

# **KBP SDK 1.4.10**

## **Release Notes**

## Revision History

Revision	Date	Change Description
KBP-SDK-1.4.10	01/15/17	BCM15K related bug fixes and enhancements
KBP-SDK-1.4.9	10/26/16	Serdes Enhancements
KBP-SDK-1.4.8	09/30/16	Bug fixes enhancements and performance improvements for Optimus Prime, silicon testing.
KBP-SDK-1.4.7	07/29/16	Update rate improvements for Jericho + NLA8865x. Bug fixes and performance improvements for Optimus Prime, silicon testing.
KBP-SDK-1.4.6	06/12/16	Update rate improvements for Jericho + NLA8865x. Bug fixes and performance improvements for Optimus Prime, silicon testing.
KBP-SDK-1.4.5	05/01/16	Bug fix for Crash Recovery
KBP-SDK-1.4.4	03/09/16	Multiple features for Jericho + NLA8865x. Bug fixes and performance improvements for Optimus Prime
KBP-SDK-1.4.3	01/11/16	Optimus Prime full feature support and bug fixes. Increased IPv4 and IPv6 scale for ARAD/Jericho customers using NLA8865x (NLA12K)
KBP-SDK-1.4.2	07/27/15	Optimus Prime full feature support and bug fixes
KBP-SDK-1.4.1	03/16/15	Optimus Prime new feature support and bug fixes
KBP-SDK-1.4.0	12/19/14	Preliminary support for Optimus Prime
KBP-SDK-1.3.5	02/23/15	NetRoute coherency bug fix Handling for algorithmic errors reported by device

---

## **Release Notes for KBP SDK Rev. 1.4.10**

### **Description:**

KBP SDK 1.4.10 contains the following enhancements and bug fixes

New APIs have been added to read the BCM15K die voltage and temperature

- *kbp\_status kbp\_device\_read\_die\_voltage(struct kbp\_device \*device, int32\_t \*voltage);*
- *kbp\_status kbp\_device\_read\_die\_temperature(struct kbp\_device \*device, int32\_t \*temperature);*
- [KBPSDK-2124] Generic and Selective broadcast support for BCM15K
  - When destroying device handles during teardown, only the main device handle should be destroyed
- [KBPSDK-2137] – BCM15K kbp\_diags has no option to connect to other KBP devices on line card
- [KBPSDK-2113] – BCM15K 640b + range cases with search mis-match
- [KBPSDK-2095] – Range expansion heuristics tuning
- [KBPSDK-2081] – KBP Driver: Need a way to specify PCI bus mapping to KBP selection
- [KBPSDK-1753] – Debug Feature Request - Result Register Read

The packaged version 1.7 of the device driver is the only compatible version with this release. The PCIE device driver must be upgraded along with the KBP SDK for BCM15K.

KBP SDK 1.4.9 resolves symbol conflict between the Broadcom Jericho SDK 5.5.6 and KBP SDK 1.4.8 which use the same Serdes IP code. In addition the following Serdes improvements have been incorporated in this version of the SDK.

- [KBPSDK-2088] - Falcon16 API names clash with version of IP code in DNX SDK
- [KBPSDK-2089] - Temperature monitoring not enabled for serdes
- [KBPSDK-2090] - Unused lanes need to be powered down
- [KBPSDK-2091] - Upgrade Falcon16 microcode to latest version D102\_02

KBP SDK 1.4.9 has been tested for all supported features on Optimus Prime (BCM15K). All features of NLA12K and Jericho internal forwarding solution are GA quality and fully supported.

The SDK contains the following enhancements/fixes delivered as part of KBP SDK 1.4.8

- Warmboot support for NetACL and NetRoute databases with and without associated data
- Aging and Hit Bits
- [KBPSDK-2038] O3S: Need XPT per NP / SMT for 2-to-1 application
- [KBPSDK-2072] BCM15K: Interface initialization enhancements for polarity inversion, asymmetric lane enables and lane swapping

The SDK contains the following bug fixes

- [KBPSDK-1981] BCM15K: PCIE xpt requires byte swapping on big endian systems
- [KBPSDK-2062] BCM15K: S\_RST and C\_RST callbacks have incorrect
- [KBPSDK-2017] BCM15K: Quad8 links not brought up correctly

The enhancement KBPSDK-2072 and KBPSDK-2038 introduced multiple interface related features. The device configuration structures defined in `include/init.h` passed to `kbp_device_interface_init` and/or `kbp_device_init` has been enhanced to accept the following fields:

1. The start lane and num\_lanes field can be specified for upto four ports. For four port the following flag `KBP_DEVICE_QUAD_PORT` should be passed in
2. *uint32\_t reverse\_lanes*: The field is interpreted as a bitmap. Each bit specifies whether the lanes of port are reversed. Bit0 if set to one will reverse all the lanes of port0, etc.
3. *int32\_t (\*get\_rx\_tx\_lanes\_bmp)(uint64\_t \*rx\_lanes\_bmp, uint64\_t \*tx\_lanes\_bmp)*: The callback function is invoked during init if provided. The user is expected to populate the RX and TX lanes bitmap if they wish to configure asymmetric number of lanes for RX and TX. If this feature is used the num\_lanes field in the port map should be provided as maximum value of the two and the callback can be used to shut down specific lanes. The field can be set to NULL if not used
4. *int32\_t (\*get\_polarity\_invert\_lanes\_bmp)(uint64\_t \*lanes\_bmp)*: The following option callback pointer can be provided by the user to invert the polarity of the lanes. Each bit in the bitmap if set to one will invert the polarity of that physical lane number. If this feature is not used the callback function should be NULL.
5. *uint32\_t reset\_only*: This bitmap is one hot encoding for each port of the device. If the bit is set all the lanes belonging to that port are configured, however, RX is not enabled. The caller can lazily enable the RX at a later point. This feature is useful if the ports need to be enabled lazily.

---

Internally discovered bugs through testing on Optimus Prime silicon have been resolved. Please refer to the Optimus Prime Silicon Errata sheet for additional details on the silicon status.

This release also supports crash recovery for NLA12K and Jericho internal forwarding solution and contains all the issues resolved as of KBP SDK 1.3.13 and KBP SDK 1.4.7. All features of NLA12K and Jericho internal forwarding solution are GA quality and fully supported.

This SDK includes –

- APIs to initialize the device, create instructions and keys, and create and manage databases.
- Multiple Reference applications that showcase use of Optimus Prime and NLA12K.
- A C-Model for NLA12K and Optimus Prime. The model is transaction, not cycle accurate.
- The SDK can be used in Blackhole mode without using the C-Model or real device. Pass the transport layer handle as null in `kbp_device_init()`.
- Dynamic resource management of databases for NLA12K and Optimus Prime
- Support for managing multiple KBP's on one line card
- Warmboot and crash recovery

#### **Optimus Prime:**

- Any existing application coded for NLA12K is expected to work with Optimus Prime C-Model
- All aspects of configuring and managing the device
- NetACL, including associated data.
- NetRoute with associated data
- Support for Massively parallel ACLs, Power controlled ACLs with associated data.
- Single cycle lookup of multi access LPM databases.
- Resolution between lookup results of databases. Typical use is for resolving the result of a match between a public and private forwarding database.
- Hit bits and Aging
- Warmboot for all database types

#### **NLA12K:**

- GA Support for NetRoute, Massively parallel ACLs, Power-controlled ACLs, and Associated data for ACLs, SMT and Cascade.
- Warmboot and crash recovery support for all database types
- Parity error correction (In field soft repair) for entries in DBA and correction of algorithmic errors reported by device

---

**Devices Supported:**

- NL12000, NLA12000, NLA88650, ERC120 , SD5452A
- SDK has been tested with Rev A2 and Rev B0 and B1 Silicon
- BCM52311 SDK has been verified against A0 and A1 silicon

**Related documents:** (available at <https://support.broadcom.com>) :

- Device Datasheets: NLA12000-DS116-R-KBP, NLA88650-DS107-R-KBP, 41CSERC120-DS115-R-KBP , 48CSSD5452A-DS115-R-KBP, 1CSCSCO-DS116-RDS-KBP 15000-DS100-R, 52311-DS101-R
- KBP SDK Reference Manual, KBP-SDK-SWUM103-R or newer

**Known Limitations / Issues:**

The following features of the Optimus Prime chip are currently not supported

- Capacity and update rates are not optimal for NetACL databases
- The KBP SDK cannot recover if a fatal error occurs when transporting messages to the KBP

**Optimus Prime C-Model Limitations:**

The C-Model is a transaction accurate, not cycle accurate model. Model does not support-

- Network / interface specific functionality such as ILA protocol, Network Byte Order, packet errors.
- ECC, Parity scan / protection, Parity Errors.
- MDIO, JTAG Boundary Tag Registers
- Response to Error Detection, Test Pattern Generator / Checker
- Multiple instances of C-Model may not be instantiated at the same time

**NLA12K C-Model Limitations:**

The C-Model is a transaction accurate, not cycle accurate model. Model does not support-

- Network / interface specific functionality such as ILA protocol, Network Byte Order, packet errors.
- ECC, Parity scan / protection, Parity Errors.
- MDIO, Cascade of processors, JTAG Boundary Tag Registers
- Upper limit on total Associated Data in responses
- Low Power Mode (see Dynamic Power Control section in Datasheet)
- Response to Error Detection, Test Pattern Generator / Checker