Mode)" and ensure that there is no Wide Area Network attached to the Base Station.
4. On the bar across the top of the AirPort Utility, select "Wireless"
5. Set Base Station to Network Mode "Create a wireless network".
6. Under the Section that says "Wireless Network Name:" change it to "Soundwave24€".
7. Under the section that says "Wireless Security:" change it to "None"
8. Click "Wireless Options".
9. Check the box that says "5 GHz Network Name" and the box should be come editable. Change the name to "Soundwave5".
10. Then under the "Radio Mode:" pull down menu select, 802.11a/n - 802.11b/g/n.
11. Then click save on the AirPort Utility. The AirPort Utility may prompt for various errors and all of them can be ignored. Be sure to click "Update" and wait for the Base Station to reboot.
12. Power on the DUT. Wait for DUT to complete booting. Note the MAC address of the DUT.
13. The DUT should begin beaconing 802.11 beacons including the Apple Device IE indicating it is unconfigured and advertising as an 802.11 network with an SSID that is descriptive of the DUT. At minimum the indication of the manufacturer should be present, with no security enabled.
14. The product mode indicator must show that the accessory is in WAC mode or else FAIL.
15. The DUT must show up in the AirPort Utility under "Other Wi-Fi Devices" as a "New Wi-Fi Device" indicating it is an unconfigured accessory or else FAIL.
16. Select the DUT in the AirPort Utility.
17. Set the DUT's Wi-Fi Network to "Soundwave24€".
18. Give the DUT the Accessory Name "WAC DUT".
19. Click next in the AirPort Utility.
20. Verify that a setup complete message is displayed.
21. If any error messages are displayed, FAIL.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

22. Wait for DUT to complete booting and indicate that it has joined a network.
23. If DUT does not join the network, FAIL.
24. Using the AirPort Utility verify the DUT has joined the network.
25. In the AirPort Utility, select the Base Station. Verify that the DUT is present as a wireless client with the Accessory Name "WAC DUT". Take note of the DUT's MAC Address and IP Address.
26. If the DUT is not present or if the MAC address is not present, FAIL.
27. On the Test Machine connected to the Base Station ping the IP address that was just noted using ping6.
28. If all the pings are successful, then PASS, else FAIL.
29. Reboot the DUT and wait for it to come back up.
30. Once connected to the network, open the AirPort, select the Base Station. Verify that the DUT is present as a wireless client with the Accessory Name "WAC DUT". Take note of the DUT's MAC Address and IP Address.
31. If the DUT is not present or if the MAC address is not present, FAIL.
32. From a Terminal window of the Test System connected to the Base Station enter:
    - `arp –a –i $INTERFACE`
    - `where $INTERFACE is the identifier for your AirPort card`
33. If all the MAC address associated with the DUT is present and the IP address displayed is of the format 169.254.XX.XX, then PASS, else FAIL.

55.12.5 Bonjour TXT Records Tests for ADD and RMV

If the DUT is not based on the Microchip Binary platform, the following test must be run. This test checks bonjour fields that contain information related to the POSIX Source release version numbers. The valid POSIX Source releases are:

- 1.14
- 1.20
- 1.22

For the Test Environment, please refer to Figure 55-1 (page 748).

1. Power on the DUT.
2. Wait for DUT to complete booting.
3. On the Test System, issue the following command in a Terminal and leave it running.

   ```
   dns–sd –B _mfi–config
   ```

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

4. From a second Terminal window of the Test System enter:

   ```
   networksetup –setairportnetwork $INTERFACE $SSID
   ```

5. Where $INTERFACE is the identifier for your AirPort card and $SSID is the DUT Network name.
6. A Bonjour record should appear with the following format:

   ```
   Timestamp, A/R, Flags, if, Domain, Service Type, Instance Name
   ```

7. The A/R should have "Add" below it, if this is true, PASS.
8. The Service Type should have `_mfi–config._tcp` below it, if this is true, PASS.
9. The Instance should have "$INSTANCENAME- below it, which must be the Friendly Name of the DUT. If this is true, PASS.
10. From the Terminal of the Test System enter:
    - `dns–sd –L $INSTANCENAME _mfi–config local`
    - `where $INSTANCENAME is collected earlier.`
11. The Bonjour record needs to contain a minimum of the following TXT records (a value is specified that is the requirement for field):
    - deviceid=<MAC address>
    - features=<Feature flag bits>
    - seed=<Configuration seed number>
    - srcvers=<Source release version> See Bonjour TXT Records Tests for ADD and RMV (page 759) for the list of valid POSIX Source release versions. (Populated by WAC source code.)
12. If all of the fields are present then PASS, else FAIL.

55.12.6 Certification Procedure

- Accessory submission must be accompanied by proof of Wi-Fi Certification for the product.
- Full product documentation must be provided in both printed and soft copy form (all customer facing documents such as user manuals and quick start guides).
- The submission must be accompanied by a completed "Wi-Fi Accessory Configuration Product Compliance Questionnaire R1".

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

56. Wi-Fi Information Sharing

Wi-Fi configuration information can be exchanged between Apple devices and accessories through an iAP2 connection.

Apple devices can share Wi-Fi configuration information with an accessory. The accessory can initiate this process, but the user must grant permission for the Apple device to share this information. The Apple device can only share information about the currently connected Wi-Fi network, and this feature will not account for other router-configured access control mechanisms such as RADIUS or MAC address filtering.

Figure 56-1 Wi-Fi Information Sharing Alert



CarPlay accessories (see CarPlay (page 381)) can also share Wi-Fi configuration information with an Apple device. This feature enables a user to quickly join a Wi-Fi network by connecting or pairing the Apple device to an accessory over a wired or Bluetooth transport. The accessory can then notify the Apple device of the Wi-Fi configuration information of an available Wi-Fi network. The accessory can initiate this process, but the user must grant permission for the Apple device to join the new Wi-Fi network.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

56.1 Wi-Fi Information Sharing Requirements

56.1.1 Apple Device to Accessory

All accessories that support the Wi-Fi Information Sharing (Apple device to accessory) feature via iAP2 must send or receive the following iAP2 control session message(s):

RequestWiFiInformation (page 876)

WiFiInformation (page 876)

An accessory designed to use this feature must comply with the following requirements:

- It must let the user initiate Wi-Fi information sharing via either a physical button or an onscreen option.
- It must not be able to initiate Wi-Fi information sharing without an explicit user action via the means described above.
- It must notify the user, visibly and/or audibly, when it has received Wi-Fi connection information.
- It must notify the user, visibly and/or audibly, when it has successfully established a Wi-Fi connection.

56.1.2 Accessory to Apple Device

All accessories that support the Wi-Fi Information Sharing (CarPlay accessory to Apple device) feature via iAP2 must send or receive the following iAP2 control session message(s):

RequestAccessoryWiFiConfigurationInformation (page 877)

AccessoryWiFiConfigurationInformation (page 877)

Apple devices will only ask CarPlay accessories for Wi-Fi information.

56.2 Wi-Fi Information Sharing Usage

56.2.1 Apple Device to Accessory

To request the Wi-Fi network information, the accessory must send the RequestWiFiInformation (page 876) message. A request must only be sent in response to direct user action such as pressing a button or selecting a menu item.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The device will send WiFiInformation (page 876) when the user has responded to the request. The Wi-Fi network SSID and passphrase are part of the WiFiInformation (page 876), and the network security information must be determined by examining the Access Point's beacon frames per the 802.11 specification.

56.2.2 Accessory to Apple Device

To provide Wi-Fi network information, the accessory must support the RequestAccessoryWiFiConfigurationInformation (page 877) and AccessoryWiFiConfigurationInformation (page 877) messages. The Apple device will send RequestAccessoryWiFiConfigurationInformation (page 877) to request the Wi-Fi configuration from the accessory. The accessory must respond with a AccessoryWiFiConfigurationInformation (page 877) message.

56.3 Test Procedures

56.3.1 iAP2 Tests

56.3.1.1 Apple Device to Accessory

1. Verify that the following iAP2 control session message(s) are sent or received:
   - RequestWiFiInformation (page 876)
   - WiFiInformation (page 876)
2. Verify that the accessory supporting this feature complies with the following:
   - It must let the user initiate Wi-Fi login sharing via either a physical button or an onscreen option.
   - It must not be able to initiate Wi-Fi login sharing without an explicit user action via the means described above.
   - It must notify the user, visibly and/or audibly, when it has received Wi-Fi connection information.
   - It must notify the user, visibly and/or audibly, when it has successfully established a Wi-Fi connection.

56.3.1.2 Accessory to Apple Device

1. Verify that the following iAP2 control session message(s) are sent or received:
   - RequestAccessoryWiFiConfigurationInformation (page 877)
   - AccessoryWiFiConfigurationInformation (page 877)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

57. iAP2 Link

Every iAP2 connection starts with the establishment of a link between an accessory and a device over a supported transport. The link protocol provides a transport-agnostic mechanism for reliable and ordered delivery of packetized data belonging to one or more iAP2 sessions. The protocol is also configurable on a per-connection basis and can be optimized for best performance on any particular transport and accessory usage profile. Certain protocol features contribute towards these goals:

- Positive acknowledgement of received packets.
- Retransmissions only require the unacknowledged packets in a sequence to be resent.
- Explicit and efficient support for iAP2 sessions.

iAP2 is supported over the following transports:

- Bluetooth (see Bluetooth (page 344))
- Serial (see Serial (page 680))
- USB Device Mode (see USB Device Mode (page 706))
- USB Host Mode (see USB Host Mode (page 712))

All devices support one or more transports that can be used to establish an iAP2 connection with an accessory. Some transports are better suited for various features than others, so transport selection should be a high priority early in the accessory design process.

Sample source code for an iAP2 link layer implementation may be found on the MFi Portal.

57.1 Packet Structure

Every link packet must start with a fixed-size 9-byte header, including checksum, and is followed by an optional variable-length data payload. Both the header and payload have their own checksums. If there is no payload data, there is no payload checksum. An example can be found in iAP2 Link Packet Structure Example (page 785).

Table 57-1 iAP2 Link Packet Structure

Start of Packet MSB (`0xFF`)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Start of Packet LSB (0x5A)
Packet Length MSB
Packet Length LSB
Control Byte
Packet Sequence Number
Packet Acknowledgement Number
Session Identifier
Header Checksum
...
Payload Data
...
Payload Checksum

57.1.1 Start of Packet

The first two bytes of every iAP2 link packet are always 0xFF 0x5A. If these bytes are detected in the transport stream then both accessory and device must attempt to parse the following bytes as a valid packet.

57.1.2 Packet Length

The next two bytes denote the Packet Length in bytes and are always expressed as an unsigned 16-bit big-endian integer. A link packet with no Payload Data always has a Packet Length of 9 bytes (from Start of Packet to Header Checksum). Otherwise, the Packet Length is measured from the Start of Packet to the last byte of Payload Data including the Payload Checksum. For example, a link packet with 1 byte of Payload Data has a Packet Length of 11 bytes. Packets have a maximum Payload Data size of 65525 bytes; a packet with such a payload has a Packet Length of 65535 bytes.

57.1.3 Control Byte

The bits in the Control Byte indicate what is present in the packet.

- The SYN, EAK, and RST bits are mutually exclusive.
- The ACK bit may be combined with the SYN bit.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- The EAK bit is always set along with the ACK bit.
- The RST bit is never set in conjunction with any other bit.
- The SLP bit is never set in conjunction with any other bit.
- All unassigned bits must be set to 0.

iAP2 Session Payloads cannot be present when any of the SYN, EAK, or RST bits are set. Conversely, link packets with iAP2 session payloads always have the ACK bit set.

Table 57-2 iAP2 Link Control Byte Bits

Bit	Name	Meaning
7	SYN	Link Synchronization Payload is present
6	ACK	Packet Acknowledgement Number is valid, and iAP2 Session Payload may be present
5	EAK	Extended Acknowledgement Payload is present
4	RST	Link Reset
3	SLP	Device Sleep

In the rest of this specification, a link packet can be described in terms of its Control Byte bits and payload. For example, a SYN packet refers to a link packet with the SYN bit set in the Control Byte, and a SYN+ACK packet refers to a link packet with both the SYN and ACK bits set in the Control Byte.

57.1.4 Packet Sequence Number

Every link packet contains a Packet Sequence Number that uniquely identifies the packet among all other packets in transit. When a link is first created, both the device and accessory must randomly pick an initial sequence number before sending their first SYN/SYN+ACK packet. Each time a packet with iAP2 Session or Link Synchronization Payload Data is sent, the sequence number is incremented by 1. Otherwise, the sequence number does not increment when a packet is sent. Retransmissions of a previously sent packet must retain the same Packet Sequence Number. The sequence number wraps back to 0 upon reaching 255.

57.1.5 Packet Acknowledgement Number

The Packet Acknowledgment Number only has meaning if the ACK bit in the Control Byte is set. If ACK is not set, the Packet Acknowledgement Number must be set to 0 by the sender, and it must be ignored by the receiver.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

If ACK is set, the Packet Acknowledgment Number indicates to a sender the Packet Sequence Number of the last in-sequence packet that was received. For example, if the Packet Sequence Numbers 1, 2, 3, and 5 were received by the accessory, the accessory's next outgoing packet's Packet Acknowledgement Number would be 3, because 5 was received out of sequence.

57.1.6 Session Identifier

The Session Identifier only has meaning if the ACK bit in the Control Byte is set and there is an iAP2 session payload present. If those two conditions are met, the Session Identifier will be a nonzero number that specifies a particular session in an iAP2 connection. Otherwise, the Session Identifier must be set to 0.

57.1.7 Header Checksum

The Header Checksum is calculated by adding together all of the following packet bytes. If the value in the Header Checksum does not match the value calculated according to Checksum Calculation (page 768), the receiver must restart packet parsing from the next detected Start of Packet sequence.

- Start of Packet MSB
- Start of Packet LSB
- Packet Length MSB
- Packet Length LSB
- Control Byte
- Packet Sequence Number
- Packet Acknowledgement Number
- Session Identifier

57.1.8 Payload Data

This section is optional, and its presence must match the state of the bits in the Control Byte. The maximum possible payload size is 65,525 bytes.

57.1.9 Payload Checksum

The Payload Checksum byte is present if and only if Payload Data is present. If Payload Data is present, all of the bytes of the Payload Data are checksummed. If the value in the Payload Checksum does not match the value calculated according to Checksum Calculation (page 768), the receiver must restart packet parsing from the next detected Start of Packet sequence.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

57.1.10 Checksum Calculation

A checksum byte, as used by iAP2, is calculated for each packet sent. The intent is for the sum of all (unsigned 8-bit) bytes being checksummed and the checksum (unsigned 8-bit) byte, ignoring any unsigned 8-bit overflow, to be equal to 0x00. This allows a fast way to verify that the packet was correctly transmitted. The checksum byte is calculated by taking the least significant byte of the sum of the (unsigned 8-bit) bytes being checksummed, and then taking the two's complement of that 8-bit value. Example code is included below:

```
uint8_t
checksum_calculation(uint8_t *buffer, uint16_t start, uint16_t length)
{
    uint16_t i;
    uint8_t  sum = 0;

    for (i = start; i < (start + length); i++) {
        sum += buffer[i];
    }
    return (uint8_t)(0x100 - sum); /* 2's complement */
}
```

57.2 Link Synchronization Payload

The Link Synchronization Payload (LSP) is used to establish a link and synchronize Packet Sequence Numbers between the device and accessory. It also contains the negotiable link parameters. An example can be found in iAP2 Link Synchronization Payload Example (page 785).

Table 57-3 Link Synchronization Payload (Version 1)

Link Version (`0x01`)
Maximum Number of Outstanding Packets
Maximum Received Packet Length MSB
Maximum Received Packet Length LSB
Retransmission Timeout MSB
Retransmission Timeout LSB

Cumulative Acknowledgement Timeout MSB
Cumulative Acknowledgement Timeout LSB
Maximum Number of Retransmissions
Maximum Cumulative Acknowledgements
iAP2 Session 1: Session Identifier
iAP2 Session 1: Session Type
iAP2 Session 1: Session Version
...
iAP2 Session N: Session Identifier
iAP2 Session N: Session Type
iAP2 Session N: Session Version

57.2.1 Link Version

- The version of the link being established. All packet payloads may vary depending on the Link Version.
- The only valid value of Link Version at this time is 1.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

57.2.2 Maximum Number of Outstanding Packets

- The maximum number of packets that may be sent without receiving an acknowledgement.
- Valid values are 1 to 127.
- This is not a negotiable parameter.
- The accessory and device may propose and use different values.
- The accessory must not send more than the device's proposed Maximum Number of Outstanding Packets without waiting for an acknowledgement from the device, and vice versa.

57.2.3 Maximum Received Packet Length

- The largest possible Packet Length in bytes.
- Valid values are 24 to 65535.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- This is not a negotiable parameter.
- The accessory and device may propose and use different values.

57.2.4 Retransmission Timeout

- The timeout value in milliseconds for retransmission of unacknowledged packets. This should be set to a value approximating the transmission time for a packet over the link transport.
- Valid values are `20` ms to `65535` ms.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

57.2.5 Cumulative Acknowledgement Timeout

- The timeout value in milliseconds after which an acknowledgment packet must be sent if another packet is not sent.
- Valid values are `10` ms to half of the Retransmission Timeout.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

57.2.6 Maximum Number of Retransmissions

- The maximum number of packet retransmissions attempted before the link is considered to be broken.
- Valid values are `1` to `30`.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

57.2.7 Maximum Cumulative Acknowledgements

- The maximum number of received acknowledgments that may be accumulated before an acknowledgement packet must be sent if another packet is not sent.
- Valid values are `0` to `127` or the Maximum Number of Outstanding Packets, whichever is smaller.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

57.2.8 iAP2 Sessions

- The iAP2 sessions that the accessory will use to communicate with the device.
- Valid session types and versions are defined in Type (page 788) chapter. Session identifiers must be unique to each defined session. 0 is not a valid session identifier.
- This is a negotiable parameter.
- Both the accessory and device must agree on the same value.

57.3 iAP2 Session Payload

The iAP2 Session Payload is specified in iAP2 Sessions (page 788). The ACK bit in the Control Byte must be set whenever an iAP2 Session Payload is present.

57.4 Extended Acknowledgement Payload

The Extended Acknowledgement Payload is used to acknowledge packets that were received out of sequence. This payload has the following attributes:

- Both the EAK and ACK bits in the Control Byte must be set.
- The Packet Acknowledge Number contains the sequence number of the last packet that was received in sequence.
- The Payload Data section contains the sequence numbers of one or more packets that were received out of sequence. Note that these are not acknowledgements for those packets. An acknowledgement will be sent separately in a later packet with the appropriate number in the ACK field.

Table 57-4 EAK Packet Payload (Link v1)

1st Out of Sequence Acknowledgement Number
...
Nth Out of Sequence Acknowledgement Number

57.5 Reset

The RST bit in the Control Byte is used by the device to reset a connection. Such a link packet does not have a payload and may only be sent by the device.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

57.6 Sleep

The SLP bit in the Control Byte is used by the device to signal that it is about to enter a sleep state. Such a link packet does not have a payload and may only be sent by the device.

The device will stop supplying Accessory Power once it enters the sleep state. If the device exits the sleep state, it will start supplying Accessory Power again.

If the accessory is capable of suspend and resume of the link layer, the accessory must wait 500 ms after Accessory Power returns before re-initializing the link. During this time, the accessory may receive an ACK from the device which signals a link layer resume and serves to synchronize the sequence number (last number sent from the device) and ACK number (the last number received by the device).

If the accessory is not capable of suspend and resume of the link layer, the accessory must wait 80 ms after Accessory Power returns before re-initializing the link.

Accessories using the Bluetooth transport will not receive a suspend packet when the Apple device enters a sleep state.

57.7 Operation

57.7.1 Record

All link implementations must store the following variables in a record that is unique to the link. These variables will be referred to in subsequent subsections describing iAP2 Link operation.

Table 57-5 iAP2 Link Operation Record Variables

Variable	Description
SentACKTimer	A timer that keeps track of the elapsed time (ms) since the last ACK packet was sent
NextSentPSN	The Packet Sequence Number of the next packet to be sent
OldestSentUnacknowledgedPSN	The Packet Sequence Number of the oldest unacknowledged packet
InitialSentPSN	The Packet Sequence Number used for the very first packet sent
LastReceivedInSequencePSN	The Packet Sequence Number of the last packet received correctly and in sequence

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Variable	Description
InitialReceivedPSN	The Packet Sequence Number of the very first packet received
ReceivedOutOfSequencePSNs[n]	An array of Packet Sequence Numbers that have been received and acknowledged out of sequence

57.7.2 Initialization

Once the transport connection is established, the accessory must confirm the presence of a device that supports iAP2 by sending the following byte sequence at 1 Hz (once a second) until a response is received from the Apple device:

```
FF 55 02 00 EE 10
```

If the device supports iAP2, the accessory will receive the same byte sequence in return. If the device is not compatible with this Specification but is compatible with the MFi Accessory Firmware Specification R46, the accessory will receive one of the following:

- An iAP1 General Lingo `iPodACK` packet with a command status of `0x04` (ERROR: Bad Parameter): `55 04 00 02 04 EE 08`
- An iAP1 General Lingo `RequestIdentify` packet: `55 02 00 00 FE`

In either case, the packet may or may not be preceded by an `0xFF` sync byte.

If the accessory is also backward compatible with the MFi Accessory Firmware Specification R46, it may proceed to initiate communication with the older device using iAP1. The StartIDPS command must be sent within 500 ms of receiving confirmation that the Apple device supports iAP1.

If the accessory receives this message but is not backward compatible, the accessory must send the following byte sequence to the device to indicate lack of backward compatibility:

```
FF 55 0E 00 13 FF FF FF FF FF FF FF FF FF FF FF FF EB
```

The accessory must treat any other response from the device as final; no retries or retransmissions are allowed.

57.7.3 Synchronization

One notable feature of the link is support for automatic negotiation of transmission parameters over any type of transport. This permits the link to scale according to the capabilities of both device and accessory. Link initialization has 3 main goals:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Determines mutually acceptable link configuration parameters for both device and accessory.
- Searches for errors in configuration parameters.
- Terminates the link if no agreement can be reached.

A link is initiated when an underlying transport connection and device compatibility have been established. When this occurs, the accessory first sends a SYN packet containing desired connection parameters in the Link Synchronization Payload, along with a randomly generated Packet Sequence Number. The device will respond with an SYN+ACK packet acknowledging receipt of the accessory's SYN packet, its own desired connection parameters, and its randomly generated Packet Sequence Number. If the two sets of connection parameters match, the accessory must send a final ACK packet of the most recent SYN+ACK from the device, and the connection is considered to be established. Otherwise, SYN+ACK packets will continue to be exchanged up to 10 times until connection parameters are agreed upon. If 10 exchanges occur without agreement on connection parameters then the device will stop responding to any further packets sent by the accessory.

Sometimes, a device will be unable to respond immediately to the receipt of a SYN packet from an accessory. In that situation, the accessory may resend the SYN packet with the same Payload Data and the same Packet Sequence Number. It may continue doing so every second until the device sends a SYN+ACK response acknowledging the previously sent Packet Sequence Number. Once this response has been received, the accessory must ignore any subsequent incoming SYN+ACK packets with the same Packet Sequence Number.

The device will send a RST packet to reset the link if the accessory's final Link Synchronization Payload contains an invalid non-negotiable parameter.

During synchronization, the following default link configuration parameters are assumed until synchronization is complete:

Table 57-6 Default link parameters during synchronization

Parameter	Default Value
Maximum Number of Outstanding Packets	1
Maximum Received Packet Length	128 bytes
Retransmission Timeout	1000 ms
Cumulative Ack Timeout	10 ms
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	0

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The following tables contain suggested link configuration parameters for some iAP2 transports. These values are just starting points for development and must not be used without verifying suitability for a particular accessory.

Table 57-7 Suggested link parameters for USB Host Mode transport (Full Speed)

Parameter	Suggested Value
Maximum Number of Outstanding Packets	5
Maximum Received Packet Length	4096 bytes
Retransmission Timeout	2000 ms
Cumulative Ack Timeout	22 ms
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	3

Table 57-8 Suggested link parameters for USB Device Mode transport (Full Speed)

Parameter	Suggested Value
Maximum Number of Outstanding Packets	5
Maximum Received Packet Length	4096 bytes
Retransmission Timeout	2000 ms
Cumulative Ack Timeout	22 ms
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	3

Table 57-9 Suggested link parameters for Bluetooth transport

Parameter	Suggested Value
Maximum Number of Outstanding Packets	5
Maximum Received Packet Length	2048 bytes
Retransmission Timeout	1500 ms
Cumulative Ack Timeout	73 ms

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Parameter	Suggested Value
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	3

Table 57-10 Suggested link parameters for 57.6 kbps serial transport

Parameter	Suggested Value
Maximum Number of Outstanding Packets	4
Maximum Received Packet Length	256 bytes
Retransmission Timeout	1000 ms
Cumulative Ack Timeout	133 ms
Maximum Number of Retransmissions	30
Maximum Cumulative Acknowledgements	2

57.7.4 Acknowledgements

To guarantee delivery of packets, the link protocol uses both positive acknowledgement and retransmission of packets. Each packet with Payload Data and SYN packets are acknowledged when they are correctly received and accepted by the receiver. ACK packets are not acknowledged. Damaged packets are discarded and not acknowledged. Packets are retransmitted by the sender when they are not acknowledged by the receiver within the time specified by the Retransmission Timeout.

There are two types of packet acknowledgements. A cumulative acknowledgement is used to acknowledge receipt of all in-sequence packets up to a specified Packet Sequence Number. The extended acknowledgement allows the receiver to acknowledge packets out of sequence. The type of acknowledgement used is simply a function of the order in which packets arrive. Whenever possible, link implementations must use the cumulative acknowledgement and use the extended acknowledgement only when packets are received out of sequence.

Receivers must generate and send an extended acknowledgement as soon as one or more out-of-sequence packets are received.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

57.7.5 Retransmissions

Packets can get lost in transmission because of loss or damage in the underlying transport. They can also be discarded by the receiver. The positive acknowledgement of packets requires the receiver to acknowledge a packet only when the packet has been correctly received and accepted.

To detect missing packets, the sender must use a retransmission timer for each outgoing packet. This timer is set to the negotiated Retransmission Timeout value. When an acknowledgement is received for a packet, the timer is cancelled for that packet. If the timer expires before an acknowledgement is received, that packet is retransmitted and the timer is restarted, up to the Maximum Number of Retransmissions negotiated during link synchronization.

57.7.6 Flow Control

The iAP2 link employs a simple flow control mechanism that is based on the number of unacknowledged packets sent and the Maximum Number of Outstanding Packets link configuration parameter. This parameter is specified by each side when the link is first created, and should be set based on the number and size of buffers that each side is willing to allocate to the link. Once it is set this parameter remains unchanged while the link is connected.

The link employs the concept of a sequence number window for acceptable Packet Sequence Numbers. The left edge of the window is the last in-sequence acknowledged Packet Sequence Number plus one. The right edge of the window is equal to the last in-sequence acknowledged Packet Sequence Number plus the Maximum Number of Outstanding Packets. The link sender sends packets until the receiver's Maximum Number of Outstanding Packets is reached; once this limit is reached, the sender may only send a new packet for each acknowledged packet.

When a received packet has a Packet Sequence Number that falls within the window, it is acknowledged. If the Packet Sequence Number is equal to the left edge (i.e., it is the next expected Packet Sequence Number), the packet is acknowledged with a cumulative acknowledgement (ACK), and the acceptance window's left and right edges are incremented by one. If the received packet's Packet Sequence Number is within the window but is out of sequence, it is acknowledged with an Extended Acknowledgement (EAK). The window is not adjusted, and the receipt of the out-of-sequence packet is recorded. Received link packets with Packet Sequence Numbers outside of the window must be discarded; they are a sign that the link is unstable and may need to be reset.

When a sender receives Extended Acknowledgements (EAK), the sender must not transmit beyond the acceptance window. This could occur if one packet is not acknowledged but all subsequent packets are received and acknowledged. This requirement will fix the left edge of the window at the sequence number of the unacknowledged packet. As additional packets are sent, the next Packet Sequence Number will approach and eventually overtake the right edge. At this point, no more packets may be sent until the unacknowledged packet is acknowledged.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

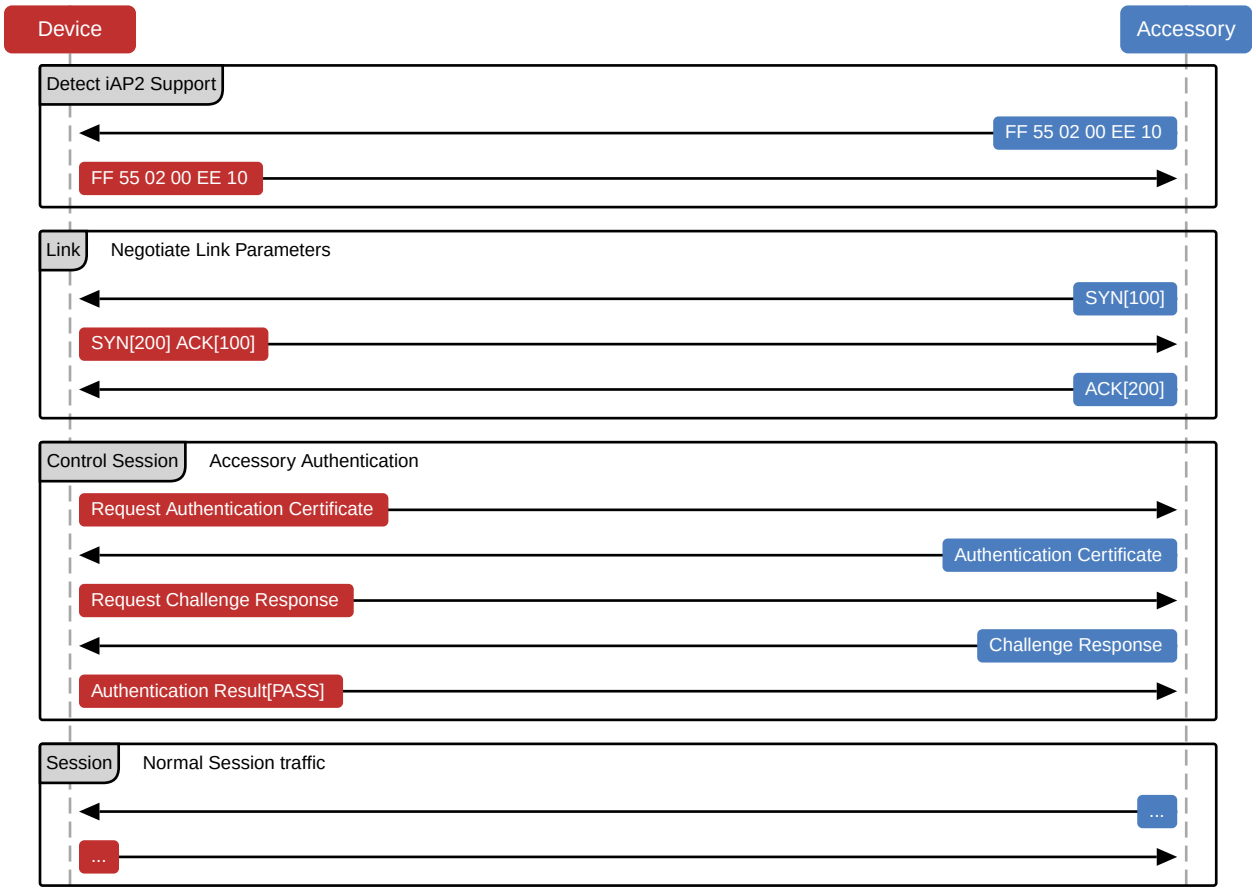
57.7.7 Reset

The device may send a RST packet to the accessory at any time. Upon receipt of this packet, the accessory must send a SYN packet to the device within 1 second of receiving the RST packet and proceed as specified in Synchronization (page 773).

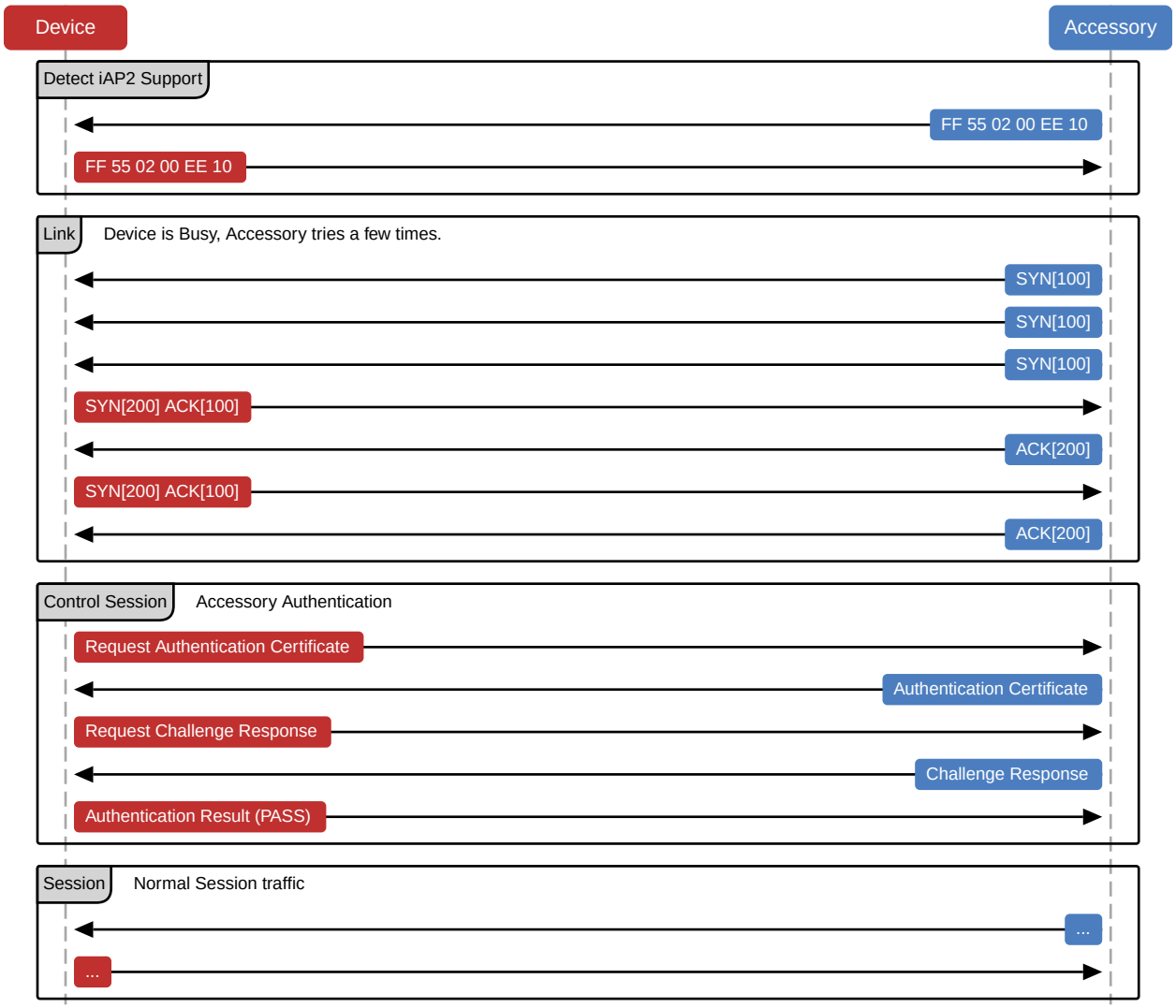
Accessories must terminate the underlying transport connection and reconnect if they wish to reset the link. This is not possible when the serial transport is active without physically disconnecting and reconnecting the accessory with the device's Lightning connector. Having to restart the link is a sign of poorly negotiated link parameters and should not occur during typical operation.

57.8 Examples

57.8.1 Typical Link Initialization



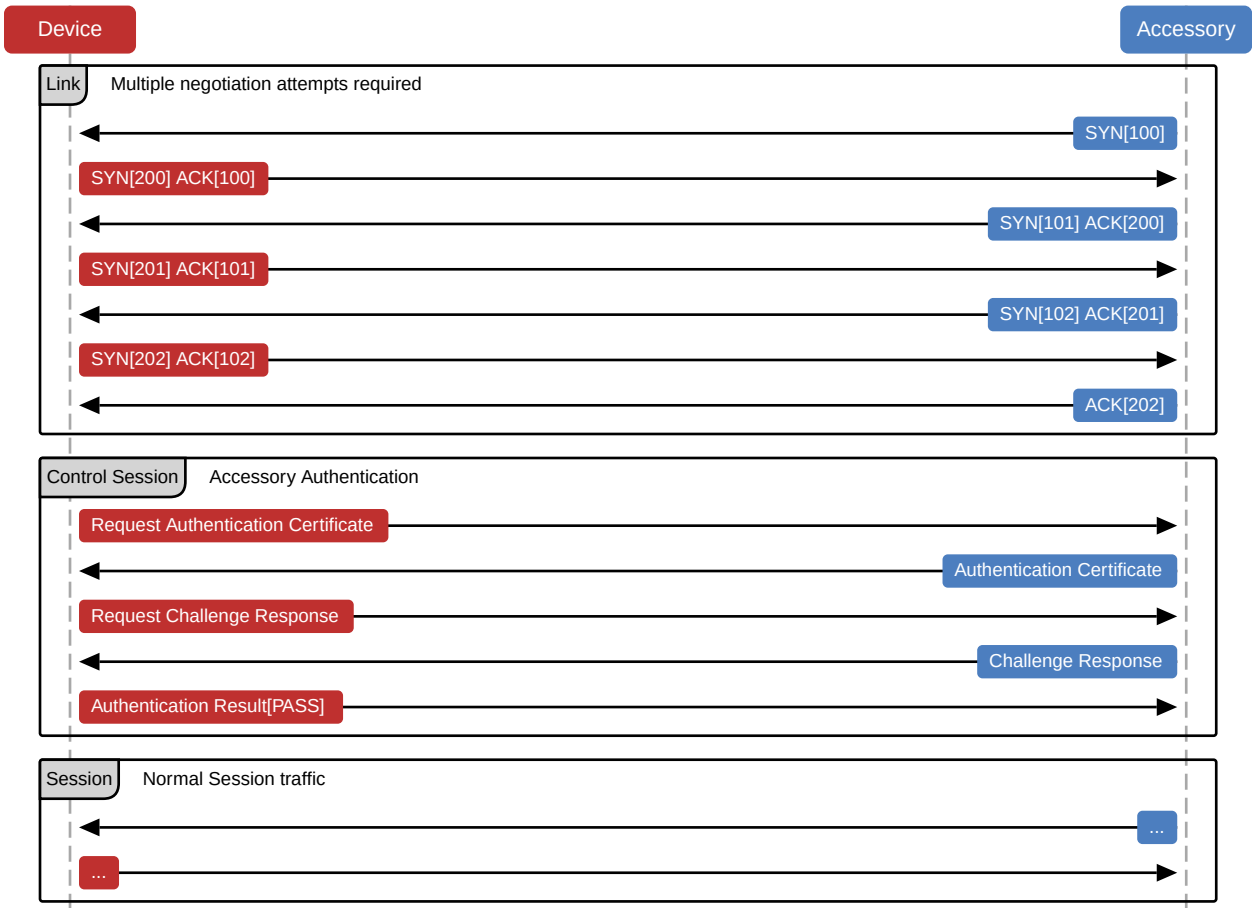
57.8.2 Connection Initialization When Device is Busy



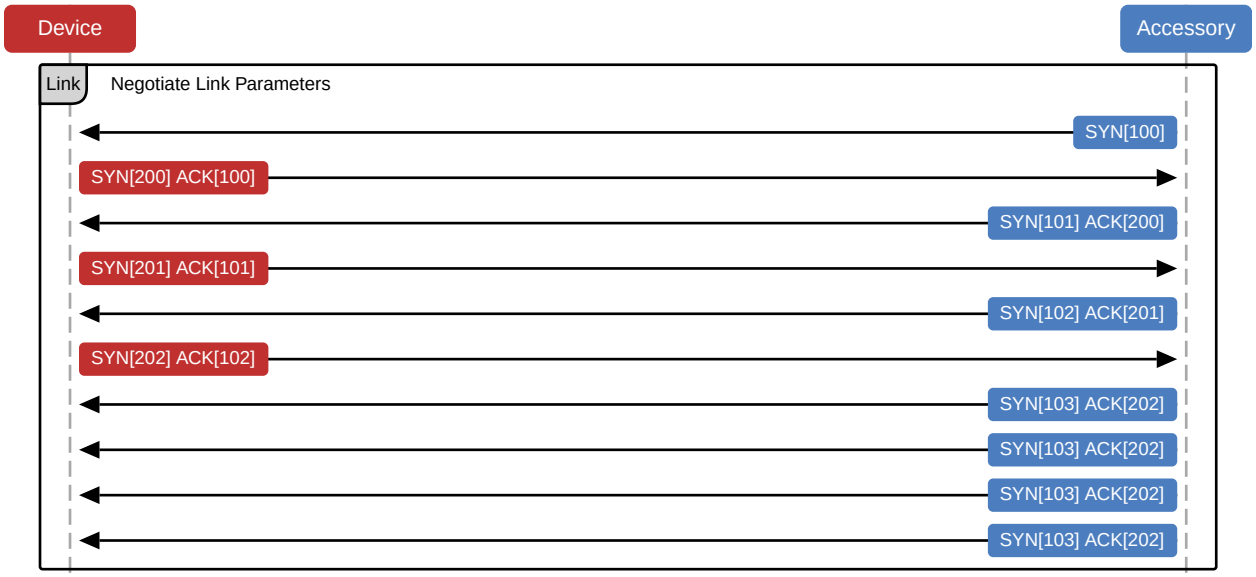
57.8.3 Connection Requiring Multiple Negotiation Attempts

Multiple negotiation attempts may occur when accessory and device do not mutually agree on iAP2 Link parameters right away.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.



57.8.4 Connection With Failed Negotiation

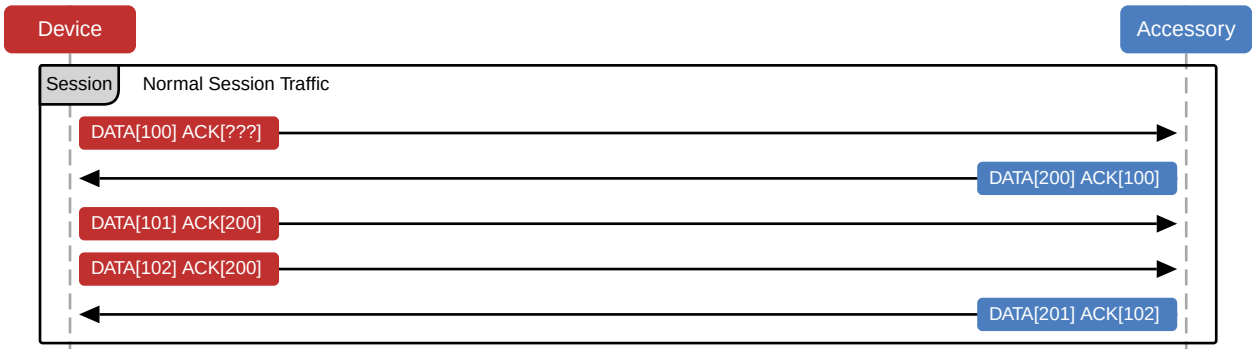


2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

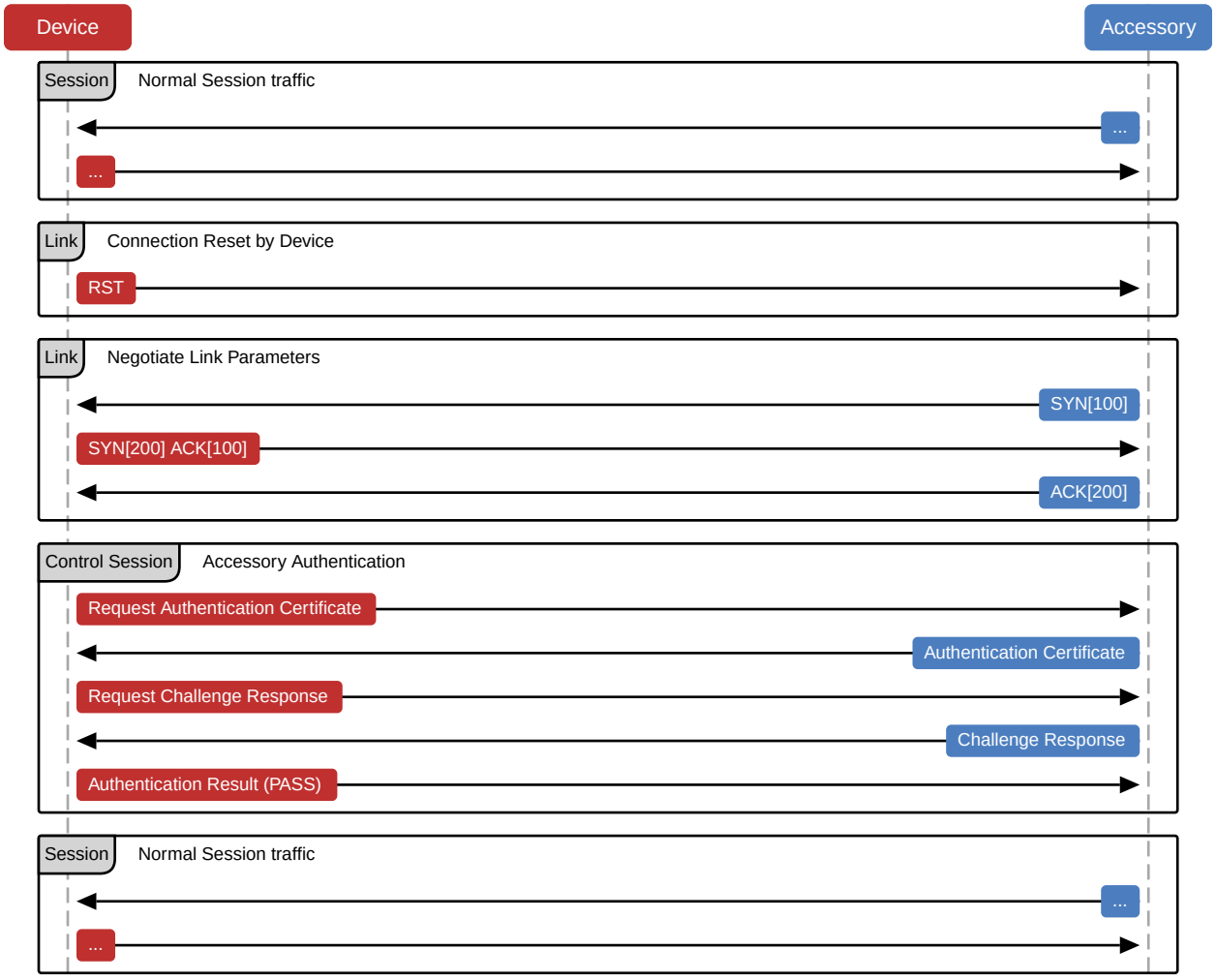
57.8.5 Normal Connection Traffic

After initialization/setup of the connection with accessory, normal runtime traffic consists of regular ACK packets with data as needed.

At any time, a RST packet may be sent by the Device to the Accessory. Accessory must reset the connection and start the connection sequence again by sending the SYN packet.

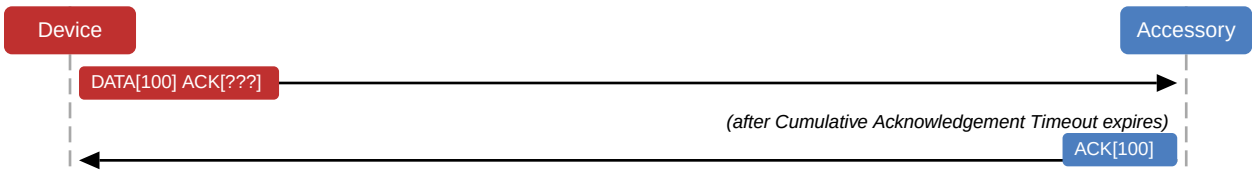


57.8.6 Device Reset of Transport Connection



57.8.7 Cumulative Ack Timeout Expired

In this example, the Maximum Cumulative Acknowledgements link parameter is set to two. The device has sent one packet but has no need to send another one at this time. After the Cumulative Acknowledgement Timeout has expired the accessory assumes that there is no need to wait for a second DATA packet and sends an ACK packet.

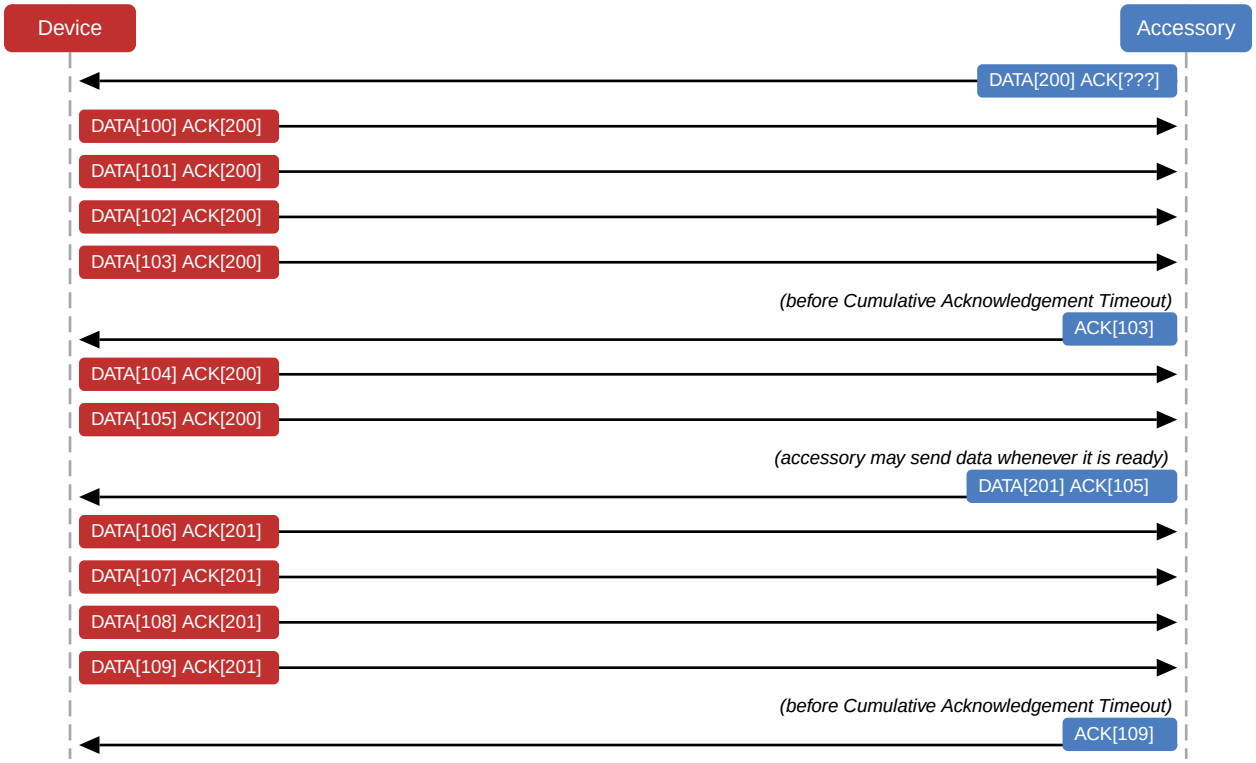


2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

57.8.8 Continuous Data Transmission with ACKs

ACK packets and Data packets with ACK are used to acknowledge receipt of a previously sent packet and to send data. In this example, the Maximum Cumulative Acknowledgement parameter is set to 4, and the Device does not put more than 4 packets in flight at any given time.

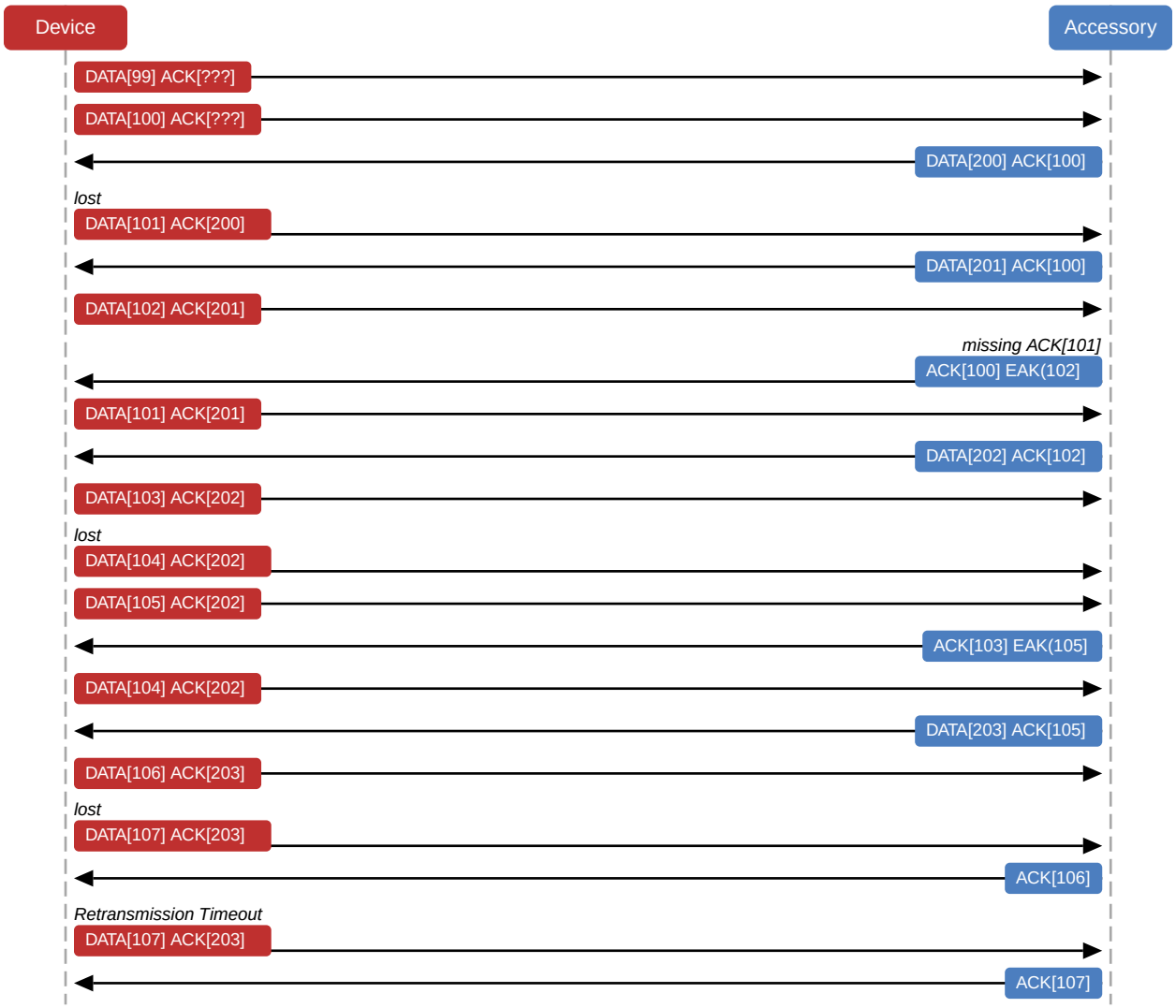
The accessory does not need to wait for Cumulative Acknowledgement Timeout to expire if it has additional packets to send.



57.8.9 Resend of missing packets using EAK

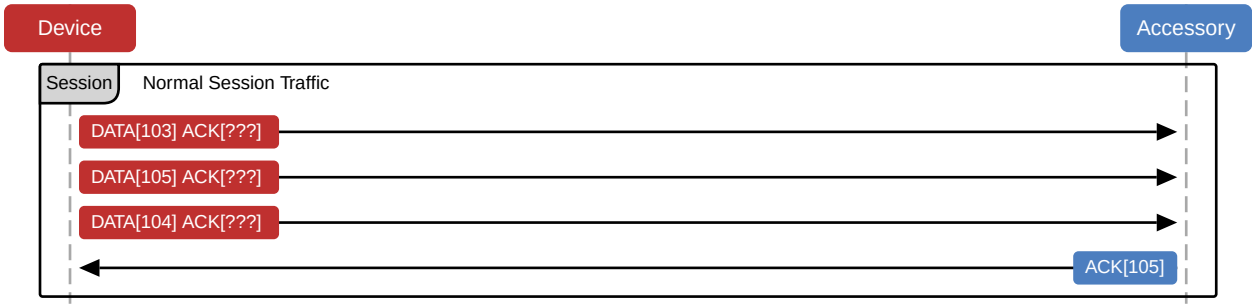
The EAK packet is used to indicate missing packets. Upon receiving an EAK packet, the recipient must resend the missing packets. The packet transfer recipient must wait for acknowledgement of all in-flight packets up to the Maximum Cumulative Acknowledgement parameter, or wait the Cumulative Acknowledgement Timeout to expire before sending an EAK.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.



57.8.10 Receiving Packets Out Of Order

The accessory will receive packets out of order and ACK the last part of the data it receives.



57.8.11 iAP2 Link Packet Structure Example

Packet Length MSB and *Packet Length LSB* refer to the size of the packet (0x001A). *Control Byte* identifies the type of payload being sent (0x80), SYN packet. *Packet Sequence Number* in this case in randomly picked (0x2B). *Packet Acknowledgement Number* and *Session Identifier* have no meaning in this context. *Header Checksum* is calculated to be 0xE2. *Payload Data* and *Payload Checksum* are described in detail in iAP2 Link Synchronization Payload Example (page 785).

Refer to Packet Structure (page 764) for more details.

Table 57-11 iAP2 Link Packet Structure Example - Accessory SYN Packet

Start of Packet MSB (FF)
Start of Packet LSB (5A)
Packet Length MSB (00)
Packet Length LSB (1A)
Control Byte (80 SYN Packet)
Packet Sequence Number (2B)
Packet Acknowledgement Number (00)
Session Identifier (00)
Header Checksum (E2)
Payload Data
(01 05 10 00 04 0B 00 17 03 03 0A 00 01 0B 02 01)
Payload Checksum (A5)

57.8.12 iAP2 Link Synchronization Payload Example

Refer to Link Synchronization Payload (page 768) for more details.

Table 57-12 Link Synchronization Payload Example

Link Version (01)
Maximum Number of Outstanding Packets (05)
Maximum Received Packet Length MSB (10)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Maximum Received Packet Length LSB (00)
Retransmission Timeout MSB (04)
Retransmission Timeout LSB (0B)
Cumulative Acknowledgement Timeout MSB (00)
Cumulative Acknowledgement Timeout LSB (17)
Maximum Number of Retransmissions (03)
Maximum Cumulative Acknowledgements (03)
iAP2 Session 1: Session Identifier (0A)
iAP2 Session 1: Session Type (00)
iAP2 Session 1: Session Version (01)
iAP2 Session 2: Session Identifier (0B)
iAP2 Session 2: Session Type (02)
iAP2 Session 3: Session Version (01)

57.9 Test Procedures

57.9.1 Link Layer (v1)

1. Verify that all accessories use only Link Version number 1.

   Note: CarPlay Accessories will use the Link Version number 2.

2. With Apple devices running iOS 7.0 or later, verify that the Link Layer session starts with the following sequence:
   - Accessory DETECT
   - Device DETECT
   - Accessory SYN
   - Device SYN+ACK
   - Accessory ACK
   - Device ACK RequestAuthenticationCertificate

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- Accessory ACK AuthenticationCertificate
- Device ACK RequestAuthenticationChallengeResponse
- Accessory ACK AuthenticationResponse
- Device ACK AuthenticationSucceeded
- Device ACK StartIdentification
- Accessory ACK IdentificationInformation
- Device ACK IdentificationAccepted
- Accessory ACK
- Additional ACKs packets with no payload (length of 9) are acceptable.

3. Verify that the Packet Sequence Number increment by 1.
4. Verify that all ACK packets without payloads (length of 9), use the Session Identifier 0. For any other ACK, packet the Session Identifier cannot be 0.

Jay Zhang
Shenzhen Goertek Electronics Co.,Ltd.
jayzhang@goertek.com

58. iAP2 Sessions

Once an iAP2 link has been established, higher level communication between an accessory and an Apple device is structured into one or more sessions. At the very minimum, all accessories must establish a control session for the purposes of authentication and identification. Once initial authentication and identification are complete, the accessory may use additional sessions.

Every session has a session identifier, session type, and session version. The session identifier is used to uniquely identify a session within an iAP2 connection. The session identifier of 0 is reserved for link use, and all sessions used by an accessory must have unique nonzero identifiers. The combination of both session type and session version dictates which session protocol variant is used for all data traffic in that session. The session identifier is part of every iAP2 link packet header.

58.1 Attributes

58.1.1 Type

Table 58-1 iAP2 Session Types

Session Type	Name
0	Control session
1	File Transfer session
2	External Accessory session

All link implementations must have one control session. All other sessions are optional depending on the features implemented. Multiple sessions of the same type are not permitted.

58.1.2 Version

The session version is used to distinguish between different variants of a particular session protocol. The exact nature of these differences depends on the session type. See the sections in the specification on each session type for more details.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

58.2 Control Session

The control session has 2 primary goals:

- Notify both the device and accessory of changes in the other's state and/or configuration.
- Provide the accessory with means to request changes in device state and/or configuration.

The control session protocol is a stateless protocol. Each message is an independent entity that has no relationship to any prior message. The device does not retain any information or status about each attached accessory from message to message. An accessory must behave the same way. Specifically, an accessory request for a change in device state and/or configuration must be treated as a request and not a command. The device may or may not choose to honor the request, and will either send an update message to confirm a change in state/configuration or not send any message at all.

All accessories must use control session version 1 with one exception. Accessories supporting the CarPlay feature (see CarPlay (page 381)) must advertise support for control session version 2.

Accessories that support control session version 1 must complete authentication (see Accessory Authentication (page 261)) before sending any other messages.

Accessories that support control session version 2 may send identification messages (see Accessory Identification (page 265)) before authentication completes, but must complete authentication before sending any other messages. They must fall back to control session 1 if the Apple device does not support control session 2 or not support those Apple devices.

Accessories must use control session messages for their specified purpose. Relying on unspecified behavior is likely to cause incompatibilities with future Apple devices and/or iOS releases.

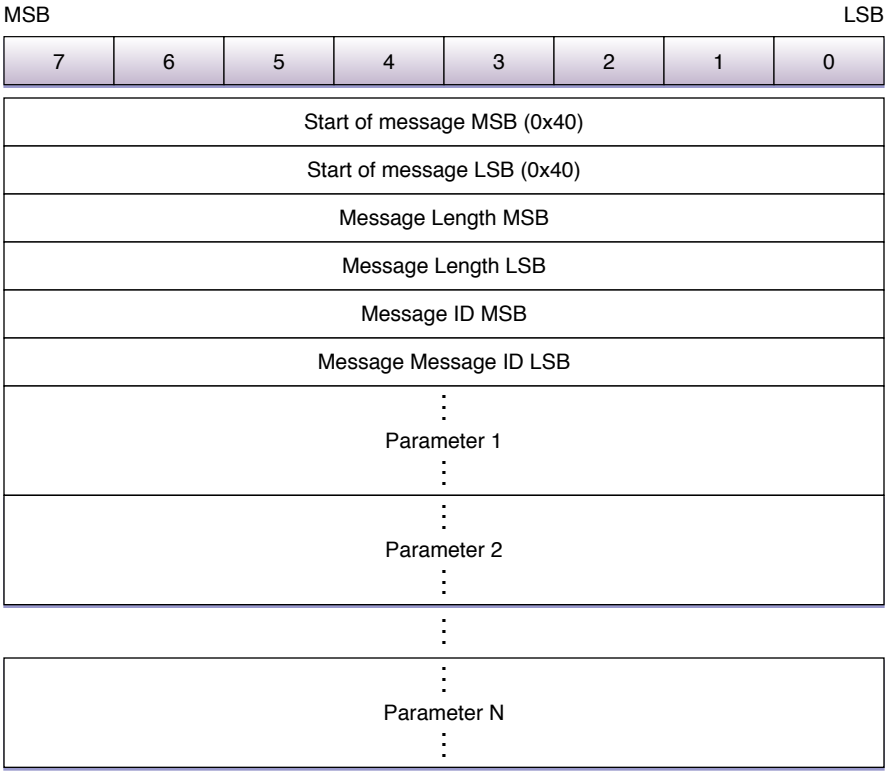
Accessories must rely on the iAP2 link layer to guarantee message delivery. Accessories should never resend messages at the control session layer.

58.2.1 Message Structure

Each message has a header followed by 0 or more parameters or parameter groups:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 58-1 Control Session Message Structure



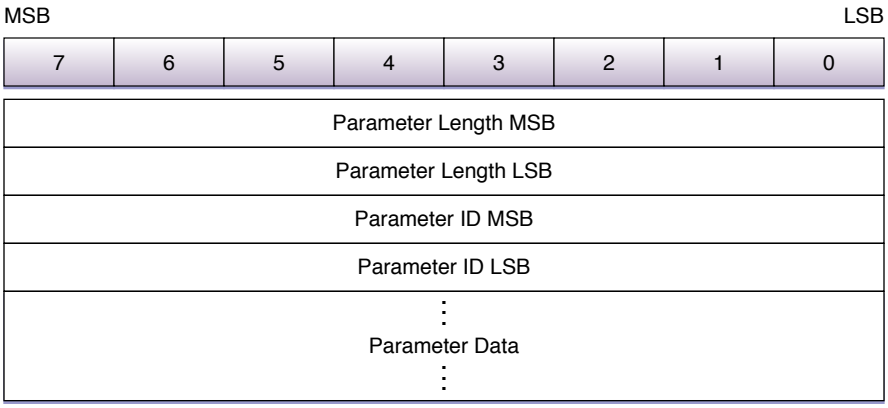
Message Length:
Size of the whole message, in bytes, including the Start of Message field, Message Length field, Message ID field and all Parameters.

Message ID:
Message Identifier

Parameter:
Parameters for this message (see below)

Parameters are structured as follows:

Figure 58-2 Control Session Message Parameter Structure



Parameter Length:
Length of parameter, in bytes, including the parameter length field, parameter ID and parameter data.

Parameter ID:
Interpretation is message specific.

Parameter Data:
Type of the data is determined by the parameter ID for this specific message ID.

58.2.2 Message Parsing

All iAP2 control session message parsers must comply with the following requirements:

- Parameters can be specified in any order.
- Received messages must be ignored if the message identifier is unrecognized.
- Received messages must be ignored if any required parameters are missing.
- Received messages must be ignored if any parameters have invalid values.
- Extra unknown parameters in a received message must be ignored and message parsing must proceed with the remaining parameters.

58.2.3 Parameter Types

All parameter bytes are transmitted and received in big-endian order.

58.2.3.1 Number

Only certain types of numbers are permitted in messages:

- **int8** - signed 8-bit byte
- **int16** - signed 16-bit word

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- **int32** - signed 32-bit long
- **int64** - signed 64-bit quadword
- **uint8** - unsigned 8-bit byte
- **uint16** - unsigned 16-bit word
- **uint32** - unsigned 32-bit long
- **uint64** - unsigned 64-bit quadword

If a number parameter has a limited range of permitted values, the range will be specified in the parameter notes. Otherwise, the full range of possible values is assumed to be valid.

58.2.3.2 Enumeration

Written in parameter lists as 'enum'. Unsigned 8-bit byte which corresponds to an exact listing of all of its possible values.

58.2.3.3 Boolean

Written in parameter lists as 'bool'. Special type of Enumeration encoded as a one byte unsigned integer where 0 indicates NO and 1 means YES. All other enumeration values are invalid.

58.2.3.4 String

Written in parameter lists as 'utf8'. Null terminated UTF-8 string. Accessories that receive string parameters must be prepared to handle any valid UTF-8 string, as Apple devices allow users to input strings in all supported languages and character sets, such as Emoji.

58.2.3.5 Blob

Written in parameter lists as 'blob'. The contents of the binary blob will be defined in the parameter notes. Typically, most blobs are arrays of custom data structures.

> **Note:** Accessories must be prepared to handle zero-length blobs.

58.2.3.6 None

Written in parameter lists as 'none'. No parameter data. The mere existence or absence of the parameter in the message has meaning.

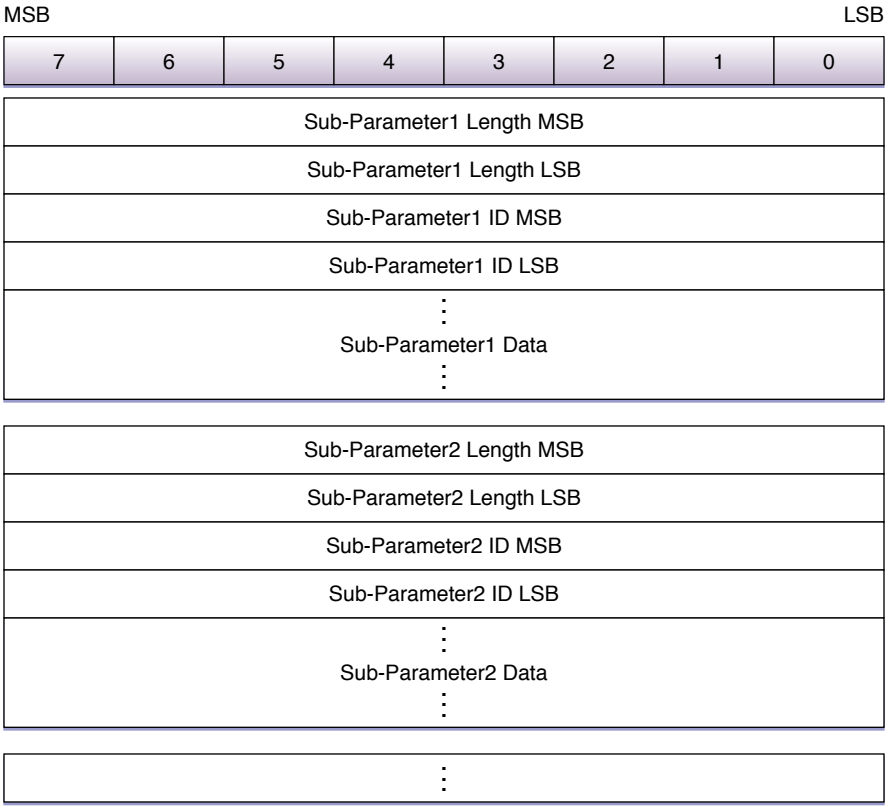
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Note: Void parameters must have a length of 4 and no payload data.

58.2.3.7 Group

Written in parameter lists as 'group'. Parameters maybe grouped together as a group. Group parameter data consists of a number of sub-parameters each having a Format consisting of Sub-parameter Length, ID, and Data. Sub-parameters may be present in any order within a group.

Figure 58-3 Group Parameter Structure



58.2.4 Message Example

Example StartExternalAccessoryProtocolSession Control Session message

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 58-4 StartExternalAccessoryProtocolSession Control Session Message Example

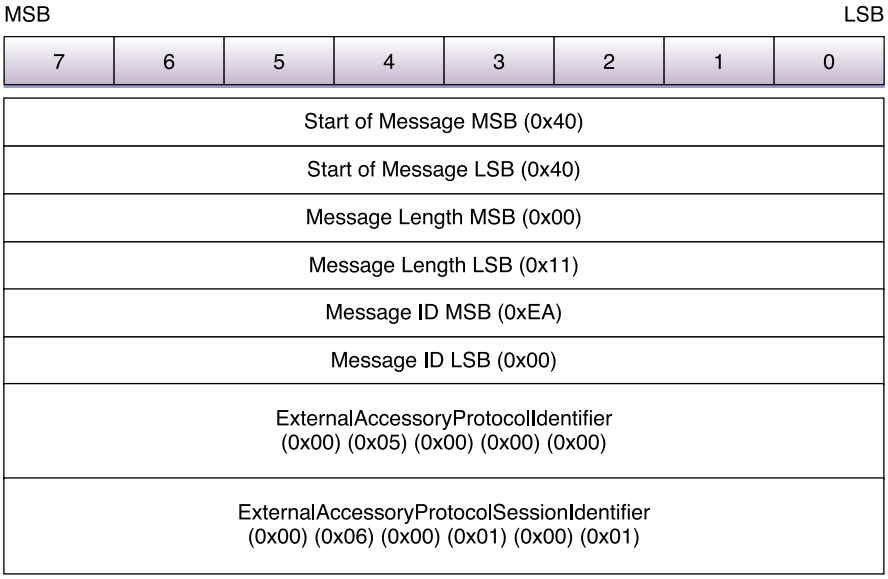


Figure 58-5 ExternalAccessoryProtocolIdentifier Parameter

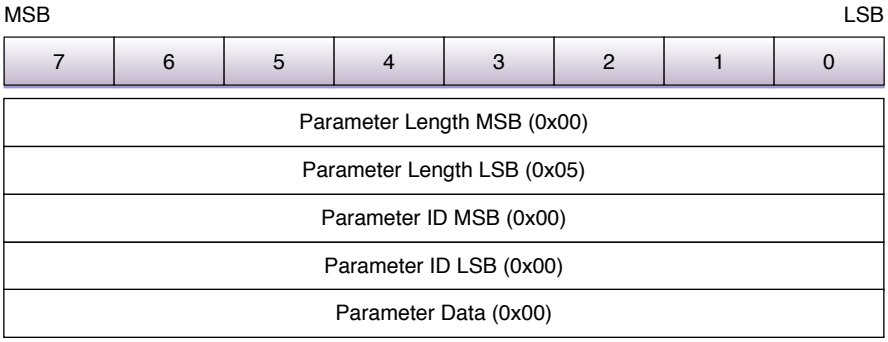
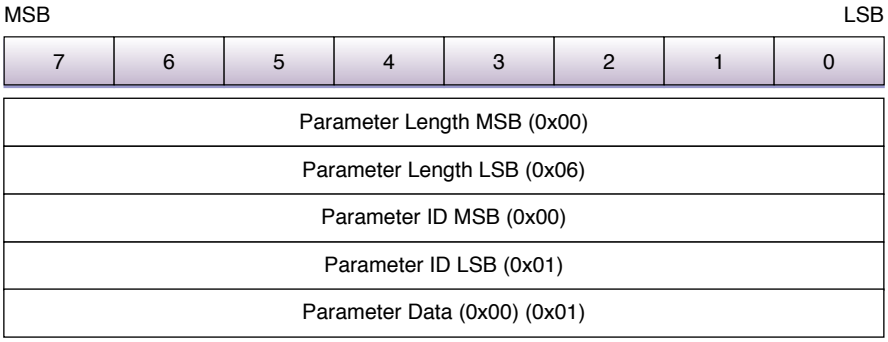


Figure 58-6 ExternalAccessoryProtocolSessionIdentifier Parameter



Example StartNowPlayingUpdates Control Session message with parameter group

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Figure 58-7 StartNowPlayingUpdates Control Session Message Example

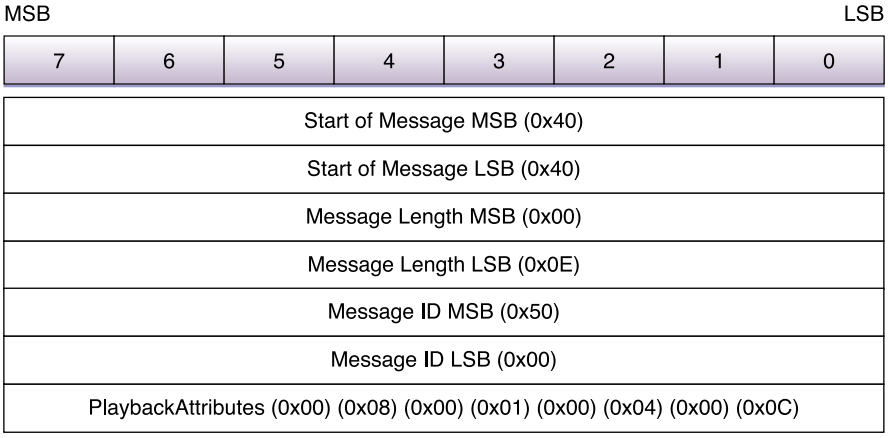
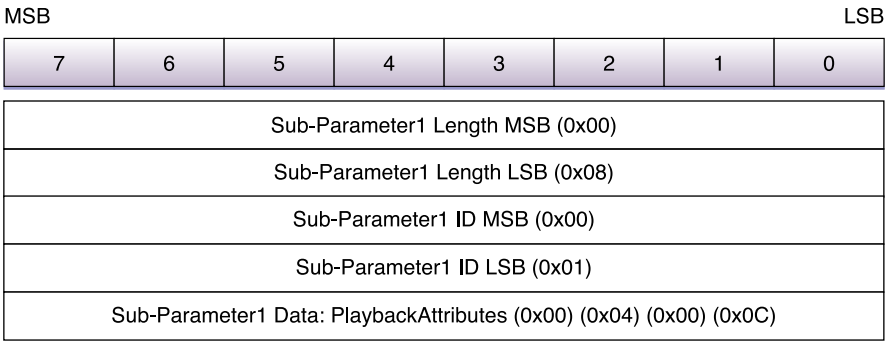


Figure 58-8 PlaybackAttributes Parameter Group



58.3 File Transfer Session

Any accessory that identifies itself as capable of sending or receiving any messages associated with file transfers must set up a file transfer session during link negotiation.

All file transfers are initiated by the sender and have a unique identifier that is also assigned by the sender. Up to 128 simultaneous transfers in each direction are possible, and transfers may be paused and/or canceled before they are complete by either the sender or the receiver.

All file transfer datagrams must fit within a single iAP2 link packet payload.

At this time, the only valid file transfer session version is version 1.

58.3.1 TransferIdentifier

Setup of a file transfer takes place in the control session. The following iAP2 messages can generate a file transfer identifier parameter:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- MediaLibraryUpdate (page 852)
- NowPlayingUpdate (page 860)

To support file transfers from the Accessory to the Device, every iAP2 connection must maintain a global unsigned 8-bit FileTransferIdentifier that starts at 0 when the connection is established. The FileTransferIdentifier must be incremented to the next inactive FileTransferIdentifer value once it has been used. Once the global FileTransferIdentifier reaches 127 the accessory must immediately reset it to 0 or the lowest inactive FileTransferIdentifier value, whichever is lower, before it initiates the next transfer.

MediaLibraryUpdate and NowPlayingUpdate messages can generate a FileTransferIdentifer parameter for transfers from the Device to the Accessory, which will have parameters ranging from 128 to 255.

58.3.2 Setup Datagram

Once a message with a FileTransferIdentifier parameter has been sent, the sender must issue a Setup datagram that contains the file size in bytes over the file transfer session using the same FileTransferIdentifier.

Table 58-2 File Transfer Session Setup Datagram

Byte	Value
0	FileTransferIdentifier
1	0x04
2	File Size Byte 0
3	File Size Byte 1
4	File Size Byte 2
5	File Size Byte 3
6	File Size Byte 4
7	File Size Byte 5
8	File Size Byte 6
9	File Size Byte 7

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

58.3.3 Start

Upon receiving the Setup datagram from the sender, the receiver may send a Start datagram back to the sender to signify that it is ready to receive file data.

Table 58-3 File Transfer Session Start Datagram

Byte	Value
0	FileTransferIdentifier
1	0x01

58.3.4 FirstData

The sender may start sending file data once it receives a Start datagram. The first datagram is unique and must take the following format:

Table 58-4 File Transfer Session FirstData Datagram

Byte	Value
0	FileTransferIdentifier
1	0x80
2	Data Byte 0
...	...
n+2	Data Byte n

Only one FirstData datagram may be sent for any given assigned FileTransferIdentifier. The sender must switch to Data datagrams for all subsequent file data until the end of the file.

58.3.5 FirstAndOnlyData

If the file is so short as to fit in one session datagram, the sender may send a FirstAndOnlyData datagram as follows:

Table 58-5 File Transfer Session FirstAndOnlyData Datagram

Byte	Value
0	FileTransferIdentifier

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Byte	Value
1	0xC0
2	Data Byte 0
...	...
n+2	Data Byte n

Only one FirstAndOnlyData datagram may be sent for any given assigned FileTransferIdentifier. The file transfer is considered complete upon receipt of the FirstAndOnlyData datagram.

58.3.6 Data

Regular file data datagrams must take the following format:

Table 58-6 File Transfer Session Data Datagram

Byte	Value
0	FileTransferIdentifier
1	0x00
2	Data Byte 0
...	...
n+2	Data Byte n

58.3.7 LastData

The last object data datagram is unique and must take the following format:

Table 58-7 File Transfer Session LastData Datagram

Byte	Value
0	FileTransferIdentifier
1	0x40
2	Data Byte 0

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Byte	Value
...	...
n+2	Data Byte n

58.3.8 Cancel

The Cancel datagram may be sent at any time by either the sender or receiver after the Setup datagram is sent or received. The FileTransferIdentifier must be marked as inactive by both the sender and receiver after it is transmitted.

The Cancel datagram must be sent by any accessory or device if any other File Transfer Session datagram is received with an inactive FileTransferIdentifier.

Table 58-8 File Transfer Session Cancel Datagram

Byte	Value
0	FileTransferIdentifier
1	0x02

58.3.9 Pause

The Pause datagram may be sent at any time by the sender or receiver after the first Start datagram has been sent by the receiver. Unlike the Cancel datagram, the FileTransferIdentifier remains active, but no FirstData, Data, or LastData datagrams may be transmitted until the end that sent the Pause datagram sends a Start datagram with the same FileTransferIdentifier. Data datagrams will resume from the point where the transfer left off.

Table 58-9 File Transfer Session Pause Datagram

Byte	Value
0	FileTransferIdentifier
1	0x03

58.3.10 Success

The Success datagram must be sent by the receiver after it has received and processed all file data. The sender may choose to delete the file or otherwise dispose of it after receiving this datagram from the receiver.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

The FileTransferIdentifier must be marked as inactive by both the sender and receiver after this datagram is transmitted.

Table 58-10 File Transfer Session Success Datagram

Byte	Value
0	FileTransferIdentifier
1	0x05

58.3.11 Failure

The Failure datagram must be sent by the receiver if it has received all file data but is unable to store it. This sender may choose to resend the file or not, depending on the circumstances.

The FileTransferIdentifier must be marked as inactive by both the sender and receiver after this datagram is transmitted.

Table 58-11 File Transfer Session Failure Datagram

Byte	Value
0	FileTransferIdentifier
1	0x06

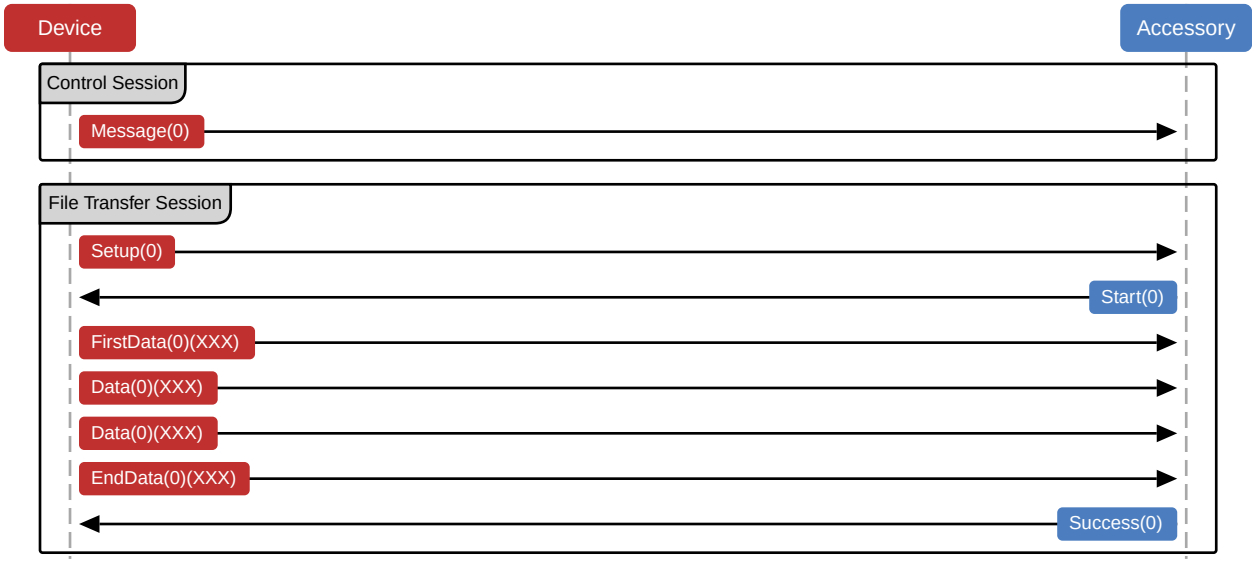
58.3.12 Examples

In all examples, the device is playing the role of the file sender, and the accessory is the file receiver. These roles can be reversed depending on the needs of a particular accessory interface feature

58.3.12.1 Typical Transfer

In this example, the device is sending one file to the accessory, and no other transfers are taking place.

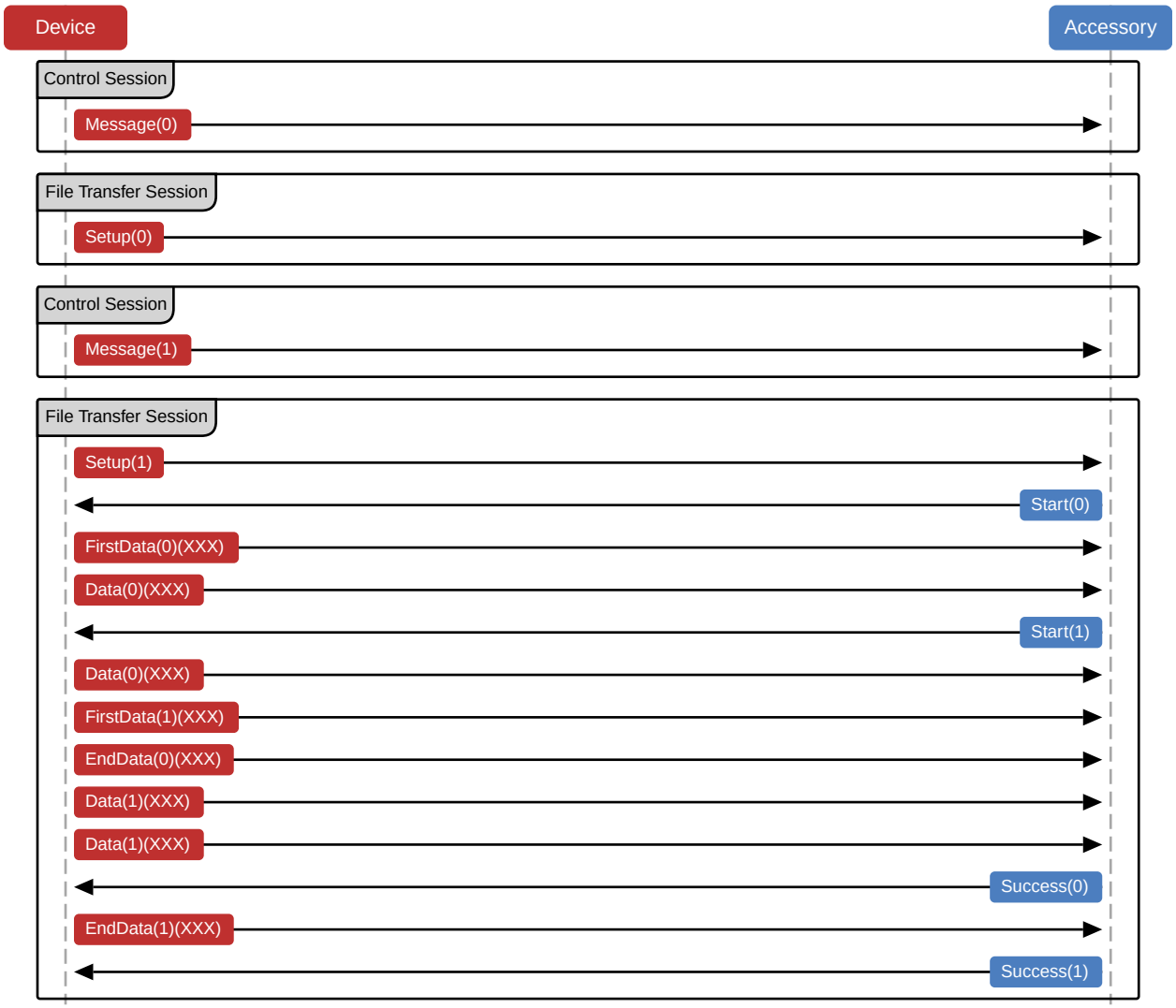
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.



58.3.12.2 Two Simultaneous Transfers

In this example, the device is sending two files to the accessory simultaneously. Note that all of the traffic can be interleaved arbitrarily, so long as the message containing the FileTransferIdentifier is sent before any file transfer session traffic starts.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.



58.4 External Accessory Session

The external accessory (EA) session is used by the External Accessory Protocol (page 535) to create one or more bidirectional serial data streams, or transfers, between an accessory and iOS apps.

At this time, the only valid external accessory session version is version 1.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

58.4.1 Setup

Setup of an EA session takes place in the control session. The accessory will receive a StartExternalAccessoryProtocolSession (page 843) message with assigned `ExternalAccessoryProtocolIdentifier` and `ExternalAccessoryProtocolSessionIdentifier` parameters.

The `ExternalAccessoryProtocolIdentifier` is defined by the accessory during identification; an accessory may define and support multiple supported protocols.

The `ExternalAccessorySessionIdentifier` is unique to each EA connection between an app and the accessory. Accessories must use this identifier to distinguish between datagrams sent over the EA session.

Accessories that declare an EA session during link synchronization must declare support for one or more `ExternalAccessoryProtocol` parameters during accessory identification as specified in External Accessory Protocol (page 535).

Accessories must not send any EA session datagrams with an unassigned transfer identifier. Conversely, accessories must ignore any EA session datagrams with unassigned session identifiers.

58.4.2 ExternalAccessorySession Datagram

All datagrams for an EA session must follow this format:

Table 58-12 ExternalAccessorySession Datagram

Byte	Value
0	ExternalAccessorySessionIdentifier Byte 0
1	ExternalAccessorySessionIdentifier Byte 1
2+	External Accessory Session Data

All EA session datagrams must fit within a single iAP2 link packet payload.

58.5 Test Procedures

58.5.1 Control Session

1. Verify that accessories exhaustively enumerate all iAP2 control session messages that they can send or receive.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

2. Authentication and Identification messages are not required to be listed as supported features in IdentificationInformation; all accessories must use them. Some claimed messages are not required to be seen. Exceptions include DeviceHIDReport, StopHID, StopPowerUpdates, StopNowPlayingUpdates, StopUSBDeviceModeAudio and StopAssistiveTouch.
3. Verify that Confirm required fields (Name, brand, etc.) are correctly filled in (no "filler" entries, such as ACME, 1234, etc.)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59. iAP2 Control Session Messages

59.1 Accessory Authentication

For more information, see Accessory Authentication (page 261).

59.1.1 RequestAuthenticationCertificate

Source	ID
Device	0xAA00

Table 59-1 `RequestAuthenticationCertificate` message parameters

Name	ID	Type	#	Notes
RequestAuthenticationCertificateSerialNumber	0	none	0/1	

59.1.2 AuthenticationCertificate

Source	ID
Accessory	0xAA01

Table 59-2 `AuthenticationCertificate` message parameters

Name	ID	Type	#	Notes
AuthenticationCertificate	0	blob	1	Accessory's X.509 certificate

59.1.3 RequestAuthenticationChallengeResponse

Source	ID
Device	0xAA02

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-3 `RequestAuthenticationChallengeResponse` message parameters

Name	ID	Type	#	Notes
AuthenticationChallenge	0	blob	1	Random #

59.1.4 AuthenticationResponse

Source	ID
Accessory	0xAA03

Table 59-4 `AuthenticationResponse` message parameters

Name	ID	Type	#	Notes
AuthenticationResponse	0	blob	1	Computed challenge response

59.1.5 AuthenticationFailed

Source	ID
Device	0xAA04

Table 59-5 `AuthenticationFailed` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.1.6 AuthenticationSucceeded

Source	ID
Device	0xAA05

Table 59-6 `AuthenticationResponseSucceeded` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.1.7 AccessoryAuthenticationSerialNumber

Source	ID
Accessory	0xAA06

Table 59-7 `AccessoryAuthenticationSerialNumber` message parameters

Name	ID	Type	#	Notes
AuthenticationSerialNumber	0	blob	1	Accessory's X.509 certificate serial number

59.2 Accessory Identification

For more information, see Accessory Identification (page 265).

59.2.1 StartIdentification

Source	ID
Device	0x1D00

Table 59-8 `StartIdentification` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.2.2 IdentificationInformation

Source	ID
Accessory	0x1D01

Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-9 `IdentificationInformation` message parameters

Name	ID	Type	#	Notes
Name	0	utf8	1	Must match the accessory's markings and packaging. A blank string is not allowed.
ModelIdentifier	1	utf8	1	Must match the accessory's markings and packaging. A blank string is not allowed.
Manufacturer	2	utf8	1	Must match the accessory's markings and packaging. A blank string is not allowed.
SerialNumber	3	utf8	1	Must match the accessory's markings and packaging. A blank string is not allowed.
FirmwareVersion	4	utf8	1	Must uniquely reflect the current revision of the accessory's firmware. A blank string is not allowed.
HardwareVersion	5	utf8	1	Must uniquely reflect the current revision of the accessory's hardware. A blank string is not allowed.
MessagesSentByAccessory	6	blob	1	The exhaustive set of unique messages that this accessory will send. This set is expressed as an array of uint16 message identifiers (uint16[]).
MessagesReceivedFromDevice	7	blob	1	The exhaustive set of unique messages that this accessory expects to receive. This set is expressed as an array of uint16 message identifiers (uint16[]).
PowerProvidingCapability	8	enum	1	See Table 59-10 (page 810). This must be 'None' if the accessory does not provide power to the Apple device.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
MaximumCurrentDrawnFromDevice	9	uint16	1	Maximum current drawn by accessory from Accessory Power pin in mA. This must be 0 if accessory does not draw power from the Apple device.
SupportedExternalAccessoryProtocol	10	group	0+	See Table 59-11 (page 810)
AppMatchTeamID	11	utf8	0/1	See App Match (page 340)
CurrentLanguage	12	utf8	1	The accessory's current active language setting. Must be one of the supported languages.
SupportedLanguage	13	utf8	1+	A language supported by the accessory. Use the ISO 639-1 designation unless it is not available, in which case use the ISO 639-2 designation. For a complete list of ISO 639-1 and ISO 639-2 codes, see http://www.loc.gov/standards/iso639-2/php/English_list.php.
SerialTransportComponent	14	group	0/1	See Table 59-16 (page 812)
USBDeviceTransportComponent	15	group	0/1	See Table 59-13 (page 811)
USBHostTransportComponent	16	group	0/1	See Table 59-15 (page 812)
BluetoothTransportComponent	17	group	0+	See Table 59-17 (page 812)
iAP2HIDComponent	18	group	0+	See Table 59-18 (page 813)
VehicleInformationComponent	20	group	0/1	See Table 59-19 (page 813)
VehicleStatusComponent	21	group	0/1	See Table 59-21 (page 813)
LocationInformationComponent	22	group	0/1	See Table 59-22 (page 814)
USBHostHIDComponent	23	group	0+	See Table 59-23 (page 815)
WirelessCarPlayTransportComponent	24	group	0+	See Table 59-25 (page 816)
BluetoothHIDComponent	29	group	0/1	See Table 59-26 (page 816)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-10 `PowerProvidingCapability` enum

Value	Meaning
0	None
1	Reserved
2	Advanced

Table 59-11 `ExternalAccessoryProtocol` parameter group

Name	ID	Type	#	Notes
ExternalAccessoryProtocolIdentifier	0	uint8	1	All ExternalAccessoryProtocol identifiers must be unique
ExternalAccessoryProtocolName	1	utf8	1	
ExternalAccessoryProtocolMatchAction	2	enum	1	See Table 59-12 (page 810) and App Match (page 340)
NativeTransportComponentIdentifier	3	uint16	0/1	Must refer to the `TransportComponentIdentifier` of a declared Table 59-15 (page 812)
ExternalAccessoryProtocolCarPlay	4	none	0/1	Must only be set by a CarPlay accessory where this protocol is intended to match with an app that supports CarPlay.

Table 59-12 `MatchAction` enum

Value	Meaning
0	No Action. The device will not prompt the user to find a matching app, and there will not be a *Find App For This Accessory* button in *Settings* > *General* > *About* > *'Accessory Name'*
1	Optional Action. The device may prompt the user to find a matching app, and there will be a *Find App For This Accessory* button in *Settings* > *General* > *About* > *'Accessory Name'*. See App Match (page 340) for more information.
2	No Alert. The device will not prompt the user to find a matching app, but there will be a *Find App For This Accessory* button in *Settings* > *General* > *About* > *'Accessory Name'*. See App Match (page 340) for more information.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Meaning
3	No Communication Protocol. The protocol is not intended for communication and does not require StartExternalAccessoryProtocolSession (page 843) or StopExternalAccessoryProtocolSession (page 843). The device may prompt the user to find a matching app, and there will be a *Find App For This Accessory* button in *Settings* > *General* > *About* > *'Accessory Name'*. See App Match (page 340) for more information.

Table 59-13 `USBDeviceTransportComponent` parameter group

Name	ID	Type	#	Notes
TransportComponentIdentifier	0	uint16	1	All TransportComponent identifiers must be unique
TransportComponentName	1	utf8	1	
TransportSupportsiAP2Connection	2	none	0/1	
USBDeviceSupportedAudioSampleRate	3	enum	0+	See Table 59-14 (page 811)

Table 59-14 `USBDeviceModeAudioSampleRate` enum

Value	Meaning
0	8000 Hz
1	11025 Hz
2	12000 Hz
3	16000 Hz
4	22050 Hz
5	24000 Hz
6	32000 Hz
7	44100 Hz
8	48000 Hz

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-15 `USBHostTransportComponent` parameter group

Name	ID	Type	#	Notes
TransportComponentIdentifier	0	uint16	1	All TransportComponent identifiers must be unique
TransportComponentName	1	utf8	1	
TransportSupportsiAP2Connection	2	none	0/1	
USBHostTransportCarPlayInterfaceNumber	3	uint8	0/1	The accessory's NCM Control Interface number. See CarPlay (page 381), CarPlay over USB (page 388) for more details.
TransportSupportsCarPlay	4	none	0/1	

Table 59-16 `SerialTransportComponent` parameter group

Name	ID	Type	#	Notes
TransportComponentIdentifier	0	uint16	1	All TransportComponent identifiers must be unique
TransportComponentName	1	utf8	1	
TransportSupportsiAP2Connection	2	none	0/1	

Table 59-17 `BluetoothTransportComponent` parameter group

Name	ID	Type	#	Notes
TransportComponentIdentifier	0	uint16	1	All TransportComponent identifiers must be unique
TransportComponentName	1	utf8	1	
TransportSupportsiAP2Connection	2	none	0/1	
BluetoothTransportMediaAccessControlAddress	3	blob	1	A valid 6-byte IEEE EUI-48 identifier (uint8[6])

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-18 `iAP2HIDComponent` parameter group

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	All HIDComponentIdentifiers must be unique
HIDComponentName	1	utf8	1	
HIDComponentFunction	2	enum	1	See Table 59-24 (page 815)

Table 59-19 `VehicleInformationComponent` parameter group

Name	ID	Type	#	Notes
Identifier	0	uint16	1	All Identifiers must be unique.
Name	1	utf8	1	
EngineType	2	enum	0+	See Table 59-20 (page 813).
DisplayName	6	utf8	1	

Table 59-20 `EngineTypes` enum

Value	Meaning
0	Gasoline
1	Diesel
2	Electric
3	CNG

Table 59-21 `VehicleStatusComponent` parameter group

Name	ID	Type	#	Notes
Identifier	0	uint16	1	All Identifiers must be unique.
Name	1	utf8	1	
Range	3	none	0/1	
OutsideTemperature	4	none	0/1	
RangeWarning	6	none	0/1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-22 `LocationInformationComponent` parameter group

Name	ID	Type	#	Notes
Identifier	0	uint16	1	All Identifiers must be unique.
Name	1	utf8	1	
GlobalPositioningSystemFixData	17	none	0/1	If present, the accessory is capable of generating NMEA GPGGA sentences.
RecommendedMinimumSpecificGPSTransitData	18	none	0/1	If present, the accessory is capable of generating NMEA GPRMC sentences.
GPSSatellitesInView	19	none	0/1	If present, the accessory is capable of generating NMEA GPGSV sentences.
VehicleSpeedData	20	none	0/1	If present, the accessory is capable of generating NMEA PASCD sentences.
VehicleGyroData	21	none	0/1	If present, the accessory is capable of generating NMEA PAGCD sentences.
VehicleAccelerometerData	22	none	0/1	If present, the accessory is capable of generating NMEA PAACD sentences.
VehicleHeadingData	23	none	0/1	If present, the accessory is capable of generating NMEA GPHDT sentences.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-23 `USBHostHIDComponent` parameter group

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	All HIDComponentIdentifiers must be unique
HIDComponentName	1	utf8	1	
HIDComponentFunction	2	enum	1	See Table 59-24 (page 815)
USBHostTransportComponentIdentifier	3	uint16	1	Must refer to a Table 59-15 (page 812)
USBHostTransportInterfaceNumber	4	uint16	1	Must match the accessory's corresponding USB device interface descriptor. If more than one USBHostHIDComponent is present, the accessory must present multiple USB HID interfaces with unique interface numbers.

Table 59-24 `HIDComponentFunction` enum

Value	Meaning
0	Keyboard
1	Media Playback Remote
2	AssistiveTouch Pointer
3	Reserved
4	Gamepad (Form-Fitting)
6	Gamepad (Non Form-Fitting)
7	Assistive Switch Control
8	Headset

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-25 `WirelessCarPlayTransportComponent` parameter group

Name	ID	Type	#	Notes
TransportComponentIdentifier	0	uint16	1	All TransportComponent identifiers must be unique
TransportComponentName	1	utf8	1	
TransportSupportsiAP2Connection	2	none	1	
TransportSupportsCarPlay	4	none	1	

Table 59-26 `BluetoothHIDComponent` parameter group

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	All HIDComponentIdentifiers must be unique
HIDComponentName	1	utf8	1	
HIDComponentFunction	2	enum	1	See Table 59-24 (page 815)
BluetoothTransportComponentIdentifier	3	uint16	1	Must refer to a Table 59-17 (page 812)

59.2.3 IdentificationAccepted

Source	ID
Device	0x1D02

Table 59-27 `IdentificationAccepted` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.2.4 IdentificationRejected

Source	ID
Device	0x1D03

Table 59-28 `IdentificationRejected` message parameters

Name	ID	Type	#	Notes
Name	0	none	0/1	
ModelIdentifier	1	none	0/1	
Manufacturer	2	none	0/1	
SerialNumber	3	none	0/1	
FirmwareVersion	4	none	0/1	
HardwareVersion	5	none	0/1	
MessagesSentByAccessory	6	blob	0/1	The set of unsupported messages sent by the accessory. This set is expressed as an array of uint16 message identifiers (uint16[])
MessagesReceivedFromDevice	7	blob	0/1	The set of unsupported messages received from the device. This set is expressed as an array of uint16 message identifiers (uint16[])
PowerProvidingCapability	8	none	0/1	
MaximumCurrentDrawnFromDevice	9	none	0/1	
SupportedExternalAccessoryProtocol	10	none	0/1	One or more of the External Accessory Protocols is not supported by the device
AppMatchTeamID	11	none	0/1	
CurrentLanguage	12	none	0/1	
SupportedLanguage	13	none	0/1	One or more of the identified languages is not supported by the device

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
SerialTransportComponent	14	none	0/1	
USBDeviceTransportComponent	15	none	0/1	
USBHostTransportComponent	16	none	0/1	
BluetoothTransportComponent	17	none	0/1	One or more of the identified Bluetooth Transport components is not supported by the device
iAP2HIDComponent	18	none	0+	One or more of the identified HID components is not supported by the device
VehicleInformationComponent	20	none	0/1	One or more of the identified Vehicle Information components is not supported by the device
VehicleStatusComponent	21	none	0/1	One or more of the identified Vehicle Status components is not supported by the device
LocationInformationComponent	22	none	0/1	One or more of the identified Location Information components is not supported by the device
USBHostHIDComponent	23	none	0+	One or more of the identified HID components is not supported by the device
WirelessCarPlayTransportComponent	24	none	0+	One or more of the identified Wireless CarPlay Transport components is not supported by the device
BluetoothHIDComponent	29	none	0/1	One or more of the identified HID components is not supported by the device

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.2.5 CancelIdentification

Source	ID
Accessory	0x1D05

Table 59-29 `CancelIdentification` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.2.6 IdentificationInformationUpdate

Source	ID
Accessory	0x1D06

Table 59-30 `IdentificationInformationUpdate` message parameters

Name	ID	Type	#	Notes
Name	0	utf8	1	
ModelIdentifier	1	utf8	1	
Manufacturer	2	utf8	1	
SerialNumber	3	utf8	1	
FirmwareVersion	4	utf8	1	
HardwareVersion	5	utf8	1	
CurrentLanguage	6	utf8	1	

59.3 App Launch

For more information, see App Launch (page 338).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.3.1 RequestAppLaunch

Source	ID
Accessory	0xEA02

Table 59-31 `RequestAppLaunch` message parameters

Name	ID	Type	#	Notes
AppBundleID	0	utf8	1	uniform type identifer (UTI) in reverse-DNS format, e.g. `com.Ajax.hello`
AppLaunchMethod	1	enum	0/1	see Table 59-32 (page 820)

Table 59-32 `AppLaunchMethod` enum

Value	Meaning
0	Launch with user alert (default)
1	Launch without user alert

59.4 AssistiveTouch

For more information, see AssistiveTouch (page 342).

59.4.1 StartAssistiveTouch

Source	ID
Accessory	0x5400

Table 59-33 `StartAssistiveTouch` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.4.2 StopAssistiveTouch

Source	ID
Accessory	0x5401

Table 59-34 `StopAssistiveTouch` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.4.3 StartAssistiveTouchInformation

Source	ID
Accessory	0x5402

Table 59-35 `StartAssistiveTouchInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.4.4 AssistiveTouchInformation

Source	ID
Device	0x5403

Table 59-36 `AssistiveTouchInformation` message parameters

Name	ID	Type	#	Notes
IsEnabled	0	bool	1	

59.4.5 StopAssistiveTouchInformation

Source	ID
Accessory	0x5404

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-37 `StopAssistiveTouchInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.5 Bluetooth Connection

For more information, see Bluetooth Connection (page 372).

59.5.1 BluetoothComponentInformation

Source	ID
Accessory	0x4E01

Table 59-38 `BluetoothComponentInformation` message parameters

Name	ID	Type	#	Notes
BluetoothComponentStatus	0	group	0+	See Table 59-39 (page 822)

Table 59-39 `BluetoothComponentStatus` parameter group

Name	ID	Type	#	Notes
ComponentIdentifier	0	uint16	1	See Table 59-17 (page 812)
ComponentEnabled	1	bool	1	`true` if the Bluetooth component is ready for connections to the Apple device

59.5.2 StartBluetoothConnectionUpdates

Source	ID
Accessory	0x4E03

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-40 `StartBluetoothConnectionUpdates` message parameters

Name	ID	Type	#	Notes
BluetoothTransportComponentIdentifier	0	uint16	1+	See Table 59-17 (page 812)

59.5.3 BluetoothConnectionUpdate

Source	ID
Device	0x4E04

Table 59-41 `BluetoothConnectionUpdate` message parameters

Name	ID	Type	#	Notes
BluetoothTransportComponentIdentifier	0	uint16	1	See Table 59-17 (page 812)
ConnectedBluetoothProfiles	1	group	0/1	See Table 59-42 (page 823)

Table 59-42 `BluetoothComponentProfiles` parameter group

Name	ID	Type	#	Notes
BluetoothHandsFree	0	none	0/1	
BluetoothPhoneBookAccess	1	none	0/1	
BluetoothAudioVideoRemoteControl	3	none	0/1	
BluetoothAdvancedAudioDistribution	4	none	0/1	
BluetoothHumanInterfaceDevice	5	none	0/1	
BluetoothiAP2Link	7	none	0/1	
BluetoothPersonalAreaNetworkAccessPoint	8	none	0/1	Apple device is in the Access Point role
BluetoothMessageAccess	9	none	0/1	
BluetoothPersonalAreaNetworkClient	12	none	0/1	Apple device is in the Client role

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.5.4 StopBluetoothConnectionUpdates

Source	ID
Accessory	0x4E05

Table 59-43 `StopBluetoothConnectionUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.6 Communications

For more information, see Communications (page 517).

59.6.1 StartCallStateUpdates

Source	ID
Accessory	0x4154

Table 59-44 `StartCallStateUpdates` message parameters

Name	ID	Type	#	Notes
RemoteID	0	none	1	
DisplayName	1	none	1	
Status	2	none	1	
Direction	3	none	1	
CallUUID	4	none	1	
AddressBookID	6	none	0/1	
Label	7	none	0/1	
Service	8	none	1	
IsConferenced	9	none	1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
ConferenceGroup	10	none	1	
DisconnectReason	11	none	0/1	

59.6.2 CallStateUpdate

Source	ID
Device	0x4155

Table 59-45 `CallStateUpdate` message parameters

Name	ID	Type	#	Notes
RemoteID	0	utf8	0/1	Remote phone number or email
DisplayName	1	utf8	0/1	Caller's display name on phone In Contacts: John Smith Not in Contacts: (408) 996-1010
Status	2	enum	1	See Table 59-46 (page 826)
Direction	3	enum	0/1	See Table 59-47 (page 826)
CallUUID	4	utf8	0/1	
AddressBookID	6	utf8	0/1	
Label	7	utf8	0/1	Caller's label In Contacts: mobile, work, home Not in Contacts: San Jose, CA
Service	8	enum	0/1	See Table 59-48 (page 826)
IsConferenced	9	bool	0/1	Whether this call is part of a conference or not
ConferenceGroup	10	uint8	0/1	Conference group number; Will only be sent if IsConferenced = 1
DisconnectReason	11	enum	0/1	See Table 59-49 (page 826); Will only be sent if Status = Disconnected

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-46 `CallStateUpdateStatus` enum

Value	Meaning
0	Disconnected
1	Sending
2	Ringing
3	Connecting
4	Active
5	Held
6	Disconnecting

Table 59-47 `CallStateUpdateDirection` enum

Value	Meaning
0	Unknown
1	Incoming
2	Outgoing

Table 59-48 `CallStateUpdateService` enum

Value	Meaning
0	Unknown
1	Telephony
2	FaceTimeAudio
3	FaceTimeVideo

Table 59-49 `CallStateUpdateDisconnectReason` enum

Value	Meaning
0	No Reason (Call Ended)
1	Call Declined

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Meaning
2	Call Failed

Table 59-50 `CallStateUpdateStatusLegacy` enum

Value	Meaning
0	Disconnected
1	Active
2	Held
3	Ringing/Sending

Table 59-51 `CallStateUpdateDirectionLegacy` enum

Value	Meaning
0	Incoming
1	Outgoing
2	Unknown

59.6.3 StopCallStateUpdates

Source	ID
Accessory	0x4156

Table 59-52 `StopCallStateUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.6.4 StartCommunicationsUpdates

Source	ID
Accessory	0x4157

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-53 `StartCommunicationsUpdates` message parameters

Name	ID	Type	#	Notes
SignalStrength	0	none	0/1	
RegistrationStatus	1	none	0/1	
AirplaneModeStatus	2	none	0/1	
CarrierName	4	none	0/1	
CellularSupported	5	none	0/1	
TelephonyEnabled	6	none	0/1	
FaceTimeAudioEnabled	7	none	0/1	
FaceTimeVideoEnabled	8	none	0/1	
MuteStatus	9	none	0/1	
CurrentCallCount	10	none	0/1	
NewVoicemailCount	11	none	0/1	
InitiateCallAvailable	12	none	0/1	
EndAndAcceptAvailable	13	none	0/1	
HoldAndAcceptAvailable	14	none	0/1	
SwapAvailable	15	none	0/1	
MergeAvailable	16	none	0/1	
HoldAvailable	17	none	0/1	

59.6.5 CommunicationsUpdate

Source	ID
Device	0x4158

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-54 `CommunicationsUpdate` message parameters

Name	ID	Type	#	Notes
SignalStrength	0	enum	0/1	See Table 59-55 (page 829); Will not be sent if CellularSupported is false
RegistrationStatus	1	enum	0/1	See Table 59-56 (page 830); Will not be sent if CellularSupported is false
AirplaneModeStatus	2	bool	0/1	
CarrierName	4	utf8	0/1	Will not be sent if CellularSupported is false
CellularSupported	5	bool	0/1	
TelephonyEnabled	6	bool	0/1	
FaceTimeAudioEnabled	7	bool	0/1	
FaceTimeVideoEnabled	8	bool	0/1	
MuteStatus	9	bool	0/1	
CurrentCallCount	10	uint8	0/1	
NewVoicemailCount	11	uint8	0/1	
InitiateCallAvailable	12	bool	0/1	Allowed to initiate a call or add a call if one is already active
EndAndAcceptAvailable	13	bool	0/1	
HoldAndAcceptAvailable	14	bool	0/1	
SwapAvailable	15	bool	0/1	
MergeAvailable	16	bool	0/1	
HoldAvailable	17	bool	0/1	You cannot place a call on hold if this is false. You can un-hold a call if the status is held even if this is false.

Table 59-55 `CommunicationsUpdateSignalStrength` enum

Value	Meaning
0	0 bars

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Meaning
1	1 bar
2	2 bars
3	3 bars
4	4 bars
5	5 bars

Table 59-56 `CommunicationsUpdateRegistrationStatus` enum

Value	Meaning
0	Unknown
1	Not Registered
2	Searching
3	Denied
4	Registered Home
5	Registered Roaming
6	Emergency Calls Only

59.6.6 StopCommunicationsUpdates

Source	ID
Accessory	0x4159

Table 59-57 `StopCommunicationsUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.6.7 InitiateCall

Source	ID
Accessory	0x415A

Table 59-58 `InitiateCall` message parameters

Name	ID	Type	#	Notes
Type	0	enum	1	See Table 59-59 (page 831)
DestinationID	1	utf8	0/1	Required for Destination call. Number to call Format: 4089961010 Format: +14089961010 Format: +4408442090611
Service	2	enum	0/1	Required for Destination call. See Table 59-60 (page 831)
AddressBookID	3	utf8	0/1	

Table 59-59 `InitiateCallType` enum

Value	Meaning
0	Destination
1	Voicemail
2	Redial

Table 59-60 `InitiateCallService` enum

Value	Meaning
1	Telephony
2	FaceTimeAudio
3	FaceTimeVideo

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.6.8 AcceptCall

Source	ID
Accessory	0x415B

Table 59-61 `AcceptCall` message parameters

Name	ID	Type	#	Notes
AcceptAction	0	enum	1	See Table 59-62 (page 832)
CallUUID	1	utf8	0/1	UUID of call to accept

Table 59-62 `AcceptCallAcceptAction` enum

Value	Meaning
0	Accept/HoldAndAccept
1	EndAndAccept

59.6.9 EndCall

Source	ID
Accessory	0x415C

Table 59-63 `EndCall` message parameters

Name	ID	Type	#	Notes
EndAction	0	enum	1	See Table 59-64 (page 832)
CallUUID	1	utf8	0/1	UUID of call to end

Table 59-64 `EndCallEndAction` enum

Value	Meaning
0	End/Decline
1	EndAll

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.6.10 SwapCalls

Source	ID
Accessory	0x415D

Table 59-65 `SwapCalls` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.6.11 MergeCalls

Source	ID
Accessory	0x415E

Table 59-66 `MergeCalls` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.6.12 HoldStatusUpdate

Source	ID
Accessory	0x415F

Table 59-67 `HoldStatusUpdate` message parameters

Name	ID	Type	#	Notes
HoldStatus	0	bool	1	
CallUUID	1	utf8	0/1	UUID of call to update hold status

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.6.13 MuteStatusUpdate

Source	ID
Accessory	0x4160

Table 59-68 `MuteStatusUpdate` message parameters

Name	ID	Type	#	Notes
MuteStatus	0	bool	1	

59.6.14 SendDTMF

Source	ID
Accessory	0x4161

Table 59-69 `SendDTMF` message parameters

Name	ID	Type	#	Notes
Tone	0	enum	1	See Table 59-70 (page 834)
CallUUID	1	utf8	0/1	UUID of call to play tone

Table 59-70 `SendDTMFTone` enum

Value	Meaning
0	Number 0
1	Number 1
2	Number 2
3	Number 3
4	Number 4
5	Number 5
6	Number 6

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Meaning
7	Number 7
8	Number 8
9	Number 9
10	Star (*)
11	Pound (#)

59.6.15 StartListUpdates

Source	ID
Accessory	0x4170

Table 59-71 `StartListUpdates` message parameters

Name	ID	Type	#	Notes
RecentsListProperties	1	group	0/1	See Table 59-72 (page 835); Sending this parameter will cause the device to send RecentsListAvailable, RecentsListCount, and RecentsList
RecentsListMax	3	uint16	0/1	Max entries to send; 0 = no limit
RecentsListCombine	4	bool	0/1	Combine calls to the same person; Default = 1
FavoritesListProperties	6	group	0/1	See Table 59-73 (page 836); Sending this parameter will cause the device to send FavoritesListAvailable, FavoritesListCount, and FavoritesList
FavoritesListMax	8	uint16	0/1	Max entries to send; 0 = no limit

Table 59-72 `RecentsListProperties` parameter group

Name	ID	Type	#	Notes
Index	0	none	1	
RemoteID	1	none	1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
DisplayName	2	none	1	
Label	3	none	0/1	
AddressBookID	4	none	0/1	
Service	5	none	1	
Type	6	none	1	
UnixTimestamp	7	none	0/1	
Duration	8	none	0/1	
Occurrences	9	none	1	

Table 59-73 `FavoritesListProperties` parameter group

Name	ID	Type	#	Notes
Index	0	none	1	
RemoteID	1	none	1	
DisplayName	2	none	1	
Label	3	none	0/1	
AddressBookID	4	none	0/1	
Service	5	none	1	

59.6.16 ListUpdate

Source	ID
Device	0x4171

Table 59-74 `ListUpdate` message parameters

Name	ID	Type	#	Notes
RecentsListAvailable	0	bool	0/1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
RecentsList	1	group	0/1	See Table 59-75 (page 837)
RecentsListCount	2	uint16	0/1	Number of entries in the list
FavoritesListAvailable	5	bool	0/1	
FavoritesList	6	group	0/1	See Table 59-76 (page 837)
FavoritesListCount	7	uint16	0/1	Number of entries in the list

Table 59-75 `RecentsList` parameter group

Name	ID	Type	#	Notes
Index	0	uint16	1	Request index of list entry
RemoteID	1	utf8	1	Remote phone number or email
DisplayName	2	utf8	1	
Label	3	utf8	0/1	Label if in contacts, location if not
AddressBookID	4	utf8	0/1	
Service	5	enum	1	See Table 59-77 (page 838)
Type	6	enum	1	See Table 59-78 (page 838)
UnixTimestamp	7	uint64	0/1	Unix timestamp in UTC
Duration	8	uint32	0/1	In seconds. Only if occurrences = 1
Occurrences	9	uint8	1	

Table 59-76 `FavoritesList` parameter group

Name	ID	Type	#	Notes
Index	0	uint16	1	Request index of list entry
RemoteID	1	utf8	1	Remote phone number or email
DisplayName	2	utf8	1	
Label	3	utf8	0/1	Label if in contacts, location if not

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
AddressBookID	4	utf8	0/1	
Service	5	enum	1	See Table 59-77 (page 838)

Table 59-77 `ListUpdateService` enum

Value	Meaning
0	Unknown
1	Telephony
2	FaceTimeAudio
3	FaceTimeVideo

Table 59-78 `ListUpdateRecentsListType` enum

Value	Meaning
0	Unknown
1	Incoming
2	Outgoing
3	Missed

59.6.17 StopListUpdates

Source	ID
Accessory	0x4172

Table 59-79 `StopListUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.7 Device Authentication

For more information, see Device Authentication (page 523).

59.7.1 RequestDeviceAuthenticationCertificate

Source	ID
Accessory	0xAA10

Table 59-80 `RequestDeviceAuthenticationCertificate` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.7.2 DeviceAuthenticationCertificate

Source	ID
Device	0xAA11

Table 59-81 `DeviceAuthenticationCertificate` message parameters

Name	ID	Type	#	Notes
DeviceAuthenticationCertificate	0	blob	1	Device's X.509 certificate

59.7.3 RequestDeviceAuthenticationChallengeResponse

Source	ID
Accessory	0xAA12

Table 59-82 `RequestDeviceAuthenticationChallengeResponse` message parameters

Name	ID	Type	#	Notes
DeviceAuthenticationChallenge	0	blob	1	Random #

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.7.4 DeviceAuthenticationResponse

Source	ID
Device	0xAA13

Table 59-83 `DeviceAuthenticationResponse` message parameters

Name	ID	Type	#	Notes
DeviceAuthenticationResponse	0	blob	1	Computed challenge response

59.7.5 DeviceAuthenticationFailed

Source	ID
Accessory	0xAA14

Table 59-84 `DeviceAuthenticationFailed` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.7.6 DeviceAuthenticationSucceeded

Source	ID
Accessory	0xAA15

Table 59-85 `DeviceAuthenticationResponseSucceeded` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.8 Device Notifications

For more information, see Device Notifications (page 526).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.8.1 DeviceInformationUpdate

Source	ID
Device	0x4E09

Table 59-86 `DeviceInformationUpdate` message parameters

Name	ID	Type	#	Notes
DeviceName	0	utf8	0/1	Device name as shown in Settings, i.e. "John Doe's iPhone"

59.8.2 DeviceLanguageUpdate

Source	ID
Device	0x4E0A

Table 59-87 `DeviceLanguageUpdate` message parameters

Name	ID	Type	#	Notes
DeviceLanguage	0	utf8	0/1	ISO 639-1 or ISO 639-2 designation for the current device language. For a complete list of ISO 639-1 and ISO 639-2 codes, see http://www.loc.gov/standards/iso639-2/php/English_list.php

59.8.3 DeviceTimeUpdate

Source	ID
Device	0x4E0B

Table 59-88 `DeviceTimeUpdate` message parameters

Name	ID	Type	#	Notes
SecondsSinceReferenceDate	0	uint64	1	Time interval in seconds since reference date (Jan 1, 1970, GMT)
TimeZoneOffsetMinutes	1	int16	1	Difference in minutes between the device time zone and GMT

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
DaylightSavingsOffsetMinutes	2	int8	1	Daylight savings time offset in minutes

59.8.4 DeviceUUIDUpdate

Source	ID
Device	0x4E0C

Table 59-89 `DeviceUUIDUpdate` message parameters

Name	ID	Type	#	Notes
UUID	0	utf8	1	This UUID will be the same over all transports as long as one iAP accessory is connected. The UUID will be changed no less than one minute after all iAP accessories disconnect.

59.8.5 WirelessCarPlayUpdate

Source	ID
Device	0x4E0D

Table 59-90 `WirelessCarPlayUpdate` message parameters

Name	ID	Type	#	Notes
Status	0	enum	1	See Table 59-91 (page 842).

Table 59-91 `WirelessCarPlayUpdateStatus` enum

Value	Meaning
0	Unavailable
1	Available

59.9 External Accessory Protocol

For more information, see External Accessory Protocol (page 535).

59.9.1 StartExternalAccessoryProtocolSession

Source	ID
Device	0xEA00

Table 59-92 `StartExternalAccessoryProtocolSession` message parameters

Name	ID	Type	#	Notes
ExternalAccessoryProtocolIdentifier	0	uint8	1	See Table 59-11 (page 810)
ExternalAccessoryProtocolSessionIdentifier	1	uint16	1	

59.9.2 StopExternalAccessoryProtocolSession

Source	ID
Device	0xEA01

Table 59-93 `StopExternalAccessoryProtocolSession` message parameters

Name	ID	Type	#	Notes
ExternalAccessoryProtocolSessionIdentifier	0	uint16	1	

59.9.3 StatusExternalAccessoryProtocolSession

Source	ID
Accessory	0xEA03

Table 59-94 `StatusExternalAccessoryProtocolSession` message parameters

Name	ID	Type	#	Notes
ExternalAccessoryProtocolSessionIdentifier	0	uint16	1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
ExternalAccessoryProtocolSessionStatus	1	enum	1	See Table 59-95 (page 844).

Table 59-95 `ExternalAccessoryProtocolSessionStatus` enum

Value	Meaning
0	SessionStatusOK
1	SessionClose

59.10 Human Interface Device

For more information, see HID (Human Interface Device) (page 580).

59.10.1 StartHID

Source	ID
Accessory	0x6800

Table 59-96 `StartHID` message parameters

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	Must refer to an identified Table 59-18 (page 813).
VendorIdentifier	1	uint16	1	Must be assigned and registered by the USB-IF for the accessory manufacturer.
ProductIdentifier	2	uint16	1	Must be unique for each accessory made by the accessory manufacturer.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
LocalizedKeyboardCountryCode	3	uint8	0/1	Only required if the HID component is a localized (non-ANSI) keyboard. Must be drawn from the list of assigned country codes defined in *Device Class Definition for Human Interface Devices (HID) Version 1.11, section 6.2.1 HID Descriptor*.
HIDReportDescriptor	4	blob	1	HID Report Descriptor.

59.10.2 DeviceHIDReport

Source	ID
Device	0x6801

Table 59-97 `DeviceHIDReport` message parameters

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	Must refer to an identified Table 59-18 (page 813)
HIDReport	1	blob	1	HID Report

59.10.3 AccessoryHIDReport

Source	ID
Accessory	0x6802

Table 59-98 `AccessoryHIDReport` message parameters

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	Must refer to an identified Table 59-18 (page 813)
HIDReport	1	blob	1	HID Report

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.10.4 StopHID

Source	ID
Accessory	0x6803

Table 59-99 `StopHID` message parameters

Name	ID	Type	#	Notes
HIDComponentIdentifier	0	uint16	1	Must refer to an identified Table 59-18 (page 813)

59.10.5 StartNativeHID

Source	ID
Device	0x6806

Table 59-100 `StartNativeHID` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.11 Location

For more information, see Location Information (page 637).

59.11.1 StartLocationInformation

Source	ID
Device	0xFFFA

Table 59-101 `StartLocationInformation` message parameters

Name	ID	Type	#	Notes
GlobalPositioningSystemFixData	1	none	0/1	If present, the accessory may provide NMEA GPGGA sentences.
RecommendedMinimumSpecificGPSTransitData	2	none	0/1	If present, the accessory may provide NMEA GPRMC sentences.
GPSSatellitesInView	3	none	0/1	If present, the accessory may provide NMEA GPGSV sentences.
VehicleSpeedData	4	none	0/1	If present, the accessory may provide NMEA PASCD sentences.
VehicleGyroData	5	none	0/1	If present, the accessory may provide NMEA PAGCD sentences.
VehicleAccelerometerData	6	none	0/1	If present, the accessory may provide NMEA PAACD sentences.
VehicleHeadingData	7	none	0/1	If present, the accessory may provide NMEA GPHDT sentences.

59.11.2 LocationInformation

Source	ID
Accessory	0xFFFB

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-102 `LocationInformation` message parameters

Name	ID	Type	#	Notes
NMEASentence	0	utf8	1+	One NMEA Sentence of the type(s) specified by StartLocationInformation (page 846). Multiple NMEA sentences may be sent by including multiple NMEASentence parameters in the same message.

59.11.3 StopLocationInformation

Source	ID
Device	0xFFFC

Table 59-103 `StopLocationInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.12 Media Library Access

For more information, see Media Library Access (page 649).

59.12.1 StartMediaLibraryInformation

Source	ID
Accessory	0x4C00

Table 59-104 `StartMediaLibraryInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

Copyright © 2015 Apple Inc. All Rights Reserved.

59.12.2 MediaLibraryInformation

Source	ID
Device	0x4C01

Table 59-105 `MediaLibraryInformation` message parameters

Name	ID	Type	#	Notes
MediaLibraryInformation	0	group	0+	See Table 59-106 (page 849)

Table 59-106 `MediaLibraryInformation` parameter group

Name	ID	Type	#	Notes
MediaLibraryName	0	utf8	1	
MediaLibraryUniqueIdentifier	1	utf8	1	
MediaLibraryType	2	enum	1	See Table 59-107 (page 849)

Table 59-107 `MediaLibraryType` enum

Value	Meaning
0	Local device library
2	iTunes Radio library

59.12.3 StopMediaLibraryInformation

Source	ID
Accessory	0x4C02

Table 59-108 `StopMediaLibraryInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.12.4 StartMediaLibraryUpdates

Source	ID
Accessory	0x4C03

Table 59-109 `StartMediaLibraryUpdates` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1	
LastKnownMediaLibraryRevision	1	utf8	0/1	If no LastKnownMediaLibraryRevision is included, a full database update will be sent.
MediaItemProperties	2	group	0/1	See Table 59-110 (page 850)
MediaPlaylistProperties	3	group	0/1	See Table 59-111 (page 851)
MediaLibraryUpdateProgress	4	none	0/1	
MediaLibraryIsHidingRemoteItems	5	none	0/1	
PlayAllSongsCapable	6	none	0/1	

Table 59-110 `MediaItemProperties` parameter group

Name	ID	Type	#	Notes
MediaItemPropertyPersistentIdentifier	0	none	1	
MediaItemPropertyTitle	1	none	0/1	
MediaItemPropertyMediaType	2	none	0/1	
MediaItemPropertyRating	3	none	0/1	
MediaItemPropertyPlaybackDurationInMilliseconds	4	none	0/1	
MediaItemPropertyAlbumPersistentIdentifer	5	none	0/1	
MediaItemPropertyAlbumTitle	6	none	0/1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
MediaItemPropertyAlbumTrackNumber	7	none	0/1	
MediaItemPropertyAlbumTrackCount	8	none	0/1	
MediaItemPropertyAlbumDiscNumber	9	none	0/1	
MediaItemPropertyAlbumDiscCount	10	none	0/1	
MediaItemPropertyArtistPersistentIdentifier	11	none	0/1	
MediaItemPropertyArtist	12	none	0/1	
MediaItemPropertyAlbumArtistPersistentIdentifier	13	none	0/1	
MediaItemPropertyAlbumArtist	14	none	0/1	
MediaItemPropertyGenrePersistentIdentifier	15	none	0/1	
MediaItemPropertyGenre	16	none	0/1	
MediaItemPropertyComposerPersistentIdentifier	17	none	0/1	
MediaItemPropertyComposer	18	none	0/1	
MediaItemPropertyIsPartOfCompilation	19	none	0/1	
MediaItemPropertyIsResidentOnDevice	25	none	0/1	
MediaItemPropertyChapterCount	27	none	0/1	

Table 59-111 `MediaPlaylistProperties` parameter group

Name	ID	Type	#	Notes
MediaPlaylistPropertyPersistentIdentifer	0	none	1	
MediaPlaylistPropertyName	1	none	0/1	
MediaPlaylistPropertyParentPersistentIdentifer	2	none	0/1	
MediaPlaylistPropertyIsGeniusMix	3	none	0/1	
MediaPlaylistPropertyIsFolder	4	none	0/1	
MediaPlaylistPropertyContainedMediaItemsFileTransferIdentifier	5	none	0/1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
MediaPlaylistPropertyIsiTunesRadioStation	6	none	0/1	

59.12.5 MediaLibraryUpdate

Source	ID
Device	0x4C04

Table 59-112 `MediaLibraryUpdate` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1	
MediaLibraryRevision	1	utf8	0/1	
MediaItem	2	group	0+	See Table 59-113 (page 853). Every MediaItem in a MediaLibraryUpdate message will contain a MediaItemPersistentIdentifier parameter. Note that not all MediaItem parameters are supported by the Media Library feature. See Table 59-110 (page 850).
MediaPlaylist	3	group	0+	See Table 59-114 (page 855). Every MediaPlaylist in a MediaLibraryUpdate message will contain a MediaPlaylistPersistentIdentifer parameter.
MediaItemDeletePersistentIdentifier	4	uint64	0+	PersistentIdentifier of the media item that has been deleted.
MediaPlaylistDeletePersistentIdentifier	5	uint64	0+	PersistentIdentifier of the playlist that has been deleted.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
MediaLibraryReset	6	none	0/1	If present, the accessory must delete all data pertaining to the specified media library before applying new updates.
MediaLibraryUpdateProgress	7	uint8	0/1	If present, indicates percentage completion for the current set of media library updates (0-100).
MediaLibraryIsHidingRemoteItems	8	bool	0/1	If present, the Apple device's UI is only presenting media items that are resident on the device to the user. Must be requested with MediaItemIsResidentOnDevice set to 1.
PlayAllSongsCapable	9	bool	0/1	If present, the Apple device is capable of playing all songs. See PlayMediaLibrarySpecial (page 858).

Table 59-113 `MediaItem` parameter group

Name	ID	Type	#	Notes
MediaItemPersistentIdentifier	0	uint64	0/1	
MediaItemTitle	1	utf8	0/1	
MediaItemMediaType	2	enum	0+	See Table 59-115 (page 855). Note that it is possible for some media items to be associated with multiple media types.
MediaItemRating	3	uint8	0/1	Shall be 0...5
MediaItemPlaybackDurationInMilliseconds	4	uint32	0/1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
MediaItemAlbumPersistentIdentifer	5	uint64	0/1	
MediaItemAlbumTitle	6	utf8	0/1	
MediaItemAlbumTrackNumber	7	uint16	0/1	
MediaItemAlbumTrackCount	8	uint16	0/1	
MediaItemAlbumDiscNumber	9	uint16	0/1	
MediaItemAlbumDiscCount	10	uint16	0/1	
MediaItemArtistPersistentIdentifier	11	uint64	0/1	
MediaItemArtist	12	utf8	0/1	
MediaItemAlbumArtistPersistentIdentifier	13	uint64	0/1	
MediaItemAlbumArtist	14	utf8	0/1	
MediaItemGenrePersistentIdentifier	15	uint64	0/1	
MediaItemGenre	16	utf8	0/1	
MediaItemComposerPersistentIdentifier	17	uint64	0/1	
MediaItemComposer	18	utf8	0/1	
MediaItemIsPartOfCompilation	19	bool	0/1	
MediaItemIsLikeSupported	21	bool	0/1	
MediaItemIsBanSupported	22	bool	0/1	
MediaItemIsLiked	23	bool	0/1	
MediaItemIsBanned	24	bool	0/1	
MediaItemIsResidentOnDevice	25	bool	0/1	
MediaItemArtworkFileTransferIdentifier	26	uint8	0/1	
MediaItemChapterCount	27	uint16	0/1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-114 `MediaPlaylist` parameter group

Name	ID	Type	#	Notes
MediaPlaylistPersistentIdentifer	0	uint64	0/1	
MediaPlaylistName	1	utf8	0/1	
MediaPlaylistParentPersistentIdentifer	2	uint64	0/1	
MediaPlaylistIsGeniusMix	3	bool	0/1	
MediaPlaylistIsFolder	4	bool	0/1	
MediaPlaylistContainedMediaItemsFileTransferIdentifier	5	uint8	0/1	Note that it is entirely possible for the playlist content file transfer to be 0 length indicating a playlist with 0 contained items
MediaPlaylistIsiTunesRadioStation	6	bool	0/1	

Note: If the MediaPlaylist parameter group does not contain a MediaPlaylistContainedMediaItemsFileTransferIdentifier parameter, the playlist represents an endless playlist type such as Genius Mix or iTunes Radio station.

Table 59-115 `MediaType` enum

Value	Meaning
0	MediaTypeMusic
1	MediaTypePodcast
2	MediaTypeAudioBook
3	MediaTypeiTunesU

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.12.6 StopMediaLibraryUpdates

Source	ID
Accessory	0x4C05

Table 59-116 `StopMediaLibraryUpdate` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1+	

59.12.7 PlayMediaLibraryCurrentSelection

Source	ID
Accessory	0x4C06

Table 59-117 `PlayMediaLibraryCurrentSelection` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1	See MediaLibraryInformation (page 849)

59.12.8 PlayMediaLibraryItems

Source	ID
Accessory	0x4C07

Table 59-118 `PlayMediaLibraryItems` message parameters

Name	ID	Type	#	Notes
ItemsPersistentIdentifiers	0	blob	1	Array of ordered uint64 MediaItemPersistentIdentifiers (uint64[])

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
ItemsStartingIndex	1	uint32	0/1	The index of the first item in PersistentIdentifiers to start playback with. If this parameter is invalid or not present the starting index defaults to 0.
MediaLibraryUniqueIdentifier	2	utf8	1	See MediaLibraryInformation (page 849)

59.12.9 PlayMediaLibraryCollection

Source	ID
Accessory	0x4C08

Table 59-119 `PlayMediaLibraryCollection` message parameters

Name	ID	Type	#	Notes
CollectionPersistentIdentifier	0	uint64	1	
CollectionType	1	enum	1	See Table 59-120 (page 857)
CollectionStartingIndex	2	uint32	0/1	The index of the first item in the collection to start playback with. If this parameter is invalid or not present the starting index defaults to 0.
MediaLibraryUniqueIdentifier	3	utf8	1	See MediaLibraryInformation (page 849)

Note: CollectionStartingIndex parameter must only be used for Playlist collection types. Behavior when used with other collection types is undefined.

Table 59-120 `MediaLibraryCollectionType` enum

Value	Meaning
0	Playlist
1	Artist

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Meaning
2	Album
3	Album Artist
4	Genre
5	Composer

59.12.10 PlayMediaLibrarySpecial

Source	ID
Accessory	0x4C09

Table 59-121 `PlayMediaLibrarySpecial` message parameters

Name	ID	Type	#	Notes
MediaLibraryUniqueIdentifier	0	utf8	1	
AllSongs	1	none	0/1	

59.13 Now Playing

For more information, see Now Playing Updates (page 657) feature.

59.13.1 StartNowPlayingUpdates

Source	ID
Accessory	0x5000

Table 59-122 `StartNowPlayingUpdates` message parameters

Name	ID	Type	#	Notes
MediaItemAttributes	0	group	0/1	See Table 59-123 (page 859)

Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
PlaybackAttributes	1	group	0/1	See Table 59-124 (page 860)

Table 59-123 `StartNowPlayingMediaItemAttributes` parameter group

Name	ID	Type	#	Notes
MediaItemPersistentIdentifier	0	none	0/1	
MediaItemTitle	1	none	0/1	
MediaItemPlaybackDurationInMilliseconds	4	none	0/1	
MediaItemAlbumTitle	6	none	0/1	
MediaItemAlbumTrackNumber	7	none	0/1	
MediaItemAlbumTrackCount	8	none	0/1	
MediaItemAlbumDiscNumber	9	none	0/1	
MediaItemAlbumDiscCount	10	none	0/1	
MediaItemArtist	12	none	0/1	
MediaItemGenre	16	none	0/1	
MediaItemComposer	18	none	0/1	
MediaItemIsLikeSupported	21	none	0/1	
MediaItemIsBanSupported	22	none	0/1	
MediaItemIsLiked	23	none	0/1	
MediaItemIsBanned	24	none	0/1	
MediaItemArtworkFileTransferIdentifier	26	none	0/1	The device may not actually initiate the artwork file transfer if the now playing information changes, including the artwork, before the transfer has started.
MediaItemChapterCount	27	none	0/1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-124 `StartNowPlayingPlaybackAttributes` parameter group

Name	ID	Type	#	Notes
PlaybackStatus	0	none	0/1	
PlaybackElapsedTimeInMilliseconds	1	none	0/1	
PlaybackQueueIndex	2	none	0/1	
PlaybackQueueCount	3	none	0/1	
PlaybackQueueChapterIndex	4	none	0/1	
PlaybackShuffleMode	5	none	0/1	
PlaybackRepeatMode	6	none	0/1	
PlaybackAppName	7	none	1	
PlaybackMediaLibraryUniqueIdentifier	8	none	0/1	
PBiTunesRadioAd	9	none	0/1	
PBiTunesRadioStationName	10	none	0/1	
PBiTunesRadioStationMediaPlaylistPersistentID	11	none	0/1	
PlaybackSpeed	12	none	0/1	
SetElapsedTimeAvailable	13	none	0/1	
PlaybackQueueListAvail	14	none	0/1	
PlaybackQueueListTransferID	15	none	0/1	
PlaybackAppBundleID	16	none	1	

59.13.2 NowPlayingUpdate

Source	ID
Device	0x5001

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-125 `NowPlayingUpdate` message parameters

Name	ID	Type	#	Notes
MediaItemAttributes	0	group	0/1	See Table 59-113 (page 853). Note that not all MediaItem parameters are supported by the Now Playing feature. See Table 59-123 (page 859)
PlaybackAttributes	1	group	0/1	See Table 59-126 (page 861)

Table 59-126 `PlaybackAttributes` parameter group

Name	ID	Type	#	Notes
PlaybackStatus	0	enum	0/1	See Table 59-127 (page 862)
PlaybackElapsedTimeInMilliseconds	1	uint32	0/1	
PlaybackQueueIndex	2	uint32	0/1	
PlaybackQueueCount	3	uint32	0/1	
PlaybackQueueChapterIndex	4	uint32	0/1	
PlaybackShuffleMode	5	enum	0/1	See Table 59-128 (page 862)
PlaybackRepeatMode	6	enum	0/1	See Table 59-129 (page 863)
PlaybackAppName	7	utf8	0/1	The name of the Now Playing app
PBMediaLibraryUniqueIdentifier	8	utf8	0/1	
PBiTunesRadioAd	9	bool	0/1	Now playing an iTunes Radio ad
PBiTunesRadioStationName	10	utf8	0/1	Name of the now playing iTunes Radio station
PBiTunesRadioStationMediaPlaylistPersistentID	11	uint64	0/1	MediaPlaylistPersistentIdentifer for the now playing iTunes Radio station

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
PlaybackSpeed	12	uint16	0/1	Fixed decimal number with 2 digits after the decimal (i.e. 100 = 1.00x, 150 = 1.50x, 25 = 0.25x, etc.). A value of 0 when PlaybackStatus is not Stopped indicates Now Playing app is not providing a value.
SetElapsedTimeAvailable	13	bool	0/1	
PlaybackQueueListAvail	14	bool	0/1	Playback queue contents are available. If false, then playback queue must be cleared
PlaybackQueueListTransferID	15	uint8	0/1	If `PlaybackQueueListAvail`, the transfer ID for the contents of the playback queue
PlaybackAppBundleID	16	utf8	0/1	Uniform type identifer (UTI) in reverse-DNS format, e.g. `com.Ajax.hello`, of the Now Playing app

Table 59-127 `PlaybackStatus` enum

Value	Meaning
0	Stopped
1	Playing
2	Paused
3	SeekForward
4	SeekBackward

Table 59-128 `PlaybackShuffle` enum

Value	Meaning
0	Off

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Value	Meaning
1	Songs
2	Albums

Table 59-129 `PlaybackRepeat` enum

Value	Meaning
0	Off
1	One
2	All

59.13.3 StopNowPlayingUpdates

Source	ID
Accessory	0x5002

Table 59-130 `StopNowPlayingUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.13.4 SetNowPlayingInformation

Source	ID
Accessory	0x5003

Table 59-131 `SetNowPlayingInformation` message parameters

Name	ID	Type	#	Notes
ElapsedTime	0	uint32	0/1	Only effective if `NowPlayingUpdate` message `SetElapsedTimeAvail` parameter is true.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Name	ID	Type	#	Notes
PlaybackQueueIndex	1	uint32	0/1	

59.14 Power

For more information, see Power (page 660).

59.14.1 StartPowerUpdates

Source	ID
Accessory	0xAE00

Table 59-132 `StartPowerUpdates` message parameters

Name	ID	Type	#	Notes
MaximumCurrentDrawnFromAccessory	0	none	0/1	
DeviceBatteryWillChargeIfPowerIsPresent	1	none	0/1	
AccessoryPowerMode	2	none	0/1	Must be present if and only if the accessory draws power from the Apple device
IsExternalChargerConnected	4	none	0/1	
BatteryChargingState	5	none	0/1	
BatteryChargeLevel	6	none	0/1	

59.14.2 PowerUpdate

Source	ID
Device	0xAE01

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-133 `PowerUpdate` message parameters

Name	ID	Type	#	Notes
MaximumCurrentDrawnFromAccessory	0	uint16	0/1	The Apple device will draw up to this amount of current (in mA) from the accessory
DeviceBatteryWillChargeIfPowerIsPresent	1	bool	0/1	
AccessoryPowerMode	2	enum	0/1	See Table 59-134 (page 865)
IsExternalChargerConnected	4	bool	0/1	This parameter is sent when an external charger is connected or disconnected
BatteryChargingState	5	enum	0/1	This parameter provides the battery charging state. See Table 59-135 (page 865)
BatteryChargeLevel	6	uint16	0/1	This specifies the battery charge level (0-100)

Table 59-134 `AccessoryPowerModes` enum

Value	Meaning
0	Reserved
1	Low Power Mode
2	Intermittent High Power Mode

Table 59-135 `BatteryChargingState` enum

Value	Meaning
0	Disabled
1	Charging
2	Charged

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.14.3 StopPowerUpdates

Source	ID
Accessory	0xAE02

Table 59-136 `StopPowerUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.14.4 PowerSourceUpdate

Source	ID
Accessory	0xAE03

Table 59-137 `PowerSourceUpdate` message parameters

Name	ID	Type	#	Notes
AvailableCurrentForDevice	0	uint16	0/1	Must be one of the following values - 0, 1000, 2100, or 2400
DeviceBatteryShouldChargeIfPowerIsPresent	1	bool	0/1	

59.15 USB Device Mode Audio

For more information, see Digital Audio (page 528).

59.15.1 StartUSBDeviceModeAudio

Source	ID
Accessory	0xDA00

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-138 `StartUSBDeviceModeAudio` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.15.2 USBDeviceModeAudioInformation

Source	ID
Device	0xDA01

Table 59-139 `USBDeviceModeAudioInformation` message parameters

Name	ID	Type	#	Notes
SampleRate	0	enum	1	See Table 59-14 (page 811)

59.15.3 StopUSBDeviceModeAudio

Source	ID
Accessory	0xDA02

Table 59-140 `StopUSBDeviceModeAudio` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.16 Vehicle Status

For more information, see Vehicle Status (page 718).

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.16.1 StartVehicleStatusUpdates

Source	ID
Device	0xA100

Table 59-141 `StartVehicleStatusUpdates` message parameters

Name	ID	Type	#	Notes
Range	3	none	0/1	
OutsideTemperature	4	none	0/1	
RangeWarning	6	none	0/1	

59.16.2 VehicleStatusUpdate

Source	ID
Accessory	0xA101

Table 59-142 `VehicleStatusUpdate` message parameters

Name	ID	Type	#	Notes
Range	3	uint16	0/1	Remaining vehicle range in kilometers
OutsideTemperature	4	int16	0/1	measured outside temperature in °C
RangeWarning	6	bool	0/1	if True, the vehicle's low range warning indicator is set

59.16.3 StopVehicleStatusUpdates

Source	ID
Device	0xA102

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-143 `StopVehicleStatusUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.17 VoiceOver

For more information, see VoiceOver (page 720).

59.17.1 StartVoiceOver

Source	ID
Accessory	0x5612

Table 59-144 `StartVoiceOver` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.17.2 StopVoiceOver

Source	ID
Accessory	0x5613

Table 59-145 `StopVoiceOver` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.17.3 RequestVoiceOverMoveCursor

Source	ID
Accessory	0x5601

Table 59-146 `RequestVoiceOverMoveCursor` message parameters

Name	ID	Type	#	Notes
CursorDirection	0	enum	1	See Table 59-147 (page 870)

Table 59-147 `VoiceOverCursorDirection` enum

Value	Meaning
0	Next
1	Previous
2	Escape

59.17.4 RequestVoiceOverActivateCursor

Source	ID
Accessory	0x5602

Table 59-148 `RequestVoiceOverActivateCursor` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.17.5 RequestVoiceOverScrollPage

Source	ID
Accessory	0x5603

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-149 `RequestVoiceOverScrollPage` message parameters

Name	ID	Type	#	Notes
ScrollDirection	0	enum	1	See Table 59-150 (page 871)

Table 59-150 `VoiceOverScrollDirection` enum

Value	Meaning
0	Left
1	Right
2	Up
3	Down

59.17.6 RequestVoiceOverSpeakText

Source	ID
Accessory	0x5606

Table 59-151 `RequestVoiceOverSpeakText` message parameters

Name	ID	Type	#	Notes
TextToSpeak	0	utf8	0/1	If not present, VoiceOver will read the currently visible onscreen text starting from the top. Otherwise, VoiceOver will read the text present in this parameter

59.17.7 RequestVoiceOverPauseText

Source	ID
Accessory	0x5608

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-152 `RequestVoiceOverPauseText` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.17.8 RequestVoiceOverResumeText

Source	ID
Accessory	0x5609

Table 59-153 `RequestVoiceOverResumeText` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.17.9 StartVoiceOverUpdates

Source	ID
Accessory	0x560B

Table 59-154 `StartVoiceOverUpdates` message parameters

Name	ID	Type	#	Notes
SpeakingVolume	0	none	0/1	
SpeakingRate	1	none	0/1	
Enabled	2	none	0/1	

59.17.10 VoiceOverUpdate

Source	ID
Device	0x560C

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-155 `VoiceOverUpdate` message parameters

Name	ID	Type	#	Notes
SpeakingVolume	0	uint8	0/1	0=muted, 255=loudest possible volume
SpeakingRate	1	uint8	0/1	0=off, 255=fastest possible speed
Enabled	2	bool	0/1	

59.17.11 StopVoiceOverUpdates

Source	ID
Accessory	0x560D

Table 59-156 `StopVoiceOverUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.17.12 RequestVoiceOverConfiguration

Source	ID
Accessory	0x560E

Table 59-157 `RequestVoiceOverConfiguration` message parameters

Name	ID	Type	#	Notes
SpeakingVolume	0	uint8	0/1	0=muted, 255=loudest possible volume
SpeakingRate	1	uint8	0/1	0=off, 255=fastest possible speed

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

59.17.13 StartVoiceOverCursorUpdates

Source	ID
Accessory	0x560F

Table 59-158 `StartVoiceOverCursorUpdates` message parameters

Name	ID	Type	#	Notes
Label	0	none	0/1	
Value	1	none	0/1	
Hint	2	none	0/1	
Traits	3	none	0/1	

59.17.14 VoiceOverCursorUpdate

Source	ID
Device	0x5610

Table 59-159 `VoiceOverCursorUpdate` message parameters

Name	ID	Type	#	Notes
Label	0	utf8	0/1	
Value	1	utf8	0/1	
Hint	2	utf8	0/1	
Traits	3	blob	0/1	An array of uint16s. See Table 59-160 (page 875) (uint16[])

The following VoiceOver item traits are defined:

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-160 VoiceOverCursorUpdate Traits

Value	Meaning
0	Button
1	Link
2	Search Field
3	Image
4	Selected
5	Sound
6	Keyboard Key
7	Static Text
8	Summary Element
9	Not Enabled
10	Updates Frequently
11	Starts Media Session
12	Adjustable
13	Back Button
14	Map
15	Delete Key

59.17.15 StopVoiceOverCursorUpdates

Source	ID
Accessory	0x5611

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-161 `StopVoiceOverCursorUpdates` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.18 Wi-Fi Information Sharing

For more information, see Wi-Fi Information Sharing (page 761).

59.18.1 RequestWiFiInformation

Source	ID
Accessory	0x5700

Table 59-162 `RequestWiFiInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.18.2 WiFiInformation

Source	ID
Device	0x5701

Table 59-163 `WiFiInformation` message parameters

Name	ID	Type	#	Notes
RequestStatus	0	enum	1	see Table 59-164 (page 877)
WiFiSSID	2	utf8	0/1	
WiFiPassphrase	3	utf8	0/1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-164 `WiFiRequestStatus` enum

Value	Meaning
0	Success
1	User Declined
2	Network Information Unavailable

59.18.3 RequestAccessoryWiFiConfigurationInformation

Source	ID
Device	0x5702

Table 59-165 `RequestAccessoryWiFiConfigurationInformation` message parameters

Name	ID	Type	#	Notes
This message has no parameters.				

59.18.4 AccessoryWiFiConfigurationInformation

Source	ID
Accessory	0x5703

Table 59-166 `AccessoryWiFiConfigurationInformation` message parameters

Name	ID	Type	#	Notes
WiFiSSID	1	utf8	1	Wi-Fi SSID
Passphrase	2	utf8	0/1	Required if SecurityType is not None
SecurityType	3	enum	0/1	see Table 59-167 (page 878)
Channel	4	uint8	0/1	

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Table 59-167 `WiFiSecurityType` enum

Value	Meaning
0	None
1	WEP
2	WPA/WPA2 Personal

Copyright © 2015 Apple Inc. All Rights Reserved.

60. Accessory Test System

Some test procedures in this specification require the use of the Accessory Test System (ATS) Lightning box (part number: MFIATS-R3-001). This box requires associated ATS software to function; this software may be downloaded from the MFi Portal. Self certification activities must always use the most recent version of the ATS software.

Additional test hardware may be required to complete certification:

Test Hardware	Description
TotalPhase Beagle USB 480 Protocol Analyzer	Part Number: TP320510. Required to capture iAP2 over USB traffic.
Frontline ComProbe BPA 100	Part Number: MFI-FL-0116-BLD-0000. Required to capture iAP2 over Bluetooth traffic.
Lightning Connector Extension Cable	Part Number: MFI677-0873.

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

61. Device Dimensional Drawings

This chapter contains the following dimensional drawings:

- Apple Watch 38 mm (page 883)
- Apple Watch 42 mm (page 884)
- iPhone 6s Plus (page 885)
- iPhone 6s (page 886)
- iPhone 6 Plus (page 887)
- iPhone 6 (page 888)
- iPhone 5s (page 889)
- iPhone 5c (page 890)
- iPhone 5 (page 891)
- iPhone 4s (page 892)
- iPhone 4 (CDMA model) (page 893)
- iPhone 4 (GSM model) (page 894)
- iPhone 3G and iPhone 3GS (page 895)
- iPhone (page 896)
- iPad Pro Wi-Fi (page 897)
- iPad Pro Wi-Fi + Cellular (page 898)
- iPad mini 4 Wi-Fi (page 899)
- iPad mini 4 Wi-Fi + Cellular (page 900)
- iPad Air 2 Wi-Fi (page 901)
- iPad Air 2 Wi-Fi + Cellular (page 902)
- iPad mini 2 & 3 Wi-Fi (page 903)
- iPad mini 2 & 3 Wi-Fi + Cellular (page 904)
- iPad Air Wi-Fi (page 905)
- iPad Air Wi-Fi + Cellular (page 906)
- iPad mini with Wi-Fi (page 907)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- iPad mini with Wi-Fi + Cellular (page 908)
- iPad with Wi-Fi (4th generation) (page 909)
- iPad with Wi-Fi + Cellular (4th generation) (page 910)
- iPad with Wi-Fi (3rd generation) (page 911)
- iPad with Wi-Fi + 4G (3rd generation) (page 912)
- iPad 2 with Wi-Fi (page 913)
- iPad 2 with Wi-Fi + 3G (page 914)
- iPad with Wi-Fi (page 915)
- iPad with Wi-Fi + 3G (page 916)
- iPod touch (6th generation) (page 917)
- iPod touch (5th generation) (page 918)
- iPod touch (4th generation) (page 919)
- iPod touch (3rd generation) (page 920)
- iPod touch (2nd generation) (page 921)
- iPod touch (page 922)
- iPod nano (7th generation) (page 923)
- iPod nano (6th generation) (page 924)
- iPod nano (5th generation) (page 925)
- iPod nano (4th generation) (page 926)
- iPod nano (3rd generation) (page 927)
- iPod nano (2nd generation) (page 928)
- iPod nano (page 929)
- iPod classic 160GB (page 930)
- iPod classic 80GB (page 931)
- iPod (5th generation) 60GB/80GB (page 932)
- iPod (5th generation) 30GB (page 933)
- iPod (4th generation) (page 934)
- iPod (3rd generation) (page 935)
- iPod photo 30GB/60GB (page 936)
- iPod photo (page 937)
- iPod shuffle (4th generation) (page 938)

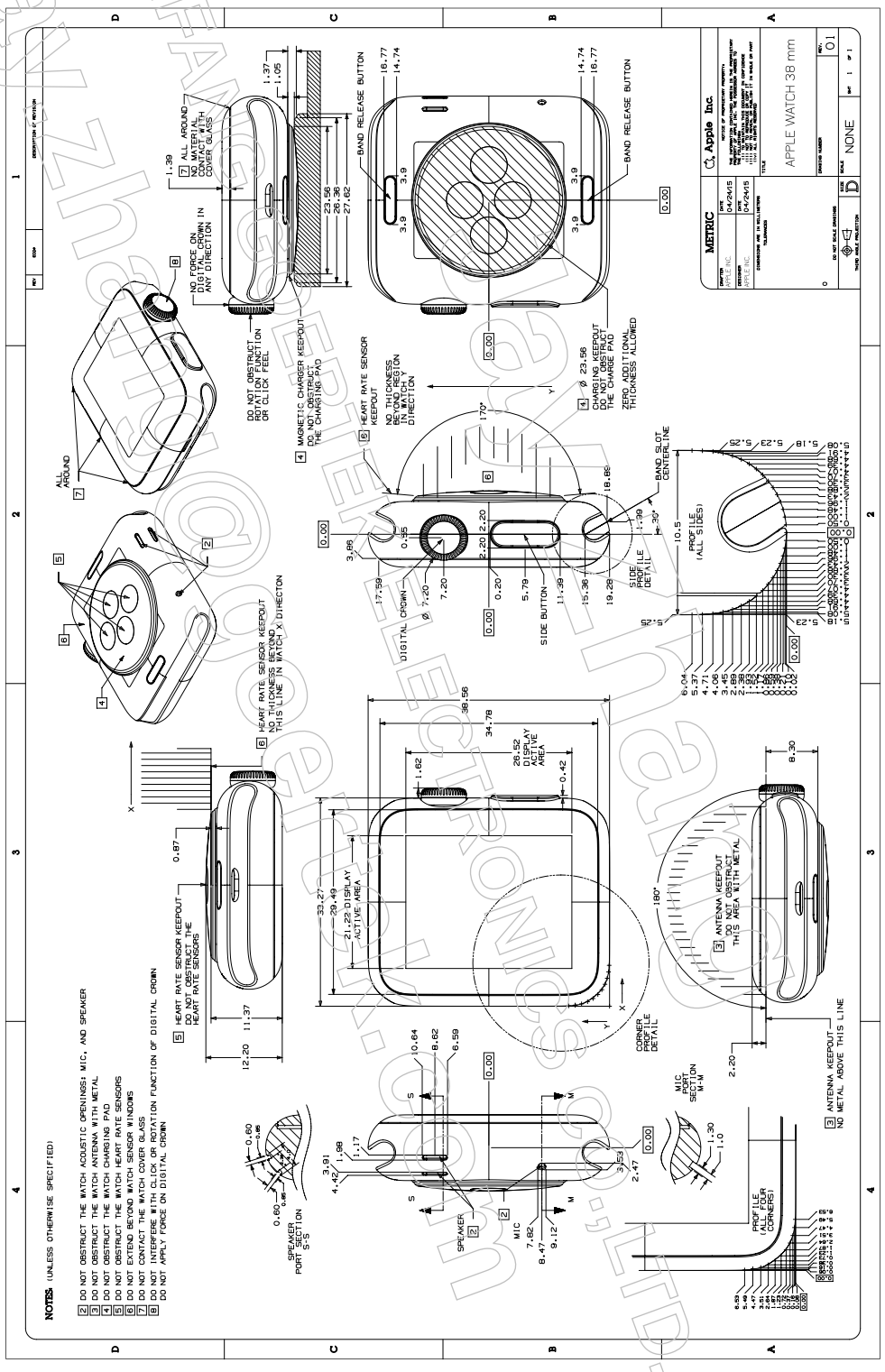
2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

- iPod shuffle (3rd generation) (page 939)
- iPod shuffle (2nd generation) (page 940)
- iPod shuffle (page 941)
- iPod mini (page 943)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

61.1 Apple Watch 38 mm

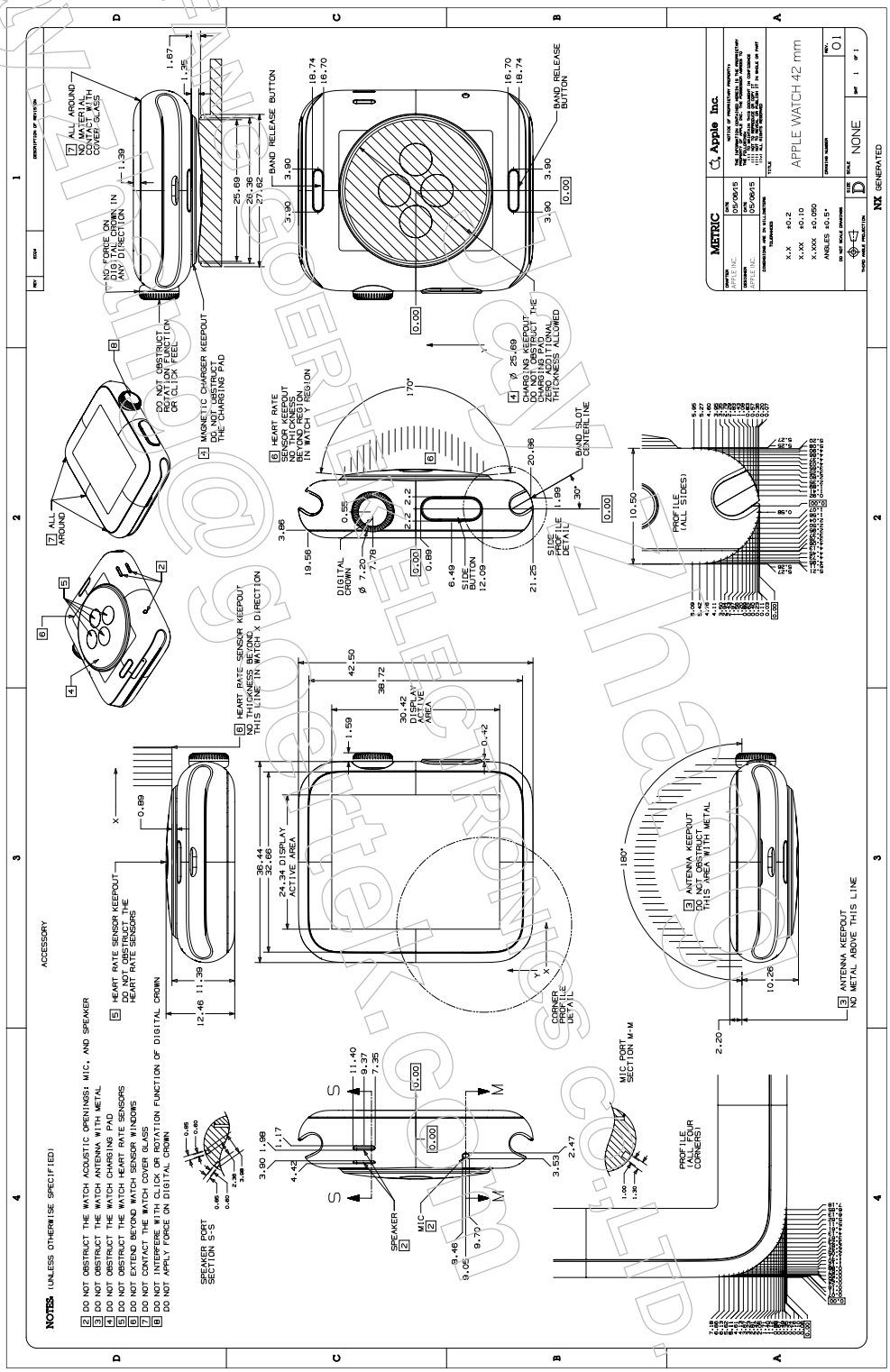
Figure 61-1 Apple Watch 38 mm Dimensional Drawing



2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

61.2 Apple Watch 42 mm

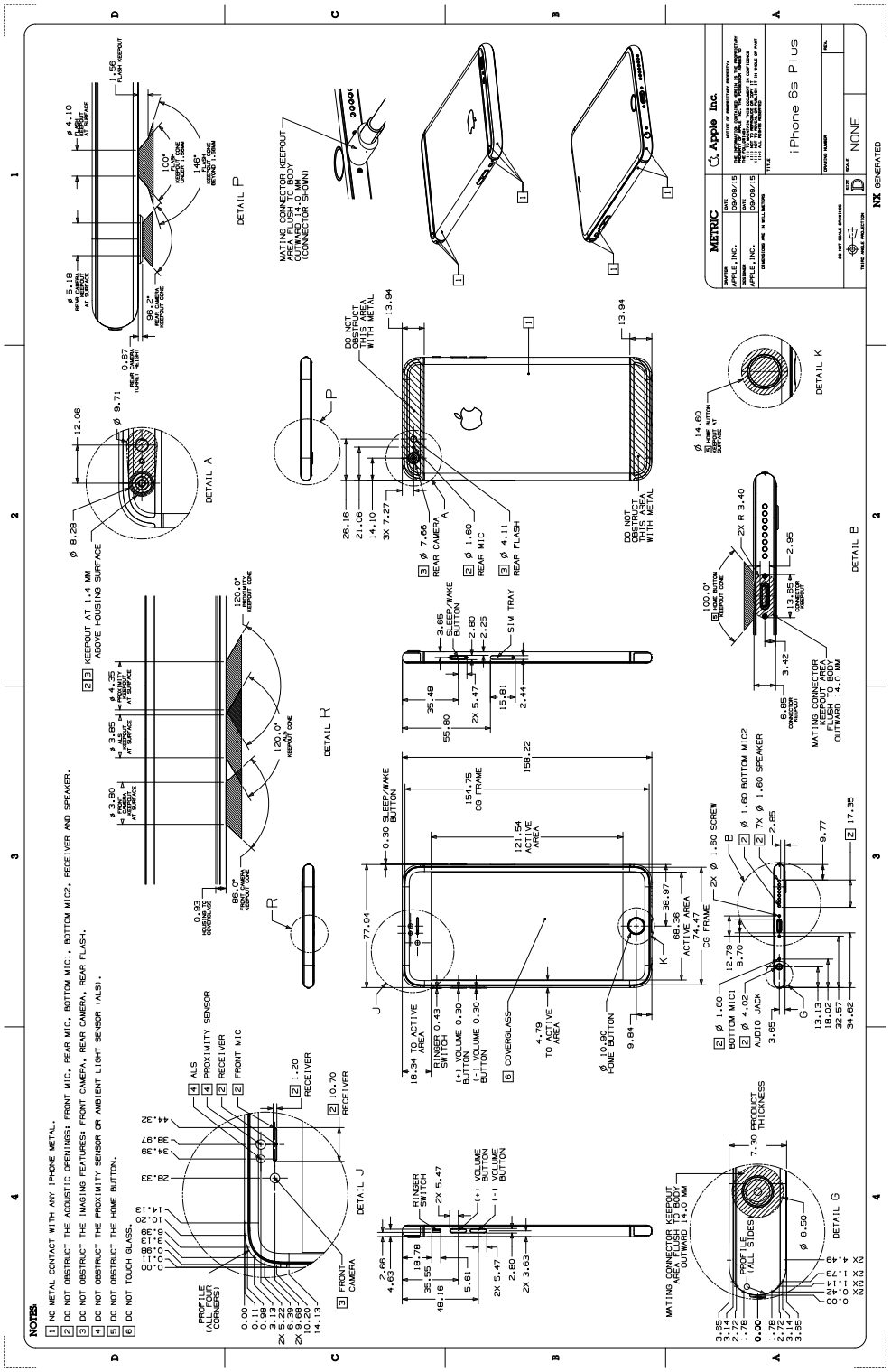
Figure 61-2 Apple Watch 42 mm Dimensional Drawing



Jiangsu Weiyuan Electronics Co., Ltd.

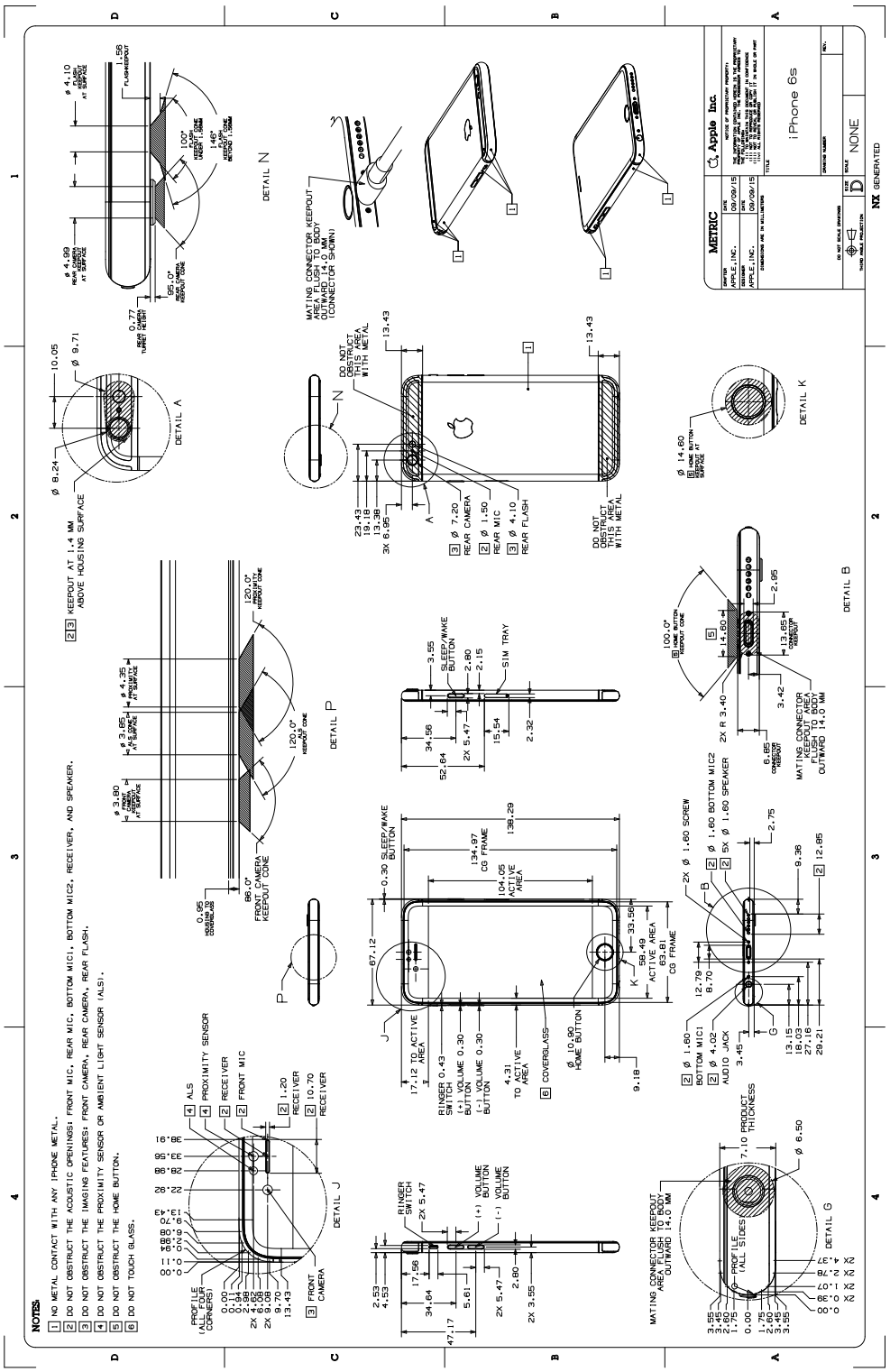
61.3 iPhone 6s Plus

Figure 61-3 iPhone 6s Plus Dimensional Drawing



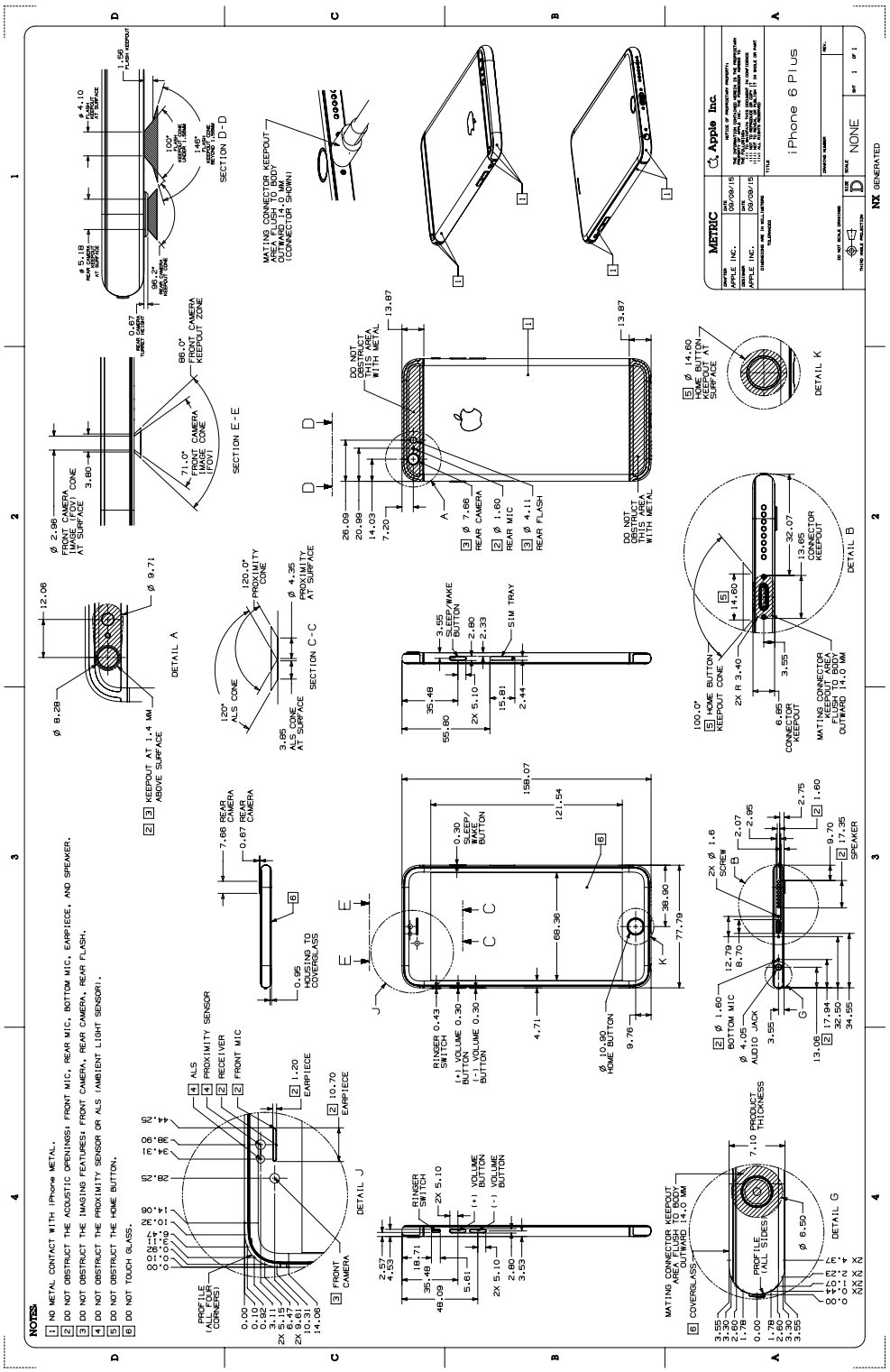
61.4 iPhone 6s

Figure 61-4 iPhone 6s Dimensional Drawing



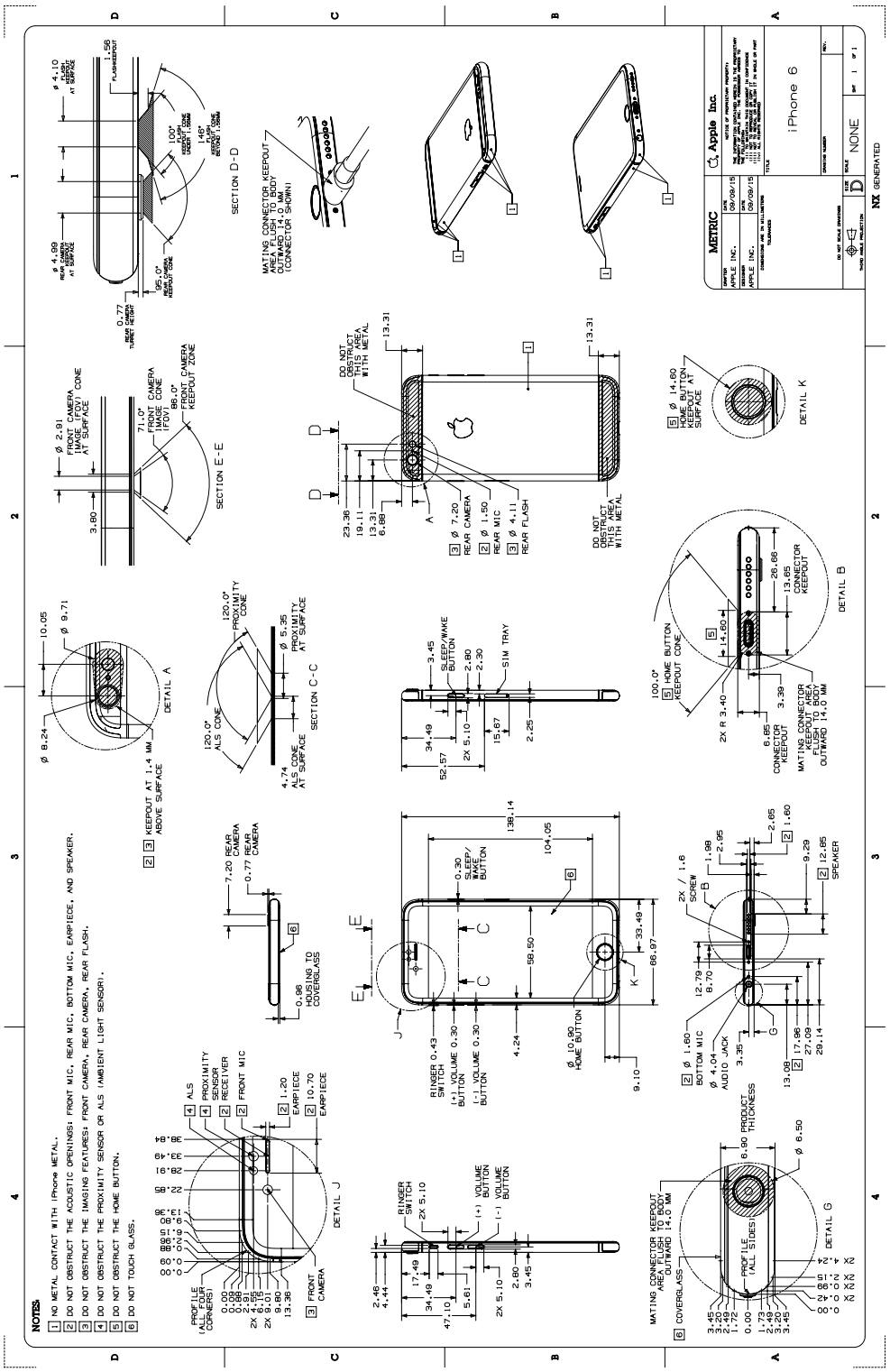
61.5 iPhone 6 Plus

Figure 61-5 iPhone 6 Plus Dimensional Drawing



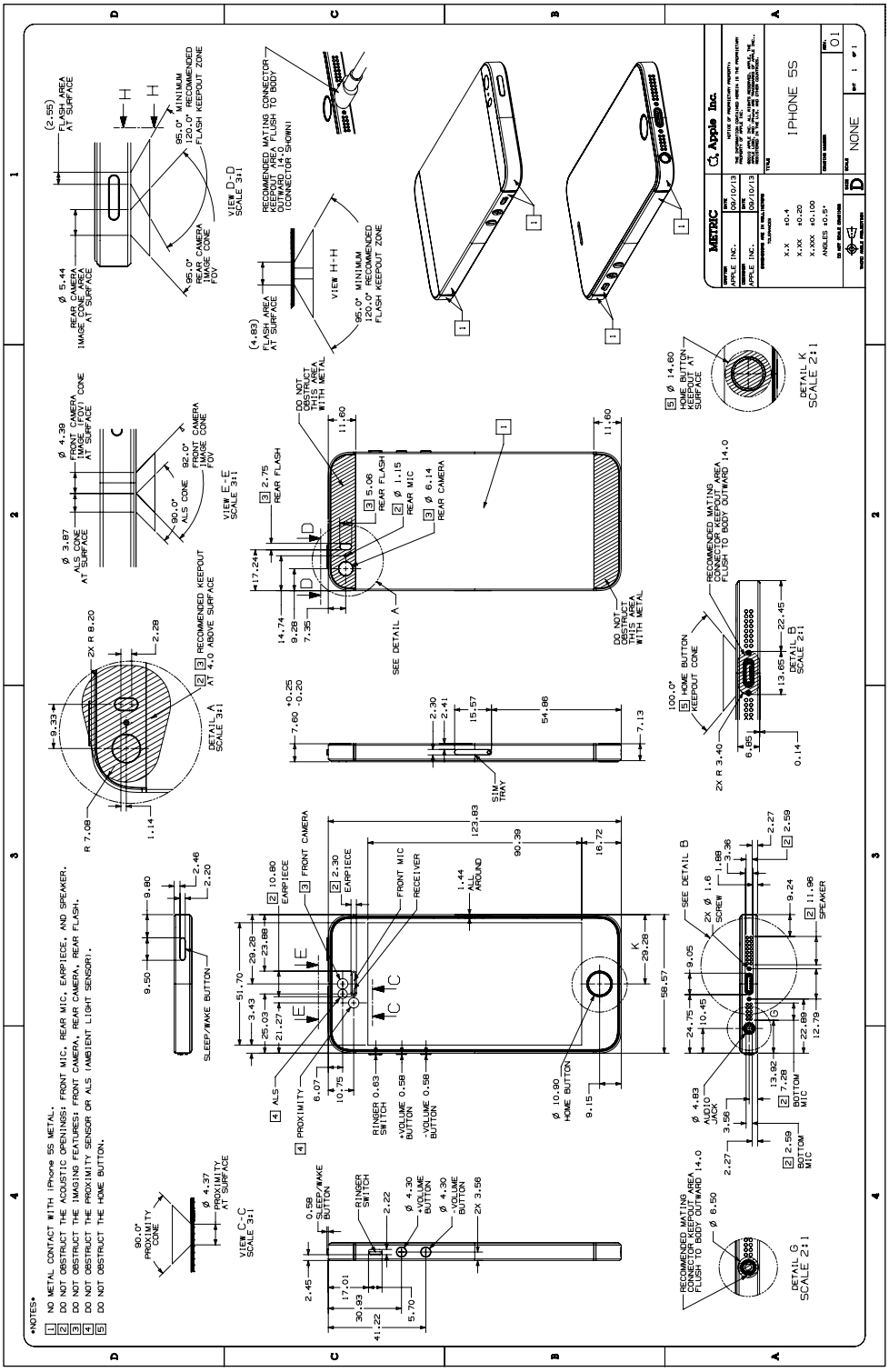
61.6 iPhone 6

Figure 61-6 iPhone 6 Dimensional Drawing



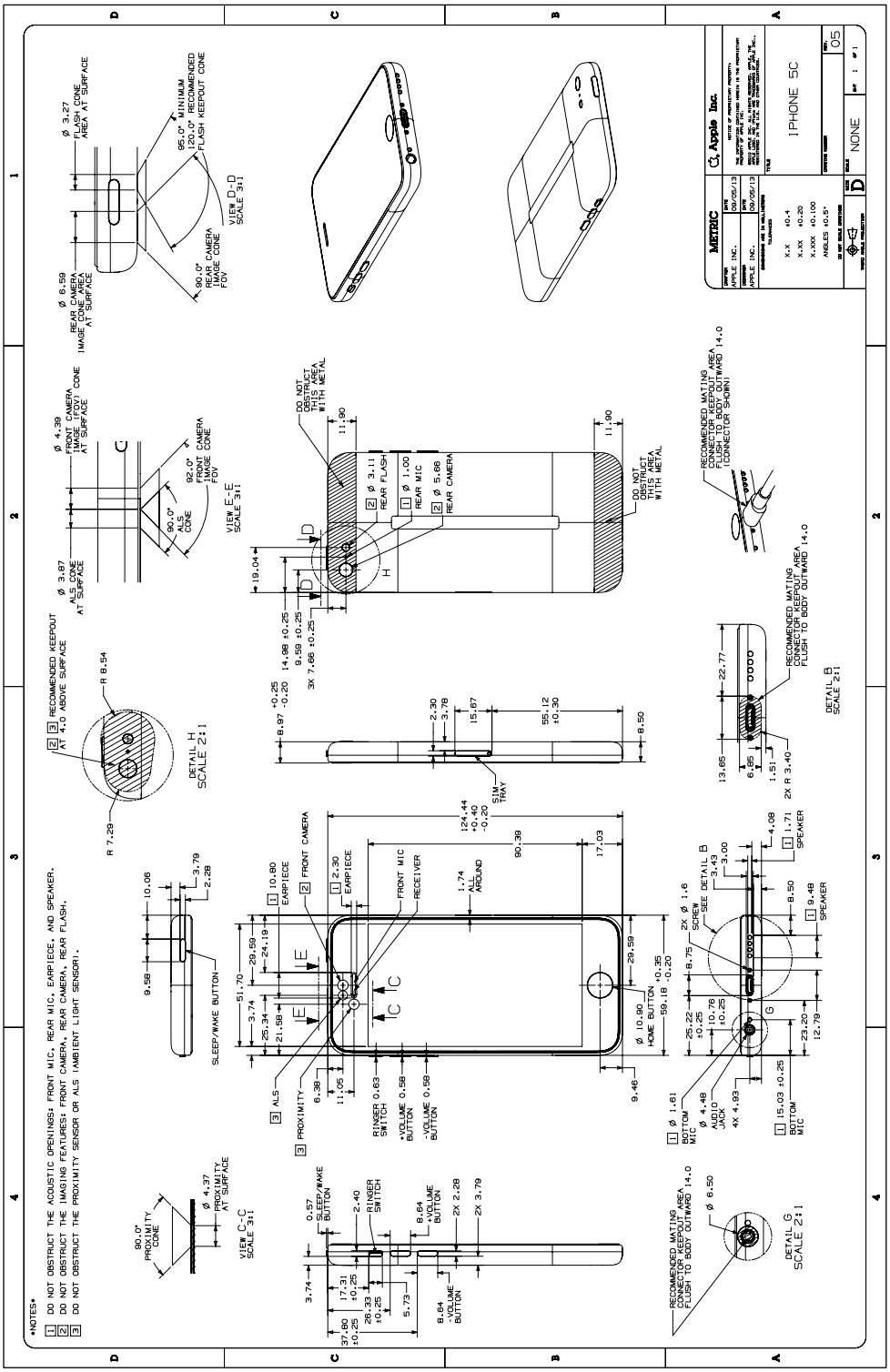
61.7 iPhone 5s

Figure 61-7 iPhone 5s Dimensional Drawing



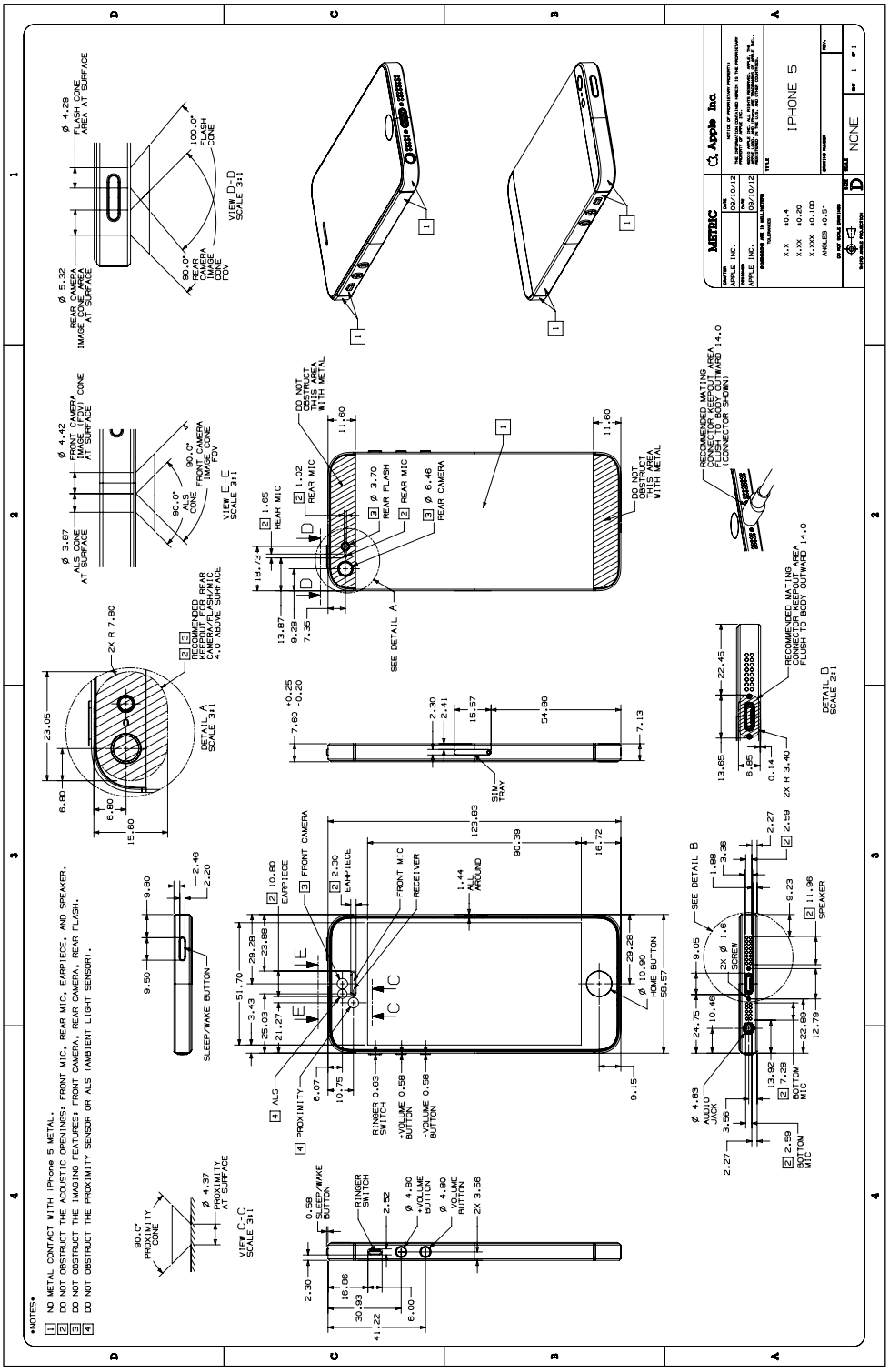
61.8 iPhone 5c

Figure 61-8 iPhone 5c Dimensional Drawing



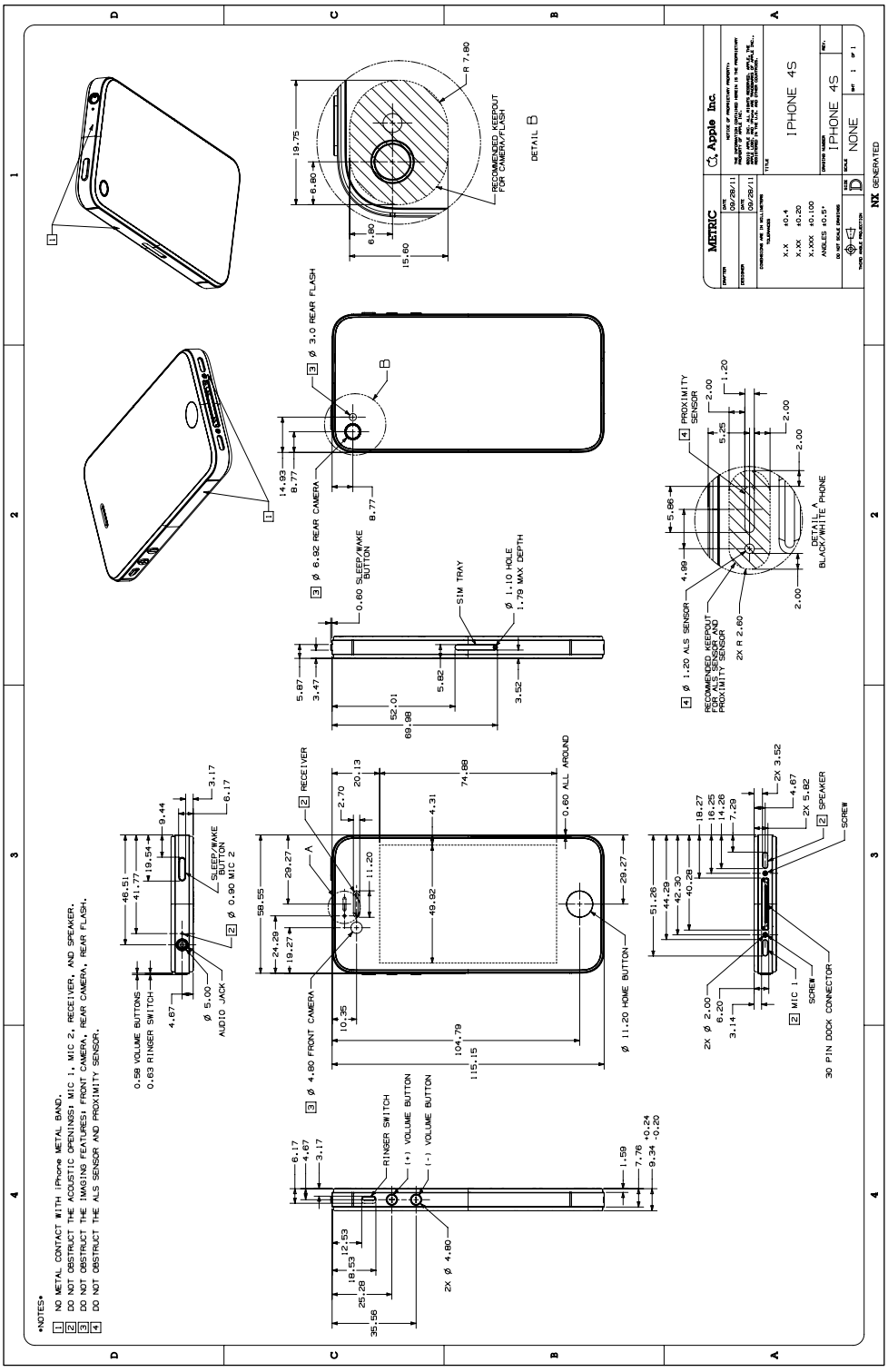
61.9 iPhone 5

Figure 61-9 iPhone 5 Dimensional Drawing



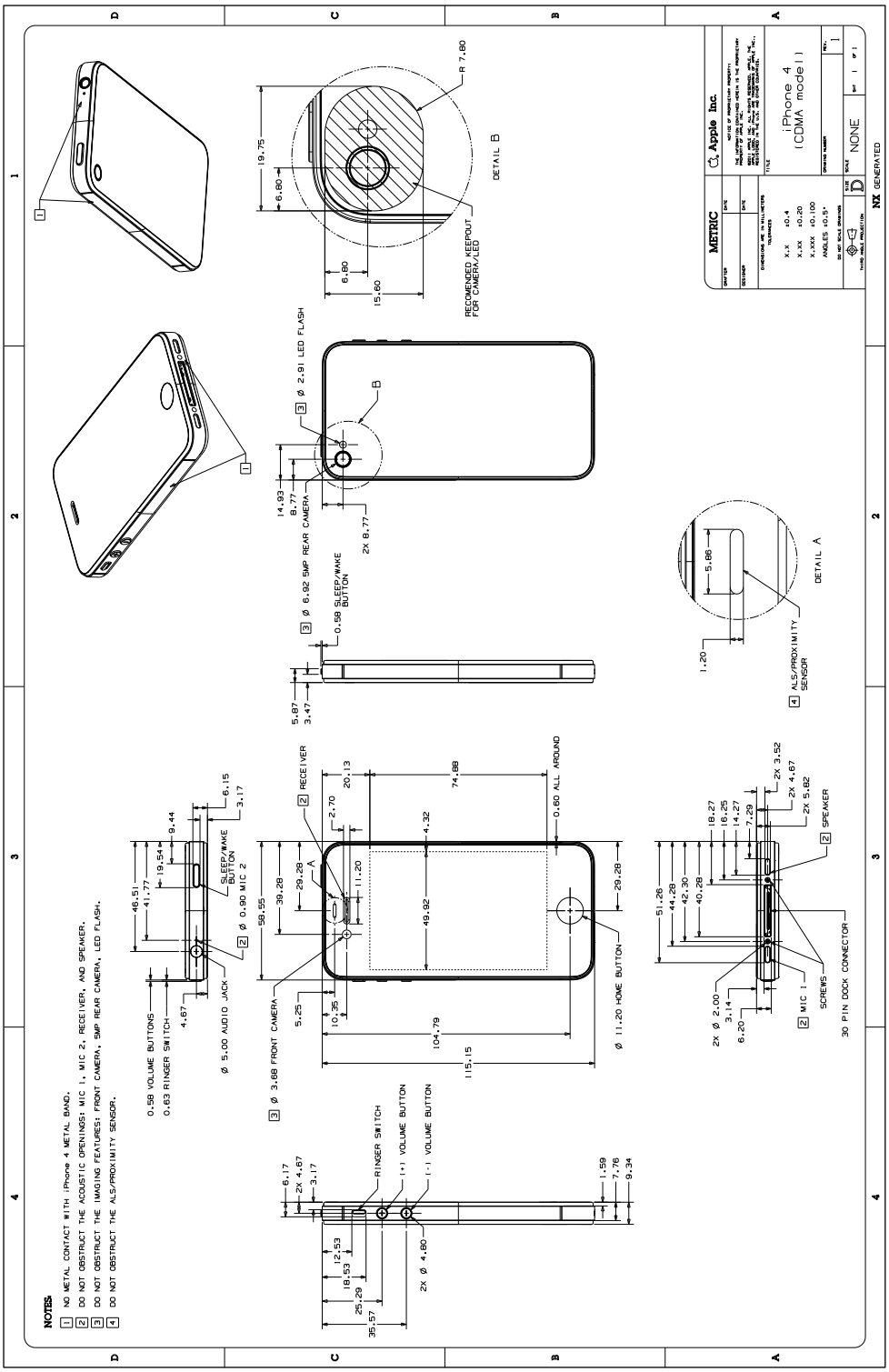
61.10 iPhone 4s

Figure 61-10 iPhone 4s Dimensional Drawing



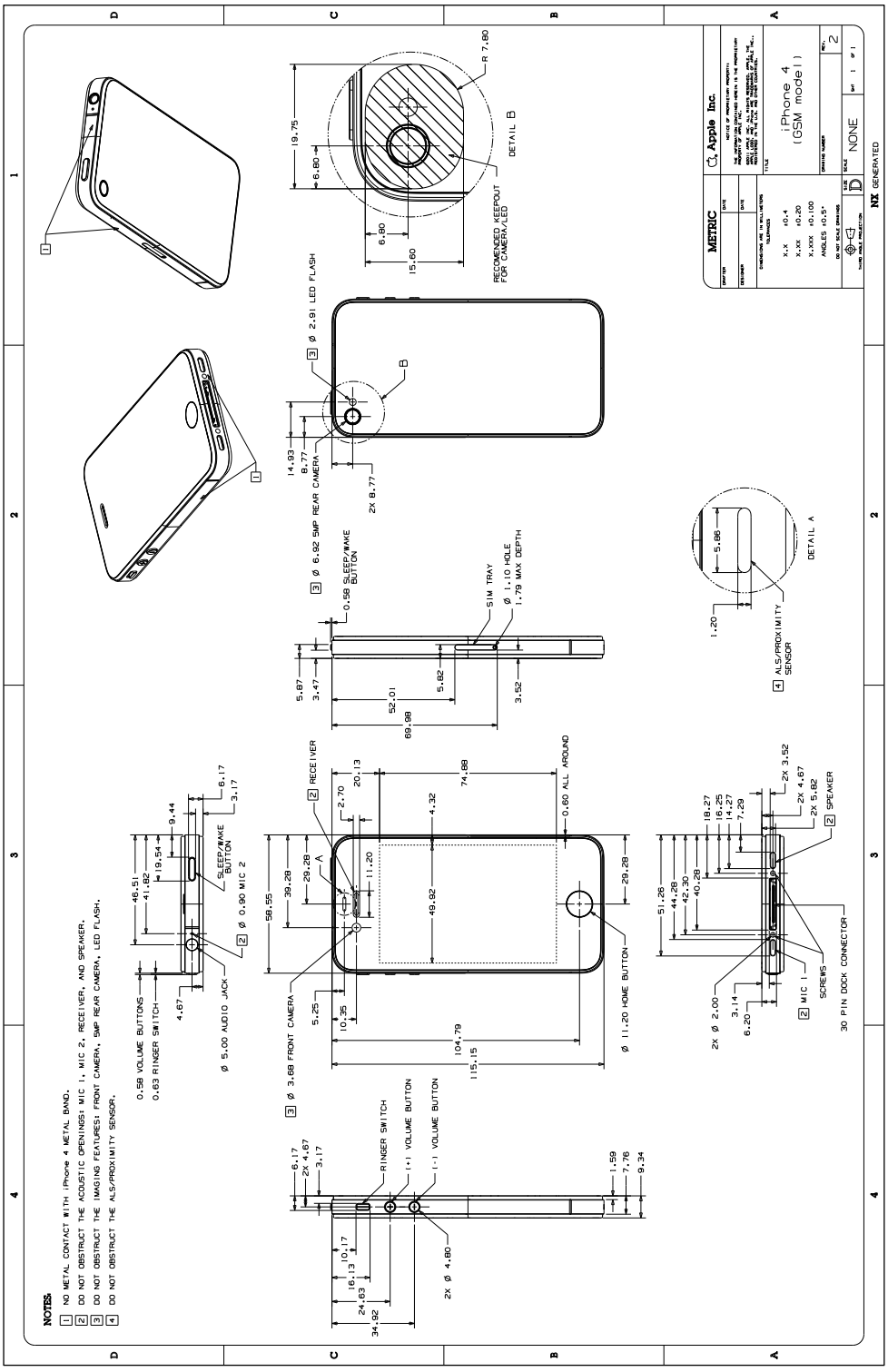
61.11 iPhone 4 (CDMA model)

Figure 61-11 iPhone 4 CDMA Dimensional Drawing



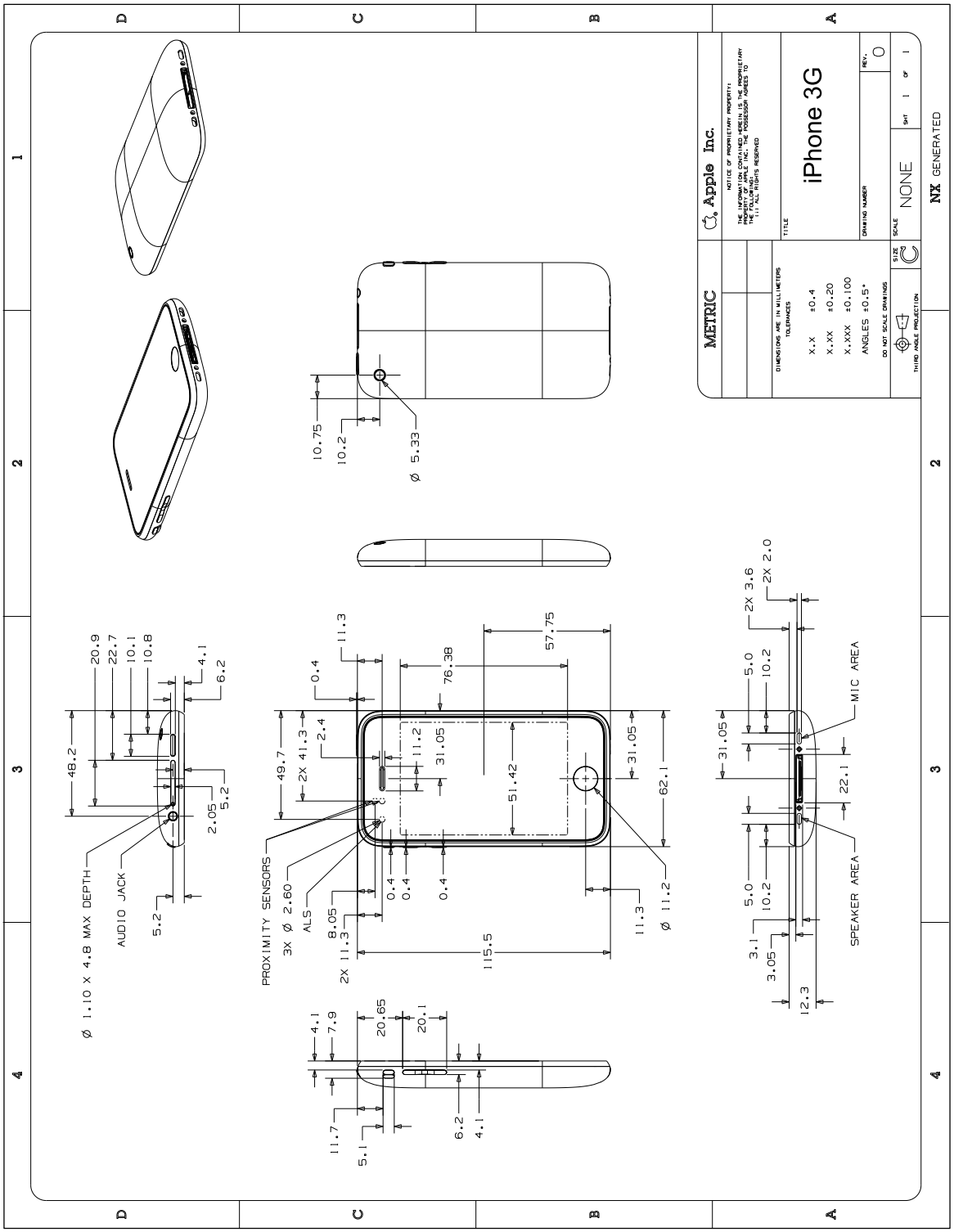
61.12 iPhone 4 (GSM model)

Figure 61-12 iPhone 4 GSM Dimensional Drawing



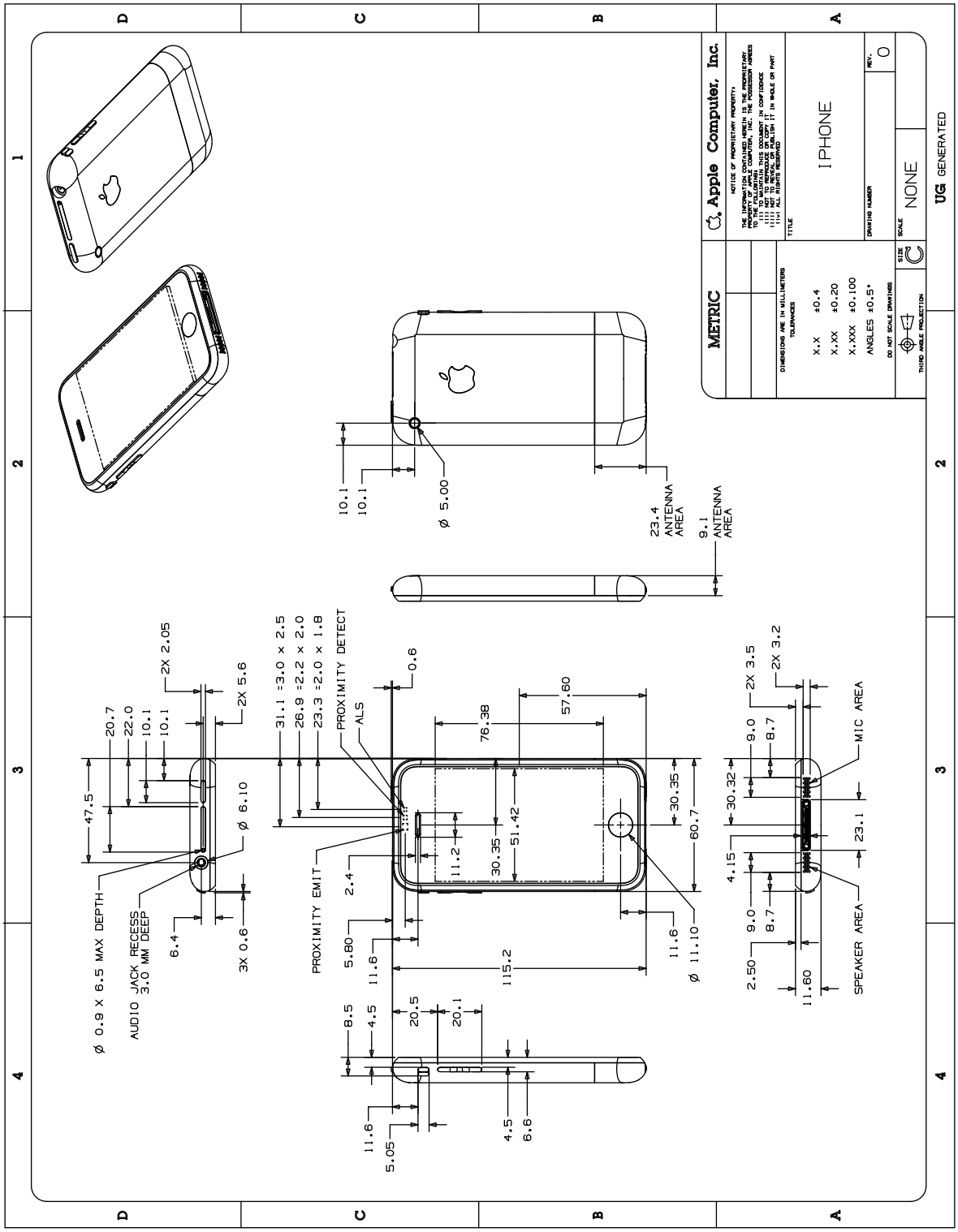
61.13 iPhone 3G and iPhone 3GS

Figure 61-13 iPhone 3G and iPhone 3GS Dimensional Drawing



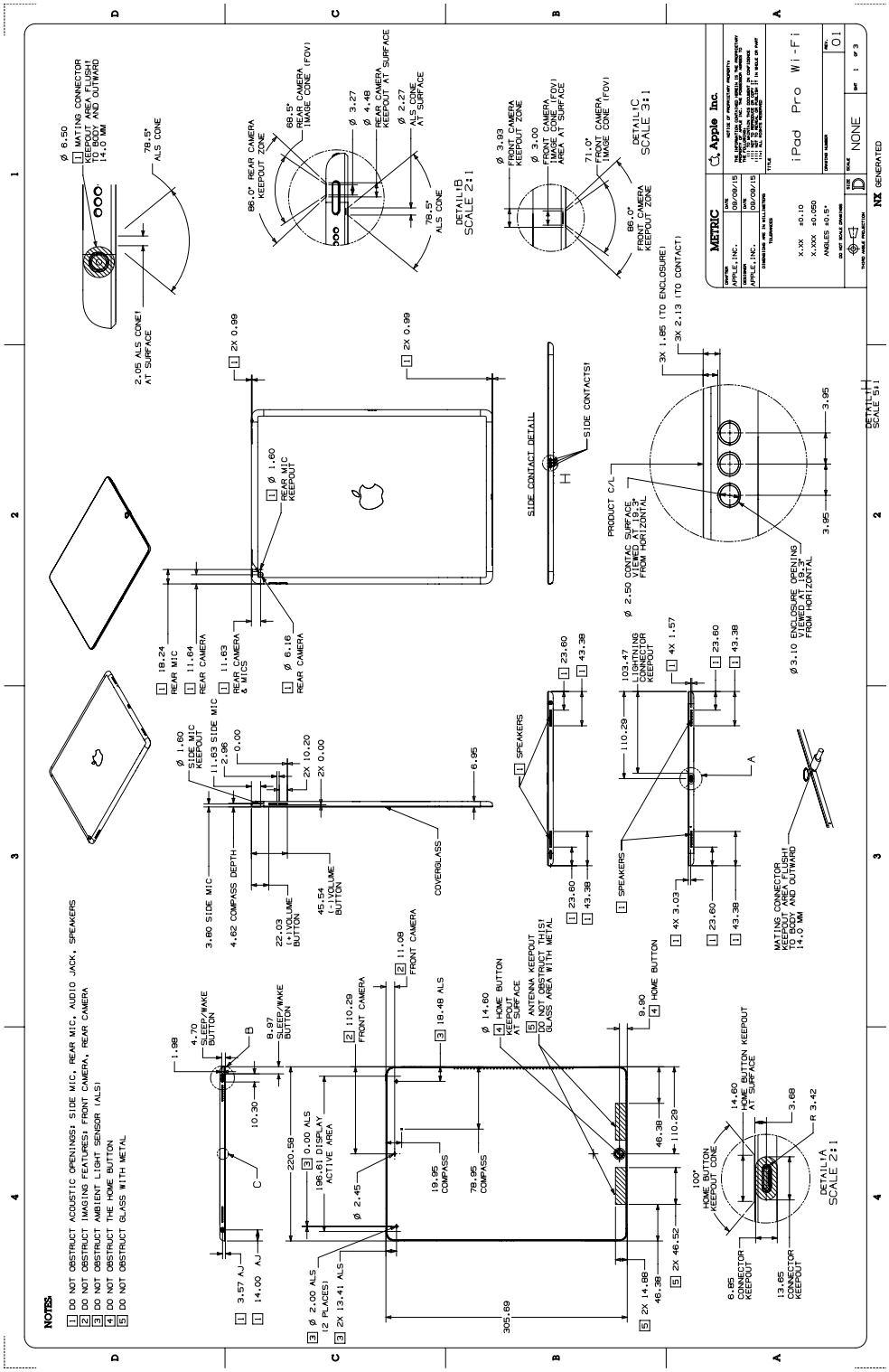
61.14 iPhone

Figure 61-14 iPhone Dimensional Drawing



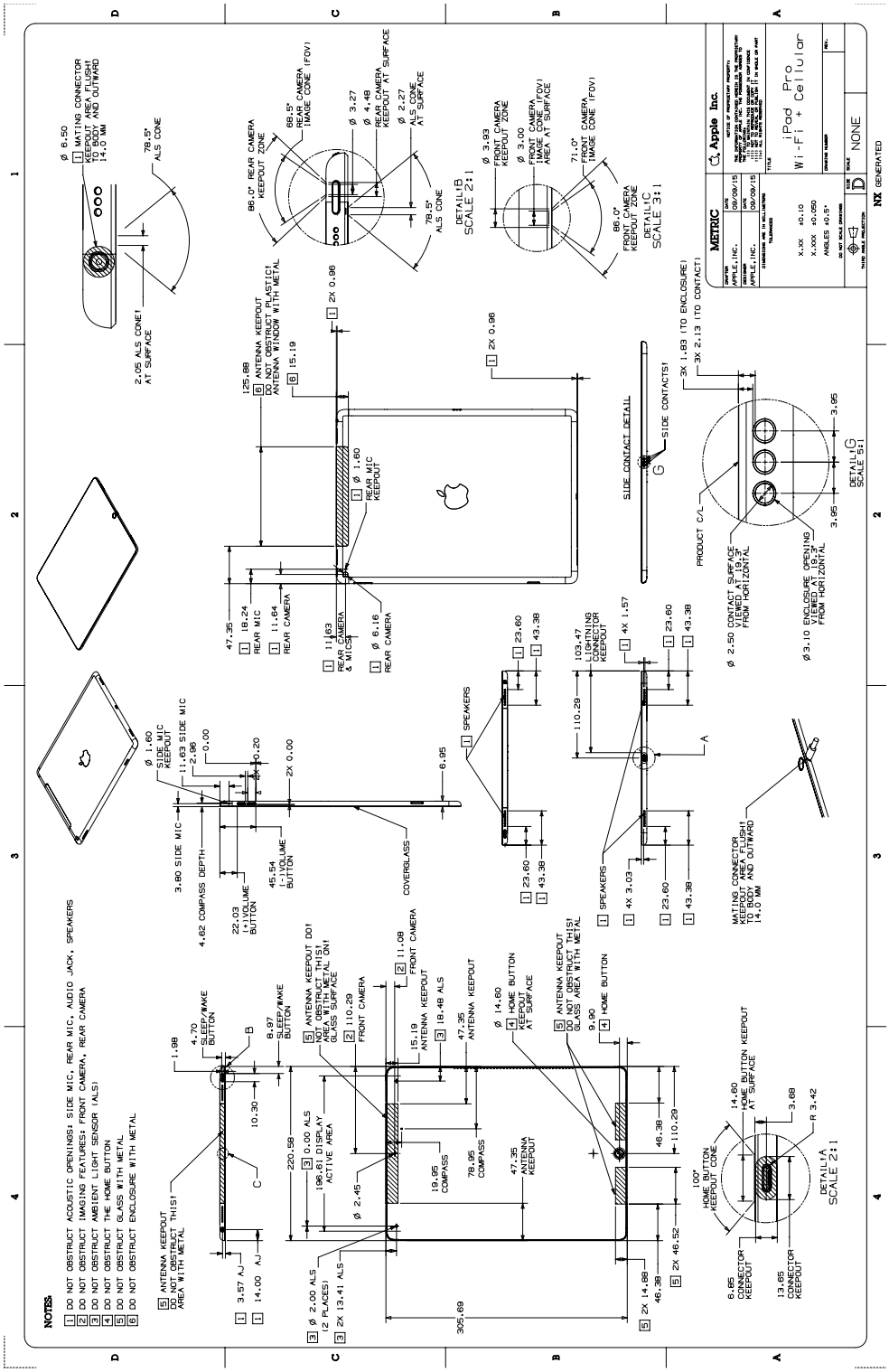
61.15 iPad Pro Wi-Fi

Figure 61-15 iPad Pro Wi-Fi Dimensional Drawing



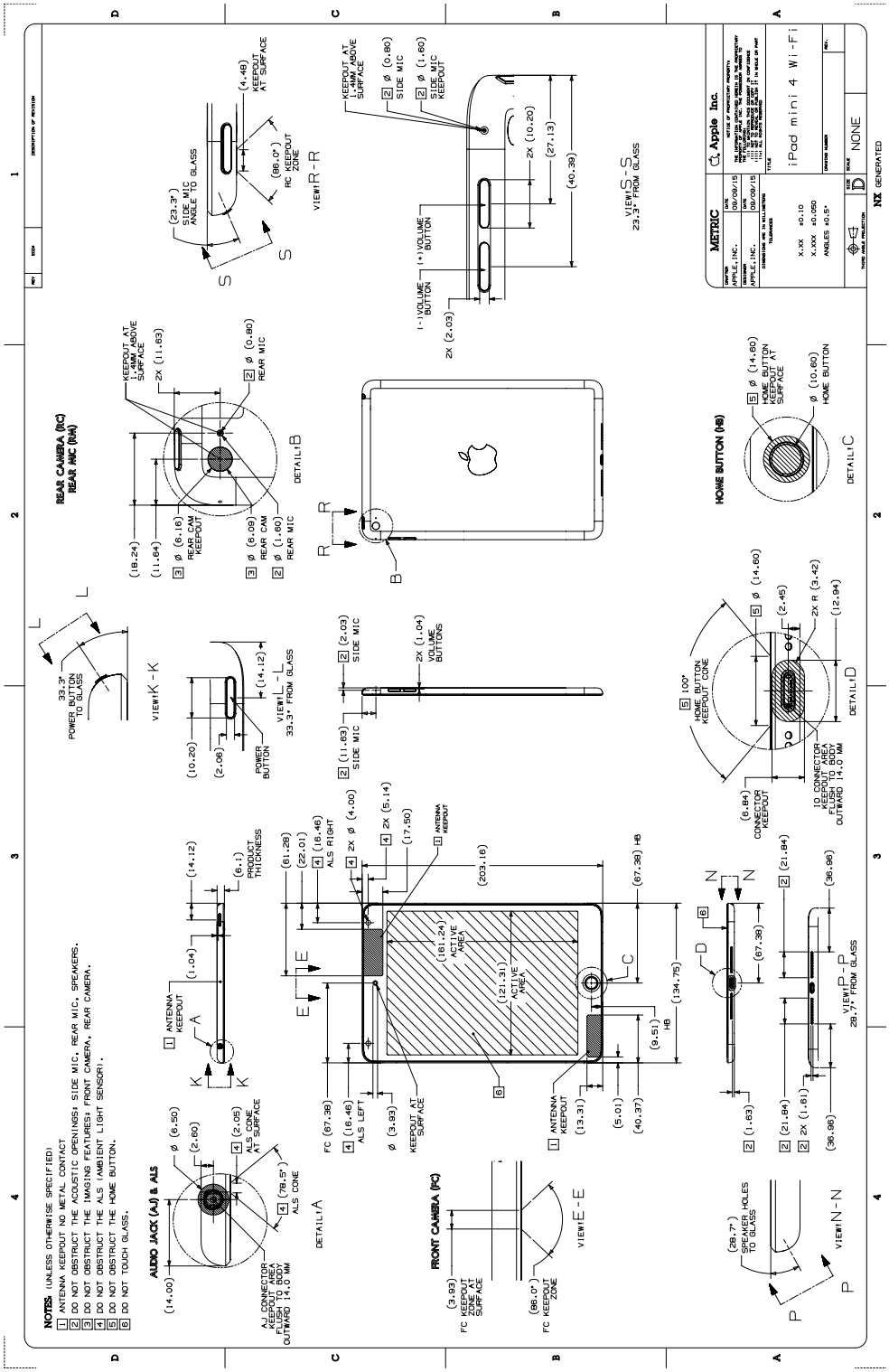
61.16 iPad Pro Wi-Fi + Cellular

Figure 61-16 iPad Pro Wi-Fi + Cellular Dimensional Drawing



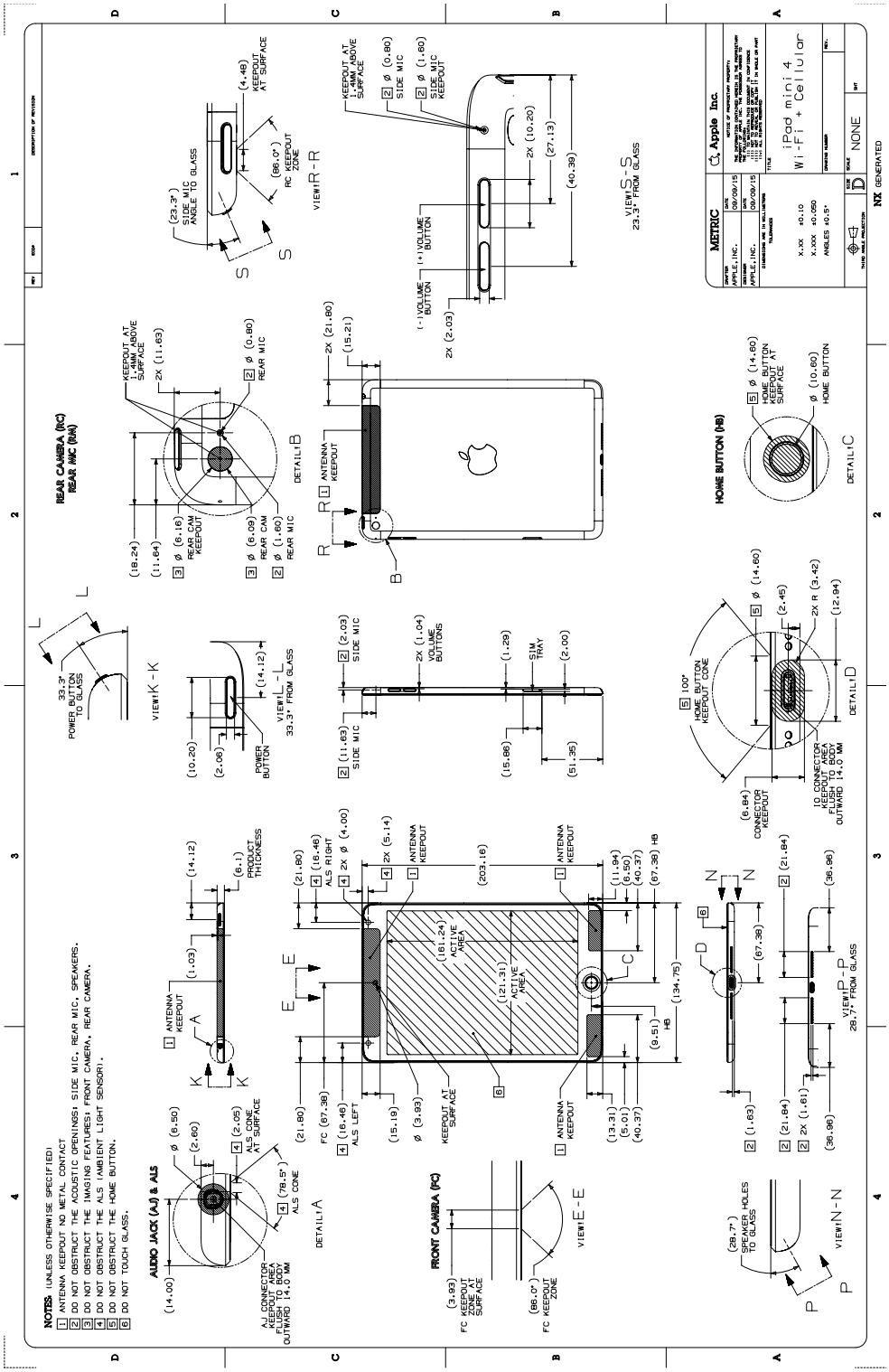
61.17 iPad mini 4 Wi-Fi

Figure 61-17 iPad mini 4 Wi-Fi Dimensional Drawing



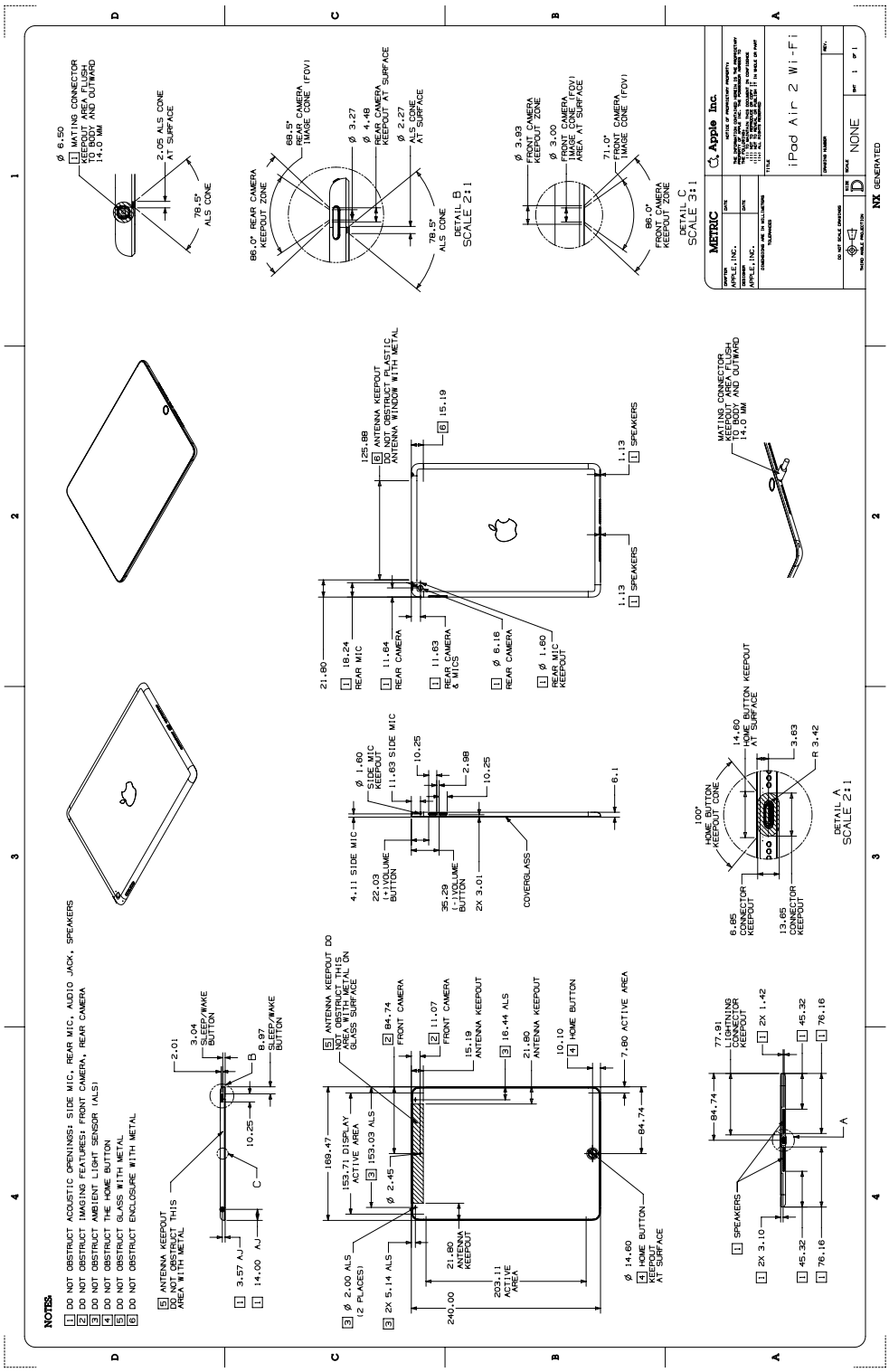
61.18 iPad mini 4 Wi-Fi + Cellular

Figure 61-18 iPad mini 4 Wi-Fi + Cellular Dimensional Drawing



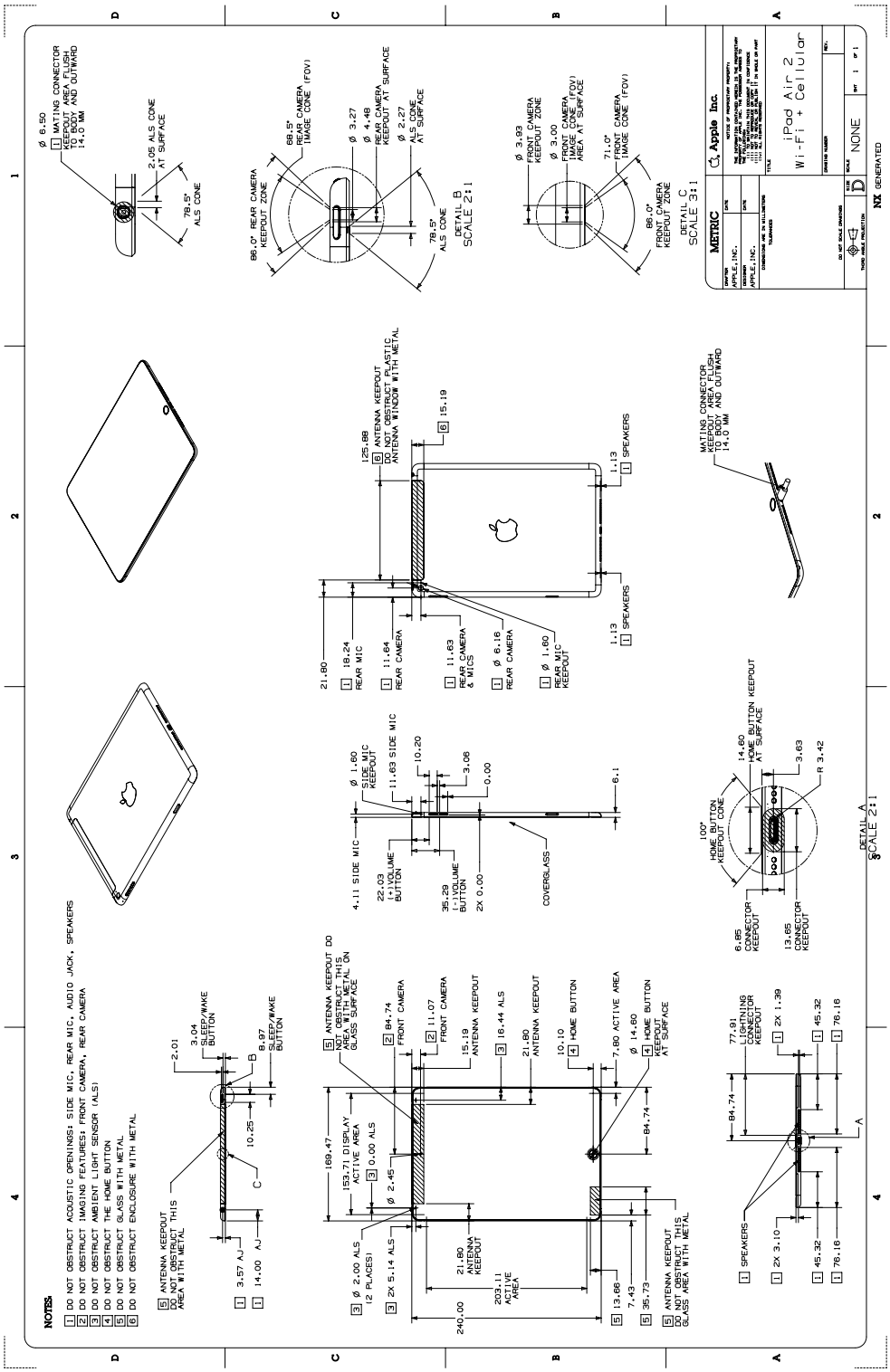
61.19 iPad Air 2 Wi-Fi

Figure 61-19 iPad Air 2 Wi-Fi Dimensional Drawing



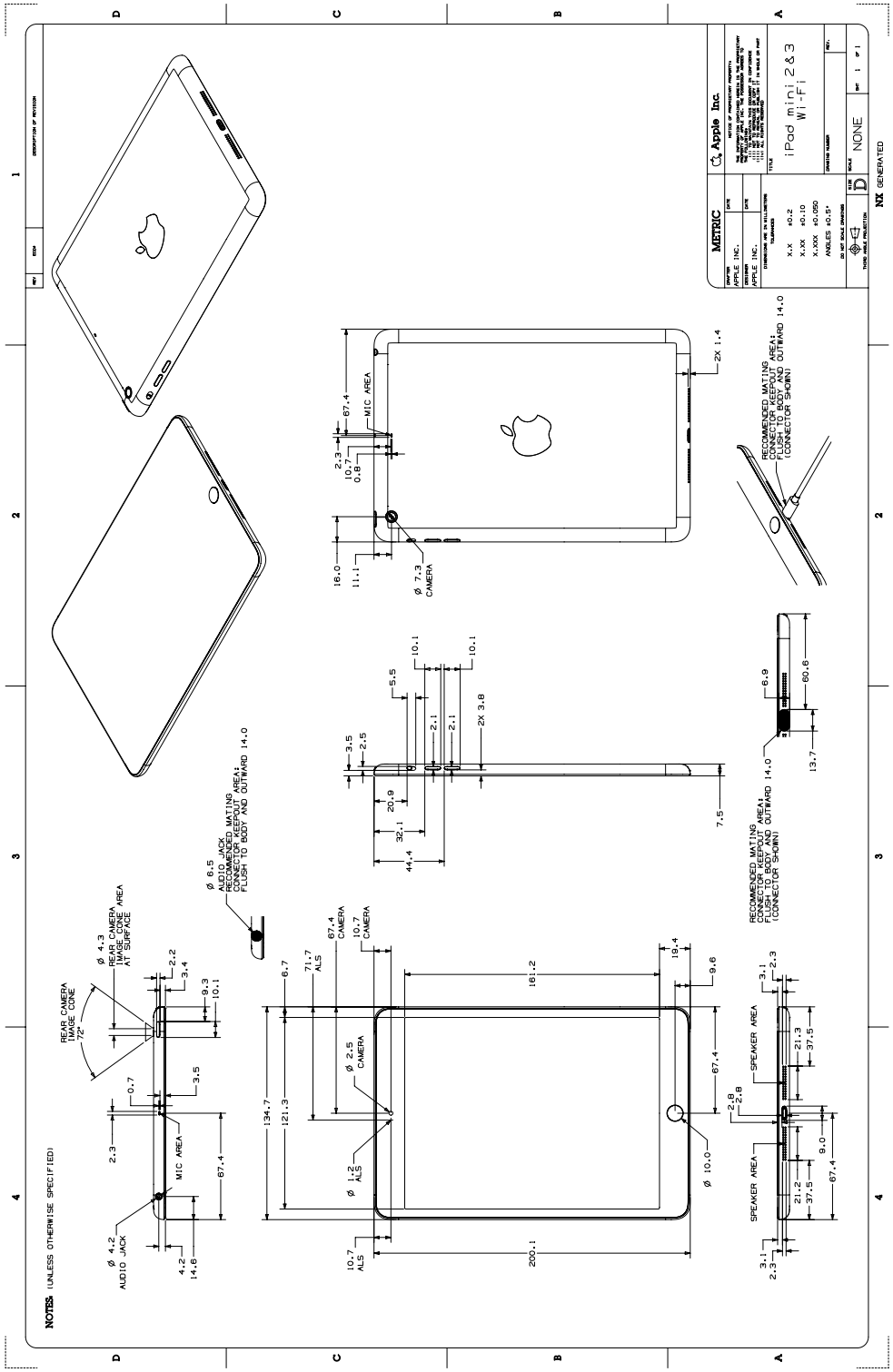
61.20 iPad Air 2 Wi-Fi + Cellular

Figure 61-20 iPad Air 2 Wi-Fi + Cellular Dimensional Drawing



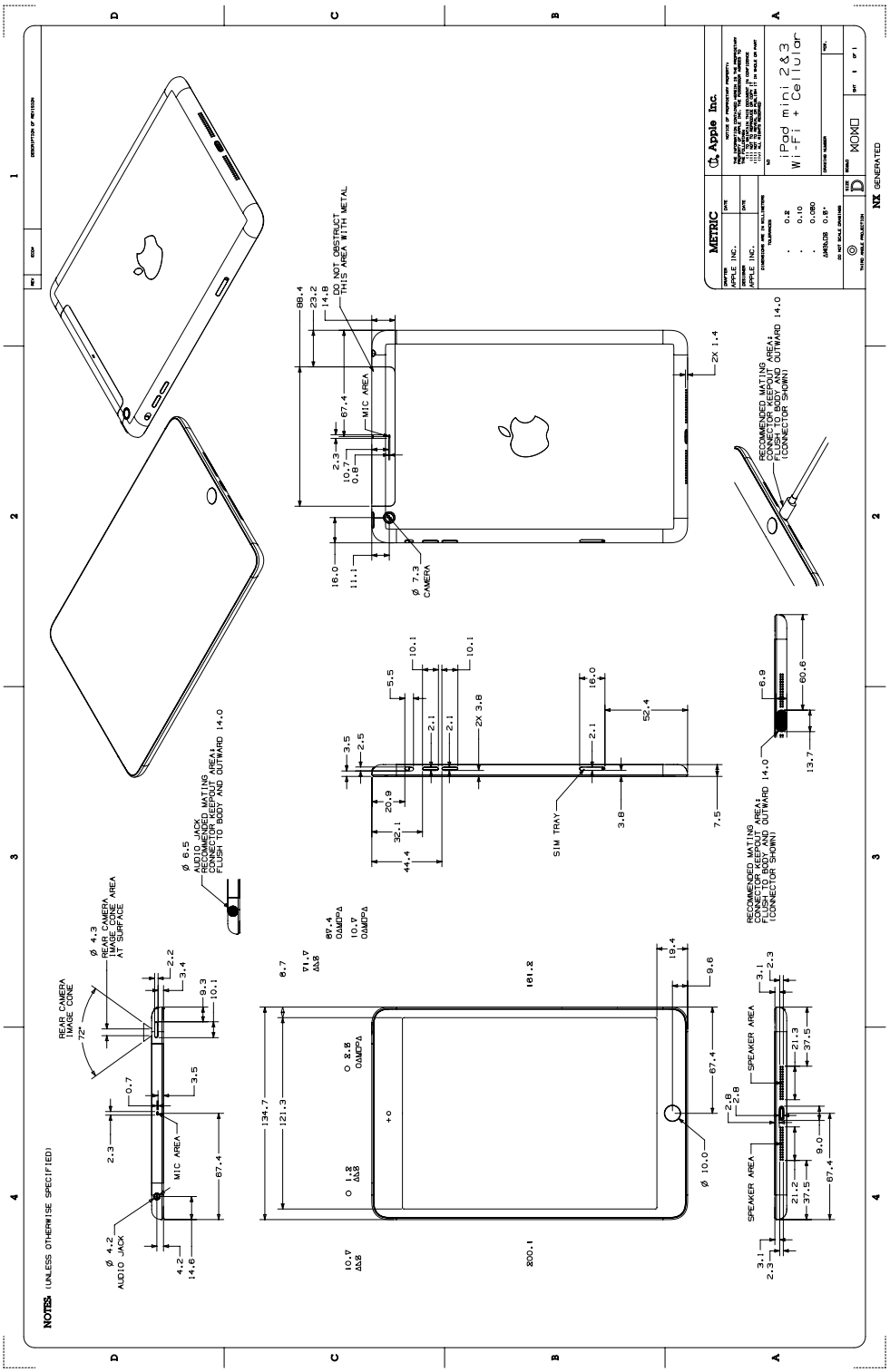
61.21 iPad mini 2 & 3 Wi-Fi

Figure 61-21 iPad mini 2 & iPad mini 3 Wi-Fi Dimensional Drawing



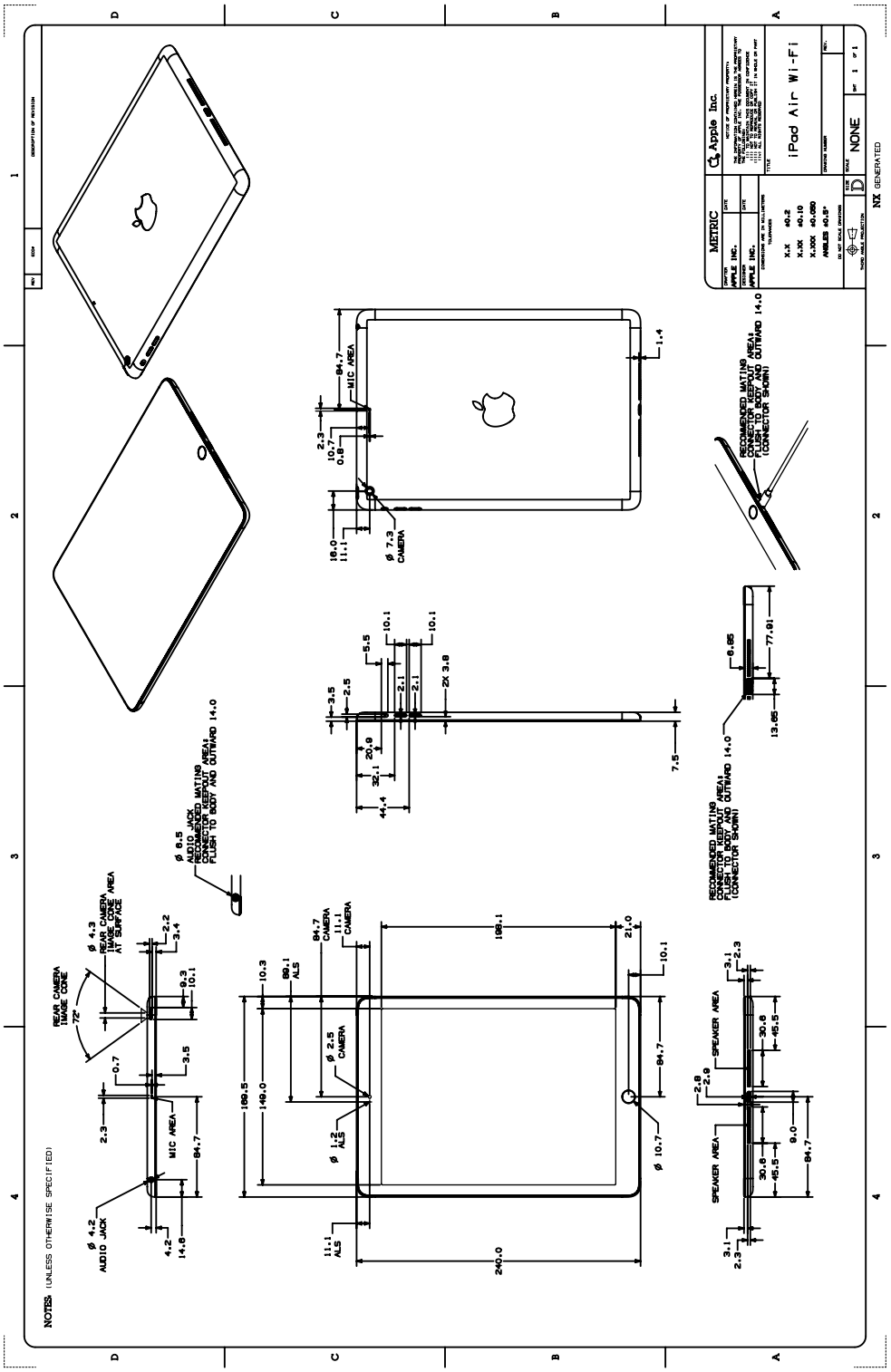
61.22 iPad mini 2 & 3 Wi-Fi + Cellular

Figure 61-22 iPad mini 2 & iPad mini 3 Wi-Fi + Cellular Dimensional Drawing



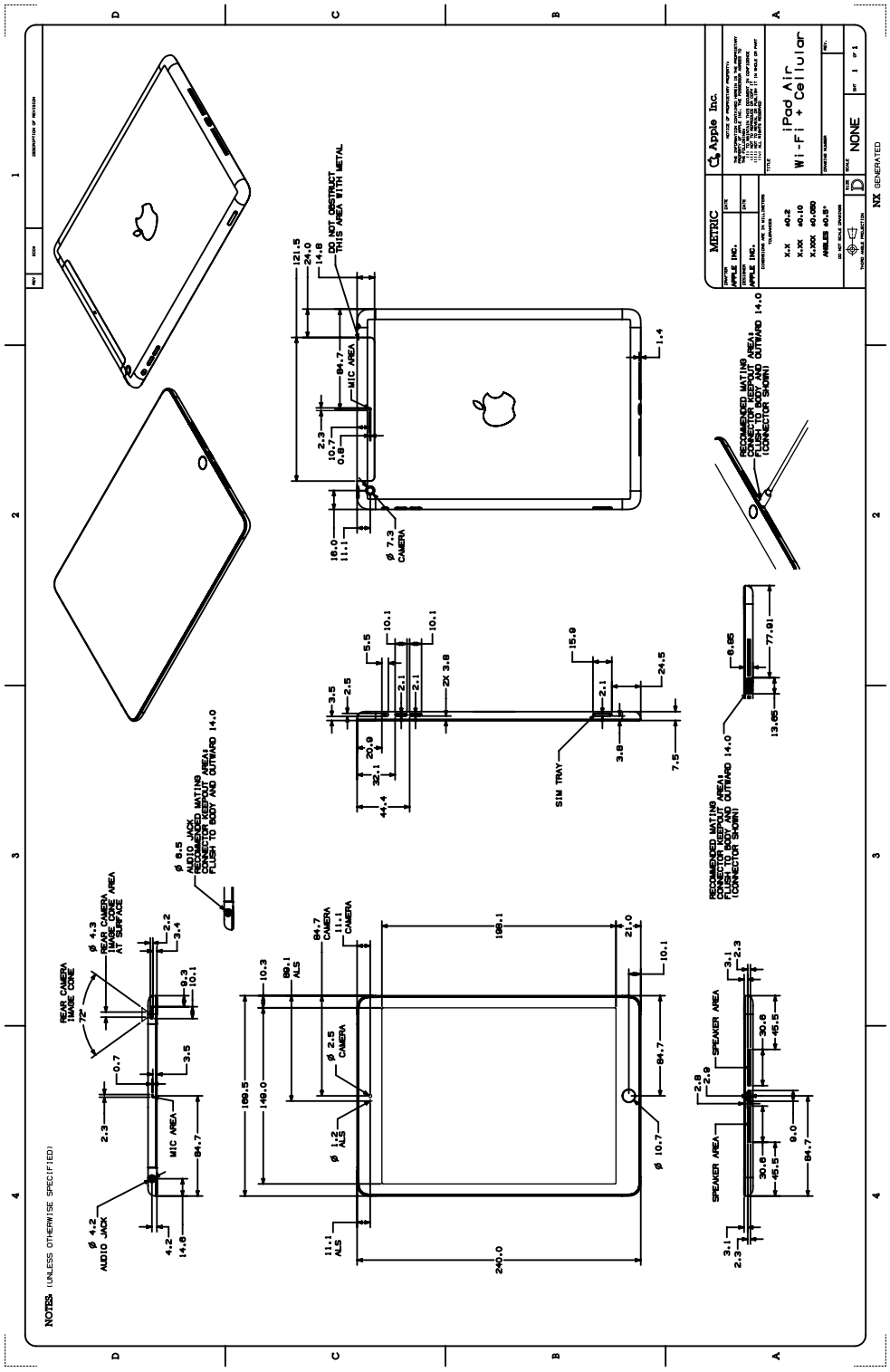
61.23 iPad Air Wi-Fi

Figure 61-23 iPad Air Wi-Fi Dimensional Drawing



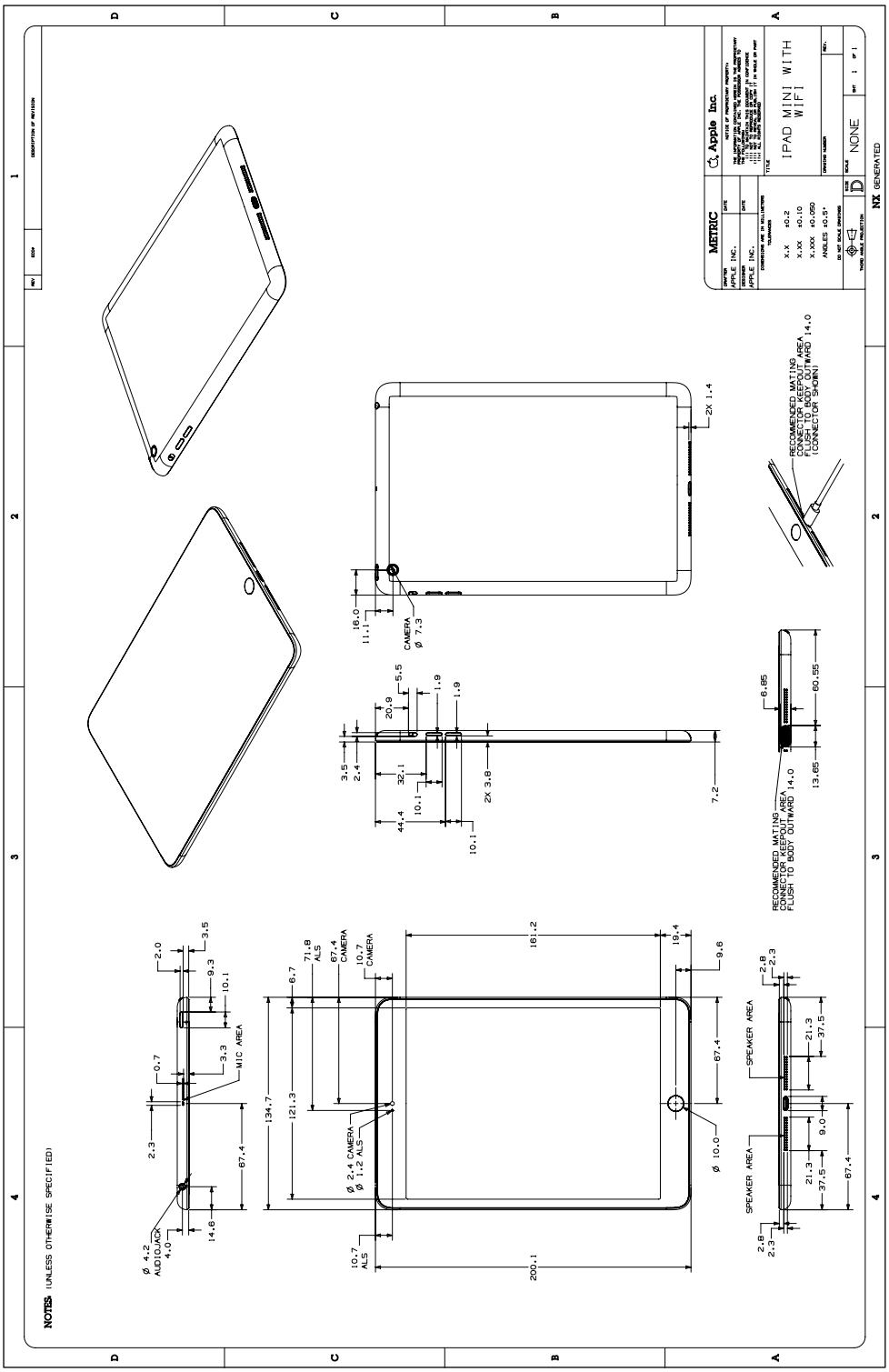
61.24 iPad Air Wi-Fi + Cellular

Figure 61-24 iPad Air Wi-Fi + Cellular Dimensional Drawing



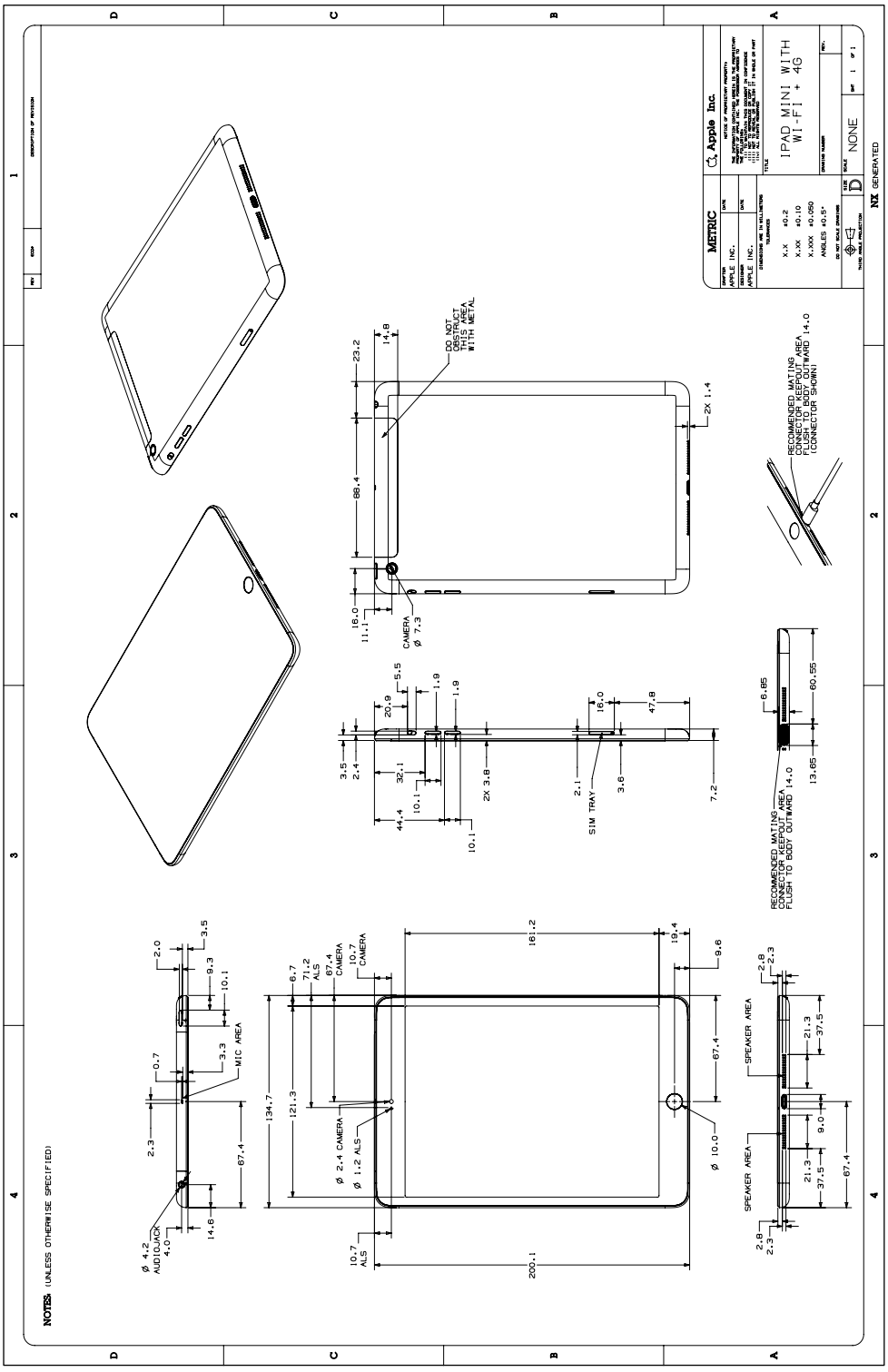
61.25 iPad mini with Wi-Fi

Figure 61-25 iPad mini with Wi-Fi Dimensional Drawing



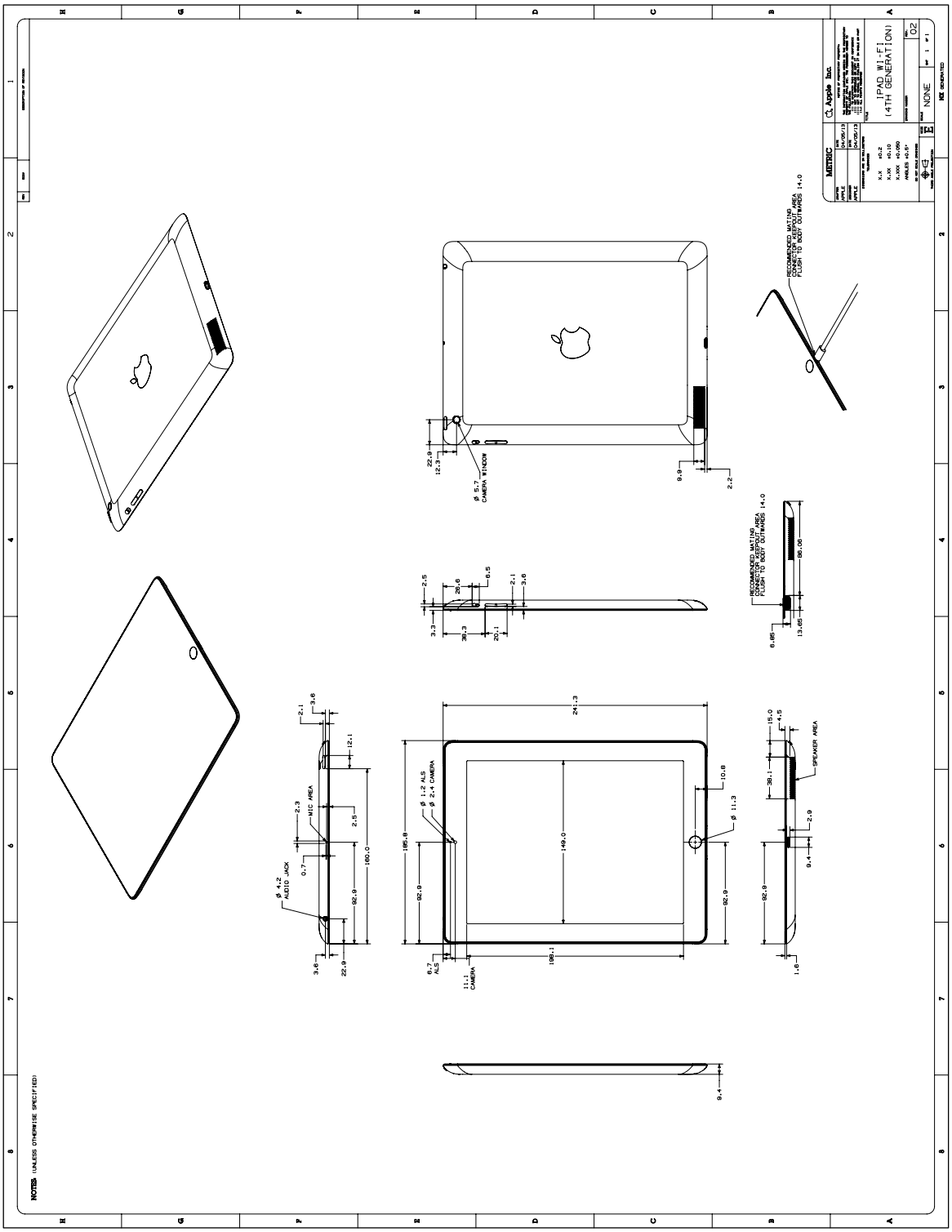
61.26 iPad mini with Wi-Fi + Cellular

Figure 61-26 iPad mini Wi-Fi + Cellular Dimensional Drawing



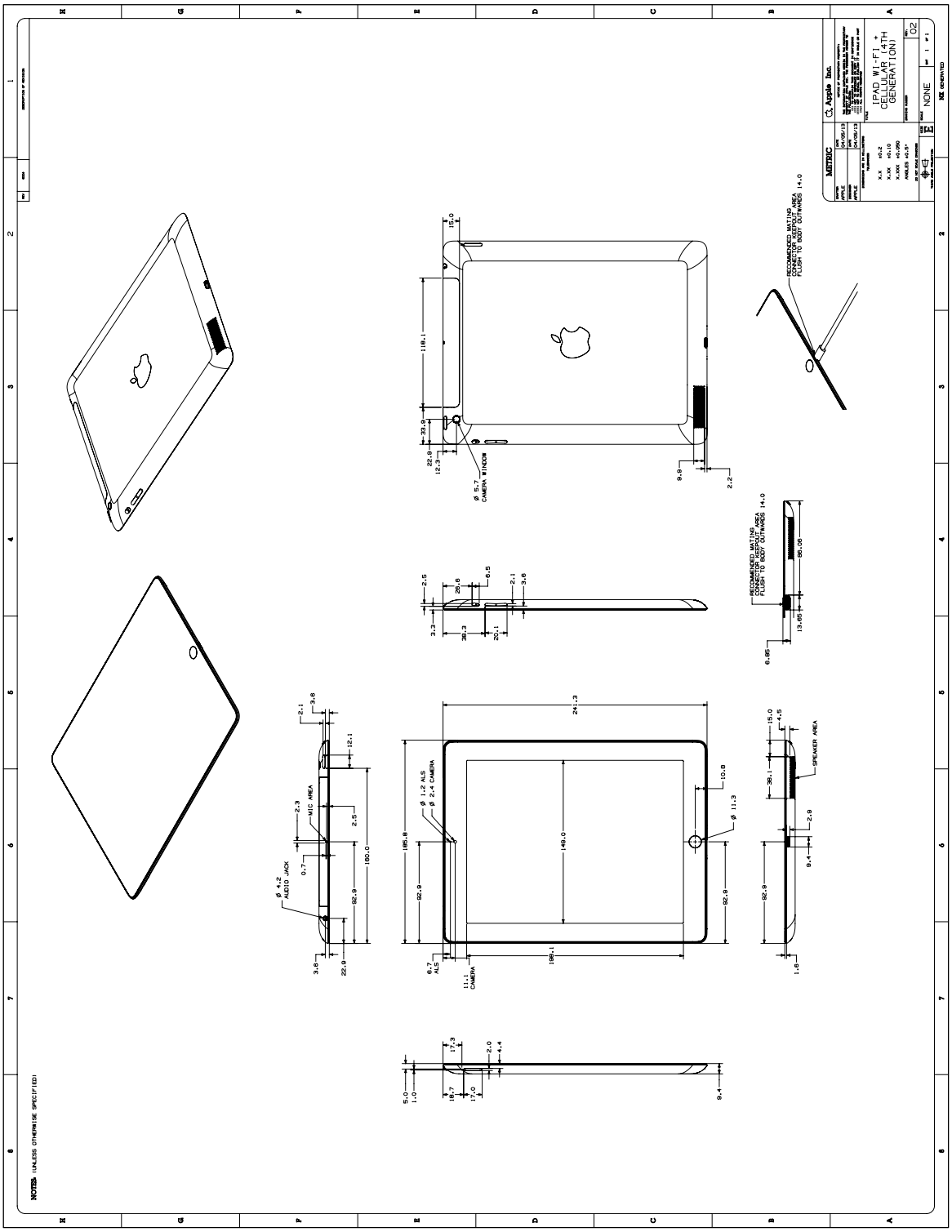
61.27 iPad with Wi-Fi (4th generation)

Figure 61-27 iPad Wi-Fi (4th generation) Dimensional Drawing



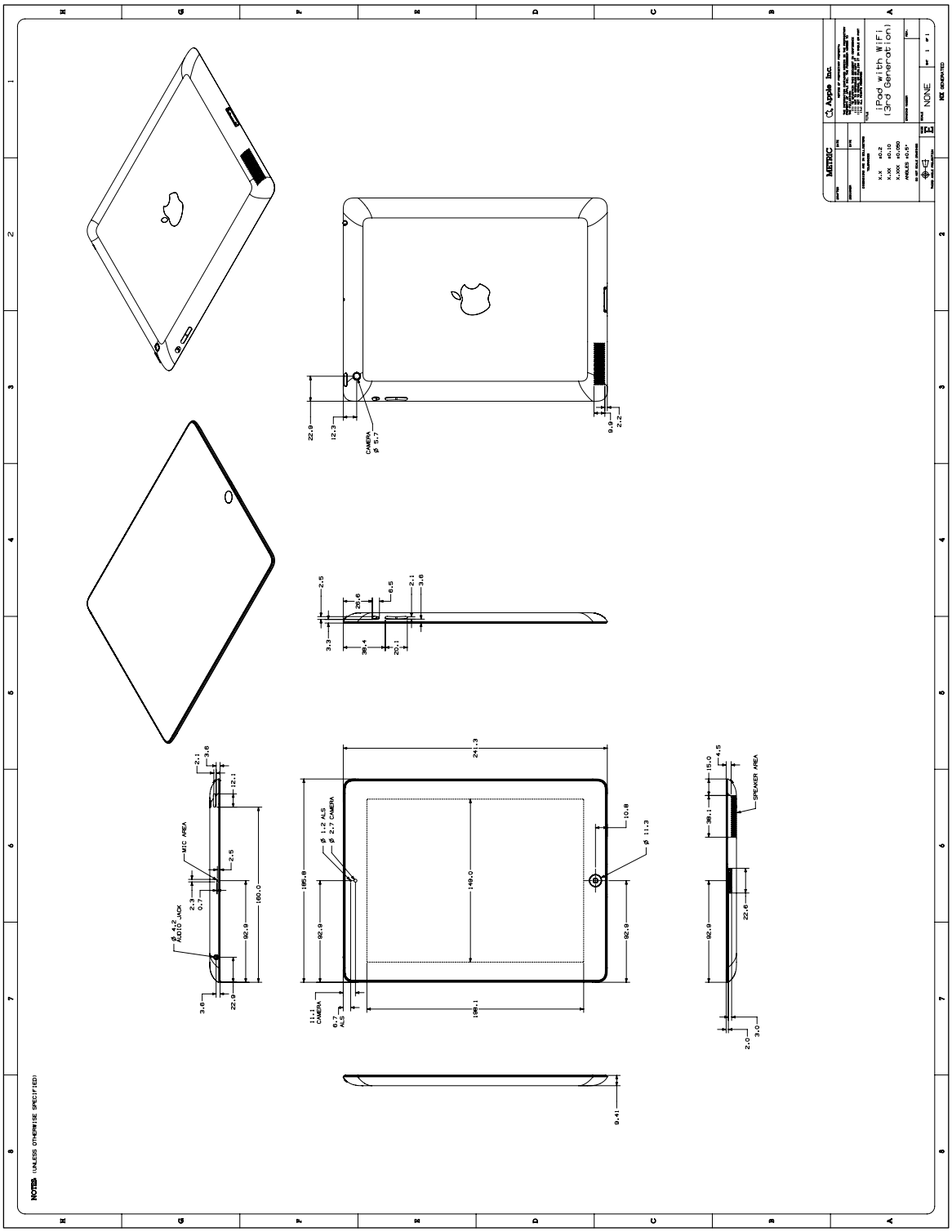
61.28 iPad with Wi-Fi + Cellular (4th generation)

Figure 61-28 iPad Wi-Fi + Cellular (4th generation) Dimensional Drawing



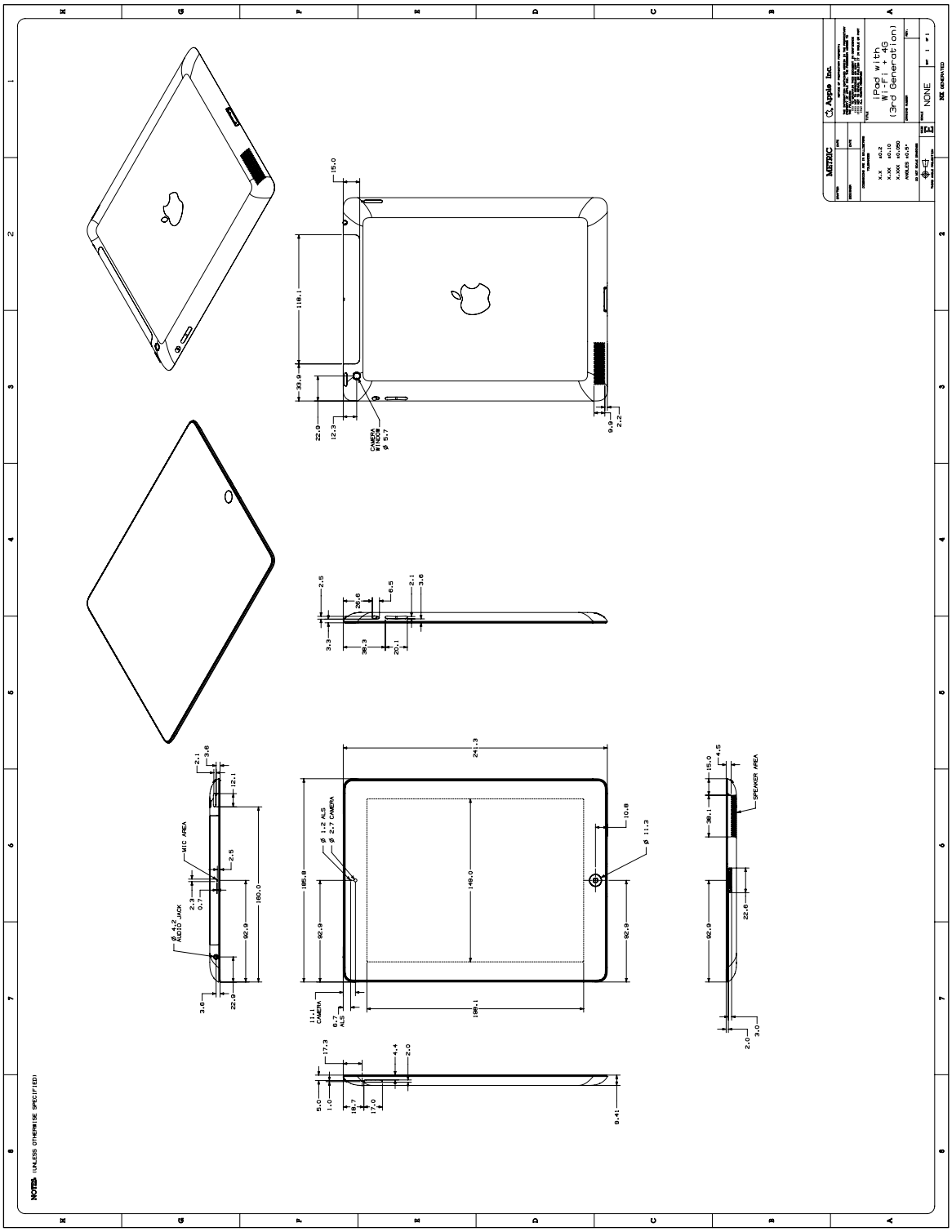
61.29 iPad with Wi-Fi (3rd generation)

Figure 61-29 iPad Wi-Fi (3rd Generation) Dimensional Drawing



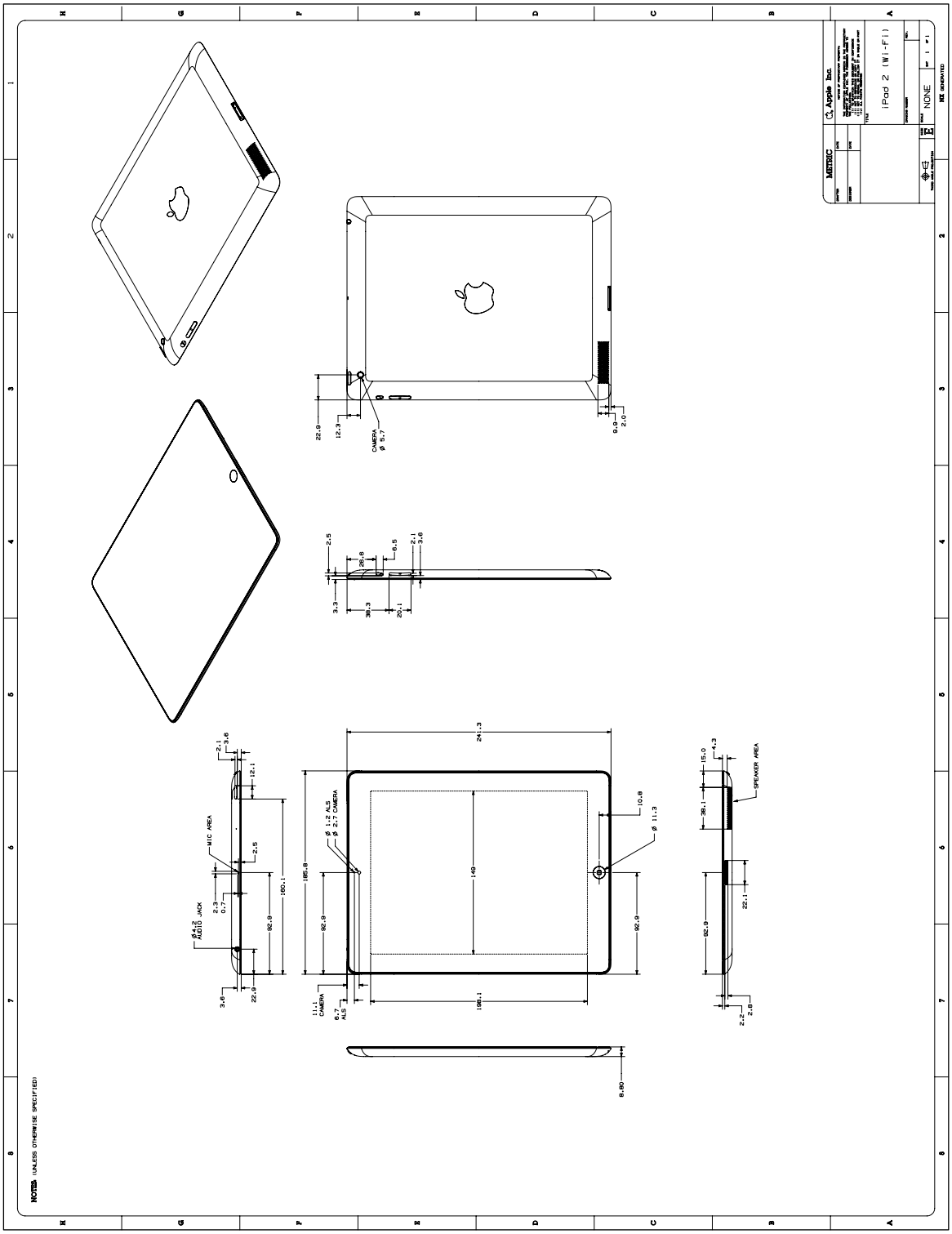
61.30 iPad with Wi-Fi + 4G (3rd generation)

Figure 61-30 iPad Wi-Fi + 4G (3rd Generation) Dimensional Drawing



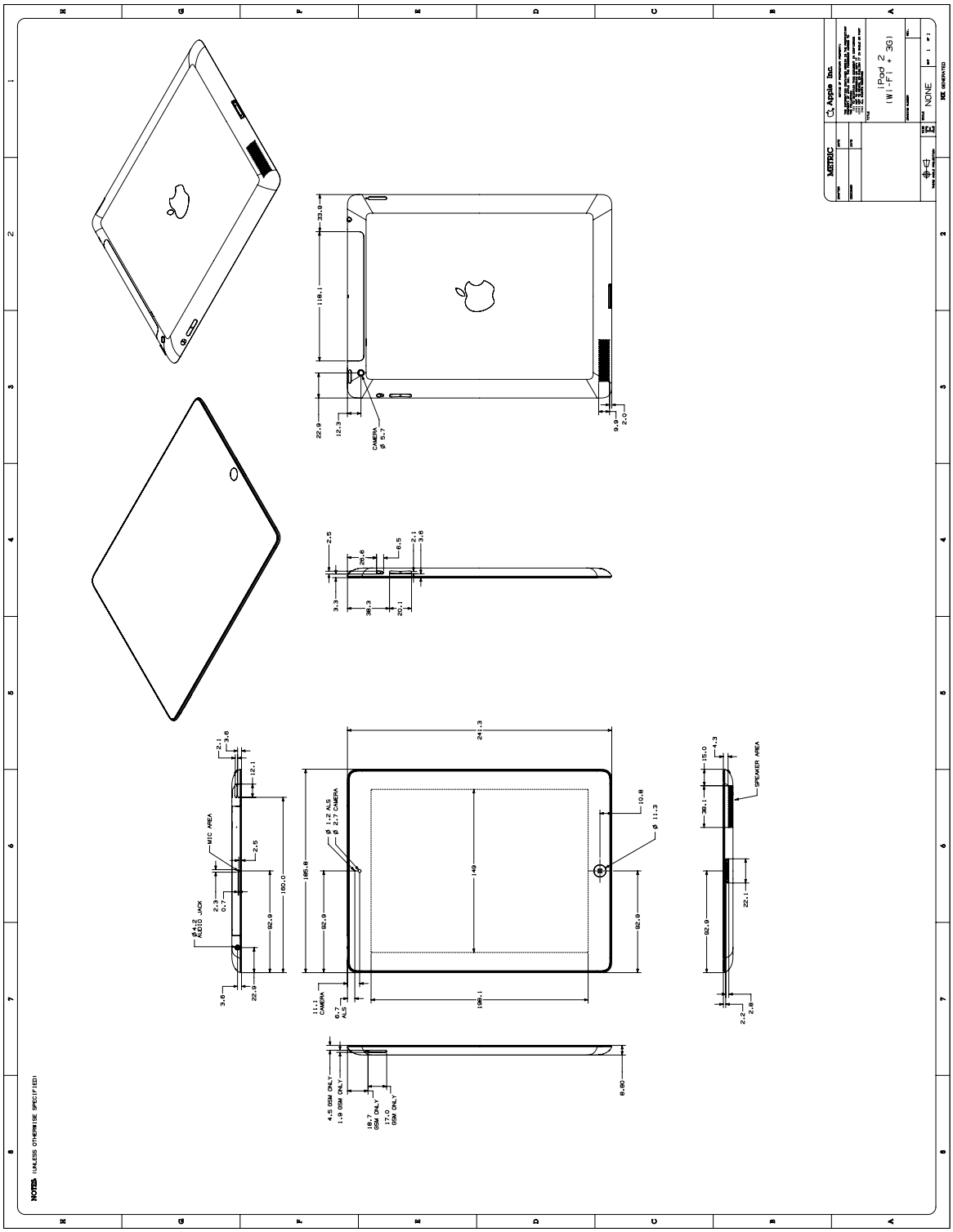
61.31 iPad 2 with Wi-Fi

Figure 61-31 iPad 2 Wi-Fi Dimensional Drawing



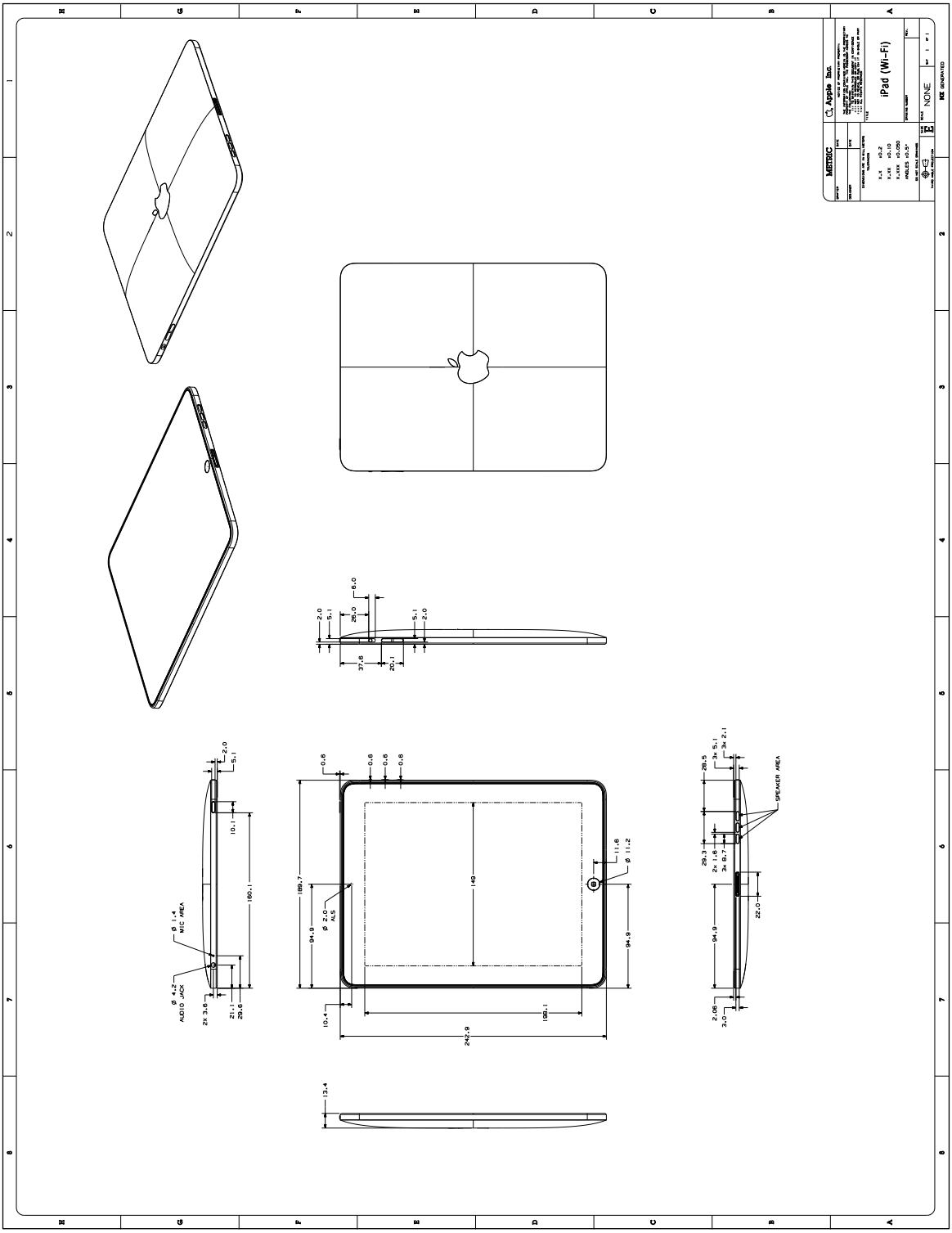
61.32 iPad 2 with Wi-Fi + 3G

Figure 61-32 iPad 2 Wi-Fi + 3G Dimensional Drawing



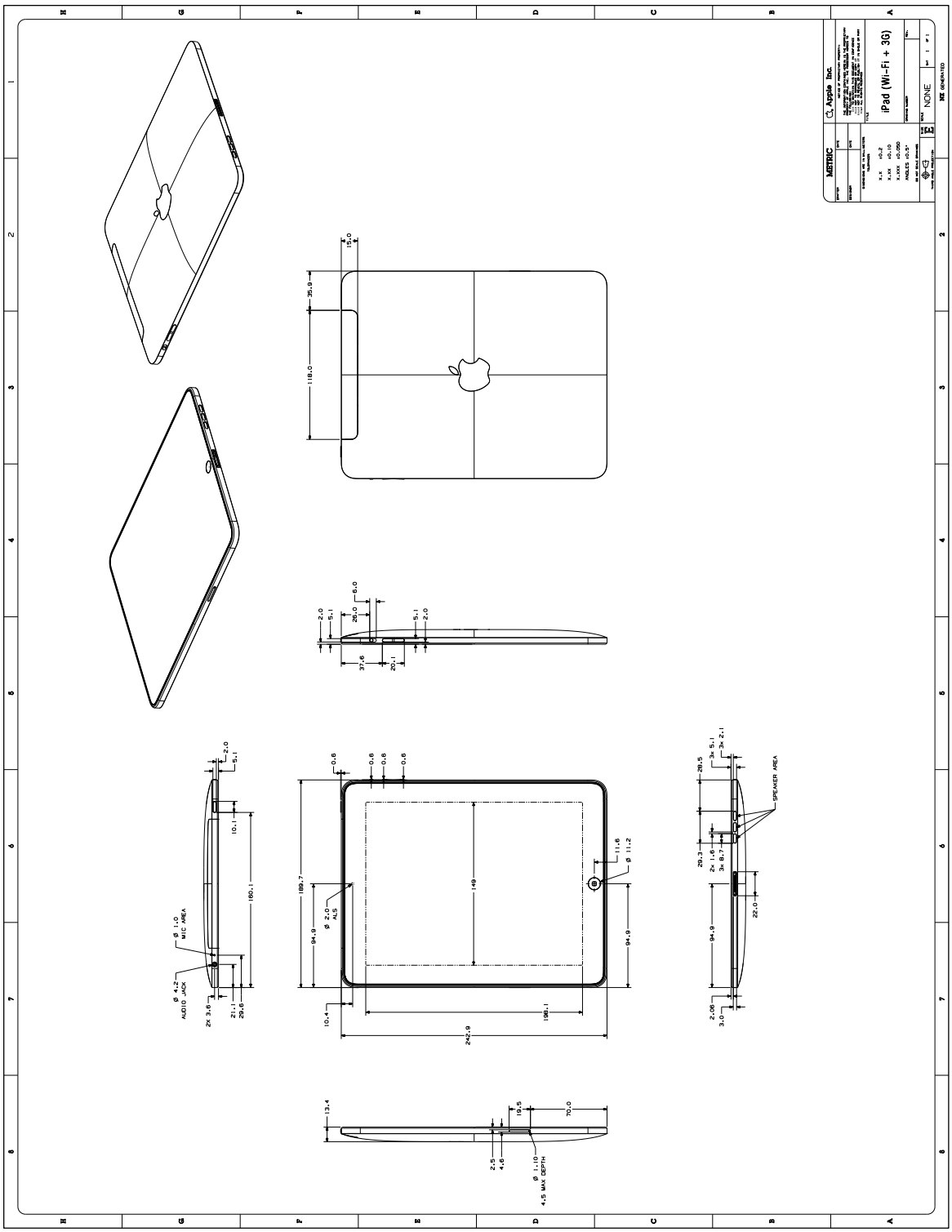
61.33 iPad with Wi-Fi

Figure 61-33 iPad Wi-Fi Dimensional Drawing



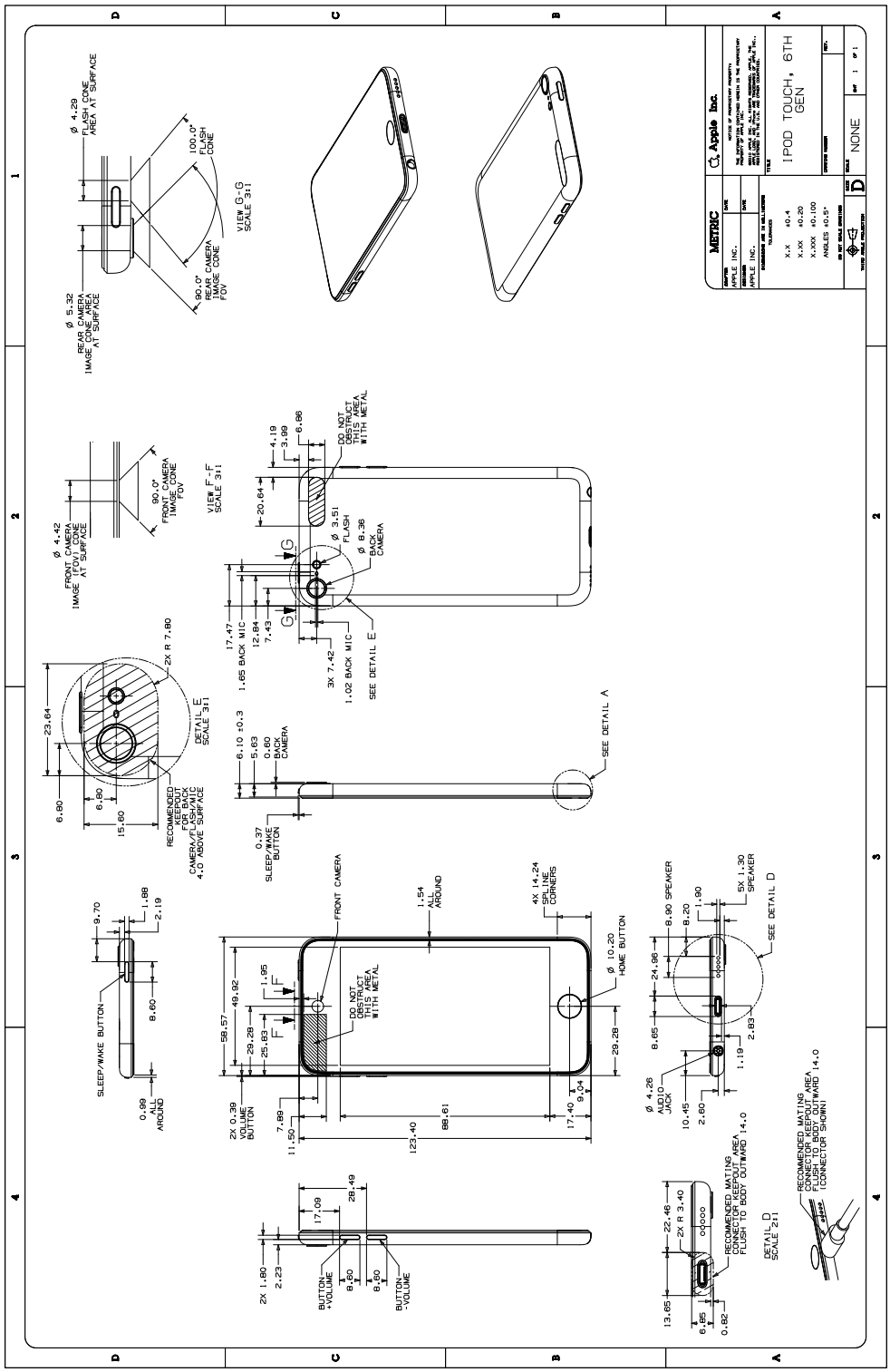
61.34 iPad with Wi-Fi + 3G

Figure 61-34 iPad Wi-Fi + 3G Dimensional Drawing



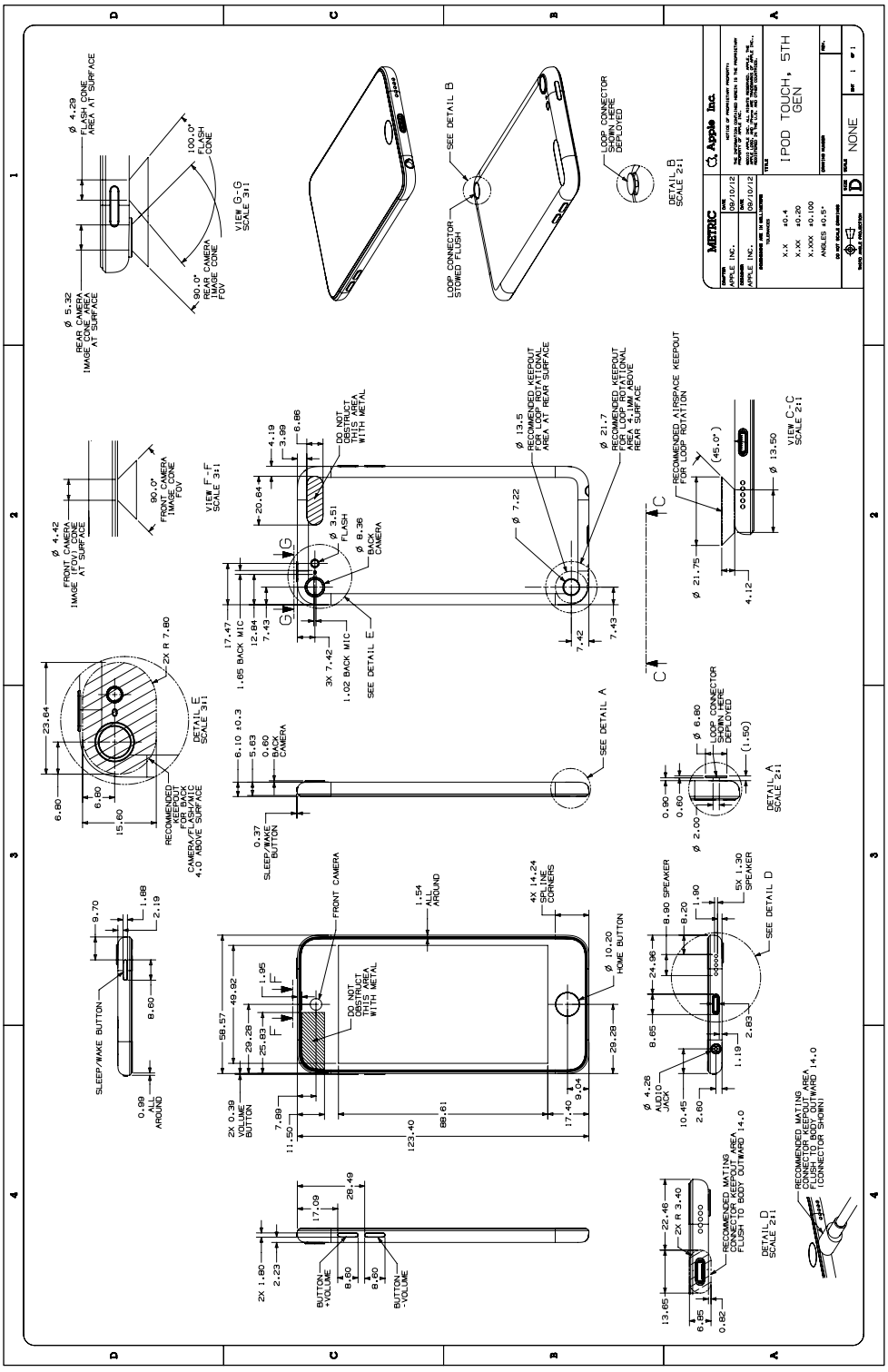
61.35 iPod touch (6th generation)

Figure 61-35 iPod touch 6th gen. Dimensional Drawing



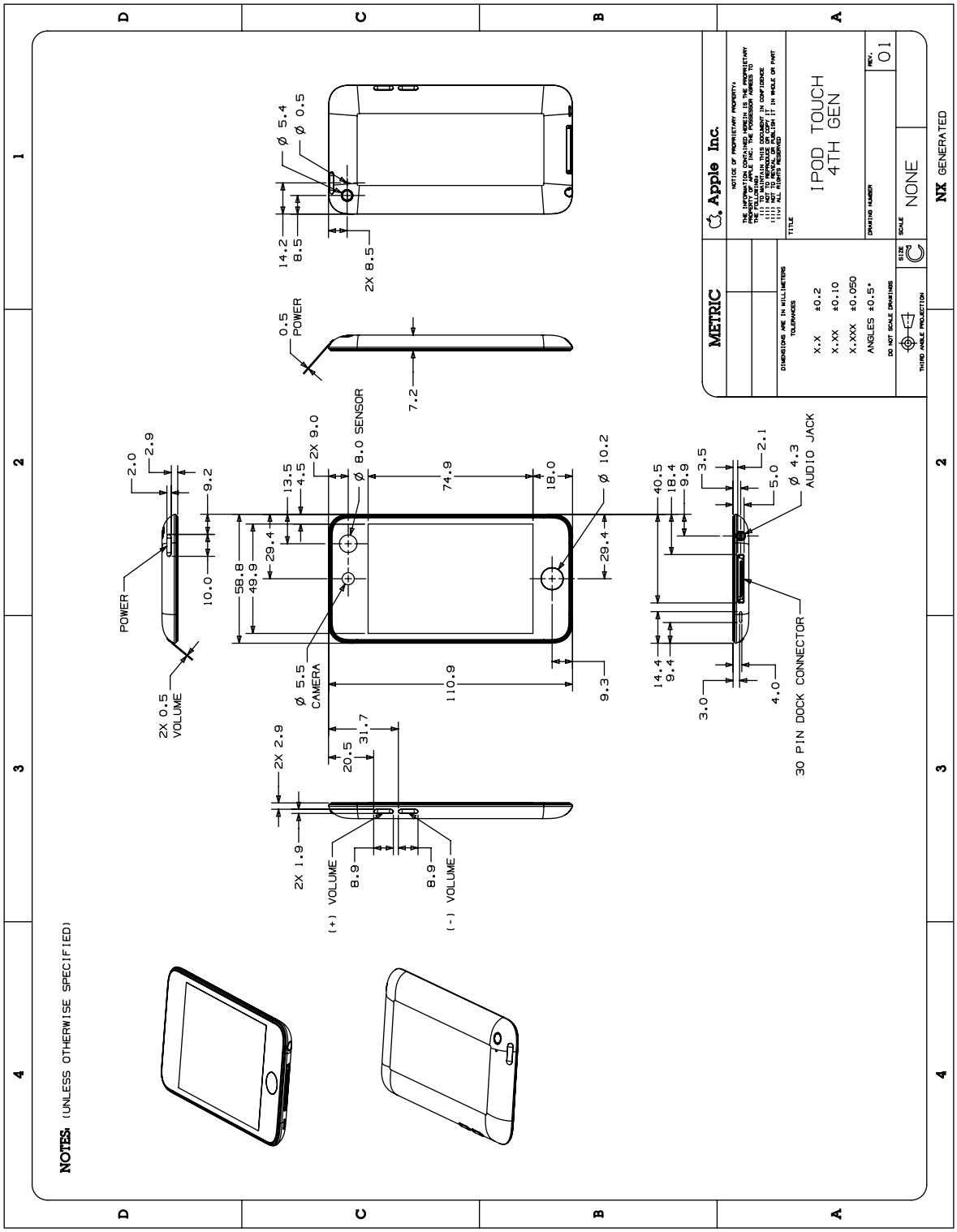
61.36 iPod touch (5th generation)

Figure 61-36 iPod touch 5th gen. Dimensional Drawing



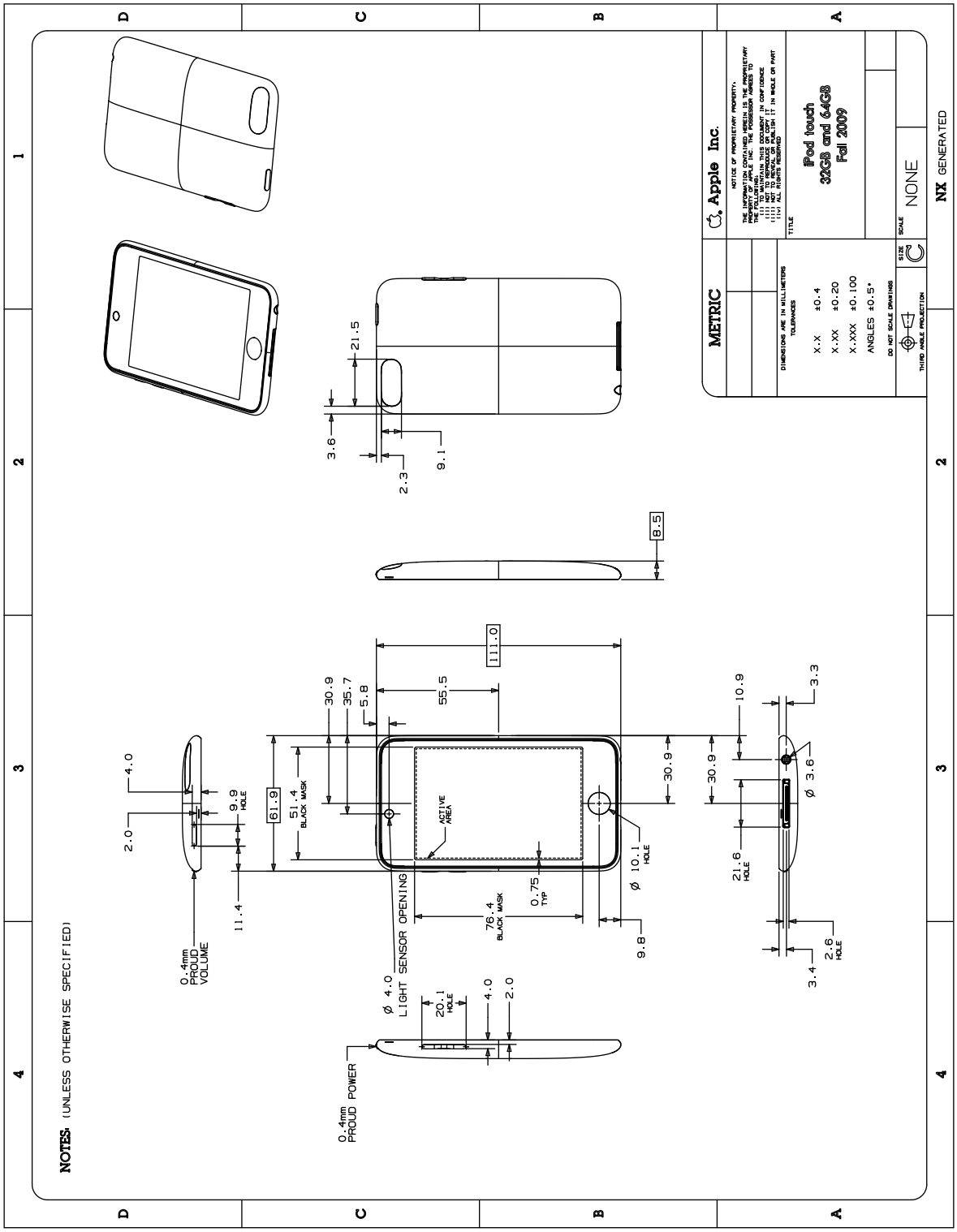
61.37 iPod touch (4th generation)

Figure 61-37 iPod touch 4th gen. Dimensional Drawing



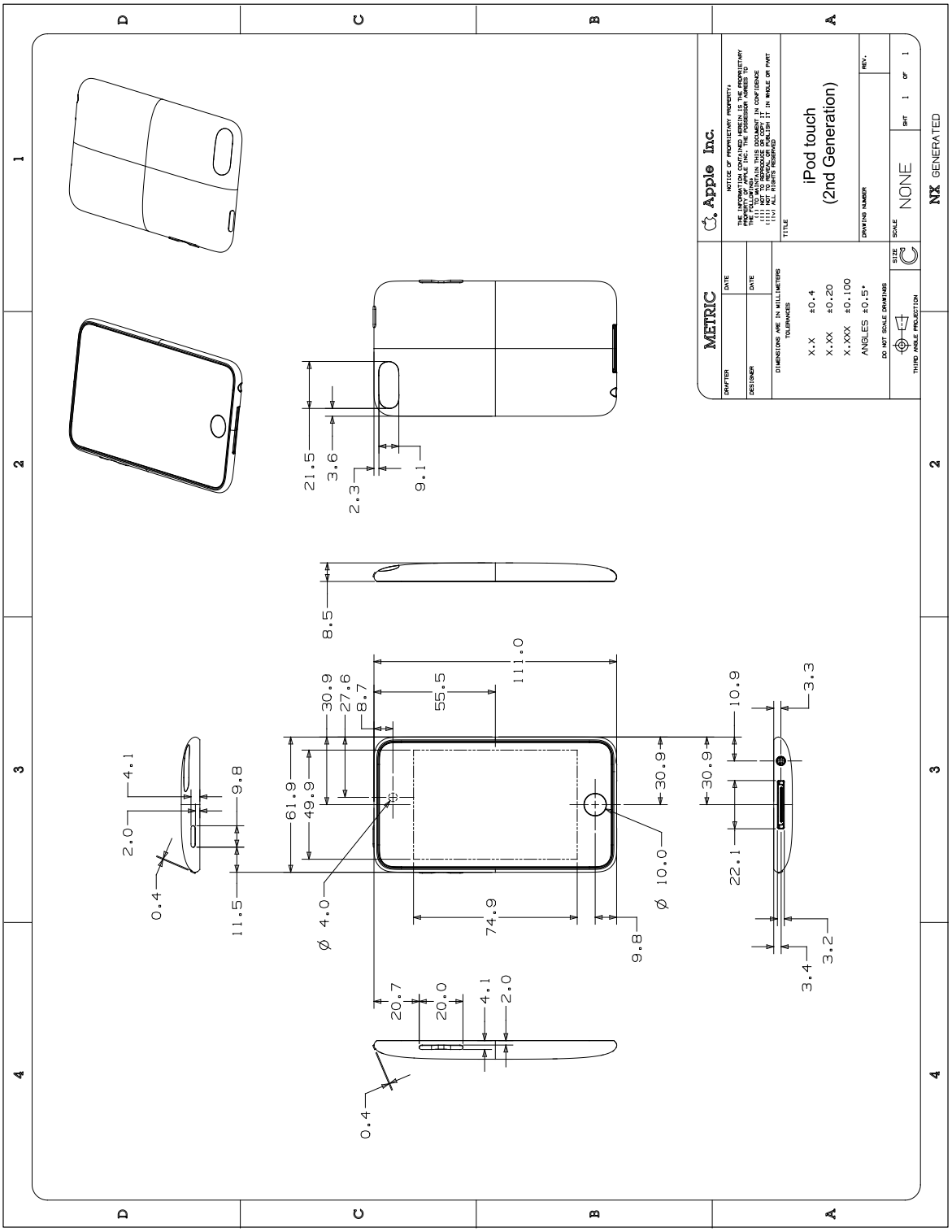
61.38 iPod touch (3rd generation)

Figure 61-38 iPod touch 3rd gen. Fall '09 32GB and 64GB Dimensional Drawing



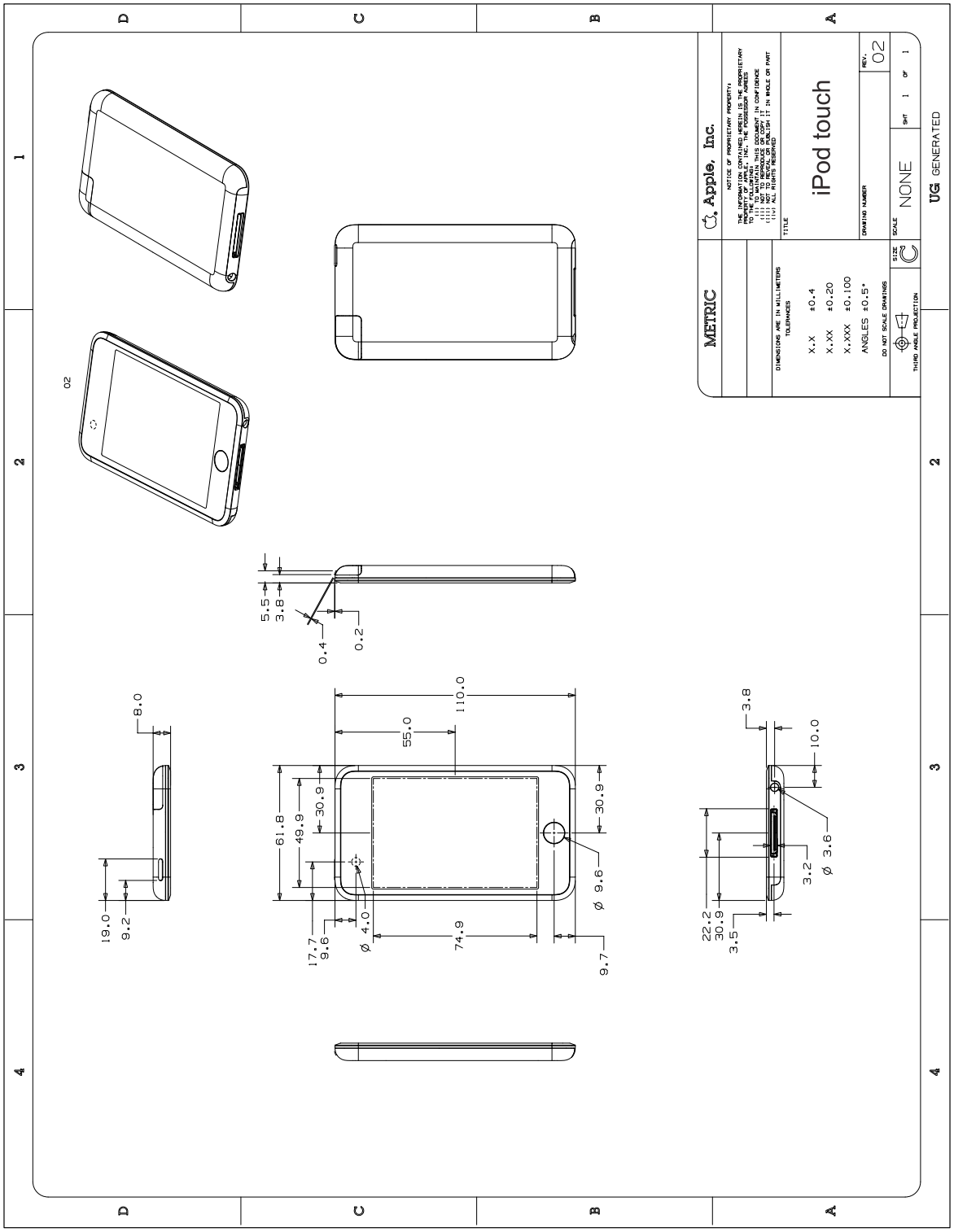
61.39 iPod touch (2nd generation)

Figure 61-39 iPod touch 2nd gen. 8GB, 16GB, 32GB Dimensional Drawing



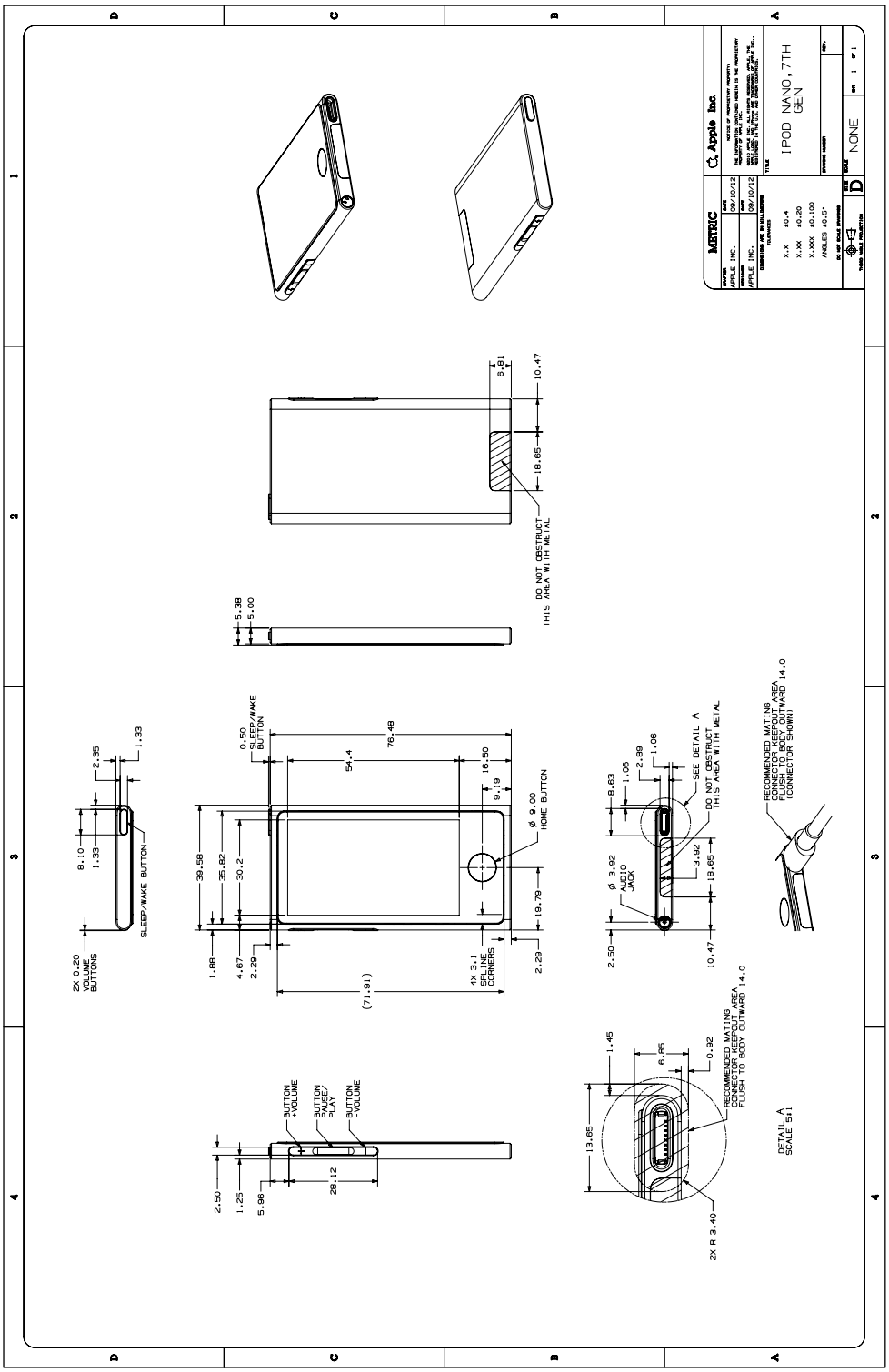
61.40 iPod touch

Figure 61-40 iPod touch Dimensional Drawing



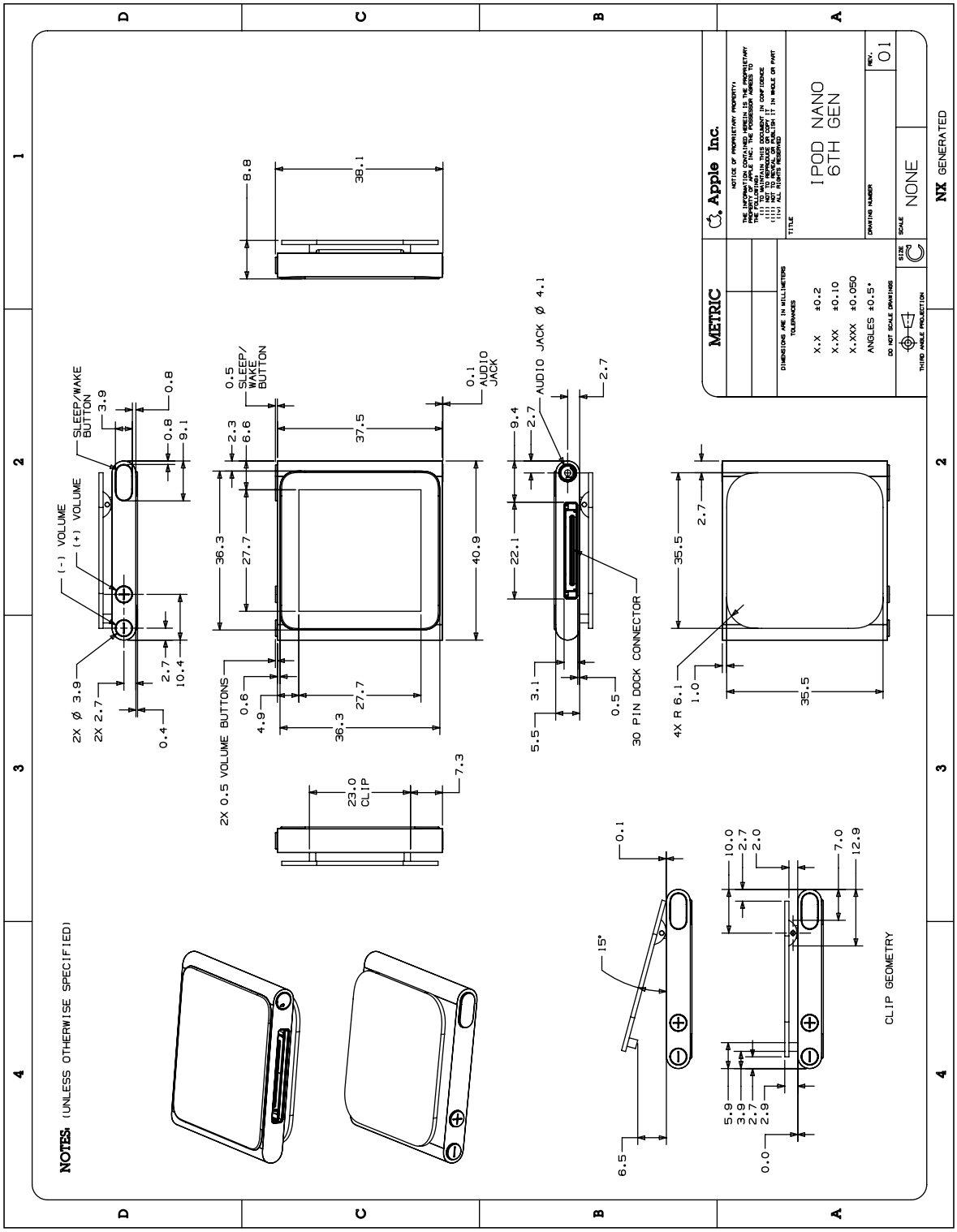
61.41 iPod nano (7th generation)

Figure 61-41 iPod nano 7th gen. Dimensional Drawing



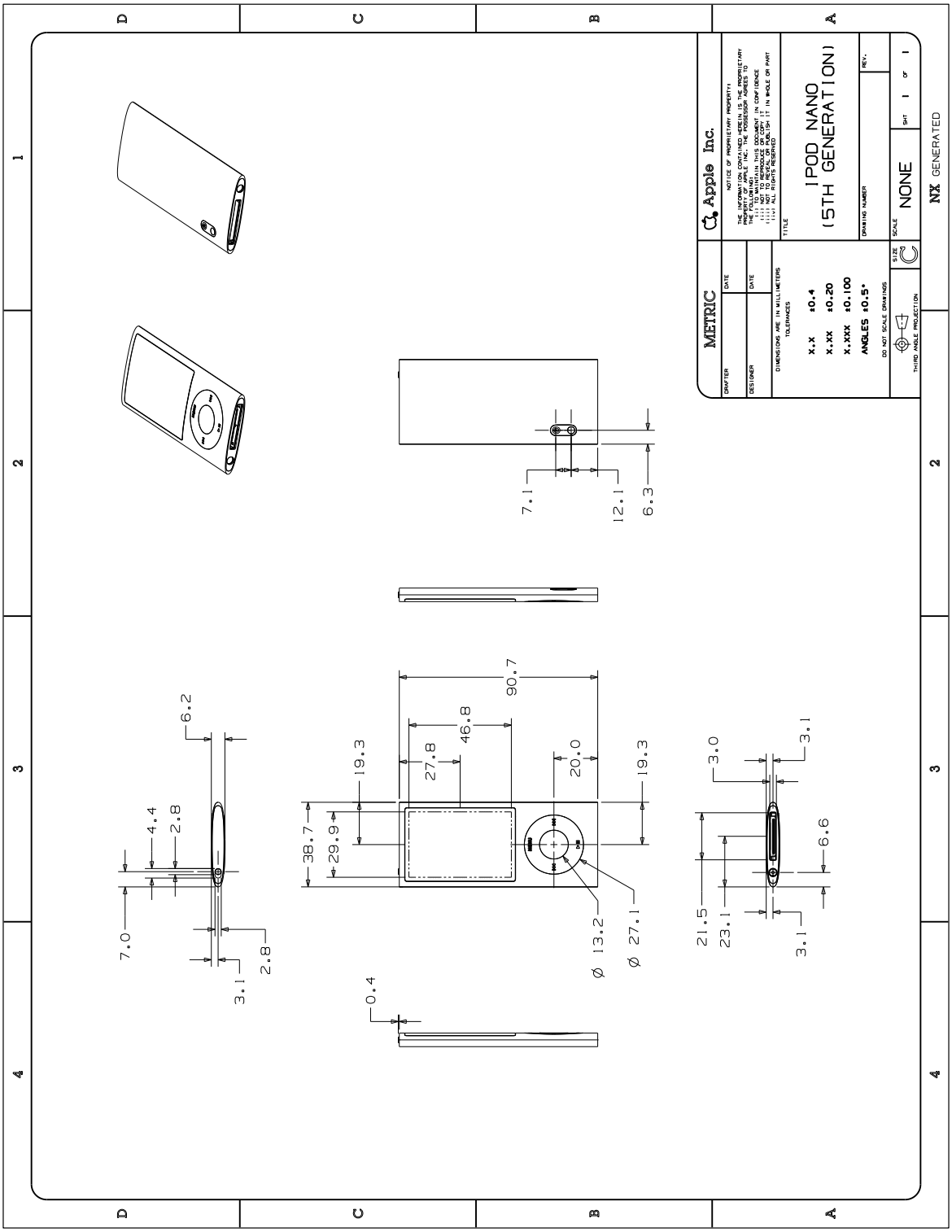
61.42 iPod nano (6th generation)

Figure 61-42 iPod nano 6th gen. Dimensional Drawing



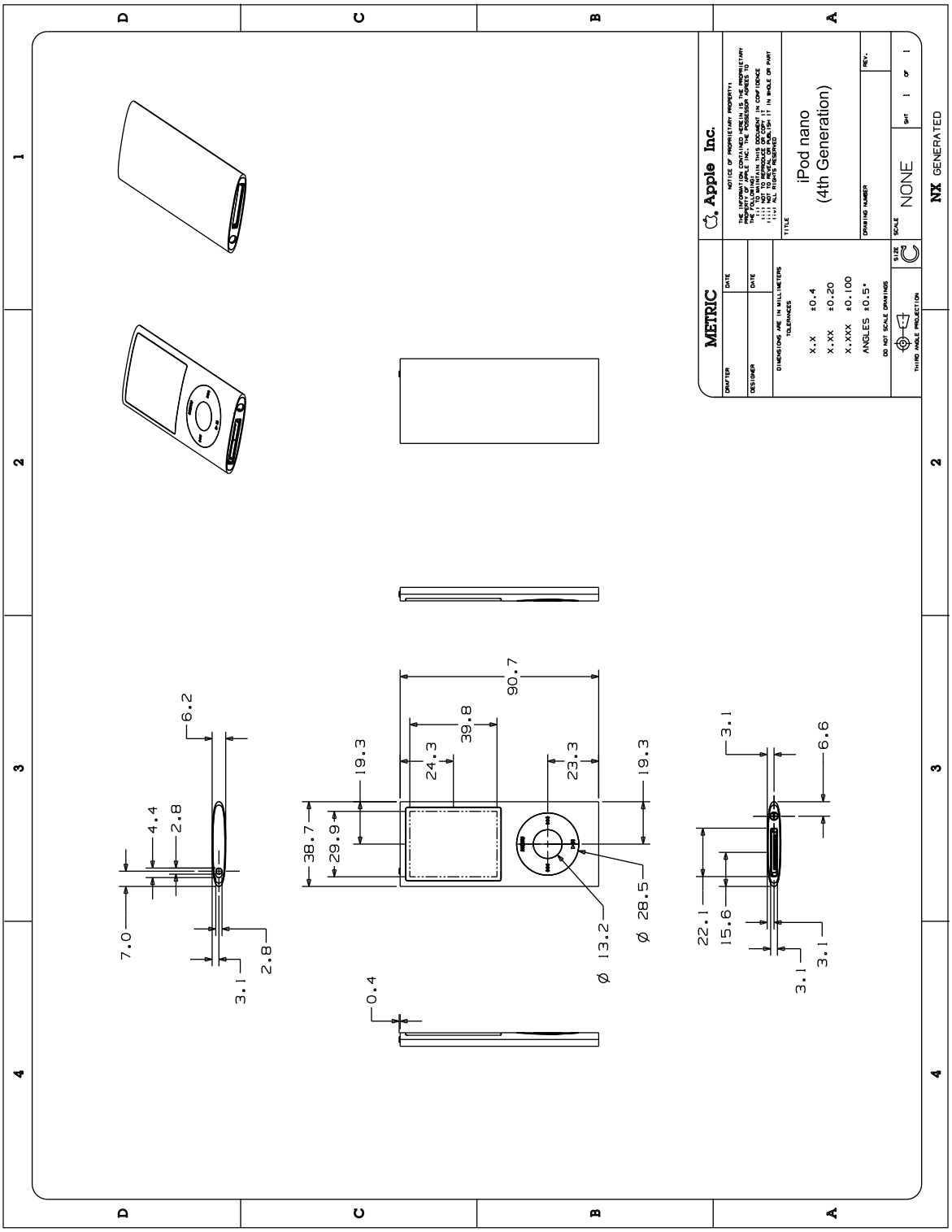
61.43 iPod nano (5th generation)

Figure 61-43 iPod nano 5th gen. Dimensional Drawing



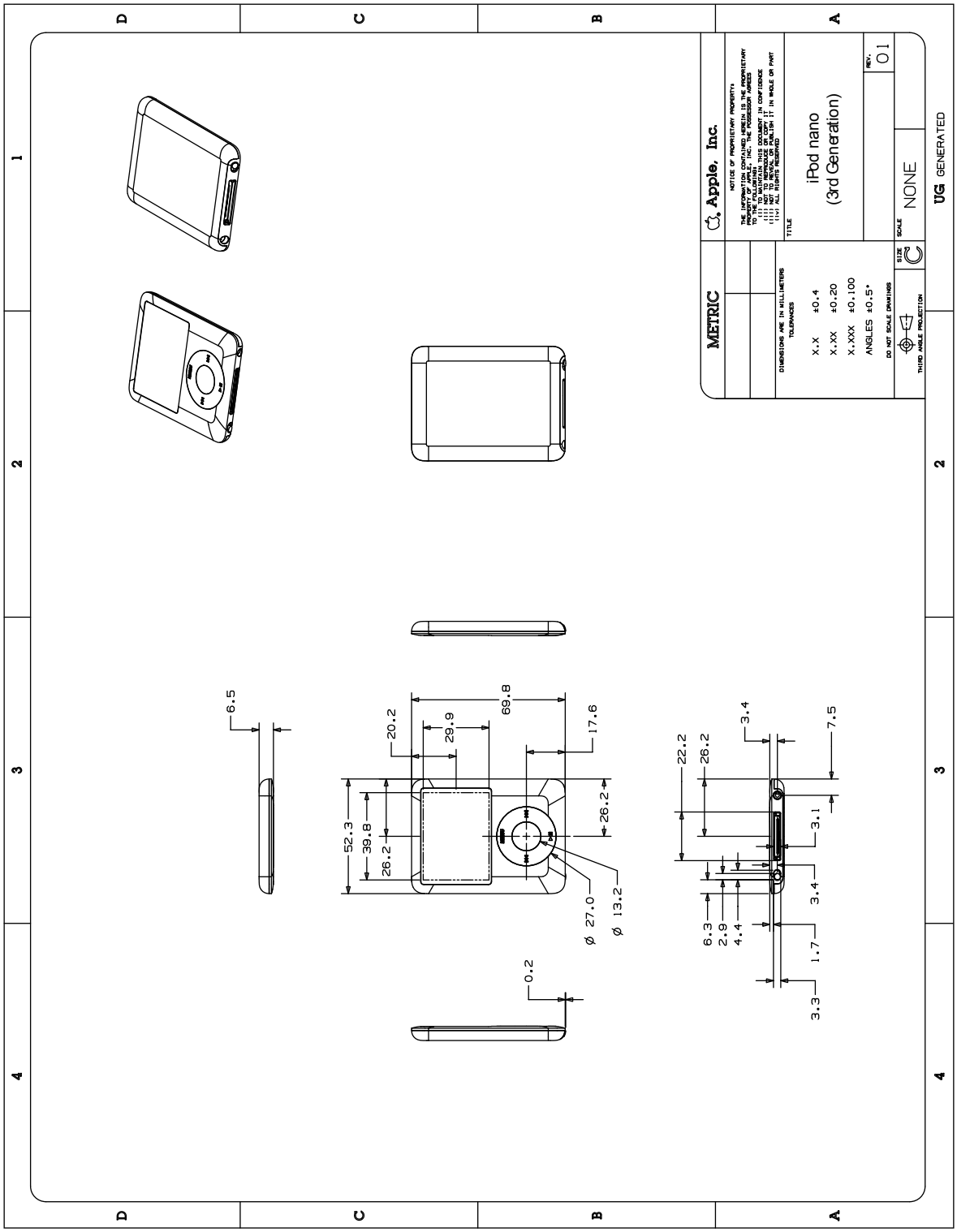
61.44 iPod nano (4th generation)

Figure 61-44 iPod nano 4th gen. Dimensional Drawing



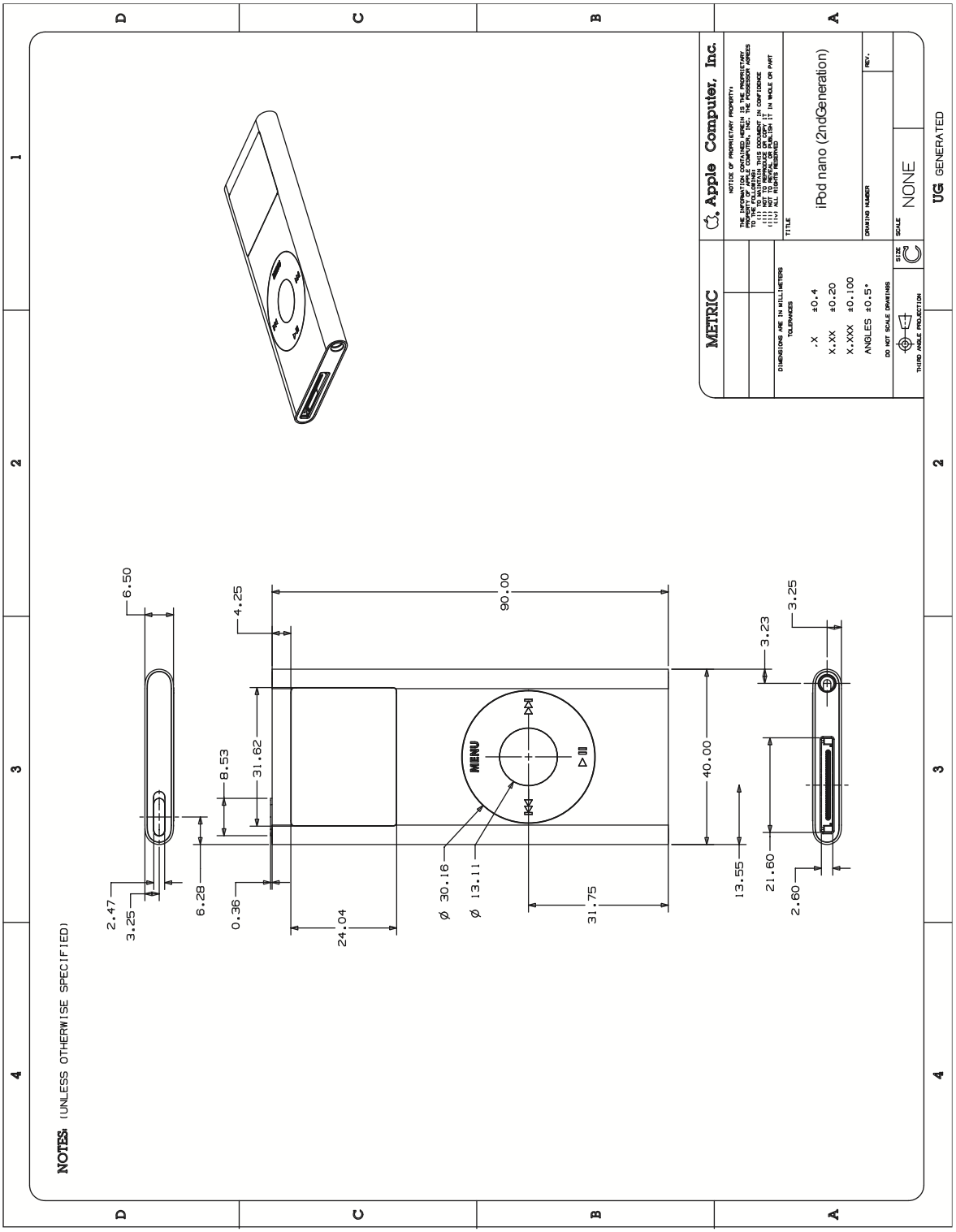
61.45 iPod nano (3rd generation)

Figure 61-45 iPod nano 3rd gen. Dimensional Drawing



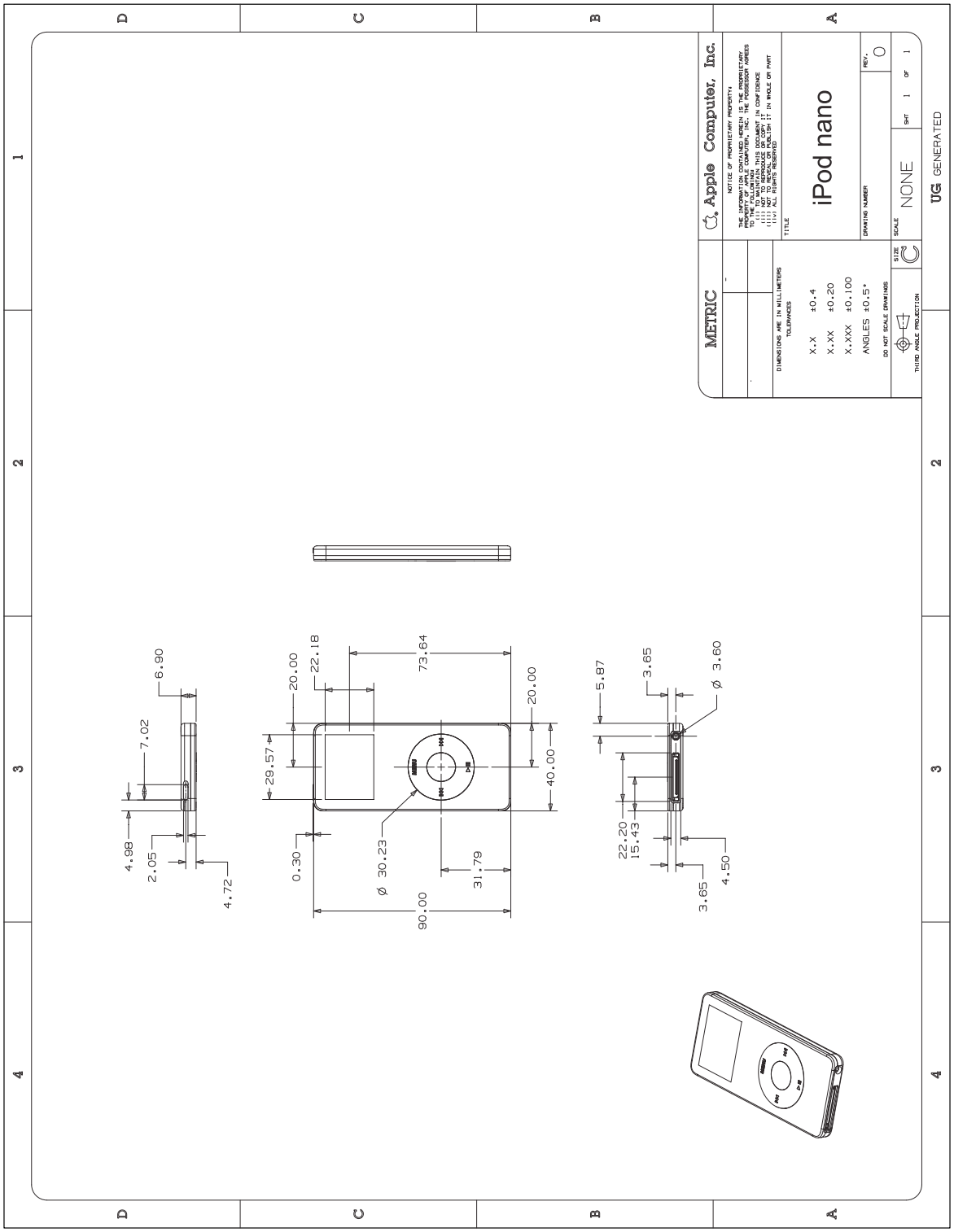
61.46 iPod nano (2nd generation)

Figure 61-46 iPod nano 2nd gen. Dimensional Drawing



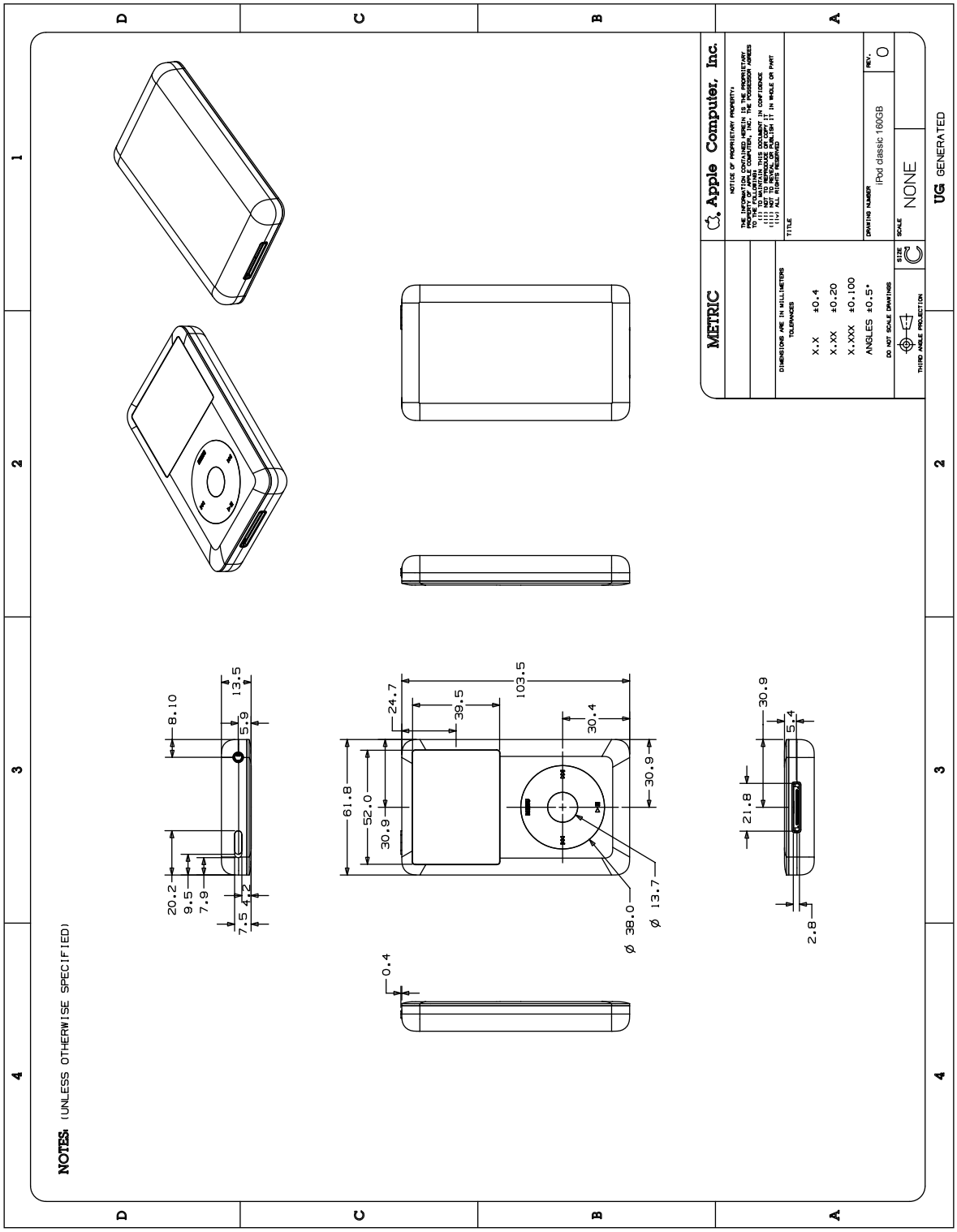
61.47 iPod nano

Figure 61-47 iPod nano Dimensional Drawing



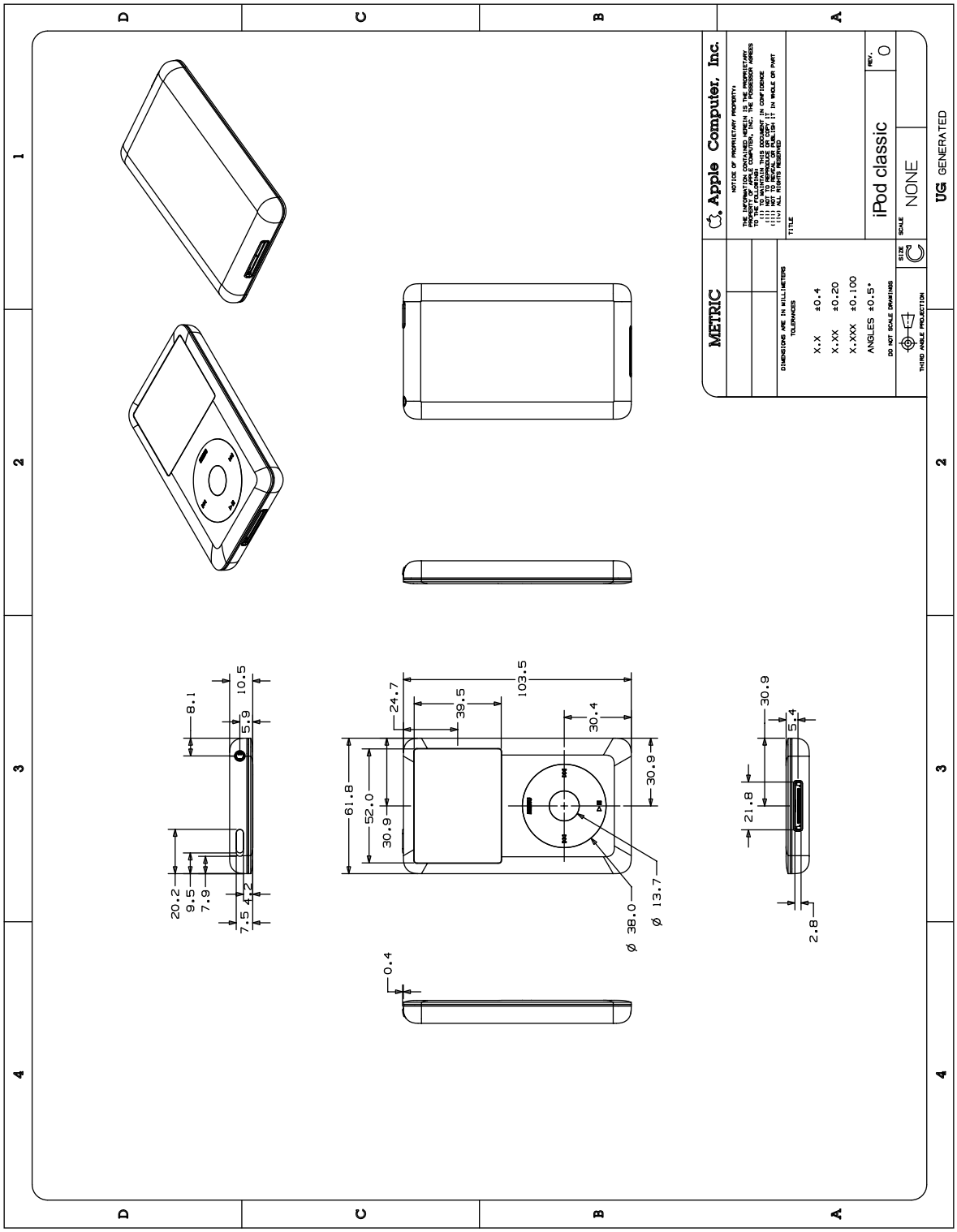
61.48 iPod classic 160GB

Figure 61-48 iPod classic 160GB Dimensional Drawing



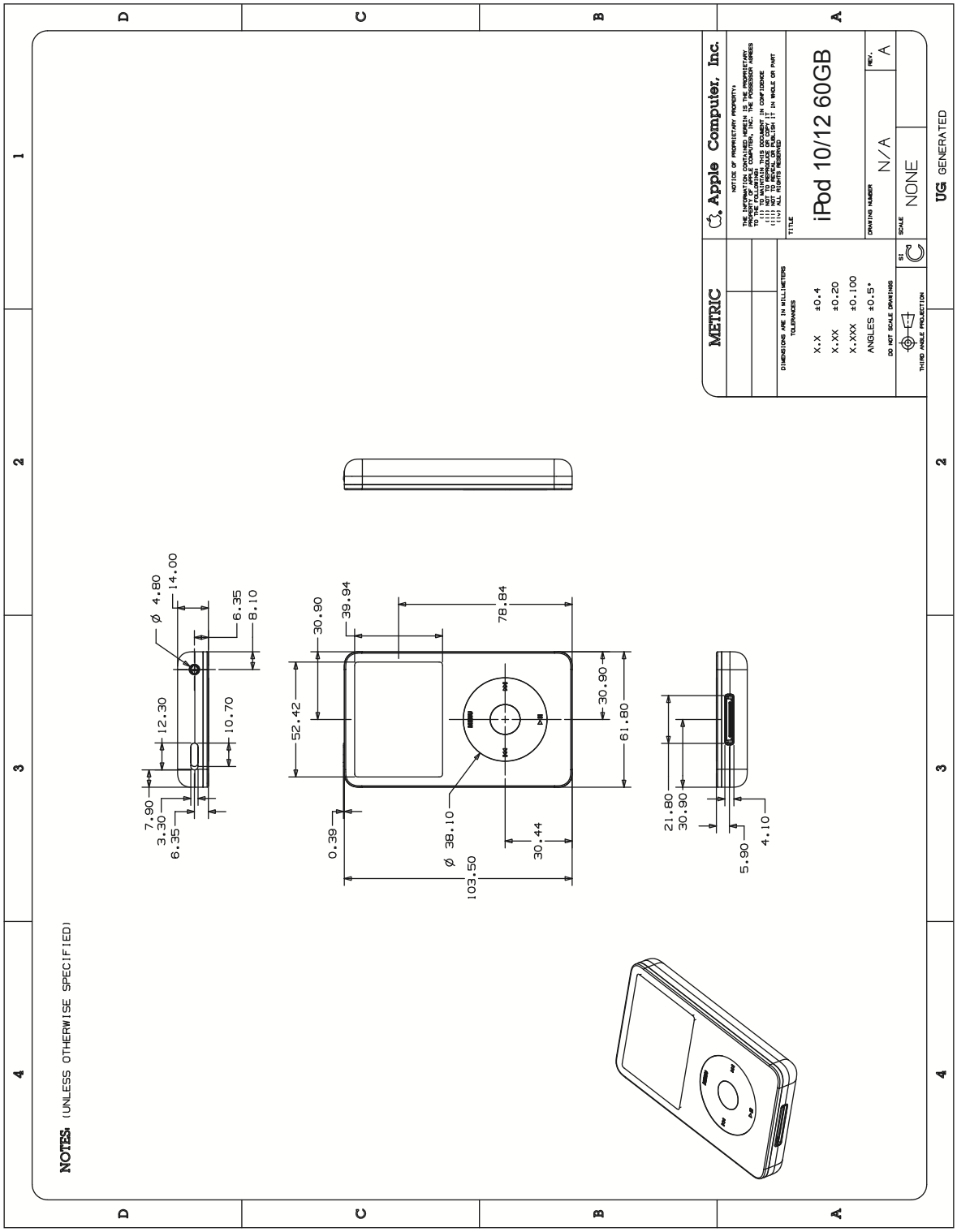
61.49 iPod classic 80GB

Figure 61-49 iPod classic 80GB Dimensional Drawing



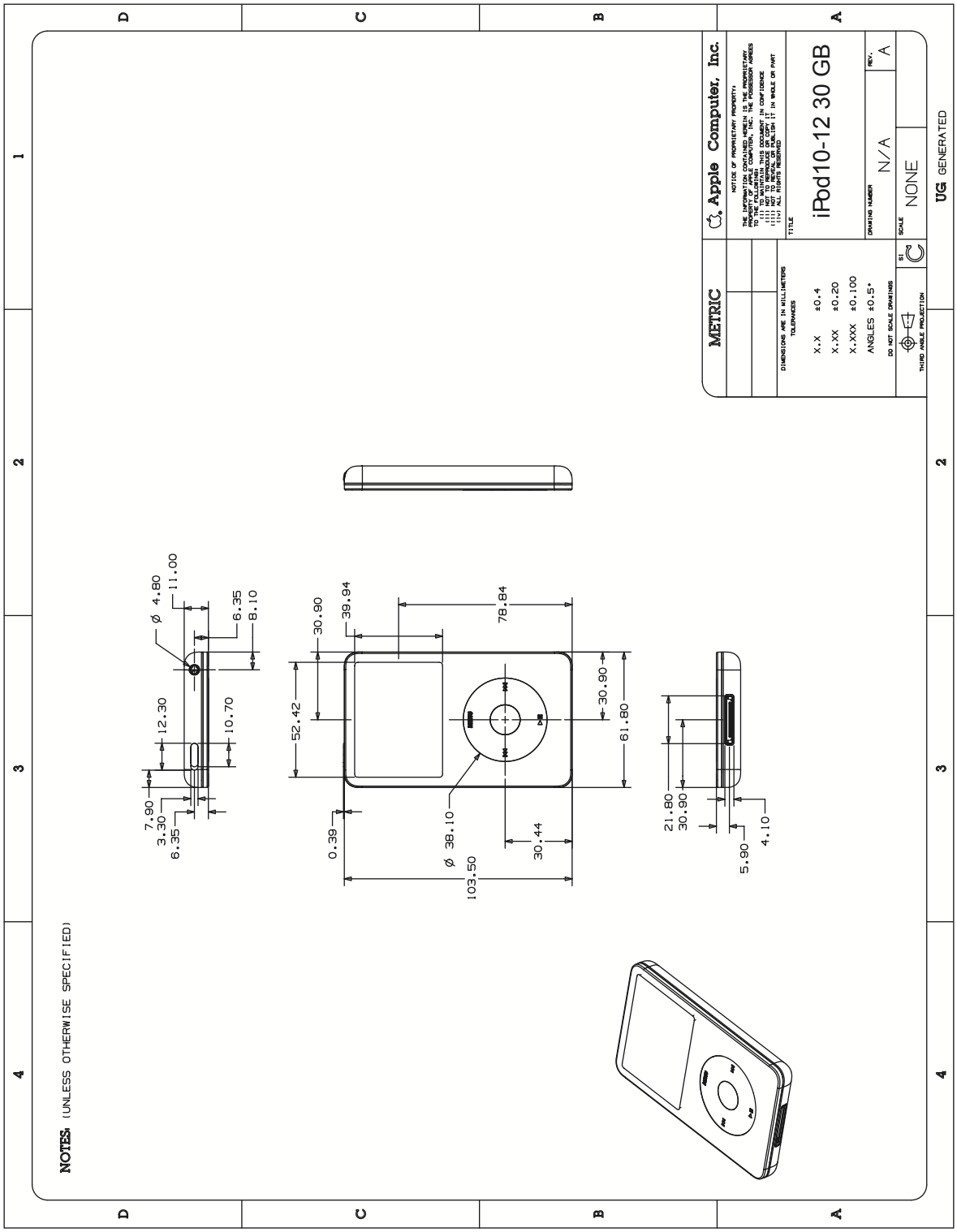
61.50 iPod (5th generation) 60GB/80GB

Figure 61-50 iPod 5th gen. 60GB/80GB Dimensional Drawing



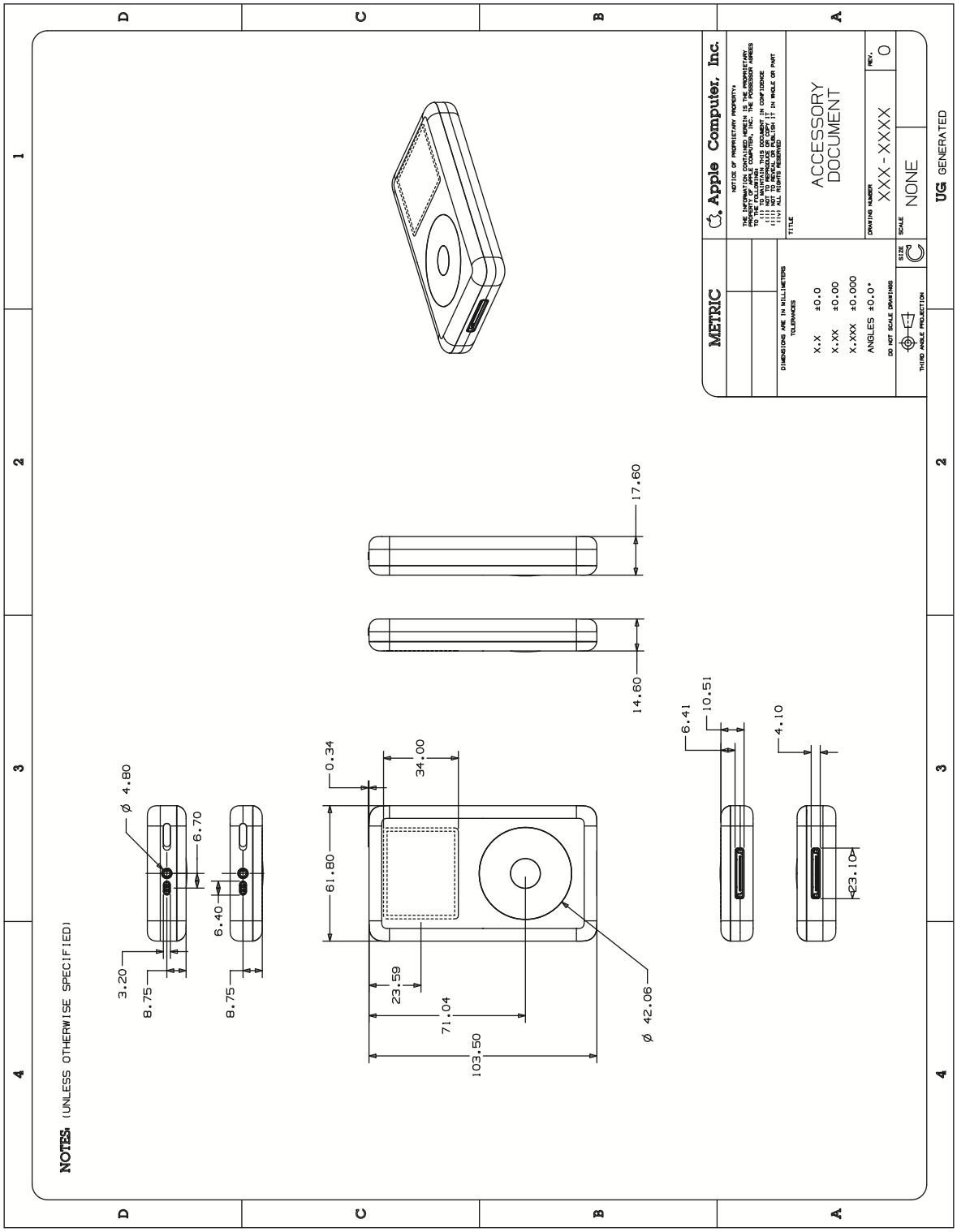
61.51 iPod (5th generation) 30GB

Figure 61-51 iPod 5th gen. 30GB Dimensional Drawing



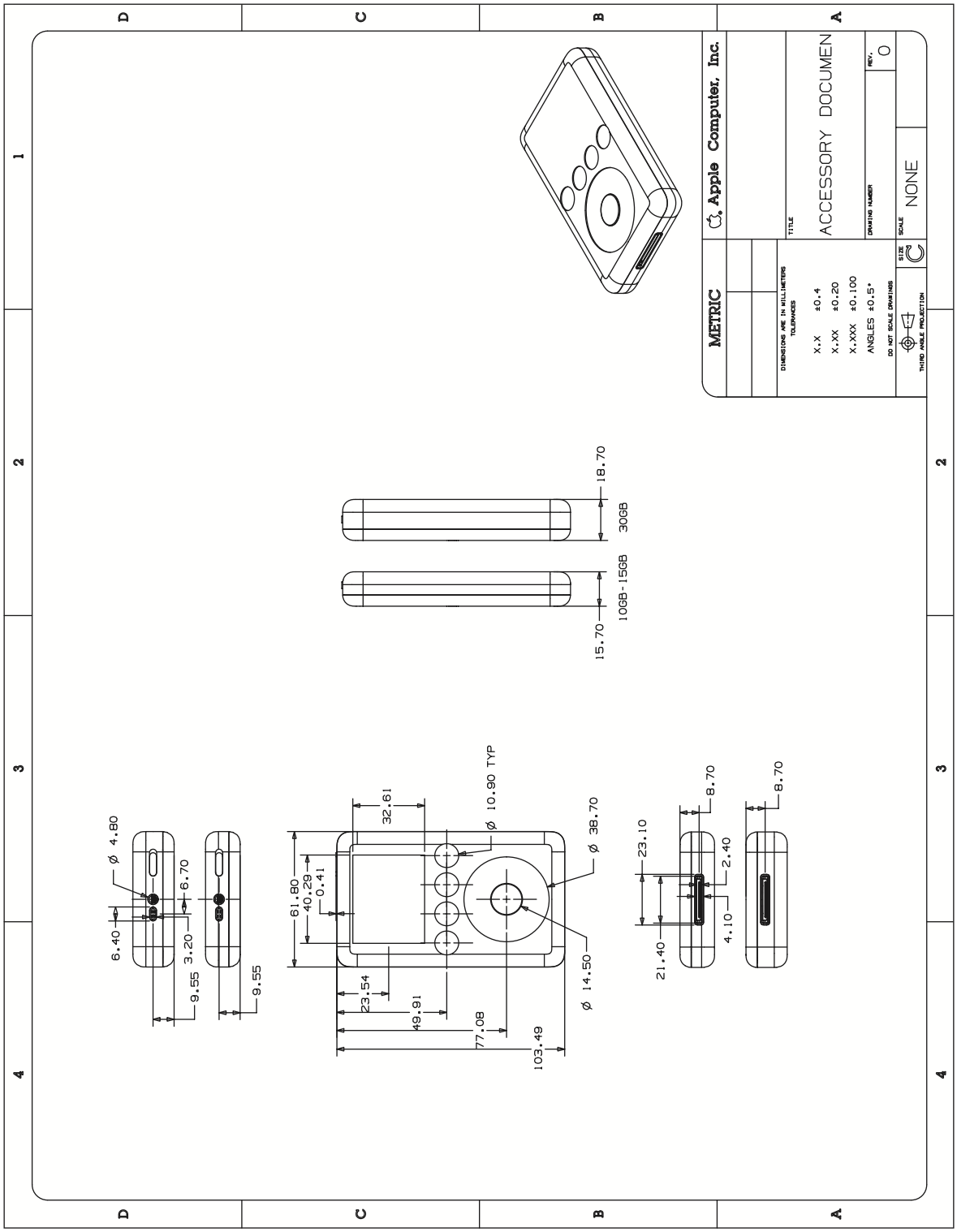
61.52 iPod (4th generation)

Figure 61-52 iPod 4th gen. Dimensional Drawing



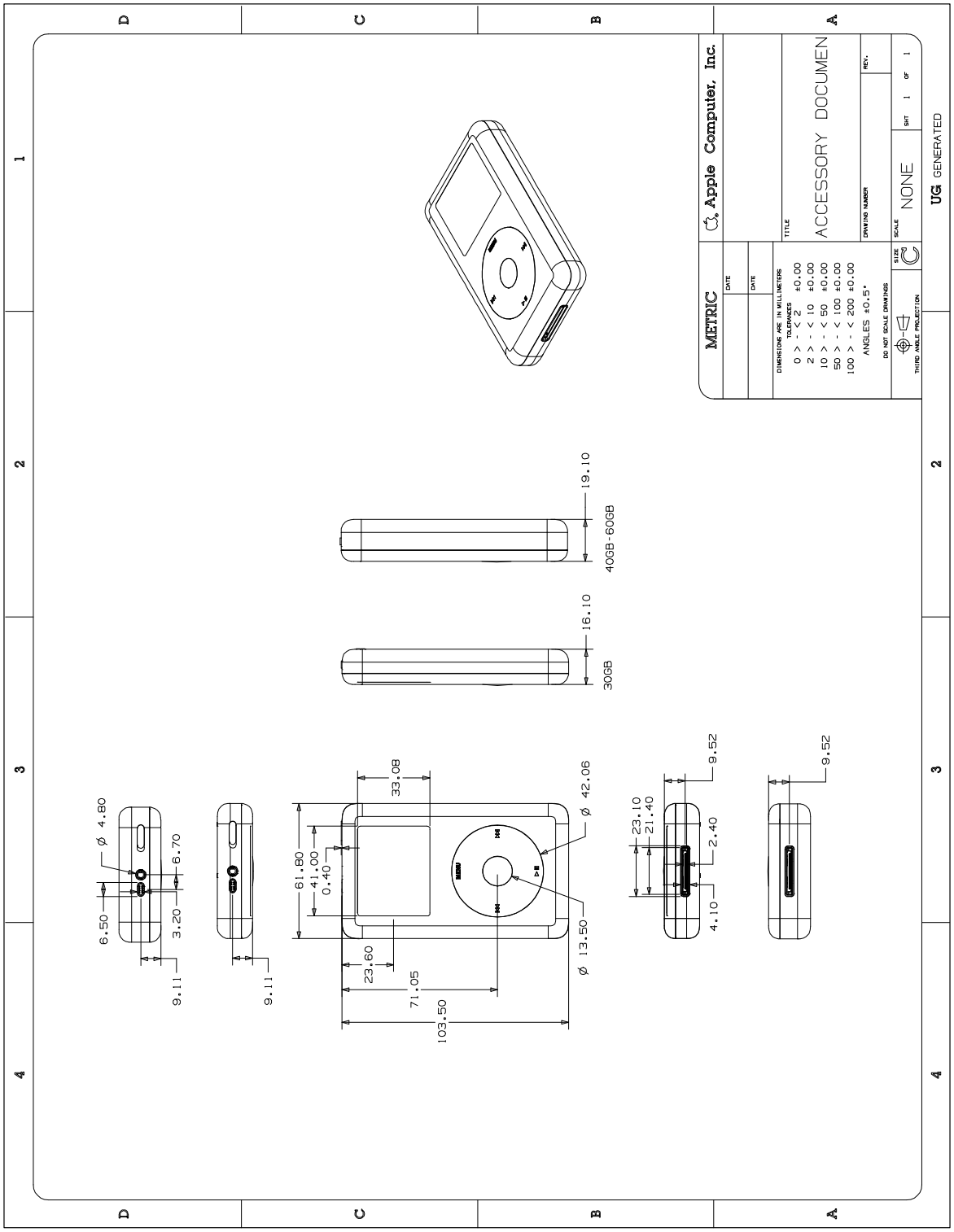
61.53 iPod (3rd generation)

Figure 61-53 iPod 3rd gen. Dimensional Drawing



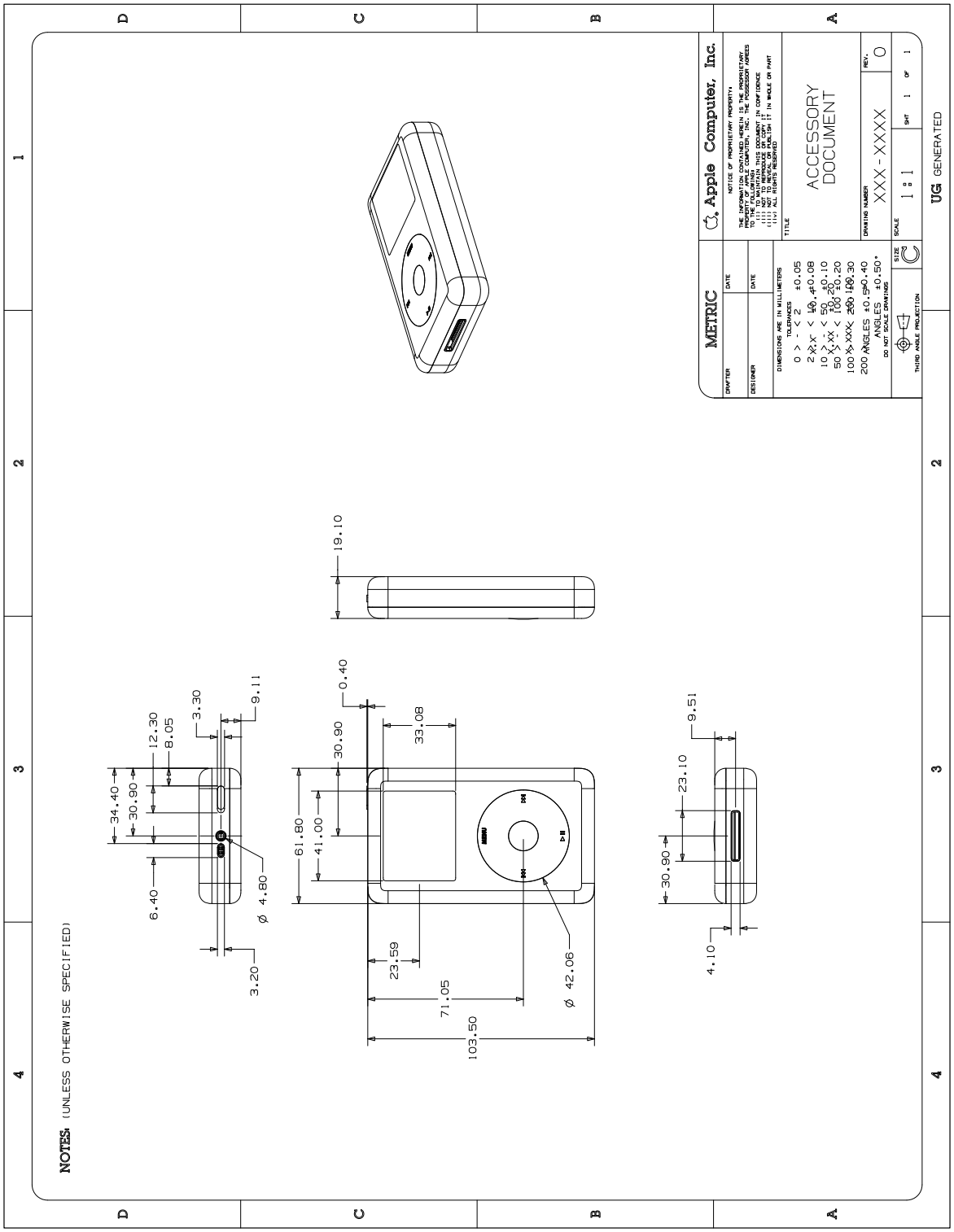
61.54 iPod photo 30GB/60GB

Figure 61-54 iPod photo 30/60GB Dimensional Drawing



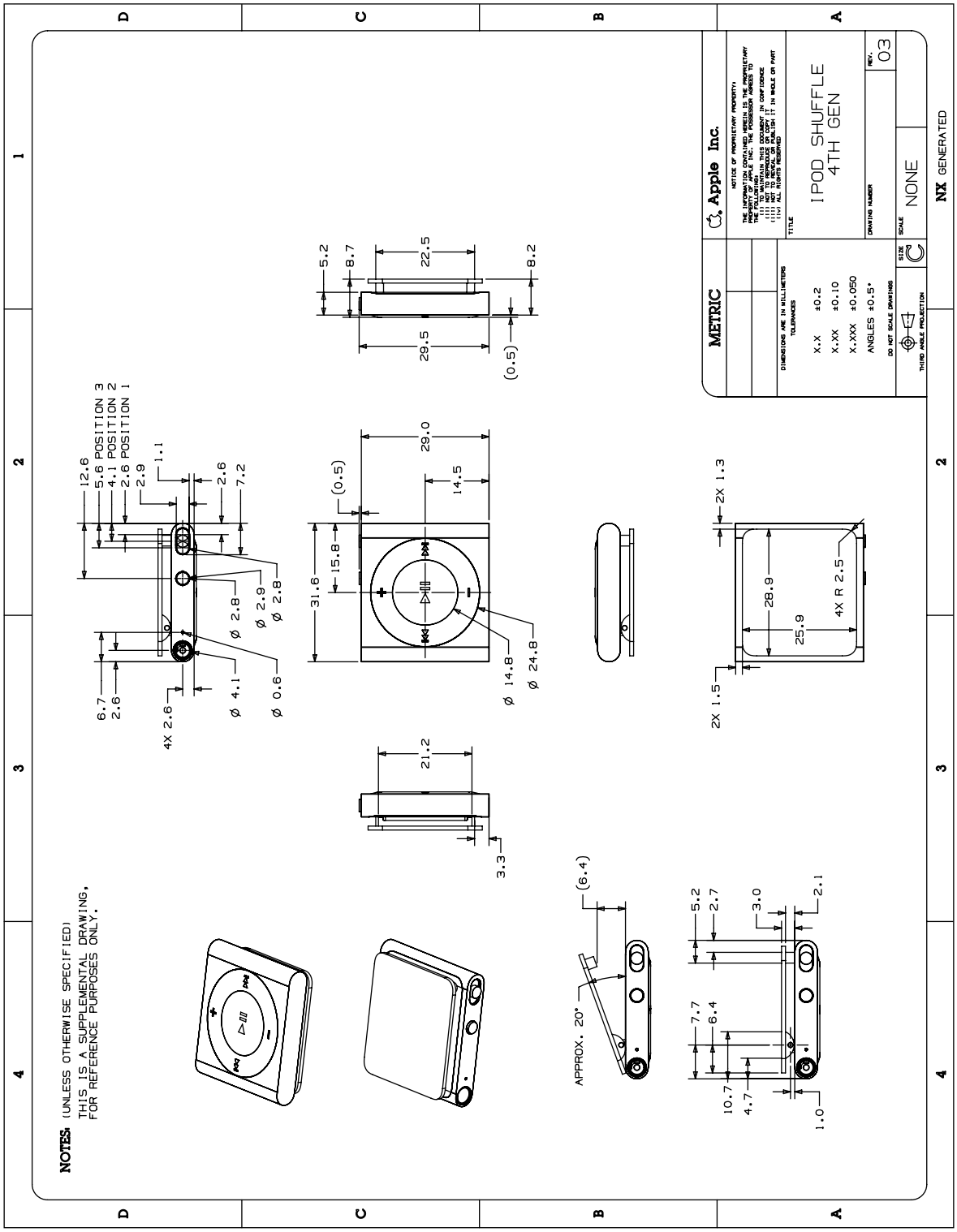
61.55 iPod photo

Figure 61-55 iPod photo Dimensional Drawing



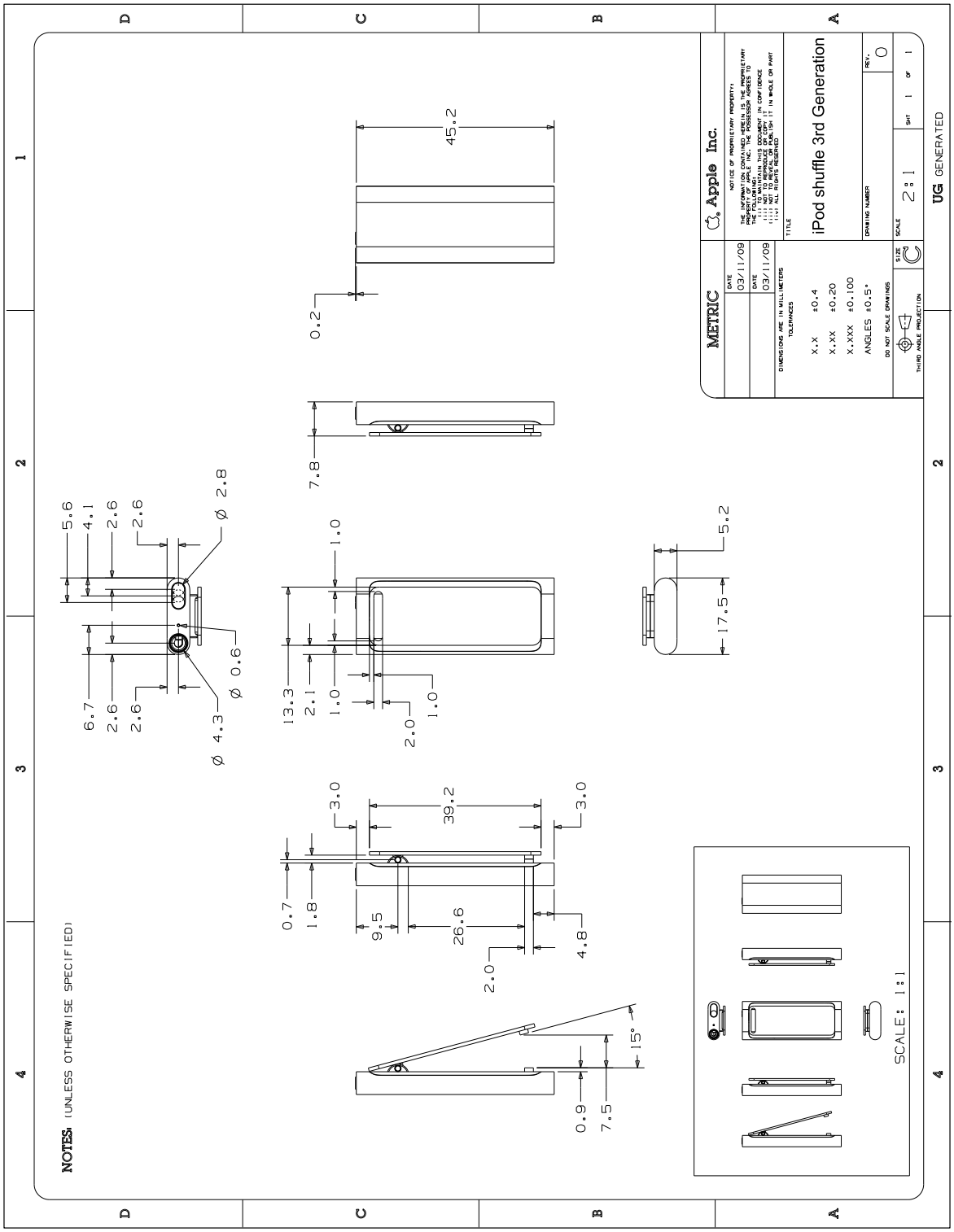
61.56 iPod shuffle (4th generation)

Figure 61-56 iPod shuffle 4th gen. Dimensional Drawing



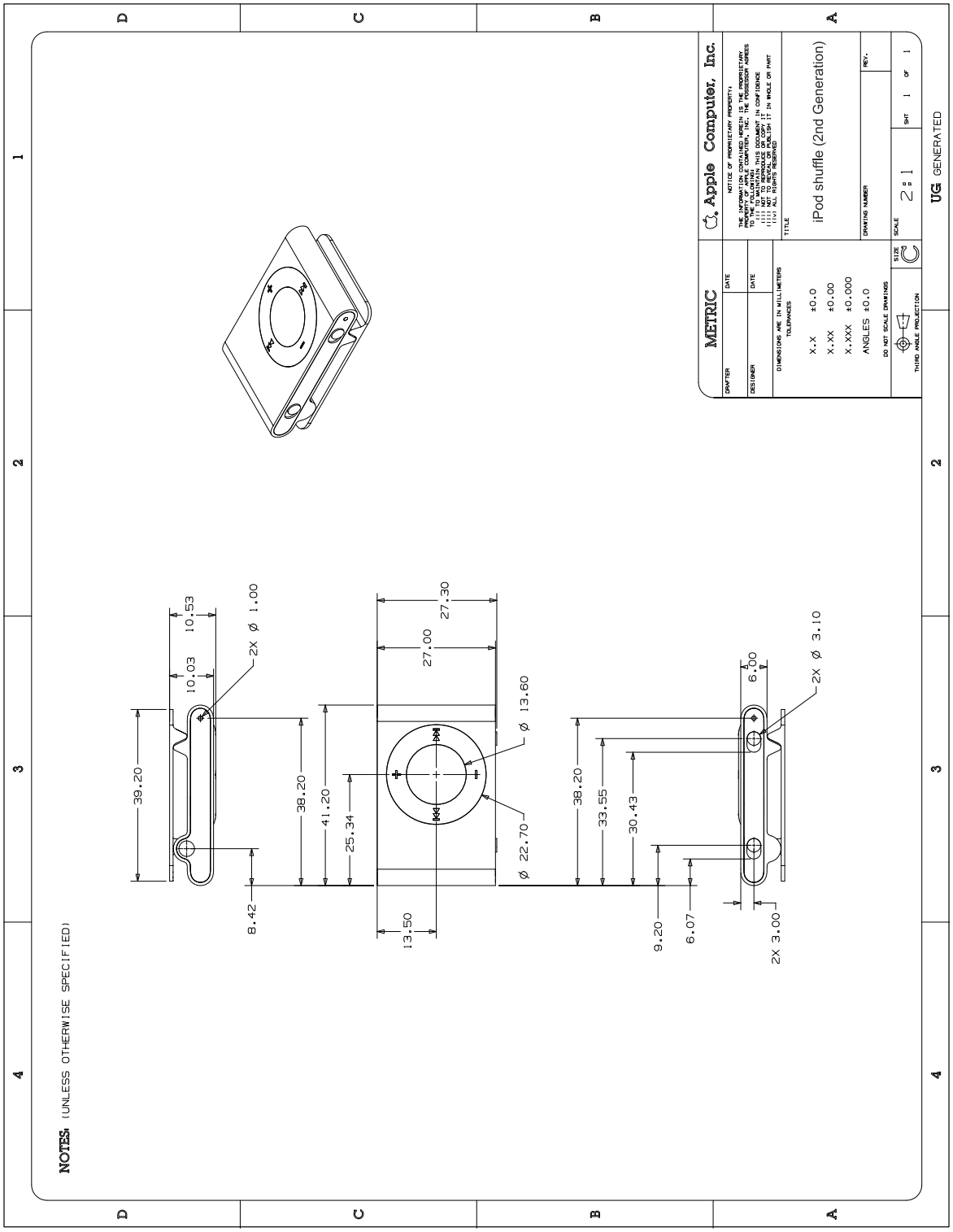
61.57 iPod shuffle (3rd generation)

Figure 61-57 iPod shuffle 3rd gen. Dimensional Drawing



61.58 iPod shuffle (2nd generation)

Figure 61-58 iPod shuffle 2nd gen. Dimensional Drawing



61.59 iPod shuffle

Figure 61-59 iPod shuffle Dimensional Drawing (1 of 2)

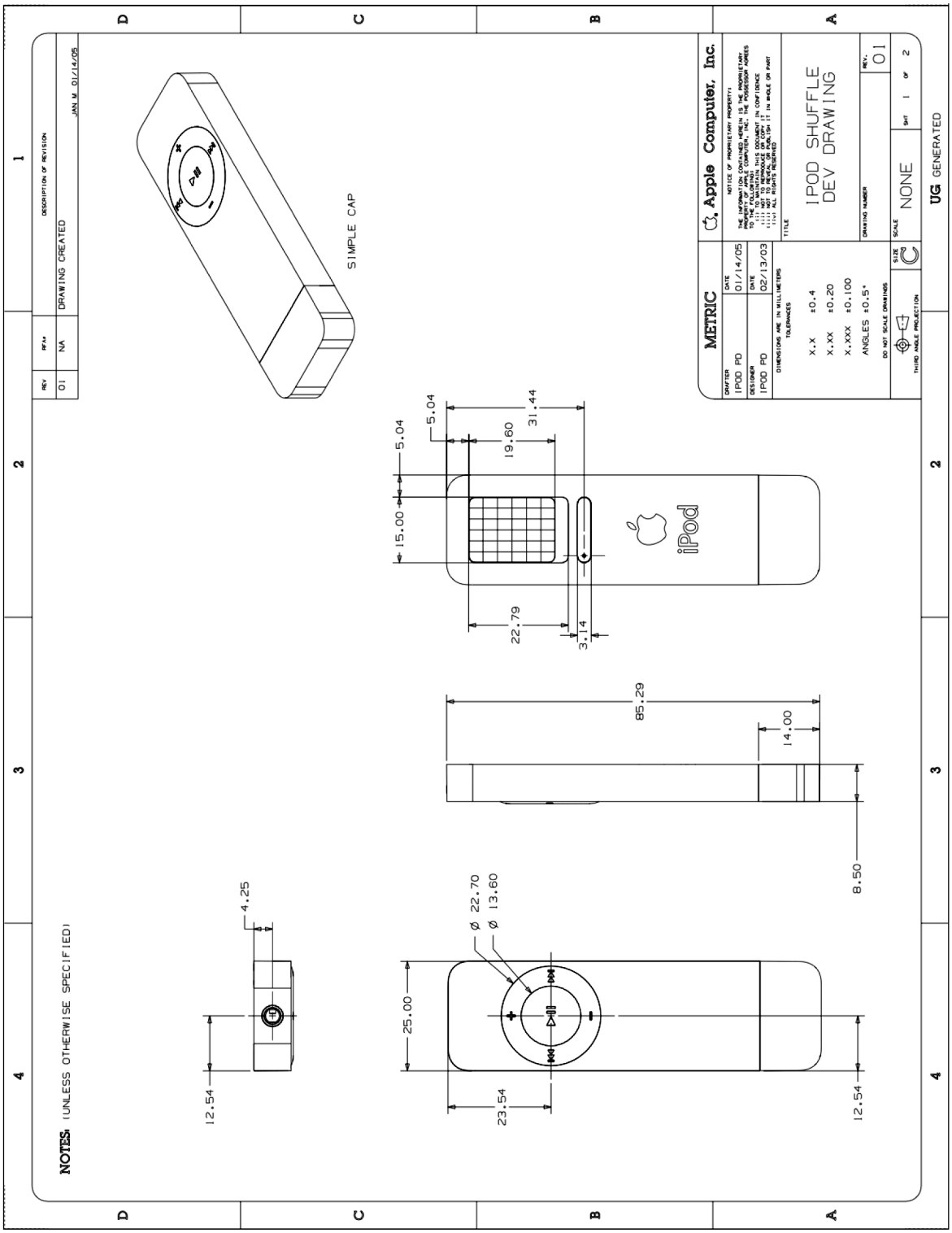
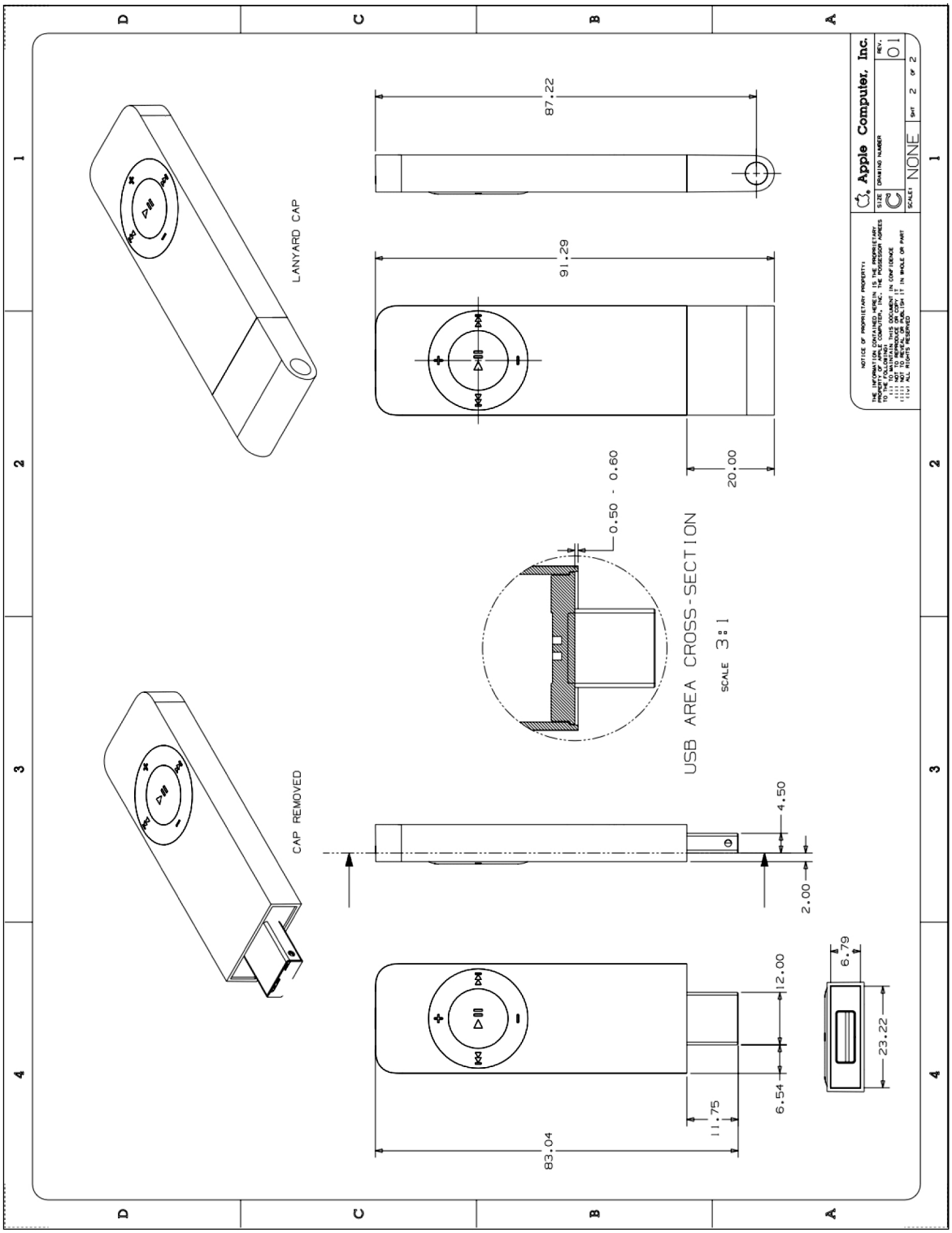
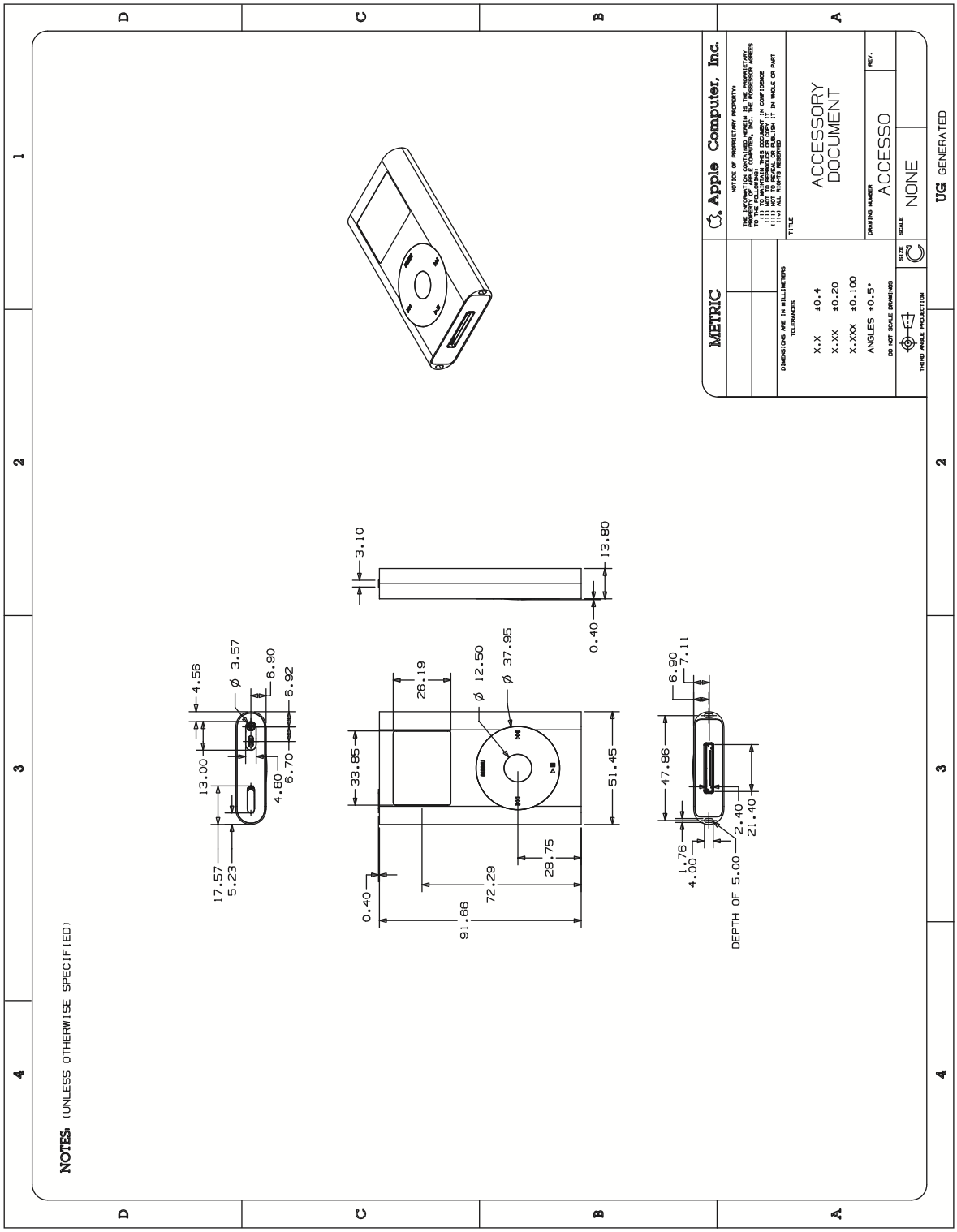


Figure 61-60 iPod shuffle Dimensional Drawing (2 of 2)



61.60 iPod mini

Figure 61-61 iPod mini Dimensional Drawing



62. Revision History

This table describes the changes to the Accessory Interface Specification.

Date	Notes
2015-12-18	*Release R22*
	Updated Register Addresses (page 72)
	Updated Cable Accessories (page 147)
	Updated Magnetic Charging Module Mechanical Requirements (page 219)
	Added Apple Smart Connector (page 251)
	Added Apple Smart Connector Module (page 253)
	Updated Table 14-1 (page 277)
	Updated User Configurable Interface (page 290)
	Updated Conformity With Bluetooth Specifications (page 344)
	Updated Pairing (page 378)
	Updated CarPlay over Wireless (page 395)
	Added Main Audio - Alert (page 420)
	Updated Main High Audio - Entertainment (page 422)
	Updated Info Message (page 430)
	Updated HID AssistiveTouch Pointer Requirements (page 588)
	Updated Switches and Position Encoders (page 597)
	Updated Table 41-3 (page 640)
	Updated Media Library Playback Usage (page 653)
	Updated High-Speed USB (page 699)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Updated App Launch (page 819)
2015-11-11	*Release R21*
	Updated iAP (page 57)
	Updated Multiple Simultaneous iAP2 Connections (page 61)
	Updated Pins and Assignments (page 194)
	Updated Passthrough USB Signal Integrity (page 198)
	Added Round-Trip DC Resistance (DCR) with Overcurrent Protection (OCP) (page 233)
	Updated CarPlay (page 381)
	Updated Test Procedures (page 496)
	Updated Communications (page 517)
	Added Support for Apple devices running iOS 8.2 or older (page 521)
	Added Game Controller Module (page 543)
	Added Electrical (page 546)
	Updated HID (Human Interface Device) (page 580)
	Updated HID Game Controller (page 591)
	Updated Test Procedures (page 609)
	Added Multiple Power States (page 669)
	Updated Multiple Connectors (page 667)
	Updated Max Current Drawn Identification (page 676)
	Updated Table 51-3 (page 716)
	Updated Wi-Fi Accessory Configuration (page 737)
	Updated Table 55-7 (page 745)
	Updated Table 55-8 (page 746)
	Updated Wi-Fi Information Sharing (page 761)
	Added Apple Device to Accessory (page 763)

Copyright © 2015 Apple Inc. All Rights Reserved.

	Added Accessory to Apple Device (page 763)
2015-09-11	*Release R20*
	Updated Applicability (page 51)
	Updated Apple Lightning Receptacle (page 186)
	Updated Apple Magnetic Charging Module (page 218)
	Updated CarPlay (page 381) chapter introduction
	Updated Siri Button (page 384)
	Updated Figure 23-13 (page 468)
	Updated Product Design (page 487)
	Updated Sensors (page 490)
	Updated Camera (page 491)
	Updated Environmental (page 494)
	Updated Near Field Communication (NFC) (page 495)
	Updated Touchscreen (page 495)
	Updated Apple Device Compatibility (page 496)
	Updated Apple Device Configurations (page 500)
	Updated RF (OTA) (page 504)
	Updated Device Notifications (page 526)
	Updated Remote Controls (page 550)
	Updated HID Game Controller Requirements (page 591)
	Updated Global Navigation Satellite System (GNSS) Mode (page 638)
	Updated Table 45-4 (page 666)
	Updated Improving Voice Recognition (page 687)
	Renamed Wi-Fi Accessory Configuration (page 737)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Updated Device Dimensional Drawings (page 880): Added dimensional drawings for iPhone 6s Plus, iPhone 6s, iPad Pro, and iPad mini 4. Updated dimensional drawings for iPhone 6 Plus and iPhone 6
2015-07-13	*Release R19*
	Updated Applicability (page 51)
	Added Minimum Apple Device Compatibility (page 55)
	Merged and updated "General Requirements, iAP1" and "General Requirements, iAP2" into iAP (page 57)
	Updated Coprocessor 2.0C Registers (page 72)
	Updated Apple Lightning Audio Module (page 93)
	Updated Cable Accessories (page 147)
	Updated Dock Accessories (page 158)
	Updated Apple Lightning Receptacle (page 186)
	Updated Apple Lightning Receptacle Controller (page 208)
	Added Apple Magnetic Charging Module (page 218)
	Updated Accessory Authentication Usage (page 262)
	Added Test Procedures (page 269)
	Added AirPlay (page 271)
	Added Test Procedures (page 358)
	Added Test Procedures (page 373)
	Updated General Requirements (page 383)
	Updated Cases (page 487)
	Updated Communications (page 517) (formerly "Telephony")
	Added Test Procedures (page 525)
	Added Test Procedures (page 533)
	Added Test Procedures (page 540)
	Added Game Controller Module (page 543)

Copyright © 2015 Apple Inc. All Rights Reserved.

Updated Headsets (page 549)
Updated Headset Plug (3.5 mm) (page 555)
Updated Headset Remote and Mic Requirements (page 559)
Updated HID (Human Interface Device) (page 580)
Updated HID Game Controller (page 591)
Updated HID Keyboard (page 621)
Added Test Procedures (page 634)
Updated Location Information (page 637)
Updated Test Procedures (page 653)
Updated Power Requirements (page 660)
Updated High-Speed USB (page 699)
Added Connecting Multiple Apple Devices to a Mac (page 707)
Added Test Procedures (page 711)
Updated USB Host Mode (page 712)
Added Test Procedures (page 722)
Added Wi-Fi (page 723)
Added Wi-Fi Accessory Configuration (page 737)
Added Test Procedures (page 763)
Updated Packet Structure (page 764)
Updated Control Session (page 789)
Added Test Procedures (page 803)
Updated Accessory Authentication (page 805)
Updated Accessory Identification (page 807)
Updated Communications (page 824)
Added Human Interface Device (page 844)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Updated LocationInformation (page 847)
	Updated MediaLibraryUpdate (page 852)
	Updated Accessory Test System (page 879)
2014-11-24	*Release R18*
	Updated Reference Designs & Development Kits (page 57)
	Updated "General Requirements, iAP2"
	Added Apple USB Power Adapters (page 61)
	Updated Apple Lightning Audio Module (page 93)
	Updated Apple Lightning Connector (page 128)
	Added Apple Lightning Receptacle (page 186)
	Added Apple Lightning Receptacle Controller (page 208)
	Updated Accessory Authentication over iAP2 Usage (page 262)
	Updated Accessory Identification Requirements (page 265)
	Updated Sniff Mode for Low Power Consumption (page 344)
	Updated Notifications (page 351)
	Added Play/Pause Button (page 351)
	Added CarPlay (page 381)
	Updated Device Protection (page 488)
	Updated Color (page 491)
	Updated Environmental (page 494)
	Updated Test Procedures (page 496)
	Updated Multiple Audio Connections (page 528)
	Updated Product Design (page 577)
	Updated HID (Human Interface Device) (page 580)
	Updated HID Game Controller Requirements (page 591)

Copyright © 2015 Apple Inc. All Rights Reserved.

	Updated Joysticks (page 602)
	Updated App Match for Controller-Enabled Games (page 606)
	Updated HID Keyboard Requirements (page 621)
	Updated HID Media Playback Remote Requirements (page 631)
	Updated Media Library Playback Requirements (page 650)
	Updated Low Power Mode Requirements (page 668)
	Updated Siri Eyes Free Mode (page 686)
	Updated Improving Voice Recognition (page 687)
	Added User Interaction with Siri Eyes Free in a Vehicle (page 690)
	Added Test Procedures (page 693)
	Updated USB Role Switch Usage (page 714)
	Added Vehicle Status (page 718)
	Updated Control Session (page 789)
	Updated Accessory Identification (page 807)
	Updated Vehicle Status (page 867)
	Updated Location (page 846)
	Updated Device Dimensional Drawings (page 880)
2014-09-19	*Release R17*
	Added Reference Designs & Development Kits (page 57)
	Updated Cables with USB Connectors (page 63)
	Updated Cables with Non-USB Connectors (page 63)
	Updated RF Transmission and Reception (page 65)
	Updated Connector Versions (page 129)
	Updated Connector Pad/Pin Configuration (page 139)
	Updated All Accessories (page 145)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Updated Cable Accessories (page 147)
Updated Lightning (C48)/Lightning (C68) Cables (page 151)
Updated Dock Accessories (page 158)
Updated Lightning (C48)/Lightning (C68) Docks (page 163)
Updated Dongle Accessories (page 167)
Updated Form-Fitting Accessories (page 167)
Updated Test Procedures (page 169)
Updated Apple Lightning Audio Module (page 93)
Updated Cable (page 102)
Updated accConfigurationInformation (0x03) (page 111)
Updated Accessory Identification of Power Capabilities (page 267)
Updated App Launch Usage (page 339)
Updated External Accessory Protocol (page 341)
Updated Cases (page 487)
Updated Access to the Headset Jack and 30-pin or Lightning Connector (page 488)
Added Device Protection (page 488)
Added Cover Glass Contact (page 488)
Added Dock Compatibility (page 489)
Updated Magnetic Interference (page 490)
Updated Touch ID Sensor (page 491)
Updated Environmental (page 494)
Added Near Field Communication (NFC) (page 495)
Updated Apple Device Compatibility (page 496)
Updated Apple Device Configurations (page 500)
Updated Device Notifications (page 526)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Updated Digital Audio Requirements (page 528)
Moved Readiness for Audio Streaming (page 528)
Moved Multiple Audio Connections (page 528)
Moved Audio Input Source Switching (page 529)
Moved Copy Protection of Digital Audio Output (page 530)
Moved USB Host Mode Audio (page 530)
Moved USB Device Mode Audio (page 531)
Updated iAP2 EA Session Requirements (page 536)
Updated Test Procedures (page 552)
Updated Headset Plug (3.5 mm) (page 555)
Updated Test Procedures (page 558)
Updated Button Detection Circuitry Adjustments (page 576)
Updated Test Procedures (page 577)
Updated App Match for Controller-Enabled Games (page 606)
Updated HID Keyboard Requirements (page 621)
Updated Standard NMEA Sentences (page 639)
Updated Media Library Access (page 649)
Updated Power (page 660)
Moved USB Host Mode Passthroughs (page 698)
Moved USB Signal Integrity (page 698)
Updated USB Embedded Host Implementation (page 706)
Updated Accessory Identification (page 807)
Updated App Launch (page 819)
Updated Device Notifications (page 840)
Updated External Accessory Protocol (page 843)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Updated Media Library Access (page 848)
	Updated Now Playing (page 858)
	Updated Power (page 864)
	Updated Device Dimensional Drawings (page 880): Added dimensional drawings for iPhone 6 Plus and iPhone 6
	Release R16
	Updated Accessory, Device, and Product (page 52)
	Updated Multiple Simultaneous iAP2 Connections (page 61)
	Updated Physical Configuration (page 83)
	Updated Connector Pad/Pin Configuration (page 139)
	Updated Apple Lightning Audio Module (page 93)
	Updated Audio Routing (page 354)
	Updated Headset Plug (3.5 mm) (page 555)
	Updated HID Requirements (page 580)
	Updated HID Game Controller Requirements (page 591)
2014-07-17	Updated Switches and Position Encoders (page 597)
	Updated Bluetooth Button (page 606)
	Added Bluetooth Button (page 613)
	Updated HID Keyboard Requirements (page 621)
	Updated Location Information (page 637)
	Updated Power Usage (page 669)
	Updated Improving Voice Recognition (page 687)
	Updated Table 48-1 (page 701)
	Updated Cumulative Acknowledgement Timeout (page 770)
	Updated Maximum Cumulative Acknowledgements (page 770)
	Updated Sleep (page 772)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Updated Synchronization (page 773)
	Updated TransferIdentifier (page 795)
	Updated Media Library Access (page 848)
	Updated Now Playing (page 858)
	Updated Communications (page 824)
2014-05-09	*Release R15*
	Removed "Accessory Test Procedures" and distributed its contents across feature chapters
	Removed "Transports"
	Updated Apple Device Detection (page 61)
	Updated Connector Versions (page 129)
	Updated Connector Pad/Pin Configuration (page 139)
	Added App Match (page 340)
	Added iAP2 (page 357)
	Updated External Accessory Protocol (page 535)
	Updated HID Game Controller (page 591)
	Updated HID Keyboard (page 621)
	Updated Location Information (page 637)
	Added Test Procedures (page 671)
	Added Serial (page 680)
	Added USB (page 698)
	Added USB Device Mode (page 706)
	Added USB Host Mode (page 712)
	Updated VoiceOver Usage (page 721)
	Updated iAP2 Link (page 764)
2014-04-07	*Release R14*

Copyright © 2015 Apple Inc. All Rights Reserved.

Updated Direct User Action (page 54)
Updated USB Signal Integrity (page 698)
Updated Cables with USB Connectors (page 63)
Updated Cables with Non-USB Connectors (page 63)
Updated Integrated Non-USB Receptacles (page 64)
Added User Supplied Cables and Power Supplies (page 64)
Updated Connector Pad/Pin Configuration (page 139)
Updated Shielding (page 152)
Added Apple Lightning Audio Module (page 93)
Added Headsets (page 549)
Updated Headset Remote and Mic (3.5 mm) (page 559)
Updated Joysticks (page 602)
Updated Bluetooth Button (page 606)
Added App Match for Controller-Enabled Games (page 606)
Added HID Headset Remote (page 616)
Updated HID Keyboard Requirements (page 621)
Updated Media Library Access Requirements (page 649)
Updated Media Library Updates Usage (page 651)
Updated USB Device Mode (page 706)
Updated Media Library Access (page 848)
Updated Communications (page 824)
Updated Test Procedures (page 169)
Added Test Procedures (page 125)
Added Test Procedures (page 496)
Added Test Procedures (page 552)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Added Test Procedures (page 629)
2014-03-06	*Release R13*
	Updated Connector Pad/Pin Configuration (page 139)
	Updated HID Game Controller Requirements (page 591)
	Updated USB Role Switch Requirements (page 714)
	Updated Signal Integrity Tests (C10A, C10B, C48A, C48B, C68A, C68B) (page 173)
	Added DC Resistance Tests (C10, C48, C68) (page 175)
	Added Test Procedures (page 577)
	Updated Test Procedures (page 609)
2014-02-04	*Release R12*
	Added Adapters and Proxies (page 58)
	Updated Cables with USB Connectors (page 63)
	Updated Cables with Non-USB Connectors (page 63)
	Updated Integrated Non-USB Receptacles (page 64)
	Updated Connector Versions (page 129)
	Updated Figure 5-9 (page 136)
	Updated Connector Mechanical Requirements (page 145)
	Updated Shielding (page 152)
	Updated Lightning (C48)/Lightning (C68) Accessory Testing Requirements (page 169)
	Added Apple 30-pin Connector (page 234)
	Updated Cases (page 487)
	Updated External Accessory Protocol Requirements (page 535)
	Updated Headset Remote and Mic Requirements (page 559)
	Updated HID Keyboard Requirements (page 621)
	Updated Power Requirements (page 660)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Added "Accessory Non-Power Source Usage"
	Added Providing Power to the Apple Device (page 669)
	Updated Bluetooth (page 344)
	Updated Human Interface Device (page 844)
	Updated Now Playing (page 858)
	Added Accessory Test System (page 879)
	Updated Over Voltage Protection Test (page 170)
2013-12-20	*Release R11*
	Added iBeacon (page 62)
	Updated Figure 5-9 (page 136)
	Updated Figure 5-10 (page 137)
	Updated "iPad mini Form-Fitting Extended Gamepad Sample"
	Updated "HID Extended Gamepad Control Layout"
	Updated Global Navigation Satellite System (GNSS) Mode (page 638)
	Updated Location Information Examples (page 646)
	Updated Flow Control (page 777)
	Updated iPad 2 with Wi-Fi + 3G (page 914)
2013-12-11	*Release R10*
	Updated Cables with USB Connectors (page 63)
	Updated Cable Encapsulation (page 150)
	Updated Shielding (page 152)
	Updated Form-Fitting Accessories (page 167)
	Updated Headset Plug (3.5 mm) (page 555)
	Updated Cable Accessories (page 173)
2013-11-01	*Release R9*

Copyright © 2015 Apple Inc. All Rights Reserved.

	Updated Applicability (page 51)
	Added Development Tools and Emulators (page 56)
	Updated Cables with Non-USB Connectors (page 63)
	Moved Apple Authentication Coprocessor 2.0C (page 66)
	Updated App Launch Requirements (page 338)
	Updated Connector Versions (page 129)
	Updated Connector Pad/Pin Configuration (page 139)
	Added Lightning (C48)/Lightning (C68) Cables (page 151)
	Updated EA Native Transport (USB Host Mode) Requirements (page 536)
	Updated HID Requirements (page 580)
	Renamed HID Assistive Switch Control (page 585)
	Updated Shoulder Buttons (page 602)
	Updated Bluetooth Button (page 606)
	Added Test Procedures (page 609)
	Updated All Accessories (page 169)
	Added Cable Accessories (page 173)
	Added iPad mini 2 & 3 Wi-Fi (page 903)
	Added iPad mini 2 & 3 Wi-Fi + Cellular (page 904)
	Added iPad Air Wi-Fi (page 905)
	Added iPad Air Wi-Fi + Cellular (page 906)
2013-09-17	*Release R8*
	Updated Applicability (page 51)
	Updated Requirements, Recommendations, and Permissions (page 50)
	Updated Connector Assemblies (page 58)
	Updated Multiple Simultaneous iAP2 Connections (page 61)

Copyright © 2015 Apple Inc. All Rights Reserved.

Added Removable Storage (page 65)
Updated Cables with Non-USB Connectors (page 63)
Added Integrated USB Receptacles (page 63)
Updated Connector Versions (page 129)
Added Table 5-1 (page 130)
Added Accessory Identification of Manufacturing Information (page 266)
Updated AssistiveTouch (page 342)
Added Bluetooth (page 344)
Added Bluetooth Accessory Identification (page 370)
Added Bluetooth Headset Battery Level Indication (page 374)
Added Cases (page 487)
Added Device Notifications (page 526)
Updated External Accessory Protocol (page 535)
Updated Headset Plug (3.5 mm) (page 555)
Added Extension Cables and Adapters (page 550)
Updated Headset Remote and Mic (3.5 mm) (page 559)
Updated HID Requirements (page 580)
Added HID Game Controller (page 591)
Updated Table 39-4 (page 626)
Updated Keyboard Example HID Report Descriptor (page 627)
Added HID Assistive Switch Control (page 585)
Updated Location Information (page 637)
Updated Now Playing Updates Usage (page 658)
Updated Power (page 660)
Added Siri (page 682)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Added Communications (page 517)
	Updated Table 51-3 (page 716)
	Updated Serial (page 680)
	Updated USB Host Mode (page 712)
	Updated Extended Acknowledgement Payload (page 771)
	Updated None (page 792)
	Updated iAP2 Control Session Messages (page 805)
	Updated Coprocessor 2.0C Overview (page 66)
	Added Device Dimensional Drawings (page 880)
2013-07-26	*Release R7*
	Updated Cables with USB Connectors (page 63)
	Updated Connector Versions (page 129)
	Updated Dongle Accessories (page 167)
	Updated Headset Remote and Mic (3.5 mm) (page 559)
2013-06-04	*Release R6*
	Update Introduction (page 50)
	Updated General Requirements and Recommendations (page 55)
	Updated Accessory Identification (page 265)
	Updated Apple Lightning Connector (page 128)
	Updated Digital Audio (page 528)
	Updated External Accessory Protocol (page 535)
	Updated Headset Plug (3.5 mm) (page 555)
	Moved material from HID (Human Interface Device) (page 580) chapter into new chapters HID AssistiveTouch Pointer (page 588), HID Keyboard (page 621), and HID Media Playback Remote (page 631).
	Updated Location Information (page 637)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Updated Media Library Access (page 649)
	Updated Power (page 660)
	Updated "Transports"
	Updated iAP2 Link (page 764)
	Updated iAP2 Sessions (page 788)
	Updated iAP2 Control Session Messages (page 805)
2013-02-20	*Release R5*
	Updated General Requirements and Recommendations (page 55)
	Updated Apple Lightning Connector (page 128)
	Updated Accessory Authentication (page 261)
	Updated Accessory Identification (page 265)
	Updated Device Authentication (page 523)
	Updated Digital Audio (page 528)
	Updated HID (Human Interface Device) (page 580)
	Updated Location Information (page 637)
	Updated Media Library Access (page 649)
	Updated Power (page 660)
	Updated USB Role Switch (page 714)
	Updated "Transports"
	Updated iAP2 Link (page 764)
	Updated iAP2 Sessions (page 788)
	Updated "Accessory Test Procedures"
2012-12-21	*Release R4*
	Updated General Requirements and Recommendations (page 55)
	Updated Apple Lightning Connector (page 128)

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

	Updated Media Library Access (page 649)
	Updated Now Playing Updates (page 657)
	Updated Power (page 660)
	Updated iAP2 Sessions (page 788)
2012-11-26	*Release R3*
	Added "Accessory Test Procedures"
	Updated General Requirements and Recommendations (page 55)
	Updated Accessory Authentication (page 261)
	Updated Device Authentication (page 523)
	Updated Headset Plug (3.5 mm) (page 555)
	Updated Power (page 660)
2012-10-30	*Release R2*
	Changed specification title
	Added Apple Authentication Coprocessor 2.0C (page 66)
	Updated Apple Lightning Connector (page 128)
	Updated Accessory Authentication (page 261)
	Updated HID (Human Interface Device) (page 580)
	Updated External Accessory Protocol (page 535)
	Updated Power (page 660)
	Updated iAP2 Link (page 764)
	Updated iAP2 Sessions (page 788)
2012-09-23	*Release R1*

2015-12-18 | Copyright © 2015 Apple Inc. All Rights Reserved.

Apple Inc.
Copyright © 2015 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to be used in the development of solutions for Apple-branded products.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, and iPhone are trademarks of Apple Inc., registered in the U.S. and other countries.

IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.