



RELEASE NOTES

JN516x ZigBee 3.0 SDK

JN-SW-4170

Build 1840

NXP Semiconductors

For the contact details of your local NXP office or distributor, refer to:

www.nxp.com

CONTENTS

RELEASE SUMMARY (v1840)	3
1. Software Components	3
2. Supported Hardware Products	3
3. Installation	3
4. Release Details	5
4.1 New Features and Changes	5
4.2 Bug Fixes	6
4.3 Modifications Required	9
4.3.1 Porting to R22 stack:	11
5. OTA Image Allocation	15
6. Application Porting	17
7. Related Documentation	17
RELEASE HISTORY (v1518)	18
8. Release Details	18
8.1 New Features and Changes	18
8.2 Bug Fixes	19
RELEASE HISTORY (v1396)	20
9. Release Details	20
9.1 New Features and Changes	20
9.2 Bug Fixes	20
9.3 Modifications Required	20
9.4 Known Issues	20
RELEASE HISTORY (v1385)	21
10. Release Details	21
10.1 New Features and Changes	21
10.2 Bug Fixes	21
10.3 Modifications Required	21
10.4 Known Issues	21

RELEASE SUMMARY (v1840)

The JN516x ZigBee 3.0 Software Developer's Kit (JN-SW-4170) contains software resources needed to develop ZigBee 3.0 applications for the NXP JN5169, JN5168 and JN5164 wireless microcontrollers. This SDK must be installed on top of the 'BeyondStudio for NXP' toolchain (JN-SW-4141), which is available via the [Wireless Connectivity](#) area of the NXP web site.

1. Software Components

This release of the JN-SW-4170 software includes the components listed in the SW_Contents_Register_JN-SW-4170.txt

2. Supported Hardware Products

This software release supports the following hardware products:

Chips	Modules	Evaluation Kits
JN5169-001	JN5169-001-M00 JN5169-001-M03 JN5169-001-M06	JN516x-EK004
JN5168-001	JN5168-001-M00 JN5168-001-M03 JN5168-001-M05 JN5168-001-M06	JN516x-EK001
JN5164-001		

3. Installation

This SDK (JN-SW-4170) must be installed on top of the 'BeyondStudio for NXP' toolchain (JN-SW-4141), available via the [Wireless Connectivity](#) area of the NXP web site.

Therefore, before installing JN-SW-4170, you must install the toolchain from:

JN-SW-4141 Beyond Studio for NXP v1308.exe

You can then install the JN516x ZigBee 3.0 SDK from:

JN-SW-4170 Zigbee 3.0 v1840.exe

For full installation instructions, refer to the *BeyondStudio for NXP Installation and User Guide (JN-UG-3098)*, available from the [Wireless Connectivity](#) area of the NXP web site. This manual also describes how to install the ZPS Configuration Editor plug-ins for Eclipse (BeyondStudio for NXP), which you must install after the SDK.

Note that the BeyondStudio for NXP toolchain includes a built-in Flash programmer that can be used to program JN516x internal Flash memory from BeyondStudio.

Alternatively, the JN51xx Production Flash Programmer (JN-SW-4107) command-line tool can be used to program JN516x internal or external Flash memory. This tool is

available from the [Wireless Connectivity](#) area of the NXP web site and is described in the *JN51xx Production Flash Programmer User Guide (JN-UG-3099)*.

4. Release Details

4.1 New Features and Changes

The following new features are included in this release:

Internal ID	Description
lpsw8035	Support for install codes to be read from OTP autonomously from the stack.
lpsw8430	Allow the application to disable route discovery when sending data packets

4.2 Bug Fixes

The following issues have been fixed in this release:

JN516x ZigBee 3.0 SDK
Release Notes

Internal ID	Description
MCUZIGBEE-1609	GPP Functionality Attribute doesn't have the right bitmap
artf571504	NXP ZCL code has not implemented transitiontime parameter as part of Recall Scene command
MCUZIGBEE-1610	GP frame counter is wrong
MCUZIGBEE-1611	GP doesn't handle reserved and manufacturer specific commands
MCUZIGBEE-1612	Zigbee stack doesn't default back to not using install codes
artf571502	[ZCL] ZCL certification failing due to OO-TC-05S as NXP ZCL code does not handle special case where minimum reporting interval is set to FFFF and maximum reporting interval is set to 0
artf571504	[ZCL] ZB 3.0 certification S-TC-04S failing as NXP ZCL code has not implemented transitiontime parameter as part of Recall Scene command which has been introduced in ZCLr7
MCUZIGBEE-1579	Wrong definition of various macros for BDB Commissioning Mode bit field in bdb_api.h
artf571087	Simultaneous Joiner optimise timings
artf566994	Leave indication should check for device type before sending an APSME update out and leave notification should check destination address.
artf566995	Run time fragmentation APDU allocation is done in the non-default cluster implementation
artf559362	Npdu leak in multimap if sending to a disabled interface
artf555858	ZPS_vNMPurgeEntry function can cause an invalid clear of MAC address
artf554519	Insecure rejoin goes to old coordinator MAC address, even though PAN has changed
artf553158	Sub Gig Channel change fails.
artf553167	The stack clears the link key on synchronous rejoin failure
artf552149	End Device join results in the Coordinator sending route request to a magic address of 0x00CF
artf550364	R22 Coordinator ignores association request
artf552096	Critical sections unsafe when using high priority interrupts
artf552095	MultiMac support needs to handle HW not present
artf552089	Purge Join resource on authentication timer
artf551817	Suppress Transport key and authentication after Orphan Notification/Re-alignment
artf547808	Applications built using OTA don't successfully validate the image and switch to the new image.
artf545795	Rx On When Idle End device address conflict resolution parent to send leave
artf545793	The API to transmit a transport key message sends the key only APS encrypted

artf545516	Reset make neighbour table and MAC table misaligned
artf545515	Leave sent from a parent whose address conflicts with the local node
artf545514	User defined default key is not persisted
lpsw8712	The APDU is not getting freed in BDB source code if ZPS function returns non-success value
lpsw8724	Route doesn't recover from a many to one route failure
lpsw8090	The stack set depth function does no copy the new depth to the beacon payload
lpsw8091	Stack form distributed network does not set depth or capacity flags in beacon payload
lpsw8407	Stack network Restore functionality doesn't restore the complete data
lpsw8408	binding table does not align to network address/neighbour table changes
lpsw8409	Rejoin and data poll request are sent to invalid address
lpsw8458	PDUM_vInit function does not set the FreeHead List index to zero
lpsw8492	NPDU leak happens with fragmentation
lpsw8498	Beacon notification on an active scan on the coordinator can result in trampling of memory
lpsw8503	GP alias doesn't get reset after sending GP packet
lpsw8504	NWK address response and IEEE address response for RxOnWhenIdle End device for extended address request is wrong
lpsw7654	BDB_bOutOfBandCommissionGetKey() returns incorrect results
lpsw8667	E_ZCL_CBET_REPORT_REQUEST will be missing if two attributes are reported together
lpsw8668	Add a function to increment PollControl timing parameters when an end device wake up from sleep
lpsw8664	The number of endpoints supported by a device should not be equal to or more than maximum endpoint ID supported
lpsw8665	Endpoint not getting registered if it contains any attribute of datatype Int40, Int48, Int56 , Int64
lpsw8666	Poll Control Cluster: Internal Variable Initialization
lpsw8681	Router and Coordinator do a bus error exception when Network leave request is received
lpsw8682	Coordinator sends a beacon response of PAN at capacity after it receives a unicast management permit join req
lpsw8692	The event ZPS_EVENT_APS_ZGP_DATA_INDICATION is not getting generated at AT_GP application and ZCL for DUT_ZGPD
lpsw8697	The function BDB_vNsStateMachine would set a wrong bdbNodeJoinLinkKeyType attribute in bdb_ns.c file
lpsw8712	The APDU is not getting freed in BDB source code if ZPS function returns non-success value
lpsw8719	A new device can not join even when an old device has left the network when using Flash TCLK feature
lpsw8742	Route doesn't recover from a many to one route failure

4.3 Modifications Required

These modifications must be made otherwise the stack will not function properly.

1. The new stack has support for better throughput and automatic buffering of data packets during route discovery. This requires the addition of a new queue in the application.

The size of the queue is defined as:

```
#define MCPS_DCFM_QUEUE_SIZE    5
```

This queue should be tied to the stack definition:

```
extern PUBLIC tszQueue zps_msgMcpsDcfm;
```

The storage of the queue must be defined:

```
PRIVATE MAC_tsMcpsVsCfmData asMacMcpsDcfm[MCPS_DCFM_QUEUE_SIZE];
```

In the APP_vInitResources function an additional queue must be added:

```
ZQ_vQueueCreate(&zps_msgMcpsDcfm,      MCPS_DCFM_QUEUE_SIZE,  
sizeof(MAC_tsMcpsVsCfmData),(uint8*)asMacMcpsDcfm);
```

2. All references to BDB_vSetKeys should be replaced with ZPS_vSetKeys.
3. The beacon filter functionality has been extended to support filtering on depth. This addition means that the filter mask field in the beacon filter structure has been changed from a u8FilterMap to u16FilterMap. The bit mask doesn't change.
4. The binding table functionality has been optimised to not store the extended address inside the binding table structure. This now has a look up field.

If the destination address mode in the binding table holds the value of 0x3 (IEEE address mode) the destination address can be got using the following reference:

```
ZPS_tsAplAib * tsAplAib = ZPS_psAplAibGetAib();  
u64Addr = ZPS_u64NwkNibGetMappedIeeeAddr(  
ZPS_pvAplZdoGetNwkHandle(), tsAplAib->psAplApsmeAibBindingTable-  
>psAplApsmeBindingTable[0].pvAplApsmeBindingTableEntryForSpSrcAddr[j].u16  
AddrOrLkUp);
```

where j is the index of the binding table entry.

If the destination address mode is (0x1) which refers to the group address then this is stored as a 16 bit value in the tsAplAib->psAplApsmeAibBindingTable->psAplApsmeBindingTable[0].pvAplApsmeBindingTableEntryForSpSrcAddr[j].u16 AddrOrLkUp.

5. There is a change in the stack behaviour during rejoins.

When a node does a rejoin with the allocate address bit set, In the previous revisions of the stack it would get a new address assigned.

The behaviour of the stack is now to only issue a new address if the address the device has allocated to itself is conflicting with something the parent has.

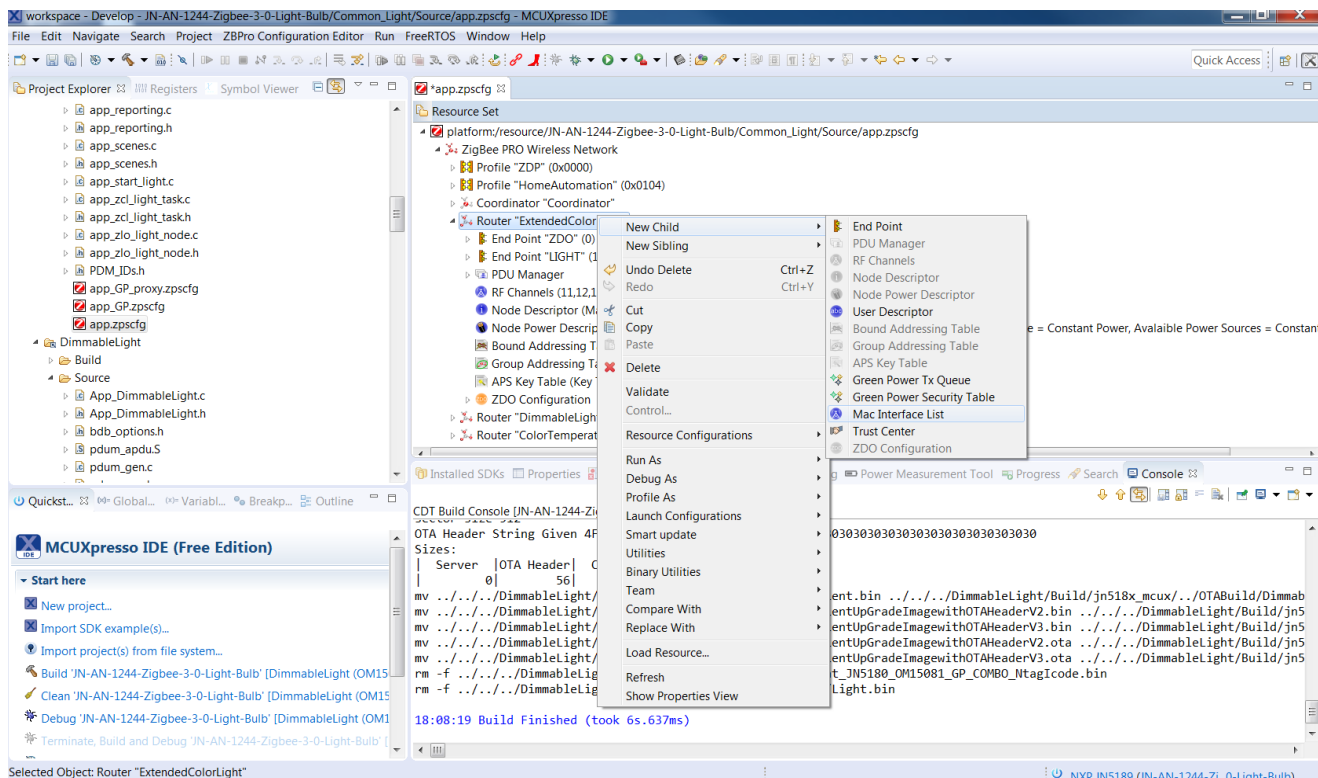
The device would be issued the same address it selected for itself if it wasn't an illegal address i.e. 0 or greater than 0xffff7 or conflicting with current addresses stored by the parent.

6. The [void ZPS_vSetOrphanUpdateDisable (bool_t bEnableOverride)] is no longer supported. This is because the stack by default treats the orphan join as a secured rejoin onto the network and hence doesn't send APSME updates or transport key to the orphaned device.

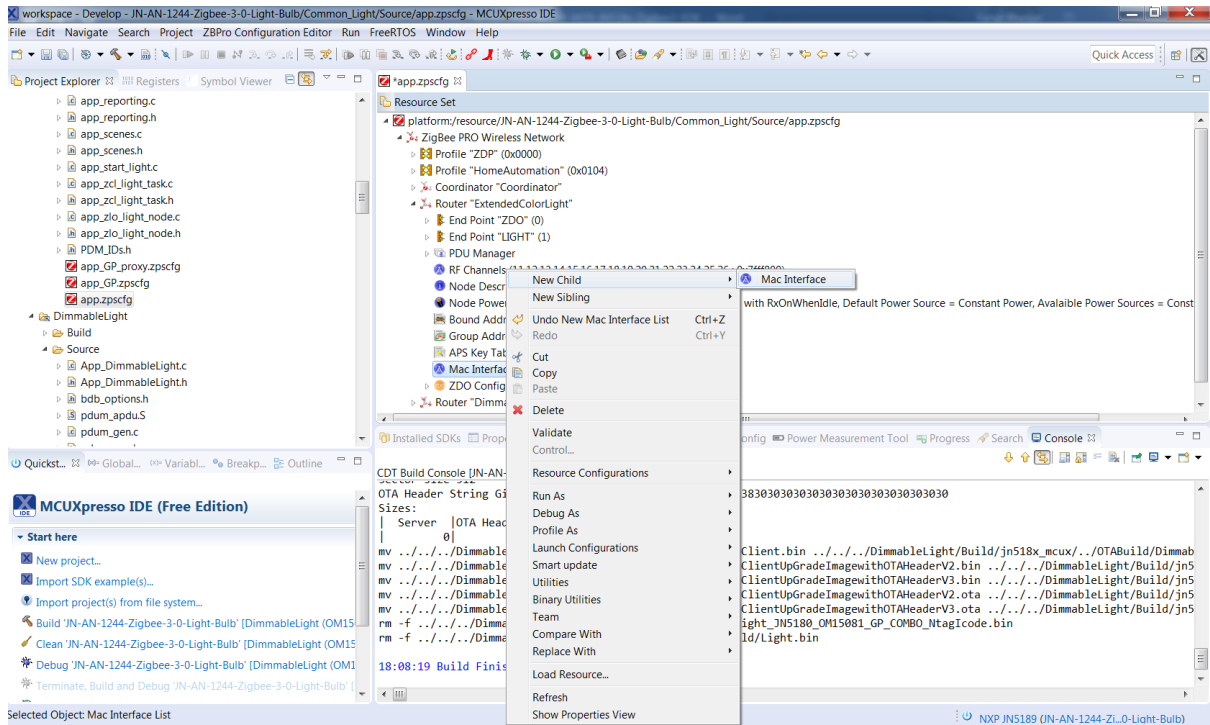
4.3.1 Porting to R22 stack:

The Zigbee PRO R22 version of the stack allows for multiple MAC interfaces to be present. This is to support both 2.4G and 868 MHz frequency bands using the single Zigbee stack. To address this a MAC interface table needs to be configured in the ZPS Config diagram.

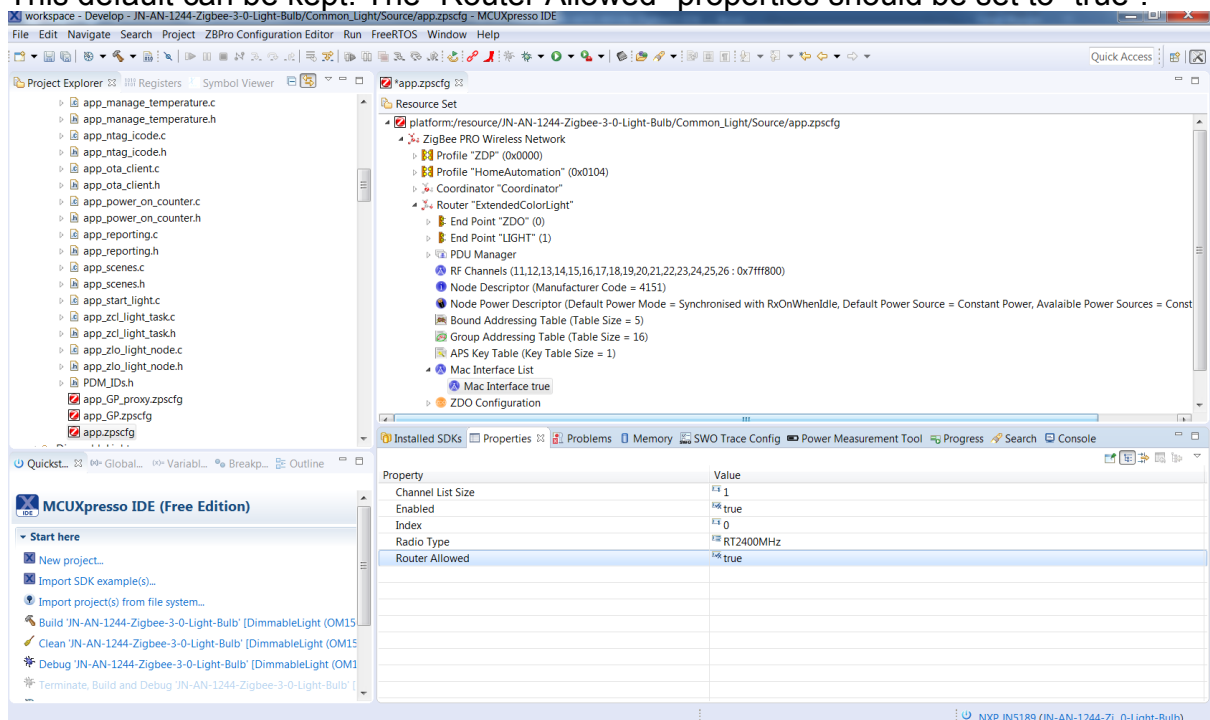
The Mac Interface list can be found as an option for the node. For e.g. if you have Zigbee network with a router node. You can select the router node and press the right mouse button to provide the options. The Mac Interface list can be found under New Child -> Mac Interface List.



After adding the Mac Interface List, select the Mac Interface list and press the right mouse button to provide the options. The Mac interface can be found under New Child -> Mac Interface.



After adding the Mac Interface, the properties can be updated. The default is 2.4G. This default can be kept. The “Router Allowed” properties should be set to “true”.



In the config diagram remove any reference to ChannelStructure
e.g.

```
<ChannelStructure FrequencyBand="2.4GHz"  
PageChannelBitmask="0x7FFFF80"/>
```

Add add

```
<MacInterfaceList>  
  <MacInterface ChannelListSize="1" index="0" RadioType="RT2400MHz"  
  Enabled="true"/>  
</MacInterfaceList>
```

before the ZDOservers

Any reference to `psActvNtEntry->uAncAttrs.bfBitfields.u3Age` should be changed to
`psActvNtEntry->u8Age`

The channel mask in the AIB is an array so this needs to change from
`psAib->u32ApsChannelMask`

to

```
psAib->pau32ApsChannelMask[0]
```

The channel mask is no longer persisted by the stack. This needs to be persisted by
the application.

local address no longer stored in the binding table

```
psAplApsmeBindingTable->u64SourceAddress
```

use

```
ZPS_u64AplZdoGetleeeAddr()
```

Any reference to `MAC_vHandleMcpsVsReqRsp` should now be changed to

```
PUBLIC void ZPS_vMacHandleMcpsVsReqRsp(void* pvMac,  
MAC_tsMcpsVsReqRsp  
*psMcpsVsReqRsp,  
MAC_tsMcpsVsSyncCfm  
*psMcpsVsSyncCfm)
```

The first element should be `NULL`.

The JPT library should be linked into each project build.

The Makefile should be updated with `LDLIBS += JPT_$(JENNIC_CHIP)`

The `PUBLIC uint32 zps_eSocMacSetTxBuffers (uint8 u8MaxTxBuffers)` has changed to
`PUBLIC uint32 ZPS_u32MacSetTxBuffers (uint8 u8MaxTxBuffers)`

The TC callback prototype has changed to

```
bool vNfTcCallback (uint16 u16ShortAddress,
```

```
uint64 u64DeviceAddress,  
uint64 u64ParentAddress,  
uint8 u8Status,  
uint16 u16MacId )
```

5. OTA Image Allocation

The function `eOTA_AllocateEndpointOTASpace()` is used in the application to allocate locations in Flash memory to store application images as part of the Over-The-Air (OTA) upgrade process. The OTA code then uses these locations to store the shadow image before switching to it, after validation.

There are two issues to consider regarding remapping and OTA upgrade:

- Whether the size of the OTA application binary file is larger than the previous version, such that it must use another sector
- Where in the memory space the OTA image is written to

Consider the following case. We have a 154KB image (5 sectors) and download a new image of the same size, starting at sector 8:

Logical Sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Physical Sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Contents	Running image					Empty			New image				Empty			

When we switch to the new image, the physical sectors are moved in the memory map by the bootloader so that the new image becomes the running image and the previous running image becomes the old image. Only the sectors that must be moved are actually moved by the bootloader, and the other sectors are left alone:

Logical Sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Physical Sector	8	9	10	11	12	5	6	7	0	1	2	3	4	13	14	15
Contents	New running image					Empty			Old running image				Empty			

If we now download a further new image that is 161KB (6 sectors) in size, it will replace the old running image and will be placed into logical sectors 8 to 13, which are physical sectors 0, 1, 2, 3, 4 and 13. However, this new image is then unusable because the *physical* sectors are not contiguous and the bootloader does not take this into account when it remaps the memory (if the new image was less than 160KB there would be no problem).

The simple solution to this problem is to replace the remapping that the bootloader has chosen with our own remapping.

Logical Sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Physical Sector	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
Contents	New running image					Empty			Old running image				Empty			

However, this does not leave any space for permanent data and it also assumes that the new image is stored at logical sector 8.

You may choose to put the new image anywhere in the Flash memory (ZigBee allows this to be configured, and a user-developed solution is free to do what it wants). So you need to adjust the remapping to match. For example, if the OTA image was placed at logical sector 7:

Logical Sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Physical Sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Contents	Running image				Empty			New image				Empty				

The purpose of this is to leave some sectors at the end of Flash memory for permanent data (otherwise you could always start the OTA image at sector 8). In such cases, the sensible approach is to calculate how much permanent data space is required, then divide the remaining space into 2 blocks of sectors, configure the OTA upgrade to start at the beginning of the second block of sectors, then force the remapping to swap all of the blocks, regardless of the actual image size.

Consider the example in which a user wants 64KB for permanent data, which requires 2 sectors. This leaves 14 sectors for applications, so we have 2 blocks of 7 sectors each (even though the application may be smaller than this):

Logical Sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Physical Sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Contents	Running image				Empty			New image				Empty		Data		

To avoid any problem with the new image growing and needing 6 sectors rather than 5 sectors, we force the remapping to swap all 7 sectors over:

Logical Sector	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Physical Sector	7	8	9	10	11	12	13	0	1	2	3	4	5	6	14	15
Contents	New running image				Empty			Old running image				Empty		Data		

This leaves sectors 14 and 15 in a fixed location.

The code to achieve this is as follows:

```

if (u8CurrentImageSector > 0)
{
    /* Remapping will not affect the current running image,
       which was already running in a continuous block at the base
       application Flash address */
    vREG_SysWrite(REG_SYS_FLASH_REMAP, 0x0dcb987);
    vREG_SysWrite(REG_SYS_FLASH_REMAP2, 0xfe654321);
}

```


6. Application Porting

Applications developed using the NXP JN516x ZigBee Home Automation/Light Link SDK can be migrated to ZigBee 3.0. Guidance on the migration of applications to ZigBee 3.0 is provided in the Application Note *ZigBee 3.0 Migration Guidelines* (JN-AN-1230).

ZigBee 3.0 applications no longer require the NXP proprietary operating system, JenOS. However, the non-OS resources that were included in JenOS (e.g. Persistent Data Manager) are included in the JN51xx Core Utilities (JCU), supplied in the JN516x ZigBee 3.0 SDK.

7. Related Documentation

The following user documentation supports this software release:

- ZigBee 3.0 Stack User Guide [JN-UG-3113]
- ZigBee 3.0 Devices User Guide [JN-UG-3114]
- ZigBee Cluster Library (for ZigBee 3.0) User Guide [JN-UG-3115]
- ZigBee Green Power (for ZigBee 3.0) User Guide [JN-UG-3119]
- JN51xx Core Utilities User Guide [JN-UG-3116]
- BeyondStudio for NXP Installation and User Guide [JN-UG-3098]
- JN51xx Production Flash Programmer User Guide [JN-UG-3099]
- ZigBee 3.0 Migration Guidelines Application Note [JN-AN-1230]

All the above manuals are available as PDF documents from the [Wireless Connectivity](#) area of the NXP web site.

RELEASE HISTORY (v1518)

8. Release Details

8.1 New Features and Changes

The following new features are included in this release:

Internal ID	Description
lpsw6319	Feature added to ZigBee PRO stack to selectively post APS retries to the MAC layer
lpsw7089	Network layer retries added to ZigBee PRO stack via Shim/MAC
lpsw7170	ZTIMER feature added to allow timers that do not block sleeping
lpsw7385	Feature added for MAC to provide beaconing and data poll PIB minBE settings
lpsw7394	Removed checks for PAN Coordinator when sending beacon notification
lpsw7410	Compile-time optimisation introduced
lpsw7562	In the ZCL, reporting functions have been changed to use new 'bound no ACK' API function
lpsw7627	Added mandatory ClusterRevision for ZLL Commissioning/ ZLL Utility and any other remaining clusters
lpsw7640	In the OTA Upgrade cluster, the ImageType attribute has been added
lpsw7641	In the OTA Upgrade cluster, a maximum block delay has been added
lpsw7658	Definition of Touchlink supported keys now allowed in the application
lpsw7659	Application now allowed to override the Touchlink master key
lpsw7737	In OTA Upgrade cluster, support added for optional ImageStamp attribute
lpsw7754	The OTA Upgrade attribute MinimumRequestDelay is now measured in seconds

8.2 Bug Fixes

The following issues have been fixed in this release:

Internal ID	Description
lpsw7204	ZTimer wake initialisation does not configure the tick timer correctly
lpsw7208	Out-of-band commissioning should send a BDB event to provide the application with a notification on success or failure
lpsw7419	MAC should ignore broadcasts during a data poll on an 'RxOnWhenIdle=FALSE' device
lpsw7423	Routing entry can be defaulted prematurely
lpsw7461	ZigBee Green Power Test Case 4.4.1.7 fails at step 1B due to not sending a channel configuration message when expected
lpsw7534	Incorrect initialisation and restore of OTA Upgrade attributes
lpsw7629	Fragmentation receive side activity counter is not balanced
lpsw7629	Address conflict can result in duplicate Neighbour table entries with the conflicted address
lpsw7630	Rejoin Request with invalid address should result in Rejoin Response with valid address even if the allocate address is set to 0
lpsw7631	In ZCL, ReadAttributeResponse has incorrect length for string type attributes
lpsw7645	In Colour Control cluster, hue is not updated when MoveToColorTemp command is used
lpsw7646	In ZCL, attribute stopped reporting manufacturer name and started reporting ZCL version
lpsw7662	Address conflict for End Device results in removal of the End Device from the parent's tables
lpsw7663	Broadcast from End Device is re-routed back to the End Device
lpsw7674	Device responding to ZCL messages with the profile ID changed to the local profile ID - now filtered on HA/Wildcard profile ID to make sure the response goes out with the original profile ID
lpsw7675	Device Announce erroneously sent out on start-up/reset of a factory-new device – Device Announce now disabled for all resets
lpsw7742	OTA upgrade block request for second block is not sent by OTA Upgrade client
lpsw7748	Leave not issued for an 'RxOnWhenIdle=TRUE' End Device on child aging
lpsw7872	Reboot after OTA upgrade time of 0xffffffff causes bus error
lpsw7927	The Touchlink commands that can be received as broadcasts are now restricted

RELEASE HISTORY (v1396)

9. Release Details

9.1 New Features and Changes

This release includes (as in the previous v1385 build):

- Certified Revision 21 of the ZigBee PRO stack
- ZigBee Cluster Library revision ZCL06
- ZigBee 3.0 Base Device implementation

9.2 Bug Fixes

The following issues have been fixed in this release:

Internal ID	Description
lpsw7056	Jennic sniffer binaries are missing from the SDK.
lpsw7057	Frames are forwarded to a child End Device after the MAC ack turnaround time and after the child's radio has been switched off.
lpsw7063	Touchlink Router initiator/target implementation is incomplete.
lpsw7067	Management Leave and Network Leave are accepted only from the parent device. If a Leave is issued by a device which is not a parent, the Leave is ignored.
lpsw7068	After an address conflict for a GP device, the network source address in the Device Announce is an old address and not the current one.

9.3 Modifications Required

No modifications are required in this release.

9.4 Known Issues

The following issues are known in this release:

Internal ID	Severity	Description
lpsw7062	Low	<p>Issue: Unnecessary route churn as a result of loss of link status due to CCA failure and environmental issues.</p> <p>Workaround: N/A</p>

RELEASE HISTORY (v1385)

10. Release Details

10.1 New Features and Changes

This release includes:

- Certified Revision 21 of the ZigBee PRO stack
- ZigBee Cluster Library revision ZCL06
- ZigBee 3.0 Base Device implementation

10.2 Bug Fixes

There are no bug fixes in this release.

10.3 Modifications Required

No modifications are required in this release.

10.4 Known Issues

The following issues are known in this release:

Internal ID	Severity	Description
lpsw7056	Low	Issue: Jennic sniffer binaries are missing from the SDK. Workaround: N/A
lpsw7062	Low	Issue: Unnecessary churn of route discovery on a large network. Workaround: N/A
lpsw7067	Low	Issue: Management Leave and Network Leave are accepted only from the parent device. If a Leave is issued by a device which is not a parent, the Leave is ignored. Workaround: N/A
lpsw7057	Low	Issue: Frames are forwarded to a child End Device after the MAC ack turnaround time and after the child's radio has been switched off. Workaround: N/A
lpsw7068	Low	Issue: After an address conflict for a GP device, the network source address in the Device Announce is an old address and not the current one. Workaround: N/A
lpsw7063	Low	Issue: Touchlink Router initiator/target implementation is incomplete. Workaround: N/A