SDK 6.5.20

Release Notes

Table of Contents

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 6.5.20.

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-6.5.20-Resolved-Issues-Improvements.xlsx in the RELDOCS directory in the release package.

Section 2: New Devices added to 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	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

Section 3: Information per Device

This release is an increment version for DPP, DNX, DNXF, DFE family devices.

The subsequent sections describe the increment in available features compared to 6.5.19, 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)**

Please use the PCIe firmware version 2.5.1-DNX with this release.
This version can be found under Firmware/pcie/dnx/v2.5.1/.

Section 3.1.1: Important Notes

Before integrating the new release, review this section thoroughly.

JIRA	Module	Description	Affected Devices
-	KBPSDK	SDK is aligned to KBPSDK version 1.5.13 Note: KBPSDK is required for DPP devices or when DNX family devices are working with external KBP. For DNX devices there is no need to to use KBPSDK for KAPs. To compile without KBPSDK remove from FEATURE_LIST KBP	88690, 88800, 88480
-	MBIST	We require that after running a Memory Built In Self Test (MBIST), that the device it runs on either be powered off or system reset (hardware reset by de-asserting the signal SYS_RST_N from the board). Without this action we do not guarantee that the device will work properly, and therefore from this release the software enforces it. This is a new requirement for BCM8869x (Jericho 2), independent of the SDK release. For BCM8880x, BCM8882x (Jericho 2C), BCM8848x (Qumran 2A) we previously required a power off, and now also accept a system reset.	88690, 88800, 88480

		Note that both powering off the device and a system reset of the device, also involve a PCIe reset of the device, and the PCIe link of the device going down; so the customer BDE needs to support this operation if done while the BDE and SDK are running. We provide a callback function skeleton dnxc_perform_system_reset() in src/appl/reference/dnxc/appl_ref_board.c that may be used to perform these operations automatically on the customer board. An example of how this function is registered in the sample application can be seen in the dnxc_perform_system_reset() in src/appl/reference/dnxc/appl_ref_init_deinit.c . To have the call back function used automatically after running MBIST, and enable to continue running the SDK later, use this new soc property: perform_system_reset_when_needed=1 . If a device power off, or system reset is not actually performed, the SDK will recognize this, fail, and not continue running, and not run when started again. The MBIST is meant to test memories when there is a suspicion of device malfunction and is not mandatory. As described here, with some effort it is possible to run MBIST before each SDK runs, but this is not the intention of the test. The changed MBIST and CPU2TAP code in the release, together with the new requirement, fix possible PCIe issues in 8869x (Jericho 2).	

Section 3.1.1.1: Backward Compatible Important Notes

SW Compatibility Guidelines 6.5.19 to 6.5.20

Please go over the list carefully.

Note: This document is written with the assumption that upgrade is done from 6.5.19 to 6.5.20. In case upgrade is done from older releases, users must first go over previous release notes.

JIRA	Module	Description	Affected Devices
SDK-219645	QOS	bcmQosControlMplsIngressPopQoSPreserve now requires flag BCM_QOS_MAP_L3 to be set with the right type (IPv4, IPv6)	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-217217	VSWITCH	bcm_vswitch_cross_connect_traverse returns now flag BCM_VSWITCH_CROSS_CONNECT_DIRECTIONAL. In previous, flag was not returned and then fails to delete cross-connection which is created with flag BCM_VSWITCH_CROSS_CONNECT_DIRECTIONAL. This release fixes the issue and changes the return flags compared to the previous release.	88480_A0, 88480_B0, 88690_B1, 88800_A0
SDK-217140	FC	Improve accuracy of API bcm_cosq_pfc_deadlock_control_set with type bcmCosqPFCDeadlockDetectionTimer. Implications are that the actual detection time will be as close to the user set value as possible, which might be different than previous behavior(depending on the value that was set). Also, a check for the pulse generator period was improved, which means that it may fail in some scenarios that were previously passing when the values of pulse generator period and detection timer are very close to each other.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-216939	INT-ET	bcm_instru_gport_control_set with bcmInstruGportControlTraceProbability was not set correctly, instead of setting percentage to X%, the API set percentage to (100-X)%. This is now fixed.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-216836	COE	PCP/DEI of COE tag can be sourced from port now by calling bcm_switch_control_port_set(unit,coe_port,bcmSwitchTagPcpDeiSrc,3). PCP/DEI in COE tag is sourced from out port's attribute as default with value zero and not from Out-AC as before.	88480_A0, 88480_B0, 88800_A0
SDK-216201	L3-IPV6-MC	Support was added for split selection for IPv4 and IPv6 forwarding type in case the IPMC routing is disabled. Using the flag BCM_L3_INGRESS_L3_MCAST_L2 when creating a RIF the user	88480_A0, 88480_B0, 88690_A0,

		can now indicate that v4MC forwarding type will be used. Default is v4Bridge if the flag is missing. Using the newly introduced flag BCM_L3_INGRESS_IP6_L3_MCAST_L2 - the user specifies v6MC forwarding type. Default is v6Bridge which will be selected if the flag is missing. In previous versions, BCM_L3_INGRESS_L3_MCAST_L2 related also to BCM_L3_INGRESS_IP6_L3_MCAST_L2 by mistake. In case user wish to allow both types (IPv4 and IPv6), it should call both flags.	88690_B1, 88800_A0
SDK-214175	TRAP	Force forward application will not work if the injected packet has ITMH header. ITMH application was refactored to not overwrite the traps with highest strength (0xF). Thus allowing force forwarded packets can reach their destination (regardless of the ITMH header). This change is true for all trap-actions of strength 0xF prior to PMF application (for ITMH injected packets).	88480_A0, 88480_B0
SDK-213934	INT-IFA	TTL field on Metadata should be outgoing TTL (incoming TTL -1) and not incoming TTL.	88480_B0, 88690_B1, 88800_A0
SDK-212714	L2	The following list of BCM APIs delete MACT entries according to one or more properties of the entries: bcm_l2_addr_delete_by_port() bcm_l2_addr_delete_by_mac() bcm_l2_addr_delete_by_vlan() bcm_l2_addr_delete_by_trunk() bcm_l2_addr_delete_by_mac_port() bcm_l2_addr_delete_by_vlan_trunk() bcm_l2_addr_delete_by_vlan_port() The above APIs did not support the flag BCM_L2_DELETE_STATIC properly. When the flag wasn't set, the APIs still deleted static entries. This is fixed. Now static entries are deleted only when BCM_L2_DELETE_STATIC is set.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-212572	LAG-PP	bcm_port_control_set bcmPortControlSystemPortInjectedMap does not support trunk-member port anymore. Instead, it is expected to be called only with local ports or trunk gports. In case of trunk gport, the mapping of trunk SPA with masked members and the relevant pp port per core is done. The API returns an error if it is called with trunk member.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-212462	COSQ-ING	VOQ bundle can be allocated in the limits of one queue region. An error is returned now in case of VOQ allocation outside the limits of a queue region.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-211382	PTP	APIs bcm_port_timesync_config_set/get, bcmPortControl1588P2PDelay are now supported according to Trunk-member (local-port) and not according to Trunk-group	88480_A0, 88480_B0,

		(trunk-gport). The change is due to give the ability for the user to configure different values per Trunk-member. APIs call are now different than the previous release, trunk-gport will return an error, while local-port should be used instead when the port is part of the LAG.	88690_A0, 88690_B1, 88800_A0
SDK-211294	SBFD	In the previous release, seamless bfd reflector didn't support local State change. In this release,this issue has been addressed. This may break backwards compatibility from previous version - in the past, State was copied from incoming packet, now State is taken from endpoint-create.	88480_A0, 88480_B0, 88690_B1, 88800_A0
SDK-210904	OAM, PMF	OAM/BFD: when injecting ITMH + PPH packets from the OAMP, ingress PMF will strip and rebuild the PPH. This allows supporting Hierarchical LM counting over LIFs in OAM (supported in SDK-200941) and BFD over VXLAN (supported in SDK-206144). The side-effect of this is that the calling sequence for is that the calling sequence for OAMP punt and protection packet configuration has changed: 1. User must now allocate a user-defined trap with ID < 255 for each punt/protection packet. Traps are still allocated with the bcm_rx_trap_type_create API. A trap code may be created with or without ID. 2. When calling bcm_rx_trap_set() use the flag BCM_RX_TRAP_TRAP. 3. In order to change the trap code of a punt packet, destroy the trap and then recreate it with a new trap code. Punt/Protection packet configuration has been modified in appl_ref_oam_init.c. As a result, punt/protection packets will arrive at the CPU with a different trap code than in previous releases. An example of how to change the trap code can be found in maid_trap_set() in cint_sand_oam.c	88480_A0, 88480_B0, 88690_B1, 88800_A0
SDK-210640	L3	IPMC flows were slightly changed. For the case of double KAPS DB, IPv6 MC LEM entry with VRF = 0: in the previous release, it was allowed, from this release it will return error as SDK doesn't support public anymore (aligns to LPM). In this JIRA, also support for working in a state of a single KAPS DB was added. The L3 and IPMC APIs can now work correctly when the capacity of public KAPS is 0. See more information in the PG document "IPMC section".	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-210026	LAG-PP	When a port is removed from a Trunk group, its PP port properties are lost and its properties are being reset to its header-type defaults. In previous versions, once such action happened, PP port properties were reset according to the PP default Application and not aligned to SDK port default configuration. This is now changed and port is now reset to its header-type defaults (the same as adding port in the system without running PP default application).	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0

		In order to mimic the application properties, an example was added. Please see CINT cint_port_packet_processing_example.c function: cint_port_packet_processing_removed_trunk_member_pp_properties_set. Bonus: An additional functionality was added to show what pp port properties should be reset before port is removed from trunk group, see function cint_port_packet_processing_pp_properties_unset. Note: Currently on this operation, API doesn't touch port learn property and its IngressMissDrop action. This may change in a future version. It is expected that the user will call bcm_port_learn_set, bcm_vlan_control_port_set bcmVlanTranslateIngressMissDrop to configure its expected values after port is created/removed from trunk.	
SDK-209800	RCY	bcm_l2_egress_get API returned local-port when dest_port attribute was given to BCM_L2_EGRESS_DEST_PORT. API also assumed that port exists in the local-device. From this release, API bcm_l2_egress_set/get uses either trunk gport or system port gport in dest_port. Other types such as Local port are not supported.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-209714	TRAP	bcmRxTrapDfltRedirectToCpuPacket which is configured on init, is now configured with flag BCM_RX_TRAP_TRAP which will result in adding FHEI header to the packet.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-209345	COSQ-ING	Changed the error type returned by bcm_cosq_gport_get API in case of non-existing VOQ. Instead of E_NONE to an actual error. This was done to be consistent with Jericho family.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-207228	COUNTERS-PP	When using VSI statistics, using API bcm_vlan_control_vlan_set, it is expected that egress_stat_id field will be equal to VSI. So far the API ignored the field. Now, it must be set with the same value with VSI once egress_stat_pp_profile is non 0.	88480_A0, 88690_A0, 88690_B1, 88800_A0
SDK-206602	FABRIC	Validation was added for all valid Fabric speed + FEC combinations so a clear error message will be returned if any invalid port resource combination is set for a fabric link.	88690_A0, 88690_B1, 88800_A0
SDK-204230	PMF	bcmFieldActionParsingStartType Action now should get input bcm_field_layer_type_t. supported values are: bcmFieldLayerTypeEth, bcmFieldLayerTypeIp4, bcmFieldLayerTypeIp6, bcmFieldLayerTypeMpls, bcmFieldLayerTypeArp, bcmFieldLayerTypeFcoe, bcmFieldLayerTypePppoe, bcmFieldLayerTypeSrv6Endpoint, bcmFieldLayerTypeSrv6Beyond, bcmFieldLayerTypeTm, bcmFieldLayerTypeForwardingMPLS, bcmFieldLayerTypeUnknown,	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0

		bcmFieldLayerTypeSctp. In order to get same functionality as before the change please use bcmFieldActionParsingStartTypeRaw instead.	
SDK-200941	OAM	In order to support hierarchical loss measurement for injected OAM over MPLS-TP, below sequence should be followed: 1. Identify y1731 OAM over PWE in egress parser by calling bcm_switch_control_indexed_set Please Refer to dnx_oam_identify_y1731_oam_over_pwe_egress in cint_dnx_utils_oam.c 2. Always enable counter in egress LIF profile by calling bcm_oam_profile_action_set. Please refer to dnx_oam_egress_with_update_counter_in_egr_profile in cint_dnx_utils_oam.c 3. For injected DMM/Rs, LMM/Rs, and piggyback CCMs (dual ended LM) over MPLS, control whether these packets should increment LM counters by calling API bcm_oam_profile_action_set with reserved egress acc profile, based on opcode. Please refer to appl_dnx_oam_hlm_egress_with_update_counter_in_egr_acc_profile in appl_ref_oam_init.c Note, with this code change, The number of available egress acc profile is reduced to 14. Profile 0 is default profile, profile 15 is reserved for injected OAM over MPLS 4. For OAM packets not mentioned above, counter increment indication can be controlled through the egress LIF profile, with bcm_oam_profile_action_set. 5. When an injected OAM packet also hits a LIF on a lower hierarchy on which another OAM endpoint exists, the OAM packet will be treated as a data packet on the lower hierarchy. For example If there are OAM MEPs with LM on a PWE and LSP LIF and the injected PWE OAM packet also hits the LSP LIF then the PWE OAM packets increment the LM counters on the LSP MEP. Note that changes from SDK-210904, SDK-213820 are also required for this JIRA. Steps 1 and 2 may affect backwards compatibility to 6.5.19. Existing OAM over PWE/MPLS applications may have to be updated accordingly.	88480_A0, 88690_B1, 88800_A0
SDK-191857	OAMP-LM-DM	In order to update the LM statistics for SLM entry upon reception of an SLR packet, the bcm_oam_loss_add() needs to be called with BCM_OAM_LOSS_UPDATE_NEXT_RECEIVED_SLR flag set and "id" set for endpoints with offloaded memory type, and "loss_id" set for endpoints with self-contained memory type. In previous releases this wasn't required for offloaded memory types, however functionality wasn't correct.	88480_A0, 88480_B0, 88690_B1, 88800_A0
SDK-191419	NIF	Changed the soc property name from "use_fabric_links_for_ilkn_nif" to "ilkn_use_fabric_links".	88690_A0, 88690_B1, 88800_A0
SDK-167463	DBAL	Change in string management. In order to maintain more efficient memory footprint	88480_A0,

		of SDK, the strings representing enum values was reduced. This change may affect SDK CMD diagnostics , since now, users doesn't need to set the full value name, but only a shorten value name for example setting value for field PORT_TERMINATION_PTC_PROFILE: was: PORT_TERMINATION_PTC_PROFILE=PORT_TERMINATION_PTC_PROFILE_PTCH2 now: PORT_TERMINATION_PTC_PROFILE=PTCH2	88480_B0, 88690_A0, 88690_B1, 88800_A0

Section 3.1.2: SDK build & load

Compile and set config files :

setenv SDK `pwd`

Example of SLK CPU compilation :

```
# Copy pre compiled mdb and kaps libraries into the relevant build folder.
# For SLK CPU big endian 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/slk_be/
cp  $SDK/libs/bin/dnx/SLK_64B_libkaps.a  $SDK/build/unix-user/slk_be/libkaps.a
cp  $SDK/libs/bin/dnx/SLK_64B_libmdb.a   $SDK/build/unix-user/slk_be/libmdb.a
# Additional mdb and kaps libraries flavors can be found under $SDK/libs/bin/.

# Compile SDK
cd $SDK/systems/linux/user/slk_be/
make MAKE_LOCAL=$SDK/make/local/dnx/Make.custom.dnx_kbp_slk
```

Common config files:

```
ln -fs $SDK/rc/rc.soc
ln -fs $SDK/rc/dnx.soc
ln -fs $SDK/rc/config-jer2pemla.bcm
ln -fs $SDK/tools/sand/db
ln -fs $SDK/rc/dnx_sku
ln -fs $SDK/rc/dnx_dram
ln -sf $SDK/rc/cmicfw/linkscan_led_fw.bin
ln -sf $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 -sf $SDK/rc/bcm88690_board.bcm
ln -sf $SDK/rc/bcm88690_legacy_interop_board.bcm

BCM8880X/BCM8882X specific links:
ln -fs $SDK/rc/config-j2c.bcm config.bcm
ln -sf $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

Run :

./bcm.user

Section 3.1.3: New Features since 6.5.19

The following list below specifies new features

JIRA	Module	Description	Affected Devices
SDK-218370	PMF-PROG-SELECT	Adding support to qualify on the MSB bit of the AcInLifWideData in CS of iPMF1. bcmQualifyAcInLifWideData should be used in bcm_field_presel_set() to qualify upon that bit. Cint example: cint_field_presel_ac_inlif_wide_data.c	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-214274	MPLS	RA label and other special labels can be trapped with below steps: 1. Disable the auto termination by bcm_switch_control_indexed_set() with bcmSwitchMplsSpecialLabelAutoTerminateDisable; 2. Create an MPLS tunnel for this label by bcm_mpls_tunnel_switch_create(); 3. Create per In-LIF trap for the tunnel LIF created in step 2 by bcm_rx_trap_lif_set(); Note: All the signals resolved by auto termination logic are reset, such as signal "force_is_oam" for RA label.	88480_B0, 88690_B1, 88800_A0
SDK-213820	PMF	ITMH_PPH application, also supports rebuilding FHEI type of VLAN-Edit if injected. For example if injected packet is ITMH_TSH_PPH_FHEI(vland-edit) then Ingress PMF rebuild all the system headers and the output will be FTMH_TSH_PPH_FHEI(vlan). This is required to support OAM Down MEP injection after the changes in SDK-210904	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0

SDK-213191	PMF	New feature add, trap profile. That allow to set 1b profile when configuring trap bcm_rx_trap_set(), later can be qualified upon in iPMF3. Tap flag was added: BCM_RX_TRAP_FLAGS2_UPDATE_ACL_PROFILE Preselector qualifier was added for IPMF3 stage: bcmFieldQualifyRxTrapProfile Cint example of qualifier usage: cint_field_presel_rx_trap_profile.c	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-212985	NIF, PHY	bcm_port_phy_control_set() can now dynamically set the TX/RX polarity for ports on both CDU and CLU, using BCM_PORT_PHY_CONTROL_TX_POLARITY or BCM_PORT_PHY_CONTROL_RX_POLARITY controls.	88480_A0, 88690_B1, 88800_A0
SDK-209506	RCY	Add support for a new port 'in header type' BCM_SWITCH_PORT_HEADER_TYPE_RCH_1 - for that port types, the SSPA is taken from the RCH (RCY header), which the pp port is updated according to a mapping table, which maps the SSPA to the pp port. The mapping table can be set with API: bcm_port_control_set() with type=bcmPortControlSystemPortInjectedMap. BCM_SWITCH_PORT_HEADER_TYPE_RCH_1 is expected to be used when the recycle is done on the Rx device, so the original SSPA is on the same device in the 2nd pass. BCM_SWITCH_PORT_HEADER_TYPE_RCH_0 should be used when the RCY is not on the RX device, so the PRT of the other device cannot map the original SSPA to pp_port.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-209030	LIF-MGMT	AC IN-LIF Wide data extension - Option to extend the wide data up to 58b, by additional lookup in ingress pipeline, when the key is the wide (generic) data from the AC IN-LIF. Can be used as ACL qualifier (bcmFieldQualifyAcInLifWideDataExtended). New APIs introduced: bcm_switch_wide_data_extension_add/delete/get/traverse New functionality is introduced in CINT: cint_inlif_wide_data.c	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-208830	PMF	Field Group type of bcmFieldGroupTypeDirectExtraction is not supported for stage bcmFieldStageEgress. Maximum key size is 32bits. Cint example: cint_field_dir_ext_epmf.c	88480_A0, 88480_B0, 88800_A0
SDK-206896	PMF-INGR	Parser now supports ETH1/2 classification according to EtherType. Classification was added as bit 14 of Ethernet Layer Qualifier for all parser stages. bcmFieldQualifyL2Format was add in PMF stage to qualify on such packets, value 0 qualifies on ETH-II while value 1 qualfiies on ETH-I.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-206144	BFD	Support for accelerated BFD over VXLAN, IPv4 and IPv6. If a VXLAN encapsulation FEC is defined, it can be used in the egress_if field when creating or modifying a BFD endpoint over UDP, including single-hop or multi-hop, and IPv4 or IPv6. Only IPv4 Underlay is supported with up to one VLAN tag for the underlay L2 header. Note: SDK-210904 is required, to support injecting ITMH + PPH rebuild by PMF (also for the case of BFD over VXLAN).	88480_B0, 88690_B1, 88800_A0

		Example usages: set use_vxlan_fec=1 or use_vxlan6_fec=1 in cint_sand_bfd.c or cint_sand_bfd_ipv6.c respectively.	
SDK-205415	METER	Added support for managing meter profile with-id, using bcm_policer_create using flag BCM_POLICER_REPLACE_SHARED with pol_cfg.entropy_id != 0 . For further details refer to UM.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-203609	PMF-QUAL-ACT	Parser now supports up to 2 IPv6 extension headers, first header gets parsed but second only gets recognized. To identify such packets in PMF bcmFieldQualifyIp6SecondAdditionalHeaderExist was add as Layer Record qualifier which can be checked in IPv6 Layer.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-199313	MIRROR-PP	Sequence number in GRE header of an ERSPAN tunnel is now supported. Tunnel API can be used for that BCM_TUNNEL_INIT_ERSPAN_WITH_SN and two new fields: stat_cmd and counter_command_id. See more information in CINT: src/examples/dnx/cint_dnx_mirror_example.c	88480_A0, 88480_B0, 88690_B1, 88800_A0
SDK-197627	COSQ-ING	New BCM API, bcm_cosq_gport_enable_set() / bcm_cosq_gport_enable_get(), was implemented to drop traffic at the ingress based on destination (system port).	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-191394	FC	Added a new type bcmPortControlPFCStatus to API bcm_port_control_get that will give the user the ability to retrieve the current received PFC status of a port. The API will return a bitmap with priorities that are currently under PFC.	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0
SDK-182658	COSQ-SCH	New shell command ('dnx tm scheduler fsM ') was added to control and read Flow Status Message counters.	88690_A0, 88690_B1
SDK-176626	METER	Meter: Support for meter expansion per TC is added.	88480_A0, 88480_B0, 88800_A0
SDK-162459	PMF-TCAM-MGMT	Entry add by location support, in order to use feature: 1) Specify bcmFieldTcamBankAllocationModeSelectWithLocation for bank_allocation_mode on tcam_params when adding a new Field Group, and specify a list of banks to pre-allocate (user can add banks later by calling 'bcm_field_tcam_bank_add()' function). 2) When adding a new entry for the field group, entry priority now points to the absolute TCAM location of the new entry to add, while still perceiving TCAM terminology for location structure (bank_id/bank_offset), even offsets are for single-sized/double-sized keys while odd offsets can be used to point to msb-half-entries for half-sized key entries	88480_A0, 88480_B0, 88690_A0, 88690_B1, 88800_A0

		cint example: cint_field_tcam_entry_add_by_location.c	
SDK-109605	COSQ-SCH	E2E port priority propagation is supported. For details, see User Manual	88480_A0, 88480_B0, 88800_A0

Section 3.2: DNXF-Family (BCM88790-Family)

Please use the PCIe firmware version 2.5.1-DNX with this release.

This version can be found under Firmware/pcie/dnx/v2.5.1/.

Section 3.2.1: Important Notes

Before integrating the new release, review this section thoroughly.

JIRA	Module	Description	Affected Devices
-	MBIST	We require that after running a Memory Built In Self Test (MBIST), that the device it ran on either be powered off or system reset (hardware reset by de-asserting the signal SYS_RST_N from the board). Without this action we do not guarantee that the device will work properly, and therefore from this release the software enforces it. This is a new requirement, independent of the SDK release. Note that both powering off the device and a system reset of the device, also involve a PCIe reset of the device, and the PCIe link of the device going down; so the customer BDE needs to support this operation if done while the BDE and SDK are running. We provide a callback function skeleton dnxc_perform_system_reset() in src/appl/reference/dnxc/appl_ref_board.c that may be used to perform these operations automatically on the customer board. An example of how this function is registered in the sample application can be seen in the dnxc_perform_system_reset() in src/appl/reference/dnxc/appl_ref_init_deinit.c . To have the call back function used automatically after running MBIST, and enable to continue running the SDK later, use this new soc property: perform_system_reset_when_needed=1 . If a device power off, or system reset is not actually performed, the SDK will recognize this, fail, and not continue running, and not run when started again. The MBIST is meant to test memories when there is a suspicion of device malfunction and is not mandatory. As described here, with some effort it is possible to run MBIST before each SDK runs, but this is not the intention of the test. The changed MBIST code in the release, together with the new requirement, fixes	88790

		possible false MBIST failures and possible PCIe issues	

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

This release contains:

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

Section 3.3.1: New Features since 6.5.19

The following list below specifies new features.

JIRA	Module	Description	Affected Devices
SDK-197627	COSQ-ING	New BCM API, bcm_cosq_gport_enable_set() / bcm_cosq_gport_enable_get(), was implemented to drop traffic at the ingress based on destination (system port).	88270_A0, 88470_A0, 88470_B0, 88670_A0, 88670_B0, 88680_A0

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

None