



# Wireless USB Framework

John S. Howard

Senior Architect

Intel Corporation

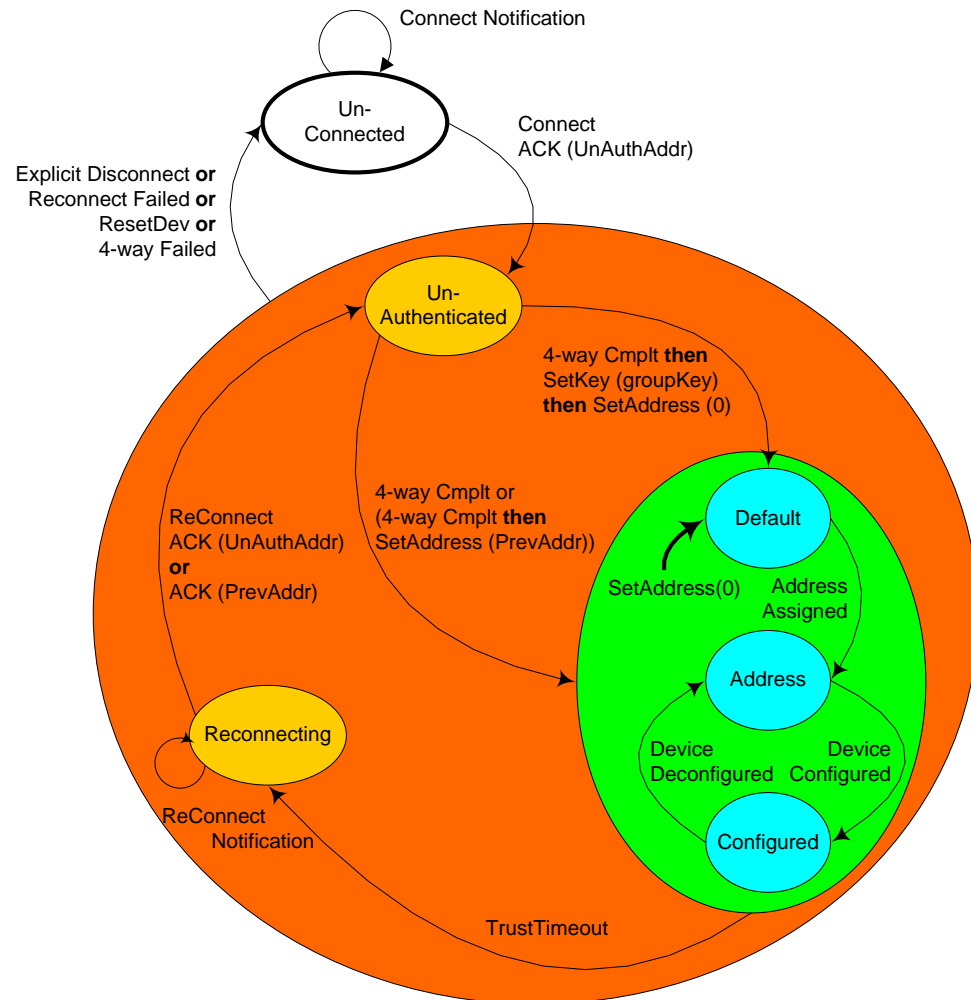


# Agenda

- Device States
- Basic Framework
- New for Wireless USB
  - Summary of Device Requests
  - Summary of Framework Descriptors
  - Summary of WUSB Channel Information Elements
  - Summary of Device Notifications

# Wireless USB Device States

- USB 2.0 Device state model is preserved within WUSB Framework
  - Encapsulated within the Secure Connection
- Additional states for establishment and re-establishment of a Secure Connection





# Basic Framework

- USB 2.0 Standard Requests and Descriptors are utilized in WUSB
- Restrictions on when requests are valid, based on whether the secure connection is in place
  - No requests that modify device information/state are allowed (except for the requests to establish the secure connection)
- New requests and descriptors added to support WUSB requirements
  - Extensions done in a way that enables future extensibility

# Basic Framework

## Summary



<b>Device States</b>	USB 2.0 Encapsulated in Secure Connection	
<b>Requests</b>	Modified	Set Address, Clear Feature, Set Feature, Get Status
	New	Loopback_data_read, Loopback_data_write, Security related requests, SetWSBData
<b>Descriptors</b>	Modified	Device, Configuration, Endpoint
	New	Security, Key, Encryption Type, BOS, Device Capability, Wireless Endpoint Companion
<b>MMC IEs</b>	All New	
<b>Device Notifications</b>	All New	

# Extensions for Device Requests

## Set/Clear Feature



USB 2.0	<b>bmRequest</b> X0000000B	<b>bRequest</b> Set/Clear Feature	<b>wValue</b> Feature Selector	<b>wIndex</b> Zero	<b>wLength</b> zero	<b>Data</b>
------------	-------------------------------	---	--------------------------------------	-----------------------	------------------------	-------------

WUSB	<b>bmRequest</b> X0000000B	<b>bRequest</b> Set/Clear Feature	<b>wValue</b> WUSB_Device	<b>wIndex</b> WUSB Feature Selector	<b>wLength</b> zero	<b>Data</b>
------	-------------------------------	---	------------------------------	---	------------------------	-------------

- Set/Clear Feature
  - Backward compatible with USB 2.0
  - Isolates all WUSB-specific features from USB 2.0 features
- New standard feature selector : WUSB\_Device
- *wIndex* used for individual WUSB feature selectors

TX DRP IE	Begin transmitting provided DRP IE
DEV XMIT_PACKET	Begin transmitting Transmit Packet
COUNT_PACKETS	Begin counting packets based on parameters provided
CAPTURE_PACKETS	Begin capturing packets base on parameters provided

# Extensions for Device Requests

## Get Status



USB 2.0	<b>bmRequest</b> 1000000B	<b>bRequest</b> Get Status	<b>wValue</b> Zero	<b>wIndex</b> Zero	<b>wLength</b> Two	<b>Data</b> Status
------------	------------------------------	-------------------------------	-----------------------	-----------------------	-----------------------	-----------------------

WUSB	<b>bmRequest</b> 1000000B	<b>bRequest</b> Get Status	<b>wValue</b> Zero	<b>wIndex</b> Status Selector	<b>wLength</b> Variable	<b>Data</b> Selector Status Data
------	------------------------------	-------------------------------	-----------------------	----------------------------------	----------------------------	-------------------------------------

- Get Status
  - Backward compatible with USB 2.0
  - Isolates all WUSB-specific Status Information
- *wIndex* used to select different status information

USB 2.0 Status	USB 2.0 Status Information
WUSB Features	Current setting for WUSB feature selectors
Channel Info	LQI with respect to last MMC
Received data	Data from Capture/Count Packet operation
MAS Availability	WiMedia MAS Availability bit map
Current Transmit Power	TPC settings for device-level managed packets (Notifications & Beacons)

# New Device Requests

## SetWUSBData



<b>bmRequest</b> 00000000B	<b>bRequest</b> SetWUSBData	<b>wValue</b> WUSB Data Selector	<b>wIndex</b> Zero	<b>wLength</b> WUSB Data Length	<b>Data</b> WUSB Data
-------------------------------	--------------------------------	--	-----------------------	---------------------------------------	--------------------------

- **SetWUSBData**
  - Used to set/update WUSB-specific descriptors and control parameters
  - *wValue* used for individual WUSB data selectors

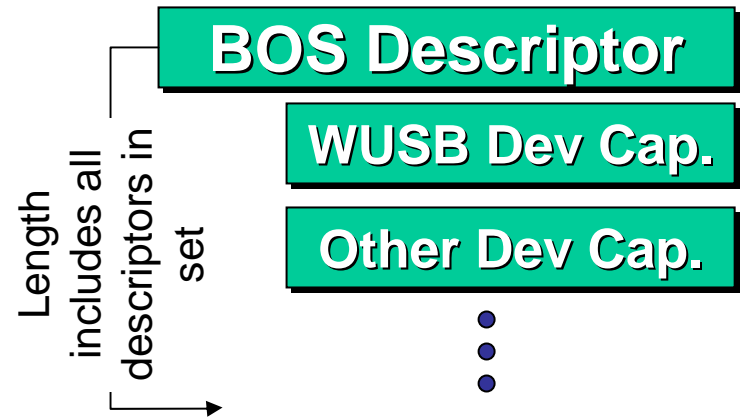
DPRIE INFO	DRPIE to be used by SBDs for WUSB Cluster
Transmit Data	Data packet to be transmitted by DBD for TX Packet feature
Transmit Params	DBD TX Packet control parameters
Receive Params	DBS RX count/capture packet parameters
Transmit Power	Host directed control for TX power for certain packet types

# Modifications for Descriptors

## Device Level



- Device Descriptor
  - bMaxPacketSize0 FFH
  - bcdUSB 0250H
- New BOS Descriptor for Device-level Extensions
  - Accessed via separate request
  - Extensible container model like Configurations
    - Set of descriptors accessed as a set



- WUSB Device Capabilities
  - P2P – DRD
  - Beacon Behavior
  - TX Rates Supported
  - TX Power Levels Supported
  - Band Groups Supported

# Modifications for Descriptors Configuration Level



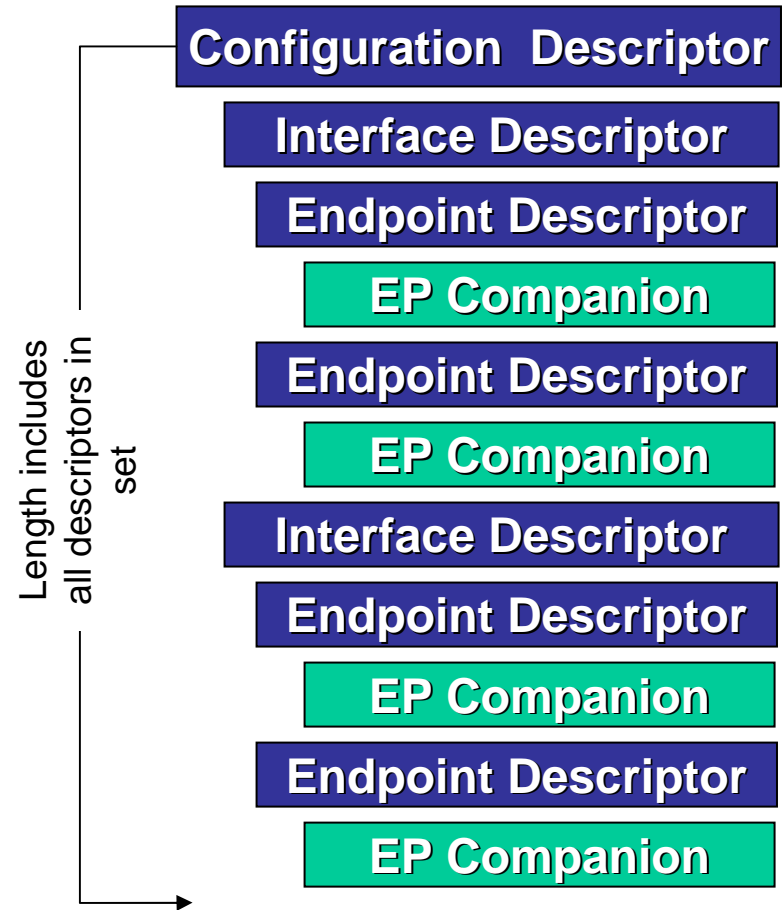
- Configuration Descriptor
  - Required settings for WUSB devices
  - New Battery Powered attribute
- Endpoint
  - Changes to :
    - bmAttributes
    - wMaxPacketSize
    - bInterval
  - Backward compatible

# Modifications for Descriptors

## New Endpoint Companion



- Every Endpoint descr. must have an Endpoint Companion descr. in every configuration set
- Endpoint companions descr. must follow the endpoint descr. they are associated with
- Endpoint companions carry new endpoint capabilities for WUSB



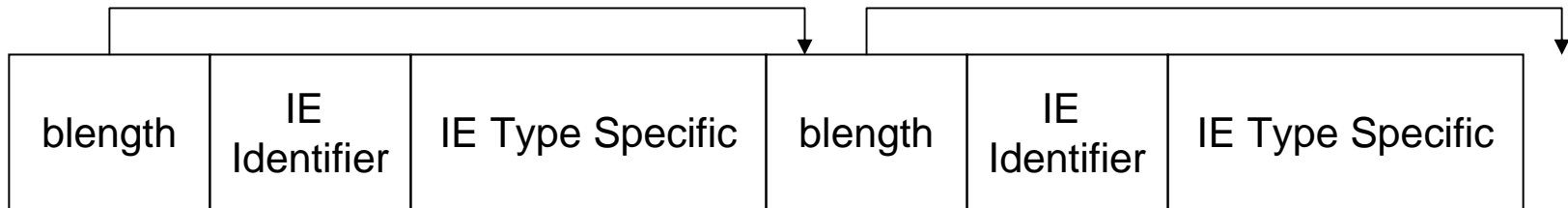
# Modifications for Descriptors

## New Security Descriptors



- See Security Session for details

# MMC Information Elements Basics



- *blength* used to find the next IE
- *IE Identifier* determines IE format (type)
- IEs are always an even number of bytes in length
- IEs addressable to multiple devices/endpoints are organized as arrays
  - Most are up to 4 elements (max)
  - Channel Time allocation IE is up to 32 elements (max)
  - May require host to add a pad byte to IE to get it to 2-byte alignment

# MMC Information Elements

## Summary

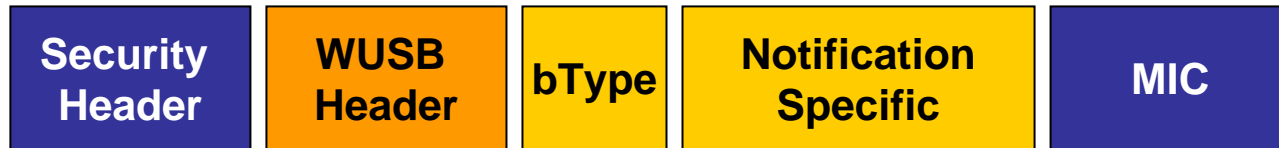


- Information in MMCs encapsulated in Information Elements

<b>WCTA_IE</b>	Reports transaction group channel time allocations
<b>WCONNECTACK_IE</b>	Host acknowledges connect requests
<b>WHOSTINFO_IE</b>	Host-specific information to uniquely identify Host and the WUSB Channel
<b>WCHCHANGEANNOUNCE_IE</b>	Information used to announce a channel change time and destination
<b>WDEV_DISCONNECT_IE</b>	Host uses this to disconnect a specific device
<b>WHOST_DISCONNECT_IE</b>	Host uses this to disconnect all devices
<b>WRELEASE_CHANNEL_IE</b>	Host uses this to tell specific device to broadcast a UDR packet (not using channel)
<b>WORK_IE</b>	Host uses this to acknowledge Device sleep requests
<b>WCHANNEL_STOP_IE</b>	Host uses this to announce the channel is stopping
<b>WDEV_KEEPLIVE_IE</b>	Host uses this to force a device to 'check-in'; usually prior to a TrustTimeout
<b>WISOCH_DISCARD_IE</b>	Host uses this on an Isochronous OUT to communicate skips
<b>WRESET_DEVICE_IE</b>	Host uses this as an analog to a USB 2.0 Port Reset signaling
<b>WXMIT_PACKET_ADJUST_IE</b>	Host uses this to communicate xmit adjustments to all devs enabled for Packet Xmit

# Device Notifications

## Basics



- Most Notifications are encapsulated in Secure Packets
- PID in WUSB Header set to DN designates as Device Notification
  - May only be transmitted by devices during Device Notification Time Slots (DNTS)
- *bType* field indicates format of notification-specific payload
- Max notification specific length is 32 bytes

# Device Notifications

## Summary



<b>DN_Connect</b>	Connect is transmitted without Secure Packet, Reconnect is
<b>DN_Disconnect</b>	Device uses this to explicitly disconnect from Host
<b>DN_EPRdy</b>	Device uses this to report that flow-controlled endpoints are ready to resume data streaming
<b>DN_MASAvailability</b>	Self-beaconing Device uses this to notify that it's MAS availability has changed
<b>DN_RemoteWake up</b>	Device uses this to emulate USB 2.0 remote-wake signaling
<b>DN_Sleep</b>	Devices uses this to inform the host that it intends to go to a Sleep state
<b>DN_Alive</b>	Device uses this in response to a WDEV_KEEPALIVE_IE or to transition from a sleep state (within a TrustTimeout period)



# Summary

- Extended the USB 2.0 Framework to support Wireless USB
- New device states to support establishing and maintaining a secure connection
  - Basic USB 2.0 defined device state preserved
- New device level capabilities descriptor (extensible)
- New endpoint companion descriptors
- Modified existing endpoint and configuration descriptor for applicability
- Wireless USB device must use bcdUSB value of 0250H
- MMC Information Elements and Device Notifications added to framework



Questions