DNX Release Notes
For
Switch Software Development Kit

SDK 6.5.15

Broadcom Corporate Headquarters: San Jose, California, USA

Table of Contents

Section 1: About This Document

This document contains release notes for DNX devices affected by the Broadcom network switching Software Development Kit (SDK) release 6.5.15.

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 details about XGS device changes, common changes and general SDK issues, refer to RN-SDK6xx-R 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 Switch Devices” and “Supported PHYs” 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 Switch Devices” and “Preview PHYs” 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: New Supported DNX Switch Devices

Family	*Devices*	*Description*
Jericho2		
Ramon B0 rev		

Section 2.2: Preview DNX Switch Devices

Family	*Devices*	*Description*

Section 2.3: Build Sequence

BCM88790 and BCM88690 code was built and tested using the following command:

```
make -C systems/linux/user/slk_be/
MAKE_LOCAL=$SDK/make/local/dnx/Make.custom.slk_dnxc
```

note: “*Make.custom.slk_dnxc*” contain TOOLCHAIN_BASE_DIR, KERNDIR pointers to hard coded folders

Section 3: New Features per Device

This release is Beta version for the BCM88690 Family product line.

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

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

Section 3.1: BCM88690-Family

Section 3.1.1: Reference Documentation

Multiple documents describing relevant HW and SW aspects are available, including:

Section 3.1.1.1: SW/Arch documentation

- 88690-DG1XX: Traffic Management Architecture
- 88690-DG2XX: BCM88690 Packet Processing Architecture Specification
- 88670-88690-AN1XX: Traffic Manager Software Compatibility Guide
- 88670-88690-AN2XX: Packet Processing Software Compatibility Guide
- DBG16S-AN100: Blackhawk Configuration and Debugging Guide for StrataDNX 16 nm Devices - BlackHawk configuration user guide
- 88690-PG2XX: Traffic Manager Programmable guide
- 88690-PG1XX: Packet Processing Programmable guide

Section 3.1.1.2: In-Package Documents

- doc/sand/88690_UM_ShellCmd.html
- doc/sand/88690_UM_SoCProperties.html
- doc/sand/88790_UM_ShellCmd.html
- doc/sand/88790_UM_SoCProperties.html

Section 3.1.2: Important notes

- **Initialization of PEMLA ucode:**

Location now reside near the bcm.user inside pemla directory i.e.:
$SDK/systems/linux/user/slk_be/pemla/jericho_2/ucode/

In case ucode file is located other place, custom_feature_ucode_path must be set accordingly.

For **JR2 mode**, standard SDK (with no special soc properties, default behavior) use following ucode:
./pemla/jericho_2/ucode/standard_1/jer2pemla/u_code_db2pem.txt

For **JR1 interop mode**, standard SDK require following soc-properties:

- system_headers_mode=0
- custom_feature_ucode_path
  Following soc-property must be set and indicate the path of the ucode file , e.g.:

custom_feature_ucode_path=standard_1/**jr2-comp-jr1-mode/**u_code_db2pem.txt
Then SDK loads the following ucode:
./pemla/jericho_2/ucode/standard_1/**jr2-comp-jr1-mode**/u_code_db2pem.txt
(available as part of the TAR)

- **Packet Processing Important notes:**

Module	JIRA	Description

KBP/KAPS	SDK-161225	SDK is aligned to KBPSDK 1.5.5.1 KAPS and OP2 (Optimus Prime 2) are functional with KBPSDK 1.5.5.1
General	SDK-154578	**Important change** (in JR2 mode only): Network headers are now terminated at the ingress ITPP module for all cases except Trap strength > 7. The decision if to terminate or not is decided by the starting offset of the egress parser (which is automatically calculated). That means that at the egress device, packet arrives without terminated network headers which up until now ETPP removed them. The feature allows better parsing capabilities (looking at network headers in more depth) as well as reducing bandwidth over fabric. This is different behavior than EA4 where the feature wasn't enabled (and tested)
General	SDK-148983	PEMLA ucode loading soc properties were changed. See Important notes section ("Initialization of PEMLA ucode").
L3, FEC	-	**Important note**: Algorithm allocation of FEC (bcm_l3_egress_create) is going to be changed in 6.5.16. Therefore we suggest in this release (6.5.15) to use without ID only. In case WITH-ID is used, please note that the sequence will be changed between versions and a case that worked in 6.5.15 (or previous versions) will not work in 6.5.16. More details will be provided as part of 6.5.16. For 6.5.15, our best suggestion is to use the "Without-ID" approach.
Traps	SDK-163271	Default trap to CPU(bcmRxTrapDfltRedirectToCpuPacket) moved from BCM initialization Tune module to Application layer example. This means that in case those traps are required to be trapped, customers require to use application layer example or configure it on their own.
VLAN	SDK-115724	Remove TX VLAN-Tag is now functional. This means in

		case of correct-configuration, packet might TX tag removed while in JR2 EA4 release it wasn't the case.
MPLS-Port, Stat-PP	-	BCM_MPLS_PORT2_STAT_ENABLE is now used only in ingress MPLS-Port, for egress use flag BCM_MPLS_EGRESS_LABEL_STAT_ENABLE in egress_label structure. This is different behavior than JR2 EA4 release.
Diagnostics	SDK-156541	Layer records information in diagnostics range now from 0-7 instead of 1-8. This is different behavior than JR2 EA4 release. After packet pass use "sig get name=Layer" to see parsing layer information for different stages
QOS Tunnel/MPLS	-	MPLS/IP-Tunnel QOS properties are now introduced. This is done by ingress_qos_model, egress_qos_model structure in APIs. Note1: QOS mapping might be changed between versions because of the change. Note2: In case QOS model is not Pipe, SDK does not allow to set EXP/DSCP/TTL fields. This is different than previous devices and previous versions.
QOS L2	-	Default (profile 0) L2 remark mapping changed. At ingress, internal nwk_qos mapping from pcp and dei. In previous versions, nwk_qos mapping from pcp only; At egress, both pcp and dei mapping from nwk_qos. In previous version, pcp mapping from nwk_qos while dei mapping from DP.
MPLS , FEC	SDK-156702	JR2 mode only: in case FEC MPLS action is required , it must be BCM_MPLS_EGRESS_ACTION_SWAP_OR_PUSH. This is different behavior than JR2 EA4 release, where SWAP action only was available.
MDB	SDK-15706	dma_desc_aggregator_* soc properties are not valid in JR2 anymore (compared to previous versions), the feature does not improve the performance of MDB. It is now disabled. Remove related SOC properties.
MPLS-Port	-	P2P service is created by setting the vsi ID to 0 and setting the flag

		BCM_MPLS_PORT2_CROSS_CONNECT. This is different than previous versions.
Traps	SDK-164022	In current version the MTU trap support only 1 compression layer type per Lif/Rif. This will be fixed for later versions which might cause API modification.

- ## TM Important notes:

Module	JIRA	Description
HBM	SDK-159301	new set of timing HBM timing parameters, **users who save and restore tune parameters must repeat the process with the new timing parameters presented in this this release.**
INIT		Multi threading init (thread per unit) isn't supported in this release.
Linkscan	SDK-160583	When linkscan callback registered by the user is executed, the bcm_port_info_t will contain only link state information.
ITM	SDK-150273	Setting global drop thresholds by bcm_cosq_gport_static_threshold_set has been changed to not accept core-specific gport. Supported gports are global VSQ gport with BCM_CORE_ALL or 0-gport.
ILKN		The BCM API **bcm_port_ilkn_lane_id_map_set** is no longer supported. Use **ilkn_lane_map_<ilkn_id>_lane<num>=<ilkn lane id>** SOC property instead.
Egress and scheduler shapers		The port, TCG and TC shapers (for both egress and scheduler shapers) of channelized ports are set according to the below formula: ***shaper_rate = ifnterface_rate / number_of_channels*** This formula is set in the example sequence, at:

		● ***dnx_tune_egr_ofp_rate_set()*** - for egress shapers. ● ***dnx_tune_scheduler_port_rates_set()*** - for scheduler shapers. User can adopt these shapers to his needs according to the above reference code.

Section 3.1.3: SDK load

After building the SDK, use following sequence to load the driver (bcm.user) :

```
cd $SDK/systems/linux/user/slk_be/
ln -fs $SDK/rc/config-sand.bcm config.bcm
ln -fs $SDK/rc/rc.soc
ln -fs $SDK/rc/dnx.soc
ln -fs $SDK/rc/config-dcmn.bcm
ln -fs $SDK/rc/config-jer2.bcm
ln -fs $SDK/rc/config-jer2pemla.bcm
ln -fs $SDK/tools/sand/db
ln -fs $SDK/rc/bcm88690_revB_board.bcm
ln -sf $SDK/rc/bcm88690_board.bcm
ln -sf $SDK/rc/cmicfw/linkscan_led_fw.bin
ln -sf $SDK/rc/cmicfw/custom_led.bin
# Following links are needed in case the driver is used for BCM88790 as well.
ln -fs $SDK/rc/config-ramon.bcm
ln -fs $SDK/rc/bcm88790_board.soc
ln -fs $SDK/rc/dnxf.soc
ln -fs $SDK/rc/bcm88790_pizza_board.soc
./bcm.user
```

Section 3.1.4: Validated features

The features listed below completed basic-validation including at least 1 scenario of traffic passed. All other features, that are not mentioned here are not tested, yet and we do not recommend to start validating them.

Section 3.1.4.1: Access Features Status

The following items validated

- Access (TR) Tests (Beta):

Test Number	Test Name	Is Validated?
1	Register reset defaults	Yes
3	Register read/write	Yes
5	MBIST	Yes
6	Memories Test	Yes
7	Memories WR First Last Test	Yes
9	Test broadcast writes	Yes
50	Memory Fill/Verify	Yes
51	Memory Random Addr/Data	Yes
52	Rand Mem Addr, write all	Yes

71	Table DMA	Yes
152	Cache Memory Test	Yes
153	Memory Ser Test	Yes

- RCPU - GA level
- MBIST - GA level
- Interrupts - GA level
- Memory Shadowing - GA level
- SER - GA level with below exception:
    - Not supported items:
        - 2bits SER for MDB memory
        - SER for TCAM

Section 3.1.4.2: Basic data path, Connectivity and Traffic Management Features Status

The following items:

- HBM - Beta level
- Ingress Cosq
    - VOQ creation - GA level
    - Ingress packet scheduler - GA level
    - Header compensation - GA level
    - Ingress congestion manager - GA level
    - System RED - GA level
- Egress Queuing
    - Egress packet scheduler - GA level
    - Egress CGM - GA level
    - Header compensation - GA level
- Egress Credit Scheduler
    - Scheduling hierarchy and shaping - GA level.
    - Shared shapers (flow priority propagation) - GA level
- Multicast - GA level.
- Mirror
    - Port Mirroring (both inbound and outbound) in GA level.
- LAG - GA level with the below exception:

    - Lag as an egress MC group destination is tested only on stand-alone device.
- Statistics and Counting
    - CRPS - GA level
    - Statistic Interface -
    - full line-rate not tested.
    - Known issue with source unmapping. See details in know issues section.
- Meter
    - Ingress Metering - GA level
    - Egress Metering - GA levell
- Dynamic port - GA level
    - Only ports with NIF ports interface or mirror-recycle ports are supporting dynamic ports add / remove. Other ports, including “simple” recycles ports will not support dynamic port add / remove.
- NIF
    - All ETH interfaces were validated
    - ILKN - GA level for NRZ rates only.
    - Autoneg
        - Works with NRZ. Not validated yet with PAM4.
    - MIB
        - Both ETH and ILKN MIB counters are GA level.
    - Eyescan - GA level.
- NIF Priority Drop - GA level
- SyncE - GA level
- LED - GA level
- Linkscan - GA level
- Fabric
    - Connectivity with Ramon and FE3200 - validated
    - Mesh - GA level.
    - Source routed cells - GA level
    - Isolation - GA level
    - Fabric CGM - GA level
- Flow Control - GA level

Section 3.1.4.3: Packet-Processing Validated features

- RAW (Force forward)
    - Useful for bandwidth tests that skip PP functionality
    - API bcm_port_force_forward_set is functional
- L2:
    - Static MACT forwarding

  - Learning
    - Ingress learning
    - Distributed
    - Centralized (DMA to CPU)

    Note: In JR1 Interop mode, learning functionality is unavailable in this release.
  - Flush machine
    - HW accelerated
    - FIFO DMA
    - Bulk operations
  - MACT global limit

  Examples:
  - cint_l2_basic_bridge_with_vlan_editing.c
  - cint_l2_traverse.c
  - cint_l2_trunk.c

  Notes:
  - Aging global configuration APIs changed from JR1. Please see L2 Backward compatible section.
  - Aging is disabled by default due to HW-Errata.
- L3
  - Forwarding according to host lookup (Exact match)
  - Forwarding according to route lookup (LPM)
  - Forwarding according to default route
  - uRPF loose/strict
  - Basic IPMC - forwarding IP compatible MC packets according to lookups in LEM, LPM and TCAM
  - Fallback-to-bridge. Note: JR1 interop mode, fallback-to-bridge is not functional.
  - ECMP (3 level)
  - ECMP RPF (using EXEM)
  - My-MAC enhancements (VRRP)
  - FEC allocation - only a very simple mode is supported today. More flexibility will be provided in the next releases.
  - Routed packets learning

  Examples:
  - cint_ip_route_basic.c
  - cint_ip_route_fwd_priorities.c
  - cint_ip_route_rpf_basic.c
  - cint_ipmc_route_basic.c
  - cint_l3_vrrp.c
  - cint_multiple_my_mac_term.c
  - cint_ecmp_basic.c
  - cint_sand_multi_device_ipv4_uc.c (JR1 Interop mode)

    - cint_sand_multi_device_ipmc.c (JR1 interop mode)

  Notes:

    - TCAM major change: BCM_IPMC_TCAM flag is introduced. It should be set in order for entry to reside in the TCAM. See backward compatible document for more details on IPMC flows and differences between JR1 and JR2.
- MDB:
    - Added preliminary support for additional MDB profiles, the MDB profile is set using the mdb_profile soc property.
      Complete list of currently supported MDB profiles:
        - **Balanced** profile offers a balanced partitioning of MDB resources between the Physical DBs used by L2 and L3 Application DBs
        - **Balanced-Exem** profile is similar to the Balanced profile, but with allocation for EXEM tables.
        - **L2-XL** profile allocates additional MDB resources to physical DBs used by L2 Application DBs, at the expense of Physical DBs used by L3 Application DB.
        - **L3-XL** profile allocates additional MDB resources to Physical DBs used by L3 Application DBs, at the expense of Physical DBs used for L2 Application DB.
        - **Ext-kbp** profile assumes the IP FIB is stored on an external device. Therefore, room can be made for other databases such ISEM, InLIF, or LEM.
    - Default profile is set to **Balanced-Exem**.
    - Minor changes to the profiles are expected in future releases.
    - Note: Split resources between KAPS Private and Public will be available in 6.5.16.
- PPMC
    - Required when more than one outlif is required in multicast replication database.
    - Expends up to 3 the number of outlifs that can be used by an MC replication.

  Examples:

    - cint_multicast_pp_basic.c
- MPLS :
    - LSR: forwarding in ILM
    - LER: termination and encapsulation of mpls tunnel (Including 4 label termination and Deep MPLS segment routing)
    - FRR termination
    - MLDP
    - L3VPN with IPv6/IPv4 forwarding
    - FEC - Push or Swap operation (without OutLIF)

  Examples:

  - cint_mpls_encapsulation_basic.c
  - cint_mpls_termination_basic.c
  - cint_sand_multi_device_mpls_lsr.c (JR1 interop mode)

  Note: LSR forwarding in ILM Arch flow changed significantly from JR1. Please read Arch document and PG document MPLS sections.

  Note2: Major changes were done in regards to QOS-Model for MPLS-Tunnels, please see QOS item.
- VPLS & VPWS :
  - Encapsulation & termination of PWE
  - Split horizon
  - Protection cases of PWE (1:1, 1+1) are not supported in this release.
  - Learning - only following cases are supported in this release:
    - MPLS-FEC + PWE-LIF (PWE-OutLIF equals PWE-InLIF)
    - PWE-LIF + Port (PWE-OutLIF equals PWE-InLIF)
    - 1+1 Protection (MC-ID)
  - PWE tagged mode Cross-Connect
  - Native VLAN editing over PWE
  - PWE Network & LIF-scope

  Examples:
  - cint_vpls_mp_basic.c
  - cint_dnx_utils_vpls.c

  Note: MPLS-Port API supports INGRESS, EGRESS only sides. Also FEC creation is done only by bcm_l3_egress_create. For more information, please see Backward Compatible document VPLS section.
  Note2: P2P service is created by setting the vsi ID to 0 and setting the flag BCM_MPLS_PORT2_CROSS_CONNECT
  Note3: Major changes were done in regards to QOS-Model for PWE, please see QOS item.

- VLAN:
  - VLAN-translation
    - IVE
    - EVE
  - VLAN-membership
  - Flooding per VLAN
  - Same-interface filtering
  - VSI = VLAN
  - VLAN Untagged lookup case: BCM_VLAN_PORT_MATCH_PORT_UNTAGGED is now supported (Incoming-Port lookup for untagged packet cases)

  Examples:
  - cint_l2_basic_bridge_with_vlan_editing.c
  - cint_vlan_membership.c
  - cint_vlan_match_untagged_packet.c

  Note 1: VLAN-translation APIs in JR2 are working in “advanced vlan translation” mode only (See backward compatible document).

  Note 2: VLAN-membership APIs (bcm_vlan_port_add, bcm_vlan_gport_add) do not handle multicast replications anymore. This should be done directly by multicast APIs. For more information see Backward compatible document "VLAN Membership API Changes"
- L2VPN & Cross-Connect:
  - Create service symmetrical ACs
  - Vswitch creation and gport attachment
  - Cross-Connect support at VLAN-Port creation
  - Cross-Connect between any AC & PWE

  Examples:
  - cint_vswitch_metro_mp.c
  - cint_vswitch_cross_connect_p2p_ac_pwe.c
- IPv4 & IPv6 Tunnels & VXLAN:
  - IPv4/6 tunnel termination with GRE/VXLAN/UDP
  - IPv4/6 tunnel encapsulation with GRE/VXLAN/UDP
  - VXLAN flows (UC, MC, BOO, ROO)
  - Notes:
    - IPv6 termination is supported both P2P and MP.
    - VXLAN-GPE is now supported.
    - VXLAN Split-Horizon is now supported
    - Major changes were done in regards to QOS-Model for IP-Tunnel, VXLAN, please see QOS item.
    - VXLAN Egress processing major change: In order to achieve VNI+AC processing, it is expected to configure Native-AC. For more information see VXLAN CINTs.

  Examples:
  - cint_ip_tunnel_term_basic.c
  - cint_ip_tunnel_encap_basic.c
  - cint_vxlan_tunnel_termination_basic.c
  - cint_dnx_ipv6_tunnel_termination_basic.c
  - cint_vxlan_roo_basic.c
  - cint_sand_multi_device_vxlan_roo.c (JR1 interop mode)
  - cint_sand_multi_device_vxlan_roo_mc.c (JR1 interop mode)
- Protection - following cases were validated:
  - 1:1 vlan port protection
  - FEC protection (including 3-level protection)
  - MPLS port protection

- Note: EVPN protection is not supported yet.

Examples:

- cint_ac_1to1_coupled_protection.c
- cint_dnx_vswitch_vpls_mpls_1plus1_protection.c
- cint_ac_fec_facility_protection.c

Note: This feature is unavailable in JR1 interop mode in this release.

- Persistent hashing / Load-balancing:
    - Load-Balancing keys per Header are validated. Please see Arch AG document for which fields are part of Load-balancing keys.
    - Fragmented IPv4 header is now considered.
    - Removing ECMP member don't affect the remaining members paths.
    - Adding ECMP member hardly affect the remaining members paths.
    - Note: Load-balancing configuration is significantly changed from previous devices, please read the PG document for full configuration sequence.

  Examples:
    - cint_ecmp_basic.c
- EVPN: UC & MC flows were validated
    - EVPN UC label termination
    - EVPN UC label encapsulation
    - EVPN MC label and ESI termination
    - EVPN MC label and ESI encapsulation
    - Example:
        - cint_dnx_evpn_basic.c
        - cint_dnx_evpn_vxlan.c
        - cint_dnx_evpn_trunk.c
        - cint_sand_multi_device_evpn.c (JR1 interop mode)
    - Note: P2P service is currently created by setting the vsi ID to 0 and setting the flag BCM_MPLS_SWITCH2_CROSS_CONNECT (see UM and example in cint_dnx_evpn_basic.c).
    - Note2: Major changes were done in regards to QOS-Model for EVPN, please see QOS item.
- QOS:
    - PHB - per hop behavior
    - Remark/Marking - PCP/DEI, DSCP, EXP
    - Egress QOS Pipe model supports both My-Namespace & Next-Namespace
    - Examples:
        - cint_qos_l2_phb.c
        - cint_dnx_qos_l2_port.c
        - cint_qos_l3_remark.c
        - cint_mpls_qos.c
        - cint_vpls_mpls_qos.c

- cint_ip_route_basic_qos.c
- Note1: QOS module was changed significantly from JR1, please read the PG QOS section.
- Note2: New field param **ingress_qos_model**/**egress_qos_model** are used for QOS/TTL Pipe/Uniform model settings over tunnels (IP/MPLS). This replace known QOS flags like BCM_MPLS_SWITCH_OUTER_EXP, BCM_MPLS_SWITCH_OUTER_TTL, BCM_MPLS_EGRESS_LABEL_EXP_SET, BCM_MPLS_EGRESS_LABEL_EXP_COPY, BCM_MPLS_EGRESS_LABEL_TTL_SET, BCM_MPLS_EGRESS_LABEL_TTL_COPY, dscp_sel, BCM_TUNNEL_TERM_USE_OUTER_TTL, BCM_TUNNEL_TERM_USE_OUTER_DSCP, BCM_TUNNEL_INIT_USE_OUTER_TTL . Examples:
  - cint_sand_ip_tunnel.c
  - cint_mpls_tunnel_initiator.c
  - cint_sand_mpls_lsr.c
- Note3: Please see known issues for QOS.

- BFD IPv4 Non-Accelerated :
  - cint_sand_bfd.c - bfd_ipv4_run_with_defaults(). Single hop, multi-hop, BFD Echo and Micro-BFD have all been verified, albeit only non-accelerated functionality.
  - Note: This feature is unavailable in JR1 interop mode in this release.
- BFD IPv6 Non-Accelerated :
  - BFD over IPv6 (non accelerated) Note: This feature is unavailable in JR1 interop mode in this release.
    - Refer to cint_sand_bfd_ipv6.c for an example.
- BFD over MPLS Non-Accelerated:
  - BFD over MPLS has been tested, however cints have not yet been ported
  - Note: This feature is unavailable in JR1 interop mode in this release.
- OAM-Classifier, CFM, Counters Non-Accelerated:
  - OAM classification has been tested, including bcm_oam_endpoint_create/get/destroy() API as well as the three profile setting APIs bcm_oam_profile_create(), bcm_oam_lif_profile_set(), bcm_oam_profile_action_set().
  - A basic example can be found in oam_run_with_defaults() in the file cint_sand_oam.c
  - Examples of using different profiles and actions can be found in the file cint_oam_action.c
  - An example of configuring the counter processor for LM can be found in set_counter_resource(), in cint_oam_basic.
  - All cints have been tested (Only non accelerated functionality).

  - OAM down MEP and up MEP looback has been tested. Both downmep and upmep loopbacks configured by the default application. For details, refer to the default application functions "appl_dnx_oam_downmep_reflector_init()" and "appl_dnx_oam_upmep_reflector_init()" located in src/appl/reference/dnx/appl_ref_oam_init.c and to section "OAM Ethernet Loopback" at the user manual (PG document).
  - Note: This feature is unavailable in JR1 interop mode in this release.
- ROO & Drop-and-Continue:
  - ROO and BUD Node flows are validated for VXLAN and MLDP
  - Examples:
    - cint_vxlan_bud_node.c - basic drop and continue flow example with VXLAN tunnel
    - cint_vxlan_roo_basic.c - contains examples of VXLAN over IPV4 and IPV6 ROO for both encapsulation and termination flows.
  - Note: Features RCH, Drop-and-Continue is unavailable in JR1 interop mode in this release.
- Reflector:
  - Four variants of reflector applications are supported by DNX SDK:
    - L2 external unicast
    - L2 external multicast
    - L2 internal multicast
    - L3 internal unicast

    All of them have been tested. Further details together with pointers to cint examples can be found at the user manual (PG document), section "RFC 2544 Reflector"
- TWAMP
  - TX/RX non-accelerated flows
  - Reflector flows
    - First pass: TWAMP sender data moved, RX timestamp stamped, SIP/DIP swapped, UDP ports swapped, TTL initialized to 255
    - Second pass: Forwarded by swapped DIP, TX timestamp and SEQ stamped
  - Examples:
    - cint_dnx_twamp_field.c - contains the PMF rule for TX/RX non-accelerated over IPv4 and TWAMP Reflector over IPV4.
    - Cint_dnx_twamp.c - contains examples of TX/RX non-accelerated over IPv4 and TWAMP Reflector over IPV4.
- Stat-PP:
  - Tested features (configuration + traffic):
    - IPV4 route & host statistics
    - Ingress VSI/RIF statistics

        - Egress RIF statistics
          (Note: Only Stat-profiles 1-3 are available)
        - L2 FWD statistics
        - MPLS Forwarding statistics
        - MPLS Termination statistics
        - MPLS Encapsulation
          (Note: Only Stat-profiles 1-3 are available)
        - MPLS PHP statistics
        - ARP
          (Note: Only Stat-profiles 1-3 are available)
        - Ingress port statistics
        - IPV4\6 host rpf statistics
        - IPV4\6 multicast statistics
        - ECMP, FEC FWD statistics
    - Cints:
        - cint_stat_pp_config.c
        - cint_stat_pp_ipv4_route.c
        - cint_stat_pp_l2_egress_fwd.c
        - cint_stat_pp_mpls_egress.c
        - cint_stat_pp_mpls_termination.c
        - cint_stat_pp_port_ingress.c
        - cint_stat_pp_ipmc_rpf
- Visibility:
    - after bcm init system is in “Selective” - mode meaning that first packet will always update signal data according to per port configuration
        - Pay attention that after each packet in order to have new packet data, old one need to be cleared. For CPU packet TX it is done implicitly, no action requested. In order to do it for others(external) packets use command “visibility resume” after you are done with current one
    - System mode may be viewed and modified via “visibility global” command or via new BCM API - *bcm_instru_control_set/get*
    - There is a new reference application “appl_dnx_sys_vis_enable” which is loaded on top of init and enables NIF, CPU & OAMP ports for signal sampling, meaning no signals for OLP packets
    - Port status may be viewed and modified also via “visibility port” command or via new BCM API - *bcm_instru_gport_control_set/get*
    - In order to mimic JR1 functionality, we suggest to use “Always” mode
- PP-Diagnostics.
    - “export” command is functional and provide a dump of the main PP functionality into XML file (located in current directory).
    - “sig get” command is functional

- “ikleap” command show the PP highlight information of visibility packet. “ikleap” for ingress KLEAP.
- “ekleap” command show the PP highlight information of visibility packet. “ekleap” for egress KLEAP.
- “pp” diagnostics highlights commands:

PP Diagnostic Command (JR2)	Parallel Command (in JR1)	Command Brief
pp vis ppi (PP VISbility PacketParsingInfo)	**diag pp pi** + **diag pp rpi**	-Last packet parsing record – its headers stuck and their qualifiers. -Incoming port information.
pp vis pkttm (PP VISbility PacKeTTM)	**diag pp pkttm**	-Information to be processed and used by Ingress TM blocks.
pp vis fdt (PP VISbility Forward Decision Trace)	**diag pp fdt**	-Traces the changes of the forwarding decision for the last packet.
pp vis fec (PP VISbility FEC) **pp inf fec** (PP INFo FEC)	**diag pp fec last_fec** **diag pp fec all**	- Dumps FECs that were hit in last packet. - Dumps all FEC entries that are configured in the device.
pp vis term (PP VISbility TERMination)	**diag pp termi**	- Dump terminated headers per-stage.
pp info sysh (PP Info SYS_Header)	**-**	- Show the system headers that are passing from ingress to egress.
pp vis ive (PP Visibility IngressVlanEditing)	**diag pp ive**	- Show Ingress VLAN editing information of the last received packet.

pp vis eve (PP Visibility IngressVlanEditing)	**diag pp eve**	- Show Egress VLAN editing information of the last received packet. Shown are both Egress VLAN edit in forwarding stage, and VLAN edit in Encap stage.
pp vis last (PP Visibility LAST)	**diag pp last**	- Shows last packet information from IRPP blocks.
pp vis ctx (PP Visibility ConTeXt)	**-**	- Shows context value and context enablers for all relevant stages for the last packet.
pp vis occ (PP Visibility OCCupation)	**diag pp occ**	-Displays inlif and outlif profile allocated resources

Note: Diagnostics-PP are in Beta quality. Commands are in early stages and might be changed in the coming releases. We encourage to use them for debug purposes but please do not use them as a format to parse into your application (because changes can still happen in the future).

- KBP OP2 interop:
  - JR2 can be connected with OP2 to extend the capacity.

  Following features were validated:
  - IPv4/6 UC packet perform lookup on external TCAM
  - IPv4 MC support
  - Up to two ACLs for IPv4/IPv6 UC packets in sequential lookup Ids
  - result size for ACLs has to be 32bit
  - total ACL key size 160 bit (master key).
  - Up to 8 FFCs for ACLs
  - Up to 2 consecutive ACLs over MPLS opcode (starting from lookup 0).

  For information on how to bring-up device with OP2 , please see PG document section “External Lookup” (Version 88690-PG102)

  Notes: Any other functionality (IPv6 or MC, Other ACL lookup sizes, number of lookups, different opcodes than IPv4 are not supported and not functional yet)

- Interop with JR1 notes:

- JR2 can be connected with Jericho/Jericho+ via soc property system_headers_mode=0 (legacy devices) and dedicate ucode file: u_code_db2pemjr2-comp-jr1-mode.txt (see important notes section) Following features were validated:
    - RAW
    - L2 static forwarding
    - L2 same-interface filtering
    - L3
    - Trap packets
    - IPMC
    - LSR, LER
    - VPLS
    - QOS
    - VLAN
    - VXLAN, VxLAN ROO
    - Split-Horizon
    - EVPN
    - PTP
    - KBP basic L3 flow
    - ITMH JR1 interop without PPH injection is validated. In order to support it define a port that support PTCH2 over ITMH JR1 header, with new header type INJECTED_PP_JR1_MODE
    - FTMH Load-balancing is now available using soc property system_ftmh_load_balancing_ext_mode (0/1)
    - FTMH stacking extension for user-header ACL processing only is now available using soc property stacking_extension_enable
    - FTMH DSP extension is available using soc-property ftmh_dsp_extension_add
- Examples:
    - cint_sand_multi_device_ipv4_uc.c
    - cint_sand_multi_device_mpls_lsr.c
    - cint_sand_multi_device_ipmc.c
    - cint_sand_multi_device_vxlan_roo.c
    - cint_sand_multi_device_vxlan_roo_mc.c
    - cint_sand_multi_device_evpn.c

Section 3.1.4.4: ACL Validated features

- PMF Feature list
    - TCAM Field Groups sizes 80/160/320 in all PMF stages
        - Examples:
            - cint_field_attach.c
            - cint_field_tcam_prefix.c
    - Direct Extraction Field groups

  - Examples:
    - cint_field_basic_de.c
    - cint_field_fem.c
- Context Selection
  - Examples:
    - cint_field_presel_fwd_layer.c
- Context Management
  - Create/Destroy
    - cint_field_presel_fwd_layer.c
  - Key Gen Variable for const Qualifiers
    - cint_field_key_gen_var.c
  - System Header profile selection
    - cint_field_trap_code_no_sys_hdr.c
  - Hash creation
    - cint_field_hashing.c
  - Compare
    - cint_field_compare_single.c
    - cint_field_compare_double.c
    - cint_field_compare_full_offset_get.c
    - cint_field_compare_specific_offset
- Cascading
  - iPMF1/2 to iPMF3
  - Ingress - Egress
  - Examples:
    - cint_field_ipmf2_ipmf3_cascading.c
    - cint_field_udh_ipmf1_epmf.c
    - cint_field_udh_ipmf1_ipmf3.c
- State Table
  - Examples:
    - cint_field_state_table.c
- Direct TCAM Field Groups
  - Examples:
    - cint_field_tcam_dt.c
- Port Profiling
  - Examples:
    - cint_field_port_profile.c
    - cint_field_general_data.c
- ACE
  - Examples:
    - cint_field_ace.c
- Exact Match:
  - Examples:
    - cint_field_exem.c

  - ACL KBP: See KBP OP2 interop section above.
  - Diagnostics
    - Command “field” - see options
  - Application list
    - ITMH
    - ITMH_PPH
    - ITMH_J1- parsing the ITMH of J1 when device is in J1 mode
  - Jericho1 inter OP
    - UDH cascading J1 mode is supported
      - Jericho1 had only 2 user header supported for use of cascading in PMF. In Jericho2 currently both header should be set with same size.

- Traps Feature List
  - Ingress Traps
    - Examples:
      - cint_rx_trap_ingress_traps.c
      - cint_rx_trap_saequalsda_trap.c
      - cint_rx_trap_1588_traps.c
  - Protocol traps
    - Examples:
      - cint_rx_trap_protocol_traps.c
  - Programmable traps
    - Examples:
      - cint_rx_trap_programmable_traps.c
  - LIF traps
    - Examples:
      - cint_rx_trap_lif_traps.c
  - ERPP Traps
    - Examples:
      - cint_rx_trap_erpp_epmf_trap_action.c
      - cint_rx_trap_erpp_traps.c
  - ETPP Traps
    - Examples:
      - cint_rx_trap_etpp_traps.c
      - cint_rx_trap_etpp_trap_actions.c
  - ETPP MTU trap
    - Examples:
      - cint_rx_trap_etpp_mtu_trap_port.c
      - cint_rx_trap_etpp_mtu_trap_lif.c
      - cint_rx_trap_etpp_mtu_trap_rif.c

  - Diagnostics

- "Trap" Command - see options

Section 3.1.5: Major Bug Fixes

Section 3.1.5.1: Packet-Processing Major Bug Fixes

JIRA-ID	Module	Description
SDK-157749	Diagnostics-PP	'export' command with scripts residing in the default directory ($SDK/tools/sand/db/jericho_2/export/dump_all.xml) The file provides the main PP diagnostic commands to dump information with.
-	Trap/CPU-Port	When packets arrives to CPU-port (header_type_out=CPU) untrapped (i.e. Is-trap is not set by the ingress processing), then in JR2 EA4 release, packet-processing didn't handle well such cases. In 6.5.15 it is expected that: 1. Fwd Header will not be changed 2. Encap Headers will be done according to OutLIF pointers (regular Packet processing) 3. System-headers will be appended
-	MPLS	Native MAC DA non zero after MPLS termination did not process well
SDK-146924	QOS	Support Egress Pipe My-Namespace
SDK-156927	MPLS	MPLS PHP action for ILM non-BOS label did not work as Egress FWD stage popped ETPS entry without applying it to the packet. In 6.5.15, Egress FWD stage does not pop ETPS entry for such tunnel allowing ENCAP stage to properly perform the PHP action
SDK-115724	VLAN	Removal TX-tag was not functional, in this version in case outgoing VLAN tag is present and equals Port's

		default VLAN, then Tag is removed. In order to disable such functional call bcm_vlan_gport_add with flag "BCM_VLAN_GPORT_ADD_UNTAGGED" unset on given Out-Port, VLAN.
SDK-156431	L3VPN	IPv6 forwarding after MPLS termination failed due to 2nd parser mis-configuration. The issue is now fixed.
SDK-148807	VXLAN	VXLAN now support correctly split-horizon
SDK-157185	Tunnel	IPv6 over UDPv6 tunnels can now be terminated correctly (Packet format: PayloadoIPv6oUDPoIPv6oETH)
SDK-158789	L3	ARP+AC object now supports also 2 VLANs in VLAN translation. New flag introduced in bcm_l3_egress_create API, BCM_L3_FLAGS2_VLAN_TRANSLATION_TWO_VLAN_TAGS.
SDK-158348	L2, GPM	bcm_l2_gport_forward_info_get is now available
-	MPLS	L3VPN with IPv4, IPv6 forwarding is now supported
-	RCH processing	RCH processing is fixed and should work for Reflector & Bud-node cases
-	CPU	CPU port can be also local-port different than 0
SDK-161515	L3	L3: Delete one RIF influences the behavior of other RIF in the device which shouldn't be. This is now fixed. The issue happens because of VSI profile mutual resources.
-	VXLAN	VXLAN encapsulation had incorrect DIP value (always zeros). This is now fixed.

Section 3.1.5.2: ACL Validated Bug Fixes

- Stage IFWD2 in diag renamed to External
- TCAM Direct Table entries are now accessed by entry_access_id as in the case of TCAM regular entries.
- “field port list” diag was fixed
- “field map” diag was removed and instead “field qual /act” now print same diags
- Bug that allowed same name for 2 or more of the same object. (For example: same name for 2 different Field Groups)
    - New restriction on “name” parameter, cannot be the same for 2 different objects
        - Field Groups
        - User created Actions
        - User created Qualifier
- User defined traps will have different HW codes.
- Ingress Egress cascading was fixed UDH header qualifiers and actions 0-3 can be used for cascading
- “field attach” diag showed wrong fes info - this was fixed
- Some LIFs/RIFs have 2b mtu profile, mtu profile allocation was adjusted to support either 2b/3b profile allocation according to LIF/RIF.
- IPv6/IPv4 qualifiers changed should no longer be used:
    - bcmFieldQualifyIpProtocol
    - bcmFieldQualifyTtl
    - bcmFieldQualifyDSCP

      Note: if will be used will now relate to IPv6 header and not IPv4 like previous version.

      instead

    - for ipv4:
        - bcmFieldQualifyIp4Protocol
        - bcmFieldQualifyIp4Ttl
        - bcmFieldQualifyIp4Tos
    - for IPv6
        - bcmFieldQualifyIp6NextHeader
        - bcmFieldQualifyIp6HopLimit
        - bcmFieldQualifyIp6TrafficClass
- To configure “bytes_to_remove” is property of iPMF2 and not iPMF3, this was fixed
    - Diagnostic is fixed
    - bcm_field_context_param_set() API is also fixed

- removed support of bcmFieldQualifyInterfaceClassPort use bcmFieldQualifyPortClassTrafficManagement instead

Section 3.1.6: Errata

Section 3.1.6.1: Basic data path, Connectivity and Traffic Management Errata

- Multi threading init isn't supported.
- SDK selects internal reference source clock for TS PLL (PLL 1). THe selection should be 25Mhz external reference clock. It will be fixed in the next release.
- STIF: bcm_dnx_stat_stif_source_mapping_set with BCM_PORT_INVALID to unmap the source is expected to fail.

Section 3.1.6.2: Packet-Processing Errata

- LIF manager: In current release, it is not possible to replace LIF information which influence the Payload fields existence in the LIF. Some of the combinations may be supported in future versions (for example moving from non-stats LIF to stats LIF). The issue exist for all LIF types: VLAN-Port, MPLS-Port, IP-Tunnel, MPLS-Tunnel, ARP, etc…
- MPLS-Port: By mistake MPLS-Port API does not return error when moving from non-protection to protection and vise versa. This will be fixed in future releases.
- VXLAN: VSI to VNI information is not fully implemented. This cause VXLAN QOS to be non functional
- VXLAN: IPv6 Bud node flow is not functional.
- VXLAN: UDP Source port is not concluded by LB-key as it should be and always set 0x5000.
- VXLAN/UDP Tunnels: It is not possible in this release to change UDP port values to recognize VXLAN / UDP tunnels. In this release, it is assumed to be 0x5555 for VXLAN.

- MDB: Egress-OAM LIF and MPLS Forwarding databases may have the wrong entry deleted when performing a delete operation at high capacity utilization of the database.
- FEC allocation: FEC allocation algorithm is not fully utilizing HW resources and its algorithm will change soon which effect WITH_ID models. We suggest in this version to work without ID approach only. Please see Important notes for more details.
- Connection to KBP OP2 - see validated section
- LIF Same interface control is not supported in this release. It is not possible to set device-scope or system-scope per LIF (useful in several VPLS cases)
- JR1 interop mode:
    - Fallback to bridge is not supported
    - Learning is not supported
    - PPHoITMH JR1 mode injection is not supported
    - UDH (user-define header) is not supported
    - OAM/BFD is not functional
    - Fallback to Bridge is not functional
    - RCH & Bud-node processing is not functional

Section 3.1.6.3: ACL Validated Errata

None.

Section 3.1.7: Features Maturity

Section 3.1.7.1: Vocabulary

- Empty cell indicates the feature is not implemented/tested yet.
  Bugs should not be reported, as those will not be fixed at this point.
- “PRE” indicates one traffic test passed. It is not recommended to start using the feature.
  Bugs should not be reported, as those will not be fixed at this point.

- “Beta” indicates the feature can be used. The majority of regression tests pass. Issues found can be reported as CFDs (Customer Found Defect) and will be fixed.
- “GA” indicates the feature can be used assuming it was fully validated. Issues found can be reported as CFDs (Customer Found Defect) and will be fixed.
- Features in green color are new features introduced in JR2.

Section 3.1.7.2: Features list

Section 3.2: BCM88790-Family

Section 3.2.1: Important Notes

Multicast: new mode

New feature is added to Ramon in Direct List-Of-Faps mode.

This new feature allows fap-ids range to be different than 0-191/95, by configuring modid offset for the FAP devices.

Fap-ids in this mode will range between [modid offset] and [modid offset + 191] instead of between 0-191 (or will range between [modid offset] and [modid offset + 95] instead of between 0-95).

The offset can be set using bcm_fabric_control_set API with the control bcmFabricControlDirectMcModidOffset.

Notes:

1. Allowed offset values are between 0 and 1856.

2. For the below APIs, need to subtract the offset value from the real modid prior to calling the API:

- bcm_fabric_multicast_set (when using modid and not local modid).
- bcm_multicast_egress_set/get.

- bcm_fabric_multicast_multi_set.

  The get function will also return [modid - base modid] value.

3. Offset value can be set only before creating the first multicast group.

4. If this API is not called, the offset will be 0, preserving the previous functionality, i.e. fap-ids range is 0-191/95.

5. This mode is not supported for INDIRECT multicast mode.

Section 3.2.2: Validated Features

Section 3.2.3: Known Issues

LCPLL

LCPLL should work on bypass mode. Use the following SoC property to configure the LCPLL to bypass mode: serdes_fabric_clk_freq_out.BCM8879X=bypass.

The input reference clock might be either 156.25MHz or 312.5MHz.

Ports

Ramon A0 supports only a single VCO on each BH.

Section 3.2.4: Bug fixes since 6.5.14 release

Easy reload

When performing a boot in easy reload mode, i.e. changing the boot flags (SOC_BOOT_FLAGS) to easy reload (BOOT_F_RELOAD) errors will occur upon init.
Issue was fixed.

Isolation

In multi stage system, when isolating and un-isolating a device:
bcm_fabric_control_set(unit,bcmFabricIsolate,1);
bcm_fabric_control_set(unit,bcmFabricIsolate,0);
There is a possibility that the device remains isolated after the un-isolation. Issue was fixed.

HW snake test

tr 131 was not working in PAM4 rate, external loopback mode. Issue was fixed.

Section 3.2.5: Compatibility changes since 6.5.14 release

LCPLL input reference clock configuration

This refers to LCPLL input reference clock configuration SoC property:

For BCM88790 stand alone or inside a mgmt Negev, the SoC property serdes_fabric_clk_freq_in is now configured according to a value read from Silab, by i2c. Supported input reference clock values are 156.25MHz and 312.5MHz.

See dnxf.soc file for details.

If the silab value is not according to the pre-defined values, 312.5MHz input clock will be set by default.

In case the user wants to avoid using this mechanism and set a specific value for the SoC property, the relevant script in dnxf.soc can be deleted and replaced by writing the SoC property in the config file.

Linkscan

When linkscan callback registered by the user is executed, the bcm_port_info_t will contain only link state information.

MAC cell shaper

The API bcm_port_rate_egress_pps_set is no longer supported.

Section 3.5: BCM88470 (Qumran-AX) Family General Availability (GA) Release

This release is the General Availability (GA) SDK version for the BCM88470 family product line. The subsequent sections describe the increment in available features compared to 6.5.13, major bug-fixes and known issues.
Before integrating the new release, review “Backward compatible important notes” section.

Section 3.5.1: Backward compatibility important notes

See Section 3.6.1: BCM88670 Family “Backward compatible important notes”

Section 3.5.2: New Features

See Section 3.6.2: BCM88670 Family “New Features” section.

Section 3.5.3: Major Bug fixes

Packet Processing

See Section 3.6.3: BCM88670 Family “Major Bug fixes” section.

Section 3.5.5: Known issues

Network Interface

Packet Processing

See Section 3.6.4: BCM88670 Family “Errata” section.

Section 3.5.6: Important Notes

None

Section 3.6: BCM88670 Family GA Release

This release is for the BCM88370/BCM88670 family product lines.
All features introduced in SDK 6.5.14 are also supported in SDK 6.5.15. The subsequent sections describe the increment in available features compared to 6.5.14, major bug-fixes and known issues.
Before integrating the new release, review “Backward compatible important notes” section.

Section 3.6.1: Backward Compatible Important Notes

Section 3.6.1.2: SW Compatibility Guidelines 6.5.X (other than 6.5.14) to 6.5.15

Before integrating the new release, review “Backward compatible important notes” sections of previous releases till 6.5.15.

For example, when upgrading from 6.5.9 to 6.5.10, review the backward compatible important notes section for SDK releases 6.5.5, 6.5.6, 6.5.7, 6.5.8, 6.5.9, 6.5.10, 6.5.12, 6.5.13,6.5.15 (this document Section 3.6.1.3).

Section 3.6.1.3: SW Compatibility Guidelines 6.5.14 to 6.5.15

Module	JIRA	Description	Affected Devices
KBP/KAPS	SDK-161225	SDK is aligned to KBPSDK 1.5.5.1 KAPS and KBP are functional with KBPSDK 1.5.5.1	All
General, L3	SDK-163191	In previous versions, the macro BCM_L3_ITF_LIF_TO_GPORT_TUNNEL won't check the LIF sub type. In the 6.5.15, this will be done, make sure a real L3-LIF type value is passed in this marco when using it.	All
Field	SDK-163363	In 6.5.14 new feature was add SDK-145728 Which allow to create Field group and indicate indicate which CE Id’s to use per qualifier. Since the newly added structure caused RPC fail, now this feature is supported with new API *bcm_field_group_config_extension_create()*	All
VXLAN IPv6	-	Application will not work natively without special soc property: custom_feature_ethernet_header_editing_size For more information, please approach AE team.	88680/88470 /88270

Section 3.6.2: New Features

Packet Processing

Module	JIRA	Description	Affected Devices
Mirror	SDK-159657	Lawful interception over IPv6 tunnel is now avaiable	BCM8867x/BCM8837X/BCM8847X/BCM8827X/BCM8868X
BFD	SDK-146129	BFD IPv6 Echo is now supported	

High Availability

Section 3.6.3: Major Bug fixes

The list below refers to major bug fixes, and does not provide a comprehensive coverage of various bug fixes on all levels.

Basic Data Path, Connectivity and Traffic Management Features

Packet Processing

Module	JIRA	Description	Affected Devices
VLAN-Translation	SDK-162540	In previous version, Ethernet editing configuration did not allow on certain packet flows to edit more than 2 VLAN tags. This is now fixed.	BCM8847X/BCM8827X/BCM8868X

MPLS	SDK-162502	PWE encapsulation: EL/ELI insertion to MPLS labels below PWE with Control Word is now functional.	BCM8867x/BCM8837X/BCM8847X/BCM8827X/BCM8868X
EVPN/MPLS	SDK-160403	When OutLIF of EVPN with wide-data or MPLS with egress protection was destroyed, other LIFs with Wide-data or protection might be influenced ...	BCM8867x/BCM8837X/BCM8847X/BCM8827X/BCM8868X

High Availability

Section 3.6.4: Errata

The list below relates to major open bugs that are not resolved.

Basic Data Path, Connectivity and Traffic Management Features

-

Packet Processing

Section 3.6.5: Important Notes

Section 3.7: BCM88680 (Jericho+) Family GA Release

This release is a sustained version for the BCM88680 family product line. The subsequent sections describe the increment in available features compared to 6.5.14, major bug-fixes and known issues.
It is extremely important to review “Backward compatible important notes” section before starting the integration of the new release.

Section 3.7.1: Backward compatible important notes

See Section 3.6.1: BCM88670 Family “Backward compatible important notes”

Section 3.7.2: New Features

Packet Processing

See Section 3.6.2: BCM88670 Family “New Features” section.

Section 3.7.3: Major Bug fixes

Packet Processing

See Section 3.6.3: BCM88670 Family “Major Bug fixes” section.

Section 3.7.4: Known issues

Traffic Management and Data Path

Packet Processing

See BCM88670, BCM88470 families known issues section.

Section 3.8: 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 for BCM88950 driver, with all major features supported.

Section 3.8.1: Important Notes

Section 3.8.2: Major Bug Fixes

Section 3.9: BCM88270 (Qumran-UX) Family GA Release

The subsequent sections describe the features validated for this release, known issues and bring-up guidelines.
Before integrating the new release, review “Backward compatible important notes” section.

Section 3.9.1: Backward Compatible Important Notes

See BCM88670-Family “Backward compatible important notes” section.

Section 3.9.2: New Features

See Section 3.5.2: BCM88470 Family “New Features” section.

Section 3.9.3: Major Bug fixes

In 6.5.15 we introduce

See Section 3.5.5: BCM88470 Family “Known issues” section.

Section 3.10: BCM88660 (ARAD+), BCM88650 (ARAD) Release

This is a sustain release of BCM88660, BCM88650 driver, with all major features supported.

Section 3.10.1: Important notes

None

Section 3.10.2: Major Bug Fixes

None

Section 3.10.3: Errata

None

Section 3.10.4: New Features

None

Section 4: General Updates

Section 4.1: Compatibility Notes

In order to support new device families, Signal DB was moved from $SDK/src/appl/diag/dcmn to $SDK/tools/sand