SDK 6.5.26-DNX1

Release Notes

Table of Contents

No table of figures entries found.

Section 1: About This Document

This document contains the release notes for DNX devices affected by the Broadcom network switching Software Development Kit (SDK) release 6.5.26-DNX.1.

Note: This release is a DNX only release (XGS devices are not supported).

The document provides a general description of the SDK and its new features. It also describes the DNX chips supported by the release, BCM API additions or changes, resolved issues, and any relevant open issues.

Only new features are described in this document. For a comprehensive review of the DNX SDK features and issues, refer to earlier release notes for SDK 6.5.x.

For the full resolved list (Both Bugs and Improvement), please reference the file SDK-6.5.26-DNX.1-Resolved-Issues-Improvements.xlsx in the RELDOCS directory in the release package.

Section 2: Devices supported in this release

For any given SDK release, support for certain devices may be provided in preview or supported status. Devices in “Supported DNX Switch Devices” have completed the full QA process and are intended for use in production systems. It is expected that customers would integrate the version of the SDK which provides "Supported" status for their use on actual development or production systems.

Devices in “Preview DNX Switch Devices” are provided to allow early integration of the customer's application with the SDK APIs that support that device. This software has not been fully tested on the physical target device and is not expected to fully function.

Section 2.1: Supported DNX Switch Devices

Family	*Devices*	*Description*
BCM8828X		Q2U - GA quality
BCM8880X/BCM8882X		J2C - GA quality
BCM8848X		Q2A - GA quality
BCM8869X		J2 - GA quality
BCM8879X		Ramon - GA quality
BCM8868X		J+ - GA quality
BCM8837X/BCM8867X		JR - GA quality
BCM8827X		QUX - GA quality
BCM8847X		QAX - GA quality
BCM8829X		Q2n - GA quality

Section 2.2: Preview DNX Switch Devices

Family	*Devices*	*Description*
BCM8883X		J2X - Pre quality

Section 3: Information per Device

This release is an incremental version for DPP, DNX, DNXF, DFE family devices.
The subsequent sections describe the increment in available features compared to 6.5.26 backward-compatible notes, major bug-fixes and known issues.

It is very important to carefully go over the release-notes prior to adapting a new release.

The following sections describe the features validated for this release, known issues and bring-up guidelines.

Section 3.1: DNX-Family

This section includes the following family devices:

- **BCM8869X-Family (Jericho2)**
- **BCM8880X/BCM8882X-Family (Jericho2C)**
- **BCM8848X-Family (Qumran2A)**
- **BCM8828X-Family (Qumran2U)**
- **BCM8829X-Family (Qumran2N)**
- **BCM8883X-Family (Jericho2X)**

Section 3.1.1: Important Notes

Before integrating the new release, review this section thoroughly.

JIRA	Module	Release-note	Affected Devices
SDK-298008	NIF-MIB	Important change in the way counters are being counted. See more information in the Backward-Compatible section.	88480_B0, 88690_B0, 88800_A0
SDK-286292	IP-BFD	NWK-QOS and Initial-TTL fields were changed for IP-BFD injected applications influence Egress-QOS. See more information in the Backward-Compatible section.	88480_B0, 88690_B0, 88800_A0

Section 3.1.1.1: Backward Compatible Important Notes

SW Compatibility Guidelines to 6.5.26-dnx.1

Please go over the list carefully.

Note: This document is written with the assumption that upgrade is done from 6.5.26. In case upgrade is done from older releases, users must first go over previous release notes.

JIRA	Module	Release-note	Affected Devices
SDK-303913	RCH	The Ethertype on RCH value is changed now to indicate that the header above is ETH. In the previous release it was 0x1234, from that release is it 0x6558 Field-Processor applications that qualify according to that number need to be adjusted accordingly.	88480_B0, 88690_B0, 88800_A0
SDK-298008	NIF-MIB	In previous releases, below counter types counted both received and transmitted packets/octets. It didn't comply with RFC 1757 in which only received packets/octets should be counted. From this release, it has been fixed. snmpEtherStatsOctets snmpEtherStatsPkts snmpEtherStatsBroadcastPkts snmpEtherStatsMulticastPkts snmpEtherStatsOversizePkts snmpEtherStatsPkts64Octets snmpEtherStatsPkts65to127Octets snmpEtherStatsPkts128to255Octets snmpEtherStatsPkts256to511Octets snmpEtherStatsPkts512to1023Octets snmpEtherStatsPkts1024to1518Octets	88480_B0, 88690_B0, 88800_A0
SDK-297089	Field-Processor	When a packet has a TCP header, the next field layer offset (unknown layer) is no longer valid. This means that any Field-Processor application that used negative offset from unknown layer to retrieve TCP fields cannot be used anymore. With this change, the following cases were handled: 1. header_size_err trap is no longer raised for a TCP packet that exceeds size limit. 2. TCP with options - its size is no longer calculated, up until this release it was calculated wrongly. Note: It's the user responsibility to check whether the data accessed inside	88480_B0, 88690_B0, 88800_A0

		the TCP header is in-bound, otherwise unexpected behavior can occur.	
SDK-295350	FL-TRAP	bcmRxTrapFwdDomainAssignmentModeTrap trap previously could be configured by bcm_rx_trap_type_create() and bcm_rx_trap_set() This trap wasn't actually supported, hence APIs now return an error as they should be.	88480_B0, 88800_A0
SDK-292216	IFA 2.0	1. By default IP-protocol 255 ("Reserved") is set to IFA2 2. IFA2 application can't run using default configurations. bcm_ifa_header_create() must be called with the desired ip_protocol.	88480_B0, 88690_B0, 88800_A0
SDK-288655	NIF	Invalid sequence which might cause EGQ stuck is now enforced: If a port's TX has already been enabled by either API bcm_port_enable_set or API bcm_port_control_set with type bcmPortControlTxEnable, enabling it again under traffic using API bcm_port_enable_set may cause the EGQ to be stuck. This is an invalid sequence. The issue was resolved by making the following changes to the APIs: API bcm_port_control_set with type bcmPortControlTxEnable will be allowed only on enabled ports. API bcm_port_enable_set will skip all the configurations if the enable parameter matches the current state of the port (enable an enabled port, disable a disabled port).	88480_B0, 88690_B0, 88800_A0
SDK-286292	IP-BFD	In the previous releases, the NWK-QOS (Network-QOS) and Initial-TTL were always constant in the case of injected IP-BFD packets. NWK-QOS was set according to bcmOamControlOampInjectedNetworkQos (by default 255) and Initial-TTL was set to 0xFF. From this release, the NWK-QOS is inherited from IPv4.TOS/IPv6.TC header and the Initial-TTL is equal to IPv4.TTL/IPv6.hoplimit. This will allow having the same process between regular Routing processing and IP-BFD processing in regards to Egress QOS. Due to that, the following cases will now work differently: 1. API bcmOamControlOampInjectedNetworkQos doesn't work for IP BFD. NWK-QOS is inherited from IPv4.TOS/IPv6.TC header only. 2. The encapsulated ETH-header (EEDB ARP + Out-AC) that is being built by the egress pipeline is now affected by this when using the QoS-Model of Uniform/Initial. The VLAN-tag QOS (PRI,CFI) may have different values. When using the default egress VLAN-editing In the previous release, the PRI is always 0x7 and CFI is 1. In this release, the PRI is equal to	88480_B0, 88690_B0, 88800_A0

		IPv4.TOS[3:1]/IPv6.TC[3:1], the CFI is equal to IPv4.TOS[0]/IPv6.TC[0]. The user can change the QoS-Model of the EEDB entry to be Pipe in order to preserve the previous behavior. 3. For IP BFD with additional encapsulation in the EEDB, the NWK-QOS and TTL in the outer-tunnel encapsulation will be affected in case of uniform/initial QOS mode. Customer can use Pipe QOS model for the TTL in outer tunnel to change the behavior. Note: The change will take affect in both interop mode and regular mode.	

Section 3.1.2: SDK build & load

Compile and set config files :

setenv SDK `pwd`

Example of Intel GTS CPU compilation:

```
# Copy pre compiled mdb and kaps libraries into the relevant build folder.
# For Intel GTS CPU 64b build flavor, Following are the relevant 2 libraries and the
#  relevant build folder (names in build folder must be libkaps.a & libmdb.a):
mkdir -p $SDK/build/unix-user/x86-64-fc28/
cp  $SDK/libs/bin/dnx/GTS_64B_libkaps.a  $SDK/build/unix-user/x86-64-fc28/libkaps.a
cp  $SDK/libs/bin/dnx/GTS_64B_libmdb.a   $SDK/build/unix-user/x86-64-fc28/libmdb.a
# Additional mdb and kaps libraries flavors can be found under $SDK/libs/bin/.

# Compile SDK
cd $SDK/systems/linux/user/x86-64-fc28/
make -j 5 MAKE_LOCAL=$SDK/make/local/dnx/Make.custom.gts
```

Example of CMODEL compilation:

```
# Copy pre compiled mdb and kaps libraries into the relevant build folder.
# For CMODEL build flavor, Following are the relevant 2 libraries and the
#  relevant build folder (names in build folder must be libkaps.a & libmdb.a):
mkdir -p $SDK/build/unix-linux-64-cmodel/
cp  $SDK/libs/bin/dnx/CModel_libkaps.a  $SDK/build/unix-linux-64-cmodel/libkaps.a
cp  $SDK/libs/bin/dnx/CModel_libmdb.a   $SDK/build/unix-linux-64-cmodel/libmdb.a
# Additional mdb and kaps libraries flavors can be found under $SDK/libs/bin/.
```

```
# Compile SDK
cd $SDK/systems/sim/dpp
make -j 5 MAKE_LOCAL=$SDK/make/local/dnx/Make.pkg.dnx_only_sim_cmodel
target_suffix=-cmodel
```

Common config files:

```
ln -fs $SDK/rc/rc.soc
ln -fs $SDK/rc/dnx.soc
ln -fs $SDK/rc/jer2pemla-ucode.bcm
ln -fs $SDK/tools/sand/db
ln -fs $SDK/rc/config-skus.bcm
ln -fs $SDK/rc/dnx_dram
ln -fs $SDK/rc/cmicfw/linkscan_led_fw.bin
ln -fs $SDK/rc/cmicfw/custom_led.bin
```

BCM8869X specific links:

```
ln -fs $SDK/rc/config-jr2.bcm config.bcm
ln -fs $SDK/rc/bcm88690_revB_board.bcm
ln -fs $SDK/rc/bcm88690_board.bcm
ln -fs $SDK/rc/bcm88690_legacy_interop_board.bcm
```

BCM8880X/BCM8882X specific links:

```
ln -fs $SDK/rc/config-j2c.bcm config.bcm
ln -fs $SDK/rc/bcm88800_board.bcm
```

BCM8848X/BCM8828X specific links:

```
ln -fs $SDK/rc/config-q2a.bcm  config.bcm
ln -fs $SDK/rc/bcm88480_board.bcm
```

BCM8883X specific links:

```
ln -fs $SDK/rc/config-j2x.bcm  config.bcm
ln -fs $SDK/rc/bcm88830_board.bcm
```

Run :

```
./bcm.user
```

Section 3.2: DNXF-Family (BCM88790-Family)

Section 3.2.1: Supported SKUs

The following SKUs are supported:

- 88790
- 88795
- 88797

Section 3.2.2: Important Notes

Before integrating the new release, review this section thoroughly.
See Section 3.1.1: Important Notes

Section 3.2.2.1: Backward Compatible Important Notes

SW Compatibility Guidelines 6.5.26 to 6.5.26-dnx.1

Please go over the list carefully.

Note: This document is written with the assumption that upgrade is done from 6.5.26 to 6.5.26. In case upgrade is done from older releases, users must first go over previous release notes.

See section Section 3.1.1.1

Section 3.3: DPP-Family - BCM88670/680/470/270 Family GA Release

This release is for:

- BCM88670 (Jericho) family product lines.
- BCM88270 (QUX) family product line
- BCM88470 (QAX) family product line
- BCM88680 (Jericho+) family product line

The subsequent sections describe the increment in available features compared to 6.5.26, major bug-fixes and known issues. Before integrating the new release, review the “Backward compatible important notes” section.

Section 3.3.1: Important Notes

Before integrating the new release, review this section thoroughly.

JIRA	Module	Release-note	Affected Devices
SDK-298008	NIF-MIB	Important change in the way counters are being counted. See more information in the Backward-Compatible section.	88270_A0, 88470_B0, 88670_B0, 88680_A0

Section 3.3.2: Backward Compatible Important Notes

SW Compatibility Guidelines to 6.5.26-dnx.1

Note: This document is written with the assumption that upgrade is done from 6.5.26. In case upgrade is done from earlier releases to 6.5.26, it must first go over previous SDK release notes.

JIRA	Module	Release-note	Affected

			Devices
SDK-298008	NIF-MIB	In previous releases, below counter types counted both received and transmitted packets/octets. It didn't comply with RFC 1757 in which only received packets/octets should be counted. From this release, it has been fixed. snmpEtherStatsOctets snmpEtherStatsPkts snmpEtherStatsBroadcastPkts snmpEtherStatsMulticastPkts snmpEtherStatsOversizePkts snmpEtherStatsPkts64Octets snmpEtherStatsPkts65to127Octets snmpEtherStatsPkts128to255Octets snmpEtherStatsPkts256to511Octets snmpEtherStatsPkts512to1023Octets snmpEtherStatsPkts1024to1518Octets	88270_A0, 88470_B0, 88670_B0, 88680_A0

Section 3.4: DFE-Family - BCM88770 (FE3600) Release

The Broadcom BCM88770 (formerly named BCM88950) is the fourth generation in the DNX product line of Fabric Element (FE) devices.

This is a sustaining release.