SDK 6.5.22

Release Notes

Table of Contents

Section 1: About This Document

Section 2: New Devices added to this release

- Section 2.1: Supported DNX Switch Devices
- Section 2.2: Preview DNX Switch Devices

Section 3: Information per Device

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

Section 3.1: DNX-Family

- Section 3.1.1: Reference Documentation
  - Section 3.1.1.1: SW/Arch documentation
  - Section 3.1.1.2: In-Package Documents
- Section 3.1.2: Supported SKUs
- Section 3.1.3: Important Notes
  - Section 3.1.3.1: Backward Compatible Important Notes
- Section 3.1.4: SDK build & load
- Section 3.1.5: IPSec and MACSEC support
- Section 3.1.6: Validated features
  - Section 3.1.6.1: Access, Basic data path, Connectivity and Traffic Management Features Validated features
  - Section 3.1.6.2: Packet-Processing Validated feature
  - Section 3.1.6.3: ACL Validated features
  - Section 3.1.6.4: High Availability Validated features
- Section 3.1.8: Errata
- IMPORTANT: The following KM article summarizes the location of all SW Errata KM articles: “DNX Software Errata SDK MAIN tracker“.
- Section 3.1.9: New Features & Bug-fixes

Section 3.2: DNXF-Family (BCM88790-Family)

Section 3.2.1: Supported SKUs

Section 3.2.2: Important Notes

Section 3.2.2.1: Backward Compatible Important Notes

Section 3.2.3: Validated Features

BCM88790 is GA level.

Section 3.2.4: Errata

Section 3.2.5: Bug fixes & New features

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

Section 3.3.1: Important Notes

Section 3.3.2: Backward Compatible Important Notes

SW Compatibility Guidelines 6.5.21 to 6.5.22 (and 6.5.22-EA4 to 6.5.22)

Section 3.3.4: New Features and Bug fixes

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

Section 3.5: Bug fixes since 6.5.21 for DNX/DNXF/DPP/DFE Families

Section 4: Compatibility

Section 4.1: SDK and PCIe FW Compatibility

Section 4.2: SDK and KBPSDK lib Compatibility

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 6.5.22.

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-6.5.22-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 6.5.21 and 6.5.22-EA4, 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)**

Section 3.1.1: Important Notes

Before integrating the new release, review this section thoroughly.

JIRA	Module	Description	Affected Devices
SDK-247716	COSQ Ref. app	Reference application adjustments: When allocating voq connectors in the DNX reference application we used the local port id of the destination port as part of the voq connector id calculation. This changes the calculation to be based on the destination physical attributes (such as pp_dsp and modid) instead of local port.	88690_B1, 88800_A0, 88480_A0, 88480_B0

Go over the Backward Compatible Important Notes as well.

Section 3.1.3.1: Backward Compatible Important Notes

SW Compatibility Guidelines 6.5.21 to 6.5.22 (and 6.5.22-EA4 to 6.5.22)

Please go over the list carefully.

Note: This document is written with the assumption that upgrade is done from 6.5.21/6.5.22-EA4 to 6.5.22. In case upgrade is done from older releases, users must first go over previous release notes.

JIRA	Module	Description	Affected Devices	From which SDK version backward compatible breakage is relevant 6.5.21 or 6.5.22-EA4 or both (i.e. 6.5.21 and 6.5.22-EA4)
SDK-245329	MPLS	Speculative parsing is done after MPLS labels at the first Parser according to the first nibble after the BOS label. SDK changed its behavior: All nibbles except 4,6 will now speculate by default ETH header instead of Unknown layer (Unknown layer was the default in previous releases). In order to get the previous behavior call: bcm_switch_control_indexed_set control_type bcmSwitchMplsSpeculativeNibbleMap index: BCM_SWITCH_DEFAULT_NIBBLE_INDEX value bcmSwitchMplsNextProtocolNone.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-249941	GTP-LIF	GTP Flow application names changed as follows: GTP_TERMINATION to GTP_TERMINATOR GTP_CLASSIFICATION to GTP_MATCH This affects bcm_flow_handle_get(), which should be called with the new names	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-249648	VXLAN	When calling to VLAN-Port Virtual-AC (bcm_vlan_port_create) ESEM entry with flags = BCM_VLAN_PORT_CREATE_EGRESS_ONLY | BCM_VLAN_PORT_VLAN_TRANSLATION and criteria = BCM_VLAN_PORT_MATCH_NAMESPACE_VSI or BCM_VLAN_PORT_MATCH_PORT_VLAN or BCM_VLAN_PORT_MATCH_NAMESPACE_PORT. A verification is added that if the entry is already created then the call is with	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4

		BCM_VLAN_PORT_REPLACE flag. This may change a sequence of calls in the case of VXLAN application. In case one created a VXLAN entry and then created a Virtual-AC with the same key, now the API of bcm_vlan_port_create will require REPLACE flag while in previous releases it does not require.		
SDK-248081	Field	Field Applications for ITMH (and ITMH-PPH) injected traffic need to be changed. ASE and TSH layers of ITMH injected packets are now placed on the same layer (1) and not in two separated layers (1,2) in the Field qualifiers layers. Application reference was changed accordingly to cover this change. It is a must to change field injected applications in order for ITMH and ITMH_PPH application to work correctly. The reason for the change is that PPH packets with unsupported extensions (extensions that aren't rebuilt) are required to hit ITMH context (instead of ITMH_PPH context) and so PPH won't be parsed, only ITMH thus preserving the extension. Reminder, PPH re-build supported extensions are: 1. FHEI Trap of size 5B. 2. FHEI Vlan of size 5B. Before this change, unsupported extensions were not allowed to be injected, otherwise Field Application ref. would corrupt the packet.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-247913	Hashing-Configured	IPv6 headers load-balancing changed due to the move of some of the IPv6 headers parsing to the second stage parser: 1. In order to simulate correctly IPv6 headers that come with extension headers or with no additional headers (additional headers can be either Tunnels like GRE or IPv4/IPv6/MPLS/ETH headers), it is required to provide to the LB offline simulator a new flag that indicates that the header is reparsed at 2nd stage (LB_SIM_HEADER_REPARSE). The flag is provided in a per packet header. This is different than previous releases where the flag wasn't required. In case the flag won't be provided, on the cases mentioned above, a mismatch between the device and the simulator will appear. 2. Due to the above, there can be cases where outer IPv6 headers load-balancing is done in the second stage parser where the configuration of bcmSwitchHashIP6InnerField takes into account. It is required to set the same configuration value for bcmSwitchHashIP6OuterField, bcmSwitchHashIP6InnerField.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4

SDK-247377	VXLAN	ESEM Namespace-VSI entry can be created by the following two APIs: bcm_vxlan_network_domain_config_add bcm_vlan_port_create In case the ESEM was created by the first API, on calling the second API, it will just update relevant AC fields (i.e out_lif_profile). If both above API were called, when the user wants to delete the ESEM Namespace-VSI entry, it needs to call of the delete APIs: Call first bcm_vlan_port_destroy. Second, call bcm_vxlan_network_domain_config_remove. Otherwise, error will be generated (on some cases in previous releases API didn't return error but some fields information might be lost).	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-246103	RFC-6374	RFC-6374 MPLS-DM can work in both one-way mode and two way mode. When using API bcm_oam_delay_get() to get delay, the fields in structure bcm_oam_delay_t has the different meaning for one-way mode and two-way mode, the details are, In one-way mode, field delay means far-end one-way delay, field delay_min and delay_max are for far-end two, while field delay_near_end means near-end one-way delay, field delay_min_near_end and delay_max_near_end are for near-end too. In two-way mode field delay means two-way delay, field delay_min and delay_max are for two-way delay too. This JIRA corrects the statistics that are returned in one way mode. Fields meaning changed from the previous releases.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-246088	VLAN	When API bcm_switch_control_set bcmSwitchIngressVlanEditClassNull is enabled, then bcm_vlan_translate_action_id_create will fail on the last action_id. SDK added validation for this mis-configuration. This is different from previous release, that vlan translate API ignored the configuration of bcmSwitchIngressVlanEditClassNull and override it.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.22-EA4
SDK-246077	L2	The default age refresh mode for the MACT was changed from hitting either the source address or the destination address into the source address only. For getting back to the previous default the following should be called bcm_switch_control_setunit, bcmSwitchMactAgeRefreshMode, bcmSwitchAgeRefreshModeSrcOrDest)	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-245352	Trunk	Several PP-port configurations were reserved/incorrect-set after trunk was destroyed (bcm_trunk_destroy) which can lead ports that are removed from trunk to have "junk" configurations. This is now changed and all information is cleared. Following fields were configurated incorrectly: Port Default VID was set to	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4

		BCM_VLAN_DEFAULT and not to 0 as it should be, Source-system-port and Ingress TPID class ID wasn't cleaned. Note: When trunk/port is created or when port is removed from trunk, it is expected to configure all port configurations and not to be depended on "junk" configurations. That is it expected to configure all the basic properties such as port default VID, VLAN-Domain and so on...		
SDK-245224	IP-Tunnel	In order to prevent IPvX tunnel termination process when the layer above it is one of the following: ICMP, TCP, IGMP, UDP, Gre keep alive or Unknown, those packets will skip the tunnel termination flow at all. In previous releases, tunnel termination was disabled by the routing enablers mechanism but LIF/VPN fields were updated even when it was hit causing packets to be with wrong metadata of InLIF and VPN. The functionality is global mode in the device and enabled by default to disable all the mentioned layers above. In order to work with the previous (routing enablers) solution, call API bcm_switch_control_set with type bcmSwitchTunnelRouteDisable and argument BCM_SWITCH_TUNNEL_ROUTE_DISABLE_* should be used.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-244466	Field	bcmFieldQualifyL2Format is not functioning correctly in this release. It will be supported differently from next release. This is due to the change in Ethernet header parsing: LLC/SNAP headers are now parsed as part of the ETH header. The Eth Qualifier holds indication about the type of the frame, it has 2 bits indicating: 0x0 - ETH2 (Ethertype is above 1500); 0x1 - LLC frame 0x2 - SNAP frame 0x3 - other type In previous releases, only ETH2 or ETH1 indication 1 bit was available. Note: SDK-252268 Errata is open. Due to the change, bcmFieldQualifyL2Format is not correct anymore and needs to be changed to be 2b. A workaround is available in SDK-252268.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-244328	Field	The behavior of system-port gport in case of Trunk changed. In case the system-port gport is Trunk, then the API will convert it to LAG-group and LAG-member-ID encoded in SSPA when qualified by bcmFieldQualifySrcPort. In case no convert is required, please use bcmFieldQualifySrcPortRaw.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4

SDK-242540	Fied	A new FTMH_MC field application to demonstrate a 2-pass MC solution with FTMH processing is introduced. When recycling the packets at the 2nd-pass ingress processing, using PMF application, FTMH fields will be updated from the first pass, for example: LB_KEY , TC and more. Note1: The application reference is enabled by default and takes PMF resources. It is possible to disable the application reference in case it is not required by soc property: appl_enable_field_ftmh_mc=0 Note2: In order to hit the context on the 2nd-pass, it is required to define a recycle port of header type STACKING (unrelated to Stacking application but it represents header type is FTMH) and set its field property bcmPortClassFieldIngressPMF1PacketProcessingPortCs to be 7.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-234612	MPLS	From this release, if the EEDB PHP entry is used to pop label (action BCM_MPLS_EGRESS_ACTION_PHP), the input qos_map_id should be encoded as PHP QoS map id (and not Remark QOS map ID as it was in previous releases). For other cases (e.g. implicit NULL label), use Remark QOS map ID.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-212797	OAM	From this release, 32 bits TOD second is inserted between system header and network header in trapped OAM DM packets. The 32 bits TOD second should account for wrap-around according to the low 34 bits TOD in FTMH ASE header. Customer should handle this. The behavior change will also affect trapped TWAMP packet and trapped RFC6374 DM packet.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4
SDK-167009	COE	bcm_port_extender_mapping_info_get() for COE, changed his logic for type bcmPortExtenderMappingTypePortVlan. Instead of calling with flags = 0, it is expected that flags param will be similar to set API (i.e. flags BCM_PORT_EXTENDER_MAPPING_INGRESS, BCM_PORT_EXTENDER_MAPPING_EGRESS need to be set)	88480_A0, 88480_B0, 88800_A0	6.5.22-EA4
SDK-159432	Stat	1. In previous releases, it was possible to be in mis-configuration stage where a LIF object was able to be configured with stat (bcm_gport_stat_set) when the LIF format includes stat fields, even if the LIF wasn't created with the appropriate stat flag (for example, BCM_VLAN_PORT_STAT_INGRESS_ENABLE in case of a VLAN-Port LIF). This was applicable for all relevant LIFs except for MPLS LIFs of all sorts, where stat configuration was only available if the appropriate stat flag was set when calling the LIF creation API. In the current release, SDK now resolves the inconsistent. The configuration of stat parameters (bcm_gport_stat_set) fails for LIFs that weren't created with the appropriate stat flag. The only exceptions are	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4

		SRv6 Tunnel Out-LIFs and Virtual AC LIFs, where the previous logic will be retained. 2. It is not possible anymore to replace ARP+AC without statistics (bcm_l3_egress_create with flag BCM_L3_FLAGS2_VLAN_TRANSLATION) to ARP+AC with statistics, since some of the fields are lost as part of the replacement process. In case this is required, it can be done only by destroy the object and re-create it with statistics.		
SDK-235564	SRv6	SRv6 Layer is now collapsed to IPv6 layer in case of SRH.SL=0. Following changes done due to that: 1. Packet structure such as IPv4(or any payload)oSRv6(SL=0)oIPv6oETH used to create in previous releases layer protocol stack of: 0:ETH, 1:IPv6, 2:SRv6_Beyond, 3: IPv4 now it will create: 0:ETH, 1:IPv6, 2:IPv4 2. The indication for SRv6 with SL=0 is set on the IPv6 qualifier. The IPv6 first additional header enumerator now has two new values: IPv6_SRv6, and IPv6_SRv6_SL_0 Note, in case the first additional header is IPv6_SRv6_SL_0, the rest of the IPv6 qualifier bits 7-13 are changed.It does require changes on the SRv6 Field-Processor applications. 3. The application reference code has updated accordingly, it now Identifies the SRv6 USP/USD flows with the updated IPv6 qualifier as a preselector-qualifier instead of the SRv6_Beyond layer protocol and its qualifier. 4. USP flow is changed and now must have two In-LIFs involved. First LIF must be created (bcm_tunnel_terminator_create) with tunnelType bcmTunnelTypeSR6, and the second must be in lower priority, and with tunnelType bcmTunnelTypeIpAnyIn6, to match the packet after the SRv6 extension is terminated. This is considered as a bug, and has a SW Errata under SDK-246424. Flow will be changed in next release. 5. For SRv6 Endpoint flow (SL!=0), the Layer above SRv6_Beyond is not set anymore on the layer protocols array (in previous release it was set). Note: SRv6.Next-header (field in packet) is available in PMF qualifier instead.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21
SDK-242856	RCH	The API bcm_l2_egress_create had an input parameter dest_encap_id, and an input flag BCM_L2_EGRESS_DEST_ENCAP_ID, both were not used in the API configuration but the customer could provide them.	88480_A0, 88480_B0, 88690_B1,	6.5.21

		Now, SDK returns error when they are used. The RCH.Outlif field is always getting the value of Outlif_0 of the current ETPP pass (which builds the RCH) and not according to API configuration. For SRv6 Extended T.Encap flow: A new configuration is now added to indicate if to update the OutLIF-Stack in the second pass of the ingress-pipe, by setting flag BCM_L2_EGRESS_EXTENDED_COPY_DEST_ENCAP.	88800_A0	
SDK-242675	OAMP	LMM/SLM packet can not be transmitted from OAMP to NIF (Network InterFace) because its length is too short. LMM/SLM packets will now come with zero padding when sending from OAMP.	88480_B0, 88690_B1, 88800_A0	6.5.21
SDK-242668	RCH	In the previous release, RCH processing of extended encapsulation uncollapse (bcmPortControlRecycleAppExtendedEncapUncollapse) was handled incorrectly at the 2nd pass. Instead of doing regular processing of PP in the 2nd pass, the application used a dedicated 2ND pass DBs (as used in the drop and continue application). The behavior is incorrect and now fixed. The processing is now doing the normal termination processing.	88480_B0, 88690_B1, 88800_A0	6.5.21
SDK-241829	Multicast	APIs unsupported flag are now enforced. The following flags are unused and deprecated. With this fix, multicast APIs will return an error if someone trying to use them: BCM_MULTICAST_TYPE_L2 BCM_MULTICAST_TYPE_L3 BCM_MULTICAST_TYPE_VPLS BCM_MULTICAST_TYPE_SUBPORT BCM_MULTICAST_TYPE_MIM BCM_MULTICAST_TYPE_WLAN BCM_MULTICAST_TYPE_VLAN BCM_MULTICAST_TYPE_TRILL BCM_MULTICAST_TYPE_NIV BCM_MULTICAST_TYPE_EGRESS_OBJECT BCM_MULTICAST_TYPE_L2GRE BCM_MULTICAST_TYPE_VXLAN BCM_MULTICAST_TYPE_EXTENDER BCM_MULTICAST_TYPE_MAC BCM_MULTICAST_TYPE_FLOW BCM_MULTICAST_TYPE_MASK BCM_MULTICAST_DISABLE_SRC_KNOCKOUT BCM_MULTICAST_INGRESS_MMC_BUFFERS	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21
SDK-240924	L2	In shell diagnostic commands "l2 add" and "l2 clear", an input argument	88480_A0,	6.5.21

		"tgid" may be used. "tgid" should be encoded according to BCM_TRUNK_ID_SET() and not only the group-ID. This is now enforced and the command will return an error in case tgid is not properly encoded. Error information like “Input parameter tgid is not properly encoded. Please use BCM_TRUNK_ID_SET() to encode” will be invoked.	88480_B0, 88690_B1, 88800_A0	
SDK-240116	PMF	Remove support from SDK for 'field key info' diagnostic. Same diagnostic can be seen using "field context info".	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21
SDK-239082	PMF-Diag	Removed support from SDK for PMF-diagnostics 'field port'. Instead, Data can be found in the several DBAL tables: for TM ports: DBAL tables FIELD_PMF_PTC_INFO_IPMF1, FIELD_PMF_PTC_INFO_IPMF3, FIELD_PMF_PTC_INFO_IFWD, FIELD_PMF_OUT_TM_PORT_INFO_EPMF for PP ports: DBAL tables: FIELD_PMF_PP_PORT_INFO_IPMF1, FIELD_PMF_PP_PORT_INFO_IPMF3 and EGRESS_PP_PORT field FIELD_PMF_PP_PORT_INFO_IFWD for LIF ports; InLIF DB (e.g. IN_AC_INFO_DB, IN_LIF_FORMAT_PWE, IN_LIF_FORMAT_EVPN, IN_LIF_FORMAT_LSP, IN_LIF_IPvX_TUNNELS, INNER_ETH_VLAN_EDIT_CLASSIFICATION, INNER_ETH_VLAN_EDIT_CLASSIFICATION_VLAN_EDIT)	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21
SDK-239080	PMF-Diag	Remove support from SDK for 'field system' and 'field key last' diagnostics. "Field system" shows the context selection rules, same can be achieved by using "field context list" diagnostic "field key last" shows the key that was constructed for last packet, same can be achieved using "field last info" diagnostic	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21
SDK-234248	VLAN-Port	BCM_VLAN_PORT_VLAN_TRANSLATION_TWO_VLAN_TAGS is obsolete and removed because now all AC formats support two VLAN TAGs. In case application used BCM_VLAN_PORT_VLAN_TRANSLATION_TWO_VLAN_TAGS , do same sequence just without BCM_VLAN_PORT_VLAN_TRANSLATION_TWO_VLAN_TAGS flag.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21
SDK-227881	WIDE-DATA	API bcm_field_presel_set, qual_type=bcmFieldQualifyAcInLifWideData - Extend the in-lif wide data size that is used for IPMF1 context selection from 1 (msb) bit to 8 (msb) bits.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21, 6.5.22-EA4

		Please note, that the mask value meaning is changed. mask=0x1, used to indicate the msb bit of the in-lif wide data. After the change, mask=0x80 indicate on the msb bit. Please note2: In 6.5.22 (compared to 6.5.22-EA4) the 8bits are located differently. In 6.5.22-EA4 8bits located in 7MSB and 1LSB of the wide-data. In 6.5.22 8bits are now located in the 8MSB.		
SDK-210913	TDM	The default context ID for all ILKN channels is 0. From this release, context_id 0 now is invalid (traffic is drop) instead of being a valid one. When new ports are set, by default they will have invalid context_id and will not forward traffic until a valid context will be assigned to them. Set a valid TDM context ID by calling API bcm_tdm_ingress_context_create with context_id > 0.	88480_A0, 88480_B0, 88800_A0	6.5.21
SDK-193159	VXLAN	VXLAN L2VPN QOS now supports egress native PCP&DEI remark profile of IP tunnel. In previous releases, QOS was determined by remark profile 0 only.	88690_B1	6.5.21
SDK-234338	SRv6	SRv6 End point PSP is now supported per packet, and not only as a global mode. The classification of a packet as a PSP packet can be done per next DIP. Due to that, significant change in Application Reference was done in iPMF. See SRv6 UM and appl_ref changes for more information	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21
-	DBAL	DBAL Field Enums types are changed from this release: 1. There is no more defines for max value of enum (DBAL_NOF_ENUM_*). Instead use dbal_fields_max_value_get API. 2. dbal_enum_value_result_type_* do not exist anymore. Instead use dbal_result_type_t. 3. DBAL_NOF_RESULT_TYPE_* do not exist anymore. Instead use API dbal_tables_table_nof_res_type_get with the DBAL table name.	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21
SDK-218577	MPLS	The explicit null Label and other reserved MPLS labels can be terminated per port (VLAN domain) as the general labels by the following configurations: 1. call bcm_switch_control_indexed_set() with index "bcmSwitchMplsSpecialLabelAutoTerminateDisable" to disable auto termination for a particulate label; 2. call bcm_mpls_tunnel_switch() to add the label termination entry. The input parameter "port" indicates the VLAN domain. Note, in this case, the special label should be BoS. In addition, all the	88480_A0, 88480_B0, 88690_B1, 88800_A0	6.5.21

		reserved MPLS labels (0~15) cannot be configured as general labels. In previous release, the functionality was global, now it is per Incoming-port (VLAN-Domain).		
SDK-238679	L2	From this release, MACT/FID limit is not enforced for static entries insertion.	88690_B1	6.5.21

Section 3.1.4: 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
```

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 3.2: DNXF-Family (BCM88790-Family)

Section 3.2.1: Important Notes

Before integrating the new release, review this section thoroughly.

None

Section 3.2.1.1: Backward Compatible Important Notes

SW Compatibility Guidelines 6.5.21 to 6.5.22 (and 6.5.22-EA4 to 6.5.22)

Please go over the list carefully.

Note: This document is written with the assumption that upgrade is done from 6.5.21/6.5.22-EA4 to 6.5.22. In case upgrade is done from older releases, users must first go over previous release notes.

JIRA	Module	Description	Affected Devices	From which SDK version backward compatible breakage is relevant 6.5.21 or 6.5.22-EA4 or both (i.e. 6.5.21 and 6.5.22-EA4)
SDK-238031	Access	Remove "access" shell command	88790	6.5.21
SDK-250092	Init	In previous releases, in case init fails the device is detached automatically. From this release, in order to align with DNX devices behavior, same sequence apply. If the DNXF device fails on init it won't go into automatic detach procedure. Instead, customer will need to call detach after failure to remove the unit.	88790	6.5.21, 6.5.22-EA4

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 6.5.21, 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.
None

Section 3.3.2: Backward Compatible Important Notes

SW Compatibility Guidelines 6.5.21 to 6.5.22 (and 6.5.22-EA4 to 6.5.22)

Note: This document is written with the assumption that upgrade is done between 6.5.21/6.5.22-EA4 to 6.5.22. In case upgrade is done from earlier releases to 6.5.21, it must first go over previous SDK release notes.

None

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.

Section 4: Compatibility

Section 4.1: SDK and PCIe FW Compatibility

Below table shows the firmware version compatible with the SDK release.

Switch SDK Release	*PCIe FW Release*
6.5.22	2.5.4

Section 4.2: SDK and KBPSDK lib Compatibility

Below table shows the KBPSDK lib compatible with the SDK release.

Switch SDK Release	*KBPSDK lib*
6.5.22	1.5.15