
Release Note

SmartDSP OS

Table of Contents

1	About This Release	3
1.1	Version Information.....	3
1.2	Installation and Licensing.....	3
1.3	Getting Help.....	3
1.4	System Requirements.....	4
1.5	Who Should Use this Release?	4
1.6	Who Should NOT Use this Release?	4
1.7	Suggested Readings.....	4
2	New/Changed Features	5
2.1	General Features	5
2.2	Kernel Features	5
2.2.1	PSC9131 kernel.....	5
2.2.2	Rearrangement of Kernel layout.....	5
2.2.3	COP API	5
2.2.4	Hardware Semaphore	7
2.2.5	Watchdog Timer (WDT)	7
2.2.6	MMU ISR behavior.....	7
2.2.7	Intercore message assertions	8
2.3	Driver Features	8
2.3.1	PSC9131 drivers.....	8
2.3.2	MAPLE.....	8
2.3.3	BMan-lite cache support	10
2.3.4	CPRI slave mode	10
2.3.5	System DMA configuration.....	10
2.3.6	AIC.....	10
2.4	Demos	11
2.4.1	PSC9131 RDB BSP	11
2.4.2	Heterogeneous BSP	11
2.4.3	Demo README files.....	12
2.4.4	PSC9131 demos.....	12
2.4.5	binary file creator	13
2.4.6	Shell demo mbf target	13
3	Performance	14
4	Important Notes.....	15
4.1	Note on supported architectures	15
4.2	Note on MSC815x ADS demos.....	15
4.3	Note on PSC9131 RDB demos	15
4.4	Note on ISR Stack size:	15
4.5	Note on Kernel Awareness:	16
4.6	Note on MSC815x/MS825x projects.....	16



4.7	Note on MSC825x binaries	16
4.8	Note on SmartDSP HEAT server	16
4.9	Note on CodeWarrior RSE	16
5	Known Issues	17
6	Resolutions and work-around	18
6.1	Level interrupt clearing	18
6.2	MBF Net checksum()	18
6.3	eMSG error	19
7	Bug Fixes	20
8	Compatibility	21
9	Deprecations	22
9.1	Deprecated in this release	22
9.1.1	OS version macros	22
9.1.2	DEPE_LTE_OBO	22
9.1.3	MAPLE MBUS map	22
9.1.4	PCIe configuration map	22
9.2	Deprecated in previous release(s)	22
10	New/Changed Features Log	28

1 About This Release

SmartDSP OS is a preemptible, real-time, priority-based operating system for MSC815x, MSC825x, MSC814x, MSBA8100, MSC8102/22/26, MSC8101/3 and MSC711x, supporting software and hardware interrupts software and hardware timers, memory management, I/O drivers and TCP/IP stack. See note on supported architectures for MSC8101/3, MSC8102/22/26 and MSC711x support.

1.1 Version Information

This release note provides important information for users of SmartDSP OS for MSC815x/MS825x/PSC9131. The current version is B04.01.00.

1.2 Installation and Licensing

The SmartDSP OS release package can be used to patch CodeWarrior by overwriting the CodeWarrior installation. The file can be downloaded externally from:

<https://www.freescale.com/cgi/go/163455322>

If patching an existing CodeWarrior installation, users should perform the following steps:

1. Delete the contents of the folder
eclipse\configuration\org.eclipse.core.runtime
2. Delete the folder com.freescale.doc.smartdsp.os.userGuide_ * (where * is a version string) from installDir\eclipse\plugins.
3. Delete the folder com.freescale.doc.smartdsp.os.api.doc_ * (where * is a version string) from installDir\eclipse\plugins.
4. Delete the folder com.freescale.doc.fast_index from
installDir\eclipse\plugins.
5. Delete the directory installDir\eclipse\configuration\org.eclipse.help.base\index.
6. Unzip the SmartDSP OS patch zip file and overwrite CodeWarrior. The base folder to unzip is installDir
7. Start CodeWarrior with the -clean flag

SmartDSP OS is released under the same licensing terms (EULA) as CodeWarrior. Specific header files may have additional Open Source license agreements embedded within.

1.3 Getting Help

For users new to Eclipse, please refer to the document [Getting Started with CodeWarrior for StarCore v10.0](#), which is in the same Compass folder as the installer. The [CodeWarrior for StarCore v10.0 FAQ](#) (also in the installer folder) also answers common questions.

If you have questions, issues, or want to provide feedback, please use the Freescale online support web page. To use this page, follow these steps:

1. In a web browser, go to <http://www.freescale.com/TechSupport>.
Freescale's **Technical Support** web page appears.
2. On this page, click the [Submit a service request online](#) link.
The **New Service Request — Category/Topic** page appears.
3. From the Category dropdown menu, select Technical Request.
4. From the Topic dropdown menu, select CodeWarrior (or other appropriate topic).
5. Click **Next**.
The **New Service Request — SR Details** page appears.

6. In this page, enter the requested information.
At a minimum, enter information in each field marked by an *.
7. Click **Submit**.
If you are already logged in, the **Service Request Confirmation** page appears. Go to the last step.
If you are not already logged in, the **Log-in** page appears.
8. If you are a registered member, login with your user name and password.
The **Service Request Confirmation** page appears. Go to the last step.
9. If you have not yet registered,
 - a. If you want to become registered member, click **Register Now** and complete the registration process.
The **Service Request Confirmation** page appears.
 - b. If you do not want to register, supply your contact information in the **I do not want to register - Provide contact information** form and click **Submit**.
The **Service Request Confirmation** page appears.
10. Click **Done**.
Your service request is submitted.

1.4 System Requirements

Recommended Configuration

- 3GHz Intel® Pentium® P4 processor or better. Dual-core processor preferable.
- Microsoft® Windows XP
- 2GB RAM (Step speed on machines with 1GB RAM is significantly slower)
- 1GB disk space

Note: 500MB of free space is required on the OS drive, regardless of the free space available on the destination drive.

Note: this release has been tested on Microsoft® Windows 7 as well as Microsoft® Windows XP

1.5 Who Should Use this Release?

- Users developing for the PSC9131 device
- Users developing for the MSC8157 device.
- Users developing for the MSC8154/6/2/1.
- Users developing for the MSC8256/4/2/1.

1.6 Who Should NOT Use this Release?

- MSC812x, MSC8101/2 and MSC711x users should use R02.04.00 available at <https://www.freescale.com/cgi/go/222885996>
- MSBA8100 users should use R03.07.00 available at <https://www.freescale.com/cgi/go/222885996>
- MSC8144 users should use R03.09.00 available at <https://www.freescale.com/cgi/go/222885996>

1.7 Suggested Readings

For more information on SmartDSP OS, refer to SmartDSP OS User Guide and SmartDSP OS API Reference Manual.

2 New/Changed Features

This release note includes all features including those that were originally introduced in SmartDSP OS R03.10.01 (including those released in A04.00.00)

2.1 General Features

2.2 Kernel Features

2.2.1 PSC9131 kernel

SmartDSP OS B04.01.00 provides beta support for the SmartDSP OS kernel for PSC9131. Users are required to provide the C preprocessor macros PSC9131 and SC3X50 in applications so that any included file that is dependent on the architecture. The project is available in `lib\projects\projects\psc9x3x`

2.2.2 Rearrangement of Kernel layout

Towards boosting reusability and maintenance of the OS across platforms, SmartDSP OS B04.01.00 rearranged the location of many of the kernel components and drivers to non-device-specific locations. The changes can be divided into two main categories:

1. Core specific. The new header files can be found (primarily) at `include\arch\starcore\core_family\sc3X00` and include the following:
 - a. `sc3x00_cache.h`
 - b. `sc3x00_mmu.h`
2. Peripherals. The new header files can be found (primarily) at `include\arch\starcore\peripherals` and include the following:
 - a. `class\class_profiling.h`
 - b. `dma\dma.h`
 - c. `pex\pex.h`
 - d. `tdm\tdm.h`
 - e. `timers\hw_timers.h`
 - f. `vsg\hw_sem.h`

In addition, the linker file has also gone the same reorganization for the same purposes.

Important: Backward compatibility is maintained; albeit a `#warning` pre-processor has been placed in various header files indicating the location of the new files. Users are encouraged to make the necessary changes in their code.

Note: this feature was originally introduced in A04.00.00

2.2.3 COP API

SmartDSP OS B04.01.00 modifies some existing features in the COP channel and adds others. All changes are noticeable in the file `include/common/os_cop.h`

Note: these features were originally introduced in A04.00.00

2.2.3.1 COP Channel Open

SmartDSP OS B04.01.00 moves the COP channel structure definition from the file `os_cop.h` to an internal header file; thus the channel handle (`cop_channel_t`) has changed to a `void*`. The purpose of this change is to allow the OS to add/modify to the COP channel behavior without impacting public header files in the future.

Towards this, the structure `cop_ch_open_params_t` has an additional member `heap` (of the type `os_mem_type`) in which the user is to define the heap from which the channel structure should be allocated. The behavior of the COP module in the kernel is:

1. In the debug versions of the kernel: assert if a non-valid heap has been passed in `cop_ch_open_params_t.heap`. This is to promote awareness to this feature.
2. In the release versions of the kernel: revert to the `OS_MEM_LOCAL` heap. This is in order to minimize the impact on customer code that is not in the debug stage.

2.2.3.2 osCopSharedChannelOpen()

SmartDSP OS B04.01.00 adds support to shared COP channels, such that one core can dispatch and another core can reap for the channel. For each channel there can be only one dispatcher (producer) and one reaper (consumer). Towards this the function `osCopSharedChannelOpen()` (defined in `include/common/os_cop.h`) has been added.

```
os_status osCopSharedChannelOpen(cop_dev_handle cop_handle,
                                cop_channel_t *cop_ch,
                                cop_ch_open_params_t *ch_open_params,
                                int mode,
                                os_queue_handle queue);
```

Each channel can be opened as a bitwise or of the following defines:

```
#define COP_CH_REAP 0x1
/**< The channel can reap jobs after the co-processor finishes
    executing them */
#define COP_CH_DISPATCH 0x2
/**< The channel can send jobs for the co-processor to execute */
```

The existing API (`osCopChanelOpen()`) will continue to open channels as (`COP_CH_REAP | COP_CH_DISPATCH`), in addition to creating (rather than receiving a pre-initialized) queue handle.

Important: if creating a shared channel, the queue **MUST** be pre-initialized by the application as a shared queue; i.e. when calling `osQueueFind()`, the `shared` parameter passed to the function must be set to `TRUE`.

Important: the address of the job handle(s) passed to `osCopChannelDispatch()` will be returned to the core on which the channel's callback function is called (i.e. the consumer channel). As such, job handles should have a 1:1 translation from the virtual to physical addresses – otherwise the application should deal with translating virtual(producer) to virtual (consumer).

Important: the addresses of the job's LLD pointer(s) passed to `osCopChannelDispatch()` will be returned to the core on which the channel's callback function is called (i.e. the consumer channel). As such, they should have a 1:1 translation from the virtual to physical addresses – otherwise the application should deal with translating virtual(producer) to virtual (consumer). Another option is to pass physical addresses on the job's LLD description and to set `maple_pe_ch_open_params_t.no_translation` to `TRUE` when opening the channel.

2.2.3.3 Individual COP Channel callback

SmartDSP OS B04.01.00 adds the option to have a separate callback function for each COP channel. By default, the channels will inherit the COP device's callback function; however the user can call `osCopChannelCtrl()` passing the channel handle, `COP_CHANNEL_CALLBACK_SET` and a new callback (casted to `void*`) in order to override this. The user may also call the same API with the command `COP_CHANNEL_CALLBACK_REMOVE` in order to revert to the COP device's callback function.

This new feature is demonstrated in `demos\starcore\msc815x\maple_tvpe`

2.2.4 Hardware Semaphore

SmartDSP OS B04.01.00 adds a driver to the VSP hardware semaphores. The driver is initialized be default as part of the device initialization. The API is outlined in `include/arch/peripherals/vsg/hw_sem.h` and was designed to closely follow the spinlock API defined in `include/common/os_multicore_sync.h`

Note: this feature was originally introduced in A04.00.00

2.2.5 Watchdog Timer (WDT)

Note: these features were originally introduced in A04.00.00

2.2.5.1 OS handled WDT

SmartDSP OS B04.01.00 adds the option for the user to define the OS handling of WDT interrupts when `OS_WATCHDOG_SUPPORT` is defined to `OS_WDT_AUTO_HANDLING` in `os_config.h`. Towards this 2 new defines may be used:

- `OS_WATCHDOG_BEHAVIOR` – describes how the WDT should behave when the WDT has expired; defaults to the backward compatible `GENERATE_RESET`
- `OS_WATCHDOG_HANDLER` – defines the callback function to be triggered when the WDT expires; defaults to the backward compatible `NULL`.

Important: in PSC913x architectures, `GENERATE_RESET` is not supported thus the user must specify `DSP_WATCHDOG_DEFAULT` (defined in each architecture's `smartdsp_os_device.h`) or specifically `GENERATE_NMI`.

2.2.5.2 Default registration of WDT

SmartDSP OS B04.01.00 removes the default registration of the kernel to the WDT. This change was done in order to remove redundant code size; as the interrupt would not be generated without the application (or the OS) calling `osHwWatchdogCreate()` regardless of the registration as part of bringing up the interrupts module.

2.2.6 MMU ISR behavior

SmartDSP OS B04.01.00 modifies the behavior of the I/DMMU ISR behavior in two ways:

1. If the relevant debug hook was installed by the application, the ISR will not induce a default `asm(" debug")`. This change was introduced in order to leave the handling of the MMU exception entirely in the hands of the application; thereby enabling recovery from such exceptions without causing the core to go to debug state.
2. Clearing the exception in the MMU will be done after returning from the debug hook or after single-stepping the core over the `asm(" debug")`. This change was introduced in order to allow users an addition view in to the cause of the exception, namely CodeWarrior's MMU Configurator view.

Note: this feature was originally introduced in A04.00.00

2.2.7 Intercore message assertions

SmartDSP OS B04.01.00 modifies the behavior of the intercore messaging to the effect that the messages will assert when trying to post asynchronously on a synchronous message and vice versa. There should be visible affect to the user's application.

2.3 Driver Features

2.3.1 PSC9131 drivers

SmartDSP OS B04.01.00 provides alpha support for the SmartDSP OS drivers for PSC9131. Users are required to provide the C preprocessor macros PSC9131 and SC3X50 in applications so that any included file that is dependent on the architecture. The project is available in `lib\projects\projects\psc9x3x`

Important: the supported drivers are listed in the sections below. Other drivers that are found in the source code are not to be assumed as verified.

Note: this feature was originally introduced in A04.00.00

2.3.2 MAPLE

2.3.2.1 MAPLE flavors

SmartDSP OS B04.01.00 provides the following defines, or the various MAPLE types. These defines are to be found in the relevant `smartdsp_os_device.h` file:

- `MAPLE_B` for MSC8156 family of DSP
- `MAPLE_B2` for MSC8157 family of DSP
- `MAPLE_B2F` for PSC9131 family of heterogeneous SoC
- `MAPLE_B2P` (not released yet) PSC9132 family of heterogeneous SoC

Note: this feature was originally introduced in A04.00.00

2.3.2.2 MAPLE-B2 μ code

SmartDSP OS B04.01.00 is released with μ code rev10 for MAPLE-B2.

2.3.2.3 MAPLE-B2F μ code

SmartDSP OS B04.01.00 is released with μ code rev3 for MAPLE-B2F. Users are encouraged to view the μ code release notes for a list of known supported and unsupported features.

2.3.2.4 DEPE driver changes

SmartDSP OS B04.01.00 makes the following modifications to the DEPE driver towards supporting the MAPLE-B2 and MAPLE-B2F μ code versions. All changes are visible in `drivers/maple/rev1/include/maple_depe.h`

- Add WiMAX 16m support to the DEPE and added the `depe_wimax16m_header_t` header
- Change the naming of `DEPE_LTE_OBO` field of DEPE BD to `DEPE_OBO`. Backward compatibility is maintained using the define below; however this functionality is deprecated

(see 9.1 below). This field is now mandatory for all types of headers used by the DEPE.

```
#define DEPE_LTE_OBO(OBO) DEPE_OBO(OBO)
```

- Added the OFFKB bit in the eFTPE BD indicating that the CPS offset of input data during repeat is indicating KB offset and not samples.

2.3.2.5 eFTPE driver changes

SmartDSP OS B04.01.00 adds the following flags to the eFTPE job descriptor. All changes are visible in drivers/maple/rev1/include/maple_b2_ftpe.h

FLAGS for maple_ftpe_job_t.first_flags

```
#define FTPE_BD_CFG0_OFFKB 0x00000001
/**< Offset in KBytes; Valid only for FFT operation and when the BD
repeat option is enabled; Will be copied by driver to BD[OBO] */

#define FTPE_BD_CFG0_OFF16 0x00000002
/**< MAPLEB2 only; Offset in 16 Bytes; Valid only for DFT/iDFT operation
and when the BD repeat option is enabled; Will be copied by driver
to BD[OBO] */
```

2.3.2.6 eTVPE driver changes

SmartDSP OS B04.01.00 adds the following flags to the eTVPE job descriptor. All changes are visible in drivers/maple/rev1/include/maple_b2_tvpe.h

FLAGS for maple_tvpe_job_t.offsets

```
#define TVPE_WIMAX_16M 0x00008000
/**< WiMAX 802.16m indication. Valid only for WiMAX and for Separate
Vectors
```

2.3.2.7 MAPLE-B2F supported PE

SmartDSP OS B04.01.00 provides support for the following PE in MAPLE-B2F: CRCPE, DEPE, eFTPE, eTVPE, PUFFT, PUSCH and PDSCH. Other PE are not supported to-date.

2.3.2.8 MAPLE PCR commands

SmartDSP OS B04.01.00 adds support for generating PCR commands. The supported commands are enumerated as maple_pcr_opcodes_t (defined in drivers/maple/rev1/include/maple.h).

There are 2 flavors for activating MAPLE PCR commands, both are used by passing the relevant command as the command parameter to osCopDeviceCtrl() and the relevant maple_pcr_opcodes_t as the params parameter:

```
#define MAPLE_CMD_PCR_ACTIVATE_WITH_POLL (0x00000500 | COP_LLD_COMMAND)
/**< Activate MAPLE PCR routine and wait till it finishes */
#define MAPLE_CMD_PCR_ACTIVATE_NO_POLL (0x00000600 | COP_LLD_COMMAND)
/**< Activate MAPLE PCR routine and don't wait till it finishes */
```

Note: this feature was originally introduced in A04.00.00

2.3.3 BMan-lite cache support

SmartDSP OS B04.01.00 adds support for cache coherency on the BMan-lite (part of the eMSG) buffers. Towards this, the users are required to provide the caching policy relevant to the specific pool in the new field `bml_channel_config_params_t.coherency_en` (declared in `initialization/arch/peripherals/bml/include/bml_init.h`). Valid settings can be found in `include/common/os_cache.h` and include:

- NOT_CACHED
- L1_CACHED
- L2_CACHED
- L1_L2_CACHED

The BMan-lite driver will flush the buffer according to the proper policy prior to putting it in the pool.

Important: this feature is not backward compatible and must be added by users to the application.

2.3.4 CPRI slave mode

SmartDSP OS B04.01.00 adds support for CPRI slave port initialization. Towards implementing this, the `cpri_sync_mode_t` enumeration was expanded and now includes:

- CPRI_MASTER_MODE
- CPRI_SLAVE_MODE
- CPRI_END_POINT_MODE_SLAVE_MODE

This sync mode and the minimal acceptable line rate are added as new API to the function `cpriLinkRateAutoNegotiate()`

All changes are in the file

`initialization/arch/peripherals/cpri/include/cpri_init.h`

Important: When using only masters or only slaves on a single MSC8157 device – there is no restriction regarding minimal and maximal acceptable rates. When using combination on masters and slaves on a single MSC8157 device, the maximal and minimal accepted rates must be the same rate.

Note: it is not expected that the function `cpriLinkRateAutoNegotiate()` be called by the application directly.

2.3.5 System DMA configuration

SmartDSP OS B04.01.00 adds `syncio` opcodes when writing the following DMA registers:

- DMACHER in `osDmaChannelStart()`
- DMACHFR in `dmaChannelFreeze()`
- DMACHDFR in `dmaChannelDefrost()`

Although this may cause these functions to run longer than in previous versions – there is no change required from the application.

2.3.6 AIC

SmartDSP OS B04.01.00 adds alpha support for the PSC9131 AIC. Bringing up the AIC hardware module consists of 3 stages:

1. Phy configuration.
2. AIC RF configuration - network protocol details, timing and synchronization are set.

3. AIC DMA configuration - including enabling downlink/uplink.

The Freescale BSP is designed such that the first two stages are handled by Linux while third handled by SmartDSP OS.

The driver was successfully tested on DL and UL paths in LTE-FDD, single antenna in bandwidth 5MHZ, 10MHZ, 15MHZ and 20MHZ.

All tests conducted in DBM (Double Buffer Mode).

The driver bundled with B04.01.00 is known to work with an engineering version of Linux; however users are requested to wait for the formal release prior to starting work with the driver.

Note: demos for AIC will be available in future releases

Known limitations:

1. ADI is supported; Maxim is not
2. Using 2 antennas is yet to be tested
3. Testing with MAPLE hasn't been carried out yet
4. Stream Mode and buffer thresholds are yet to be tested

2.4 Demos

2.4.1 PSC9131 RDB BSP

SmartDSP OS B04.01.00 adds a BSP for the PSC9131 RDB. The project can be found in `demos\starcore\psc9x3x\rdb_support\rdb_project`

Note: this feature was originally introduced in A04.00.00

2.4.2 Heterogeneous BSP

PSC9131 is a heterogeneous SoC, including one Power e500 core and one StarCore SC3850 core. Freescale will provide a heterogeneous BSP for this device, including Linux and SmartDSP OS. The flow of this BSP is for das-uboot to boot Linux which will boot SmartDSP OS. The SmartDSP OS initialization flow (`osInitialize()`) is dependent on a structure in a predefined location in memory. The location is passed to the SmartDSP OS kernel as

`SOC_HET_CTRL_BASE` in `os_config.h` and it equals (in the SmartDSP OS demo suite) to the value of the linker command file (lcf) symbol `__SHARED_CTRL_b`. The system architect of the heterogeneous software solution should ensure that SmartDSP OS and Linux agree on the location of this structure.

The structure `os_het_control_t` is defined in `include/arch/starcore/psc9x3x/heterogeneous/psc913x_heterogeneous.h`.

Note: this feature was originally introduced in A04.00.00

2.4.2.1 PSC9X3X_SC_ONLY_BSP

SmartDSP OS B04.01.00 libraries can be compiled with the flag `PSC9X3X_SC_ONLY_BSP`; which indicates that the libraries are compiled to run that without the Linux BSP in the system. When this flag is present, the OS will skip any functionality that is related to the heterogeneous BSP.

Note: this feature was originally introduced in A04.00.00

Important: this flag is not pre-compiled in the libraries.

2.4.2.2 OS_MIMIC_PA_ARCH

SmartDSP OS B04.01.00 applications can be compiled with defining

```
#define OS_MIMIC_PA_ARCH ON
```

in `os_config.h`. In this case, the OS will perform the necessary initialization hooks in order to mimic the behavior of the Linux BSP for cases where it is not released or the user wants to debug the StarCore application in a stand-alone environment.

Users are called to add the following line in `main()` prior to invoking `osInitialize()` in order to have their software run in any combination of the flags `PSC9X3X_SC_ONLY_BSP` and `OS_MIMIC_PA_ARCH`:

```
#if !defined(PSC9X3X_SC_ONLY_BSP) && (OS_MIMIC_PA_ARCH == ON)
    psc913xHetBspStubInitialize((void *)SOC_HET_CTRL_BASE, NULL);
#endif // PSC9X3X_SC_ONLY_BSP && OS_MIMIC_PA_ARCH
```

Note: this feature was originally introduced in A04.00.00

Important: the function is declared in

`demos/starcore/psc9x3x/rdb_support/psc9x3x_het_bsp_stub.h`; and users are required to specifically `#include` this file in order to avoid runtime errors.

2.4.3 Demo README files

SmartDSP OS B04.01.00 introduces a unified README format for all the demos. This format provides all the information regarding the demo, its setup, changes to the board, various targets and expected results. The format is expected to provide a better user experience and we welcome any and all feedback.

Note: this feature was originally introduced in A04.00.00

2.4.4 PSC9131 demos

SmartDSP OS B04.01.00 includes the following families of demos. Many of these demos have multiple targets. Please refer to the relevant README files

2.4.4.1 General kernel demos

```
demos\starcore\psc9x3x\basic_demo
demos\starcore\psc9x3x\cpp_demo
demos\starcore\psc9x3x\debug_hooks
demos\starcore\psc9x3x\hw_timer
demos\starcore\psc9x3x\multitask_simple
demos\starcore\psc9x3x\watchdog
```

2.4.4.2 MAPLE-B2F demos

```
demos\starcore\psc9x3x\maple_crcpe
demos\starcore\psc9x3x\maple_depe
```



```
demos\starcore\psc9x3x\maple_ftpe  
demos\starcore\psc9x3x\maple_pdsch  
demos\starcore\psc9x3x\maple_pusch  
demos\starcore\psc9x3x\maple_pufft  
demos\starcore\psc9x3x\maple_tvpe
```

2.4.5 binary file creator

SmartDSP OS B04.01.00 provides a utility for creating binary images that are loadable by the Freescale Linux BSP in accordance to the hardware boot flow. The utility can be found in `demos\starcore\psc9x3x\psc913x_bin_creator.exe`

Running the utility with the `-h[elp]` flag will produce the following:

```
Usage: SmartDSP\demos\starcore\psc9x3x\psc913x_bin_creator.exe  
[-cw <path>] [-dir <path>] [-h[elp]]  
Creates an image that can be loaded by the Freescale Linux BSP for  
PSC913x; using sc100-elf2xx.exe  
  
-h[elp]: Prints this message  
-cw:      Valid path that CodeWarrior would evaluate as ${SC_TOOLS_HOME}  
          Default: C:\Program Files\Freescale\CW SC v10.2.2\SC  
-dir      Path where to search for eld binaries. Default is current  
location
```

All SmartDSP OS PSC913x demos include the following line in the post-build steps:

```
${CW_Project}\..\..\psc913x_bin_creator.exe -cw '${SC_TOOLS_HOME}' -dir  
'${CW_Project}'
```

2.4.6 Shell demo mbf target

SmartDSP OS B04.01.00 adds a multi-buffered-frame (mbf) target to the shell demo `demos\starcore\msc815x\shell_demo`.

Please refer to the README file for specifics.

3 Performance

SmartDSP OS B04.01.00 kernel performance figures are shown in Table 1.

The mode of calculation was such that each measurement was taken 5 times; once with cold instruction caches and the other without.

The WCS (worst case scenario) is the cold cache result.

The BCS (best case scenario) is the best of the results.

The Avg. is the average of the remaining results.

		MSC8156			PSC9131			MSC8157		
		WCS	BCS	Avg.	WCS	BCS	Avg.	WCS	BCS	Avg.
HWI prolog/epilog from HWI	HWI	88	87	87	90	87	87	87	87	87
	Epilog									
	Prolog	157	123	123	140	123	123	138	123	123
HWI prolog/epilog from SWI	HWI	173	172	172	175	172	172	180	172	172
	Epilog									
	Prolog	146	123	123	141	123	123	147	123	123
HWI prolog/epilog from Task	HWI	147	140	140	144	140	140	155	140	140
	Epilog									
	Prolog	158	126	126	148	126	126	153	126	126
High priority SWI from low priority SWI	SWI	485	475	475	477	477	477	476	476	476
	Epilog									
	Prolog	315	282	282	319	283	283	327	283	283
High priority Task from low priority Task	Task	450	439	439	447	440	440	447	440	440
	Epilog									
	prolog	467	410	410	457	410	410	473	410	410
Low priority SWI from high priority SWI	SWI	183	183	183	183	183	183	190	180	183
	Epilog									
	Prolog	253	242	242	255	242	242	265	242	242
SWI from Task	SWI	183	183	183	183	183	183	190	183	183
	Epilog									
	Prolog	399	323	323	387	323	324	429	323	323

Table 1 SmartDSP Performance Figures

4 Important Notes

4.1 Note on supported architectures

- SmartDSP OS B04.01.00 supports PSC9131, MSC815x/MS825x.
- SmartDSP OS R03.10.00 is that last release of SmartDSP OS that will introduce new features for the MSC8144 device. Maintenance and bug fixes will be performed based on this release.
- SmartDSP OS R03.06.01 is that last release of SmartDSP OS that will introduce new features for the MSBA8100 device. Maintenance and bug fixes will be performed based on this release.
- SmartDSP OS R02.04.00 is the last release of SmartDSP OS that will introduce new features for MSC8101/3, MSC8102/22/26 and MSC711x devices. Maintenance and bug fixes will be performed based on this release.

4.2 Note on MSC815x ADS demos

Some demos (TDM Framer, Ethernet) require special (not default) settings of the board. Please refer to readme files.

In some demos, the BSP will try to read and write BCSR registers in order to override the ADS dip switch settings to the ones required for the demo.

The file `SmartDSP/demos/starcore/msc815x/ads_support/common/marvell_phy.c` can be configured by setting:

```
#define PILOT_8156ADS ON    /* for pilot board */
#define PROTO_8156ADS ON  /* for proto board */
#define PILOT_8157ADS ON  /* for pilot board */
#define PROTO_8157ADS ON  /* for proto board */
```

If none is set, the BSP will try and identify the board revision by reading the BCSR.

4.3 Note on PSC9131 RDB demos

The demos running on the PSC9131 RDB were tested using a ONCE UTAP connector connected to the 14 pin ONCE HEADER on the RDB. The switch settings that are known to work for this connectivity are {0-ON, 1-OFF}:

```
SW1: 01010110
SW2: 11111111
SW3: 10011011
SW4: 00011111
SW5: 01111111
SW6: 00000000
```

CodeWarrior supports this connection with the system title `PSC9131_SC`.

4.4 Note on ISR Stack size:

ISR stack size is defined in `os_config.h` and not in the linker file (as opposed to the CodeWarrior stationary configuration)

```
#define OS_STACK_SIZE    <size>
```

4.5 Note on Kernel Awareness:

SmartDSP releases include Kernel Awareness. The log visualizer has been moved from being bundled with the IDE to being a SmartDSP component. In order to run visualization the user should `SmartDSP\tools\CommExpert\bin\CommExpert.exe` prior to running the application.

To activate, select in the menus: **Runtime -> Kernel Awareness -> Listen to KA plug-ins** and then press the **Start** button.

The default port for listening is 40000. If changed in the `CommExpert` tool or in the IDE launch file, the other location must be modified to match.

4.6 Note on MSC815x/MSC825x projects

MSC815x/MSC825x projects must add the relevant SoC (MSC8157, MSC8156, MSC8154, MSC8152, MSC8151, MSC8256, MSC8254, MSC8252 or MSC8251) as well SC3X50 as compiler preprocessors.

4.7 Note on MSC825x binaries

The MSC8256 kernel and driver share the same binaries as the MSC8156.
 The MSC8254 kernel and driver share the same binaries as the MSC8154.
 The MSC8252 kernel and driver share the same binaries as the MSC8152.
 The MSC8251 kernel and driver share the same binaries as the MSC8151.

4.8 Note on SmartDSP HEAT server

The SmartDSP HEAT is compiled and delivered for Microsoft Windows, with a Makefile to enable compilation under Linux. Under Windows (see [link](#)) if a file is not opened, using `fopen()`, specifically as binary or text, it defaults to the non-standard default of `_fmode` which defaults (see [link](#)) to text. This has ramifications on how carriage return–linefeed combinations are returned from the server to the client.

4.9 Note on CodeWarrior RSE

Starting with CodeWarrior release 10.1.8, CodeWarrior uses Remote System Explorer (RSE) as a means of defining a set of connection attributes; on a per-connection basis. This removes the necessity to redefine the same attributes in each project's individual launch files.

SmartDSP OS launch files use the following naming convention for the systems it connects to: `<device name><single space character><board type>`. For example:

- MSC8156 EVM
- MSC8156 AMC
- MSC8157 ISS
- PSC9131 RDB

5 Known Issues

1. Due to a limitation in the MAPLE-B2F μ code, the demo `demos\starcore\psc9x3x\maple_pusch` performs direct write to the MAPLE eTVPE registers directly after disabling the MMU. This feature is expected to be added to the `maple_api` functionality provided by the μ code team.
2. Due to a limitation in the MAPLE-B2F μ code, the demo `demos\starcore\psc9x3x\maple_pdsch` runs only a single BD and does not verify external symbols
3. Due to a known bug (MTWX49475), the function `testReadParams()` in `demos\starcore\psc9x3x\maple_pusch\maple_pusch_demo.c` is compiled with no optimization in both the debug and release targets of the demo.

6 Resolutions and work-around

6.1 Level interrupt clearing

MSC815x, MSC825x and PSC913x have a known hardware issue with clearing level interrupts in a heavily congested system. Please refer to the SoC documentation for an in-depth description.

Towards solving this issue SmartDSP OS, starting at R02.03.00 introduced a new macro for clearing level interrupts – `CLEAR_LEVEL_INTERRUPT`. The behavior of this macro is dependent on a compilation flag which should be provided by the user. It has three possible modes:

`CLEAR_LEVEL_INTERRUPT_ULTRA_SAFE`: Extra cautious. This provides a 100% solution for all systems however it comes at the price of increasing the interrupt handling latency.

`CLEAR_LEVEL_INTERRUPT_SAFE`: Cautious. This provides a solution for most systems however it comes at the price of increasing the interrupt handling latency (less than the previous solution).

Backward Compatible (default): This provides a solution for systems that are not heavily congested, however provides no assurance as to the hardware behavior. It does, however, allow the parallelism between clearing the status register and execution of the interrupt handler.

This macro is provided in:

- SmartDSP\include\arch\starcore\msc815x\msc815x_hwi.h.
- SmartDSP\include\arch\starcore\psc9x3x\psc9x3x_hwi.h.

SmartDSP OS is preconfigured to the default behavior. Users wishing to change the default behavior should modify this file by unmarking one of the following lines:

```
//#define CLEAR_LEVEL_INTERRUPT_ULTRA_SAFE
//#define CLEAR_LEVEL_INTERRUPT_SAFE
```

6.2 MBF Net checksum()

The MBF version of the `checksum()` function (implemented in SmartDSP/source/net/net_ip_checksum.c) has been modified so that that it always runs in O0.

```
#ifndef FRAME_SINGLE_BUFFER_ONLY
#pragma opt_level = "O0"
#endif
```

This is temporary workaround that is implemented to provide a solution to a bug (MTWX42274) that is still under investigation. The ramification of this bug is that stale buffers reach the BIO Rx callback function (primarily `etherInput()`). As in the default setting of the Net stack release version data validity was not checked (i.e. `IP_VALIDATE_LENGTH` is set to `OFF` in SmartDSP/source/net/include/net_config.h); and as in such a case there is a discrepancy between the length actually received and indicated by the QE to what is in the (stale) cache; there are scenarios where the lack of this workaround can lead to MMU exceptions in accessing null pointers.

6.3 eMSG error

Due to the integration of the MSC8157 SoC, it is possible that the eMSG error interrupt be asserted while the register MUEDR is equal 0x0000000. This is indicative of an error on the sRIO bus. Users are advised to invoke the function `srioClearPortErrors()` if the eMSG error callback function is called with `status_register` equal to 0.

7 Bug Fixes

Issue	Abstract
MTWX49881	Missing syncio on DMA enable/freeze/defrost commands
MTWX49703	MAPLE-B2F eFTPE driver configures the hardware-reserved set #7
MTWX49420	Wrong assertion on MMU flexible size for large segments
MTWX49255	Error in FCM offset in eMSG driver
MTWX48885	CPRI Rx BFN callback calls the HFN callback
MTWX48683	sc3x00_system_mmu.c makes use of use of osGetCoreID() before it is initialized
MTWX48678	Error in supporting sRIO after boot over sRIO
MTWX48673	net ipsec debug target uses the release target appli file
MTWX48060	eMSG frames pool is depleted when working with MBF and MemPart buffers
MTWX48058	Data offset might be disregded in eMSG type9 inbound
MTWX48054	osFrameOffsetSet does not support offsets larger than 128bit
MTWX48054	osFrameOffsetSet() does not support offset larger than 128bit
MTWX48053	No automatic freeing of scatter/gather entries in the BMan-lite driver
MTWX47742	Incorrect "Build configuration" value from "Debug Configurations" setting in SDOS demos
MTWX47706	osHwiGpioIrqConfigure() not publically declared



8 Compatibility

- All libraries and applications were compiled and tested with CodeWarrior for StarCore 10.2.4, Build 197.
- SmartDSP OS B04.01.00 should be source compatible with SmartDSP OS R03.10.01 and A04.00.00 except for changed features (see New/Changed Features).

9 Deprecations

9.1 Deprecated in this release

9.1.1 OS version macros

SmartDSP OS B04.01.00 modified the version macros and they are now prefixed with `OS_VERSION_` (e.g. `OS_VERSION_MAJOR`) (defined in `source/common/include/os_version.h`).

Users are not expected to use these macros directly, rather through the function `osGetVersionInfo()` (defined in `include/common/os_general.h`)

9.1.2 DEPE_LTE_OBO

SmartDSP OS B04.01.00 declares the macro `DEPE_LTE_OBO()` as deprecated. This macro (defined in `drivers/maple/rev1/include/maple_depe.h`) is replaced by `DEPE_OBO()`, as the field is no longer LTE specific.

Users are called to make the transition to the new macro as the deprecated one will be removed in one of the future releases.

9.1.3 MAPLE MBUS map

SmartDSP OS B04.01.00 declares the field `maple_mbus_memmap_t.cpre` as deprecated and replaced by `maple_mbus_memmap_t.crpe`.

Users are called to make the transition to the new name as the deprecated one will be removed in one of the future releases.

9.1.4 PCIe configuration map

SmartDSP OS B04.01.00 declares the field `pci_configuraton_t.type_1_bridge_cotrol` as deprecated and replaced by `pci_configuraton_t.type_1_bridge_control`.

Users are called to make the transition to the new name as the deprecated one will be removed in one of the future releases.

9.2 Deprecated in previous release(s)

WDT macros

SmartDSP OS R03.10.01 declared the following new macros (defined in `include/common/os_hw_watchdogs.h`).

```
#define OS_WDT_AUTO_HANDLING    0
#define OS_WDT_USER_HANDLING    1
#define OS_WDT_DISABLED         2
```

They are replace the following deprecated macros (which are kept for backward compatibility):

```
#define DISABLED                 OS_WDT_DISABLED
#define AUTO_OS_HANDLING         OS_WDT_AUTO_HANDLING
#define MANUAL_USER_HANDLING     OS_WDT_USER_HANDLING
```

Users are called to change their applications to use the new macros, as the old ones may not be supported in future releases.

The reason for this deprecation is explained in MTWX45832 that postulates that `DISABLED` is too global a name for a macro.

rtcp_api.h

SmartDSP OS R03.10.01 removed the file `include\net\rtcp_api.h`. The entire content of this file is duplicated in `include\net\net_rtcp.h` with many additional API.

m815x_srio_dma_init.h

SmartDSP OS R03.10.01 removed the file `initialization\arch\m815x\include\m815x_srio_dma_init.h` as it is not supported.

Old os_frame API removal

SmartDSP OS R03.10.01 removed support for `os_frame` API that has been deprecated in previous releases (`include\common\os_frame.h`). This is done in order to keep code maintenance simpler.

Old os_mem API removal

SmartDSP OS R03.10.01 removed support for memory manager API that has been deprecated in previous releases (`include\common\os_mem.h`). This is done in order to keep code maintenance simpler.

osDmaChainDestroy removal

SmartDSP OS R03.10.01 removed support for `osDmaChainDestroy()` that has been deprecated in previous releases (`include\common\os_dma.h`). This is done in order to keep code maintenance simpler.

ip_reassemble_init

SmartDSP OS R03.10.01 declared the datatype `ip_reassemble_init` as deprecated (declared in `include\net\net_in.h`). It was replaced by `ip_reassemble_init`. Users are called to change their applications to use the datatype, as the old one may not be supported in future releases.

MAPLE debug libraries

SmartDSP OS 3.9.0 deprecated the debug versions of the MAPLE libraries (`lib\projects\maple`) as per the deprecation warning in the 3.8.1 release notes. Users are called to change their applications to use the release targets. This will provide smaller code footprint, faster initialization and forward compatibility.

MBF IPsec Net stack

SmartDSP OS 3.9.0 removed the multi-buffered-frame targets from the IPsec version of the net stack. There is no hardware support for this in the Security Engine and thus the binaries serve no purpose.

Debug No-Assert targets of the kernel(s)

SmartDSP OS 3.9.0 declared the `debug_noassert` targets of all supported architectures kernels' as deprecated.

Release 3.9.0 removed support for:

- `os_msc8x57_debug_noassert.elb`
- `os_msc8x57_debug_noassert_log.elb`

Future releases will remove support for the following binaries:

- `os_msc8156_debug_noassert.elb`
- `os_msc8156_debug_noassert_log.elb`
- `os_msc8154_debug_noassert.elb`
- `os_msc8154_debug_noassert_log.elb`
- `os_msc814x_debug_noassert.elb`

- os_msc814x_debug_noassert_log.elb

MMU_NEXT_LINE_PREFETCH

SmartDSP OS 3.8.1 declared `MMU_NEXT_LINE_PREFETCH` (defined in `include/arch/starcore/msc815x/msc815x_mmu.h`) as deprecated.

Users are encouraged to replace the old macro with the newer `MMU_PROG_NEXT_LINE_PREFETCH`.

OS appli files

SmartDSP OS 3.8.1 changed the names of the MSC815x application configuration files for the MSC815x kernel and drivers towards providing a clearer distinction between the MSC8156 and MSC8157 families of DSP.

TVPE extrinsic map

SmartDSP OS 3.8.1 declared the TVPE extrinsic memory map as deprecated.

crcpe_ring

SmartDSP OS 3.8.1 declared the union `crcpe_ring` (defined in `include/arch/peripherals/maple/rev1/maple_b_memmap.h`) as deprecated. It has been replaced by `crcpe_bd`.

DSP subsystem memory structures

SmartDSP OS 3.8.1 declared the architecture specific DSP subsystem memory map structure naming as deprecated. They will be replaced with an architecture independent prefix `dsp_plat`, which is portable across architectures. The new structures are defined in `include/arch/starcore/core_family/sc3X00/sc3x00_platform_memmap.h`.

MAPLE TVPE CRC Polynomial

SmartDSP OS 3.8.1 deprecated `maple_tvpe_crc_t` enumerations as per deprecation warning in the 3.7.0 release notes.

BIO Channel Callback

SmartDSP OS 3.8.1 changed BIO channel callback functionality for Rx channels as per deprecation warning in the 3.6.1 release notes.

The parameter `bio_ch_open_params_t.callback` is documented (in `SmartDSP/include/common/os_bio.h`) as follows:

```
/**< Application's callback for this channel;
    for Tx, the data is the confirmed frame; for Rx, data is always 0*/
```

Towards improving performance on BIO receive channels, SmartDSP OS will pass the received frame to the callback function in the callback's `data` parameter (similar to Tx channels) without enqueueing it in the BIO module.

osMalloc alignment on cacheable heaps

SmartDSP OS 3.8.1 removed the implicit the `ARCH_CACHE_LINE_SIZE` alignment that it placed on allocations from cacheable heaps (`OS_MEM_CACHEABLE_TYPE` encoded in `os_mem_type`) as per deprecation warning in the 3.6.1 release notes.

MSC815X_PEX_EP

SmartDSP OS 3.8.1 removed support for the `MSC815X_PEX_EP` (used in `os_config.h`) as per deprecation warning in the 3.3.0 release notes. It has been replaced by `MSC815X_PEX`.

os_malloc_file_hanlde

SmartDSP OS 3.8.1 removed support for the `os_malloc_file_hanlde` (defined in `SmartDSP/include/common/os_mem.h`) as per deprecation warning in the 3.2.2 release notes. It has been replaced by `os_malloc_file_handle`.

Default allocation for Background task

SmartDSP OS 3.8.1 removed support of the automatic allocation of a single task in `osTasksInitialize()` as per the deprecation warning in the 3.6.1 release notes. Users are called upon to make sure that `OS_TOTAL_NUM_OF_TASKS` (set in each application's `os_config.h` file) takes this into account and allocates enough tasks including the background task.

MAPLE TVPE CRC Polynomial

SmartDSP OS R03.07.00 declares the `maple_tvpe_crc_t` enumerations as deprecated. Users are encouraged to replace the older enumerations with the newer ones as the backward compatible definitions will be deprecated in future releases.

Heaps

SmartDSP OS 3.6.1 declares the specific heap files (e.g. `msc815x_shared_ddr0_mem.c`) as deprecated. They will be replaced by `msc815x_all_heap.c`. Users are encouraged to replace the older files with the new one in their applications and to use the new file for all future development.

osMalloc alignment on cacheable heaps

SmartDSP OS 3.6.1 warns against the future deprecation of implicit the `ARCH_CACHE_LINE_SIZE` alignment that it placed on allocations from cacheable heaps (`OS_MEM_CACHEABLE_TYPE` encoded in `os_mem_type`) deprecated. Users are called to use the `ALIGNABLE_SIZE(SIZE, ALIGNMENT)` and `ALIGN_ADDRESS(ADDRESS, ALIGNMENT)` macros (defined in `SmartDSP/include/common/os_mem.h`) to enforce alignment requirements.

BIO Channel Callback

SmartDSP OS 3.6.1 warns that a future change will be introduced into the BIO channel callback functionality for Rx channels.

The parameter `bio_ch_open_params_t.callback` is documented (in `SmartDSP/include/common/os_bio.h`) as follows:

```
/**< Application's callback for this channel;
    for Tx, the data is the confirmed frame; for Rx, data is always 0*/
```

Future releases, towards improving performance on BIO receive channels, SmartDSP OS may pass the received frame to the callback function in the callback's `data` parameter (similar to Tx channels).

Default allocation for Background task

SmartDSP OS 3.6.1 warns that the automatic allocation of a single task in `osTasksInitialize()` will be deprecated in future releases. Users are called upon to make sure that `OS_TOTAL_NUM_OF_TASKS` (set in each application's `os_config.h` file) takes this into account and allocates enough tasks including the background task.

MSC815X_PEX_EP

SmartDSP OS 3.3.0 declares `MSC815X_PEX_EP` (used in `os_config.h`) as deprecated. It will be replaced by `MSC815X_PEX`. Users are encouraged to `MSC815X_PEX_EP` in demos/applications that require PEX.

os_malloc_file_hanlde

SmartDSP OS 3.2.2 declares `os_malloc_file_hanlde` (defined in `SmartDSP/include/common/os_mem.h`) as deprecated. It will be replaced by `os_malloc_file_handle`. Users are encouraged to use `os_malloc_file_handle` where needed.

type_t structre

SmartDSP OS 3.2.2 removed the member's `tvpe_extrinsics_map` and `tvpe_map` from `tvpe_t` structure. These were not used and their removal saves memory space.

RIONET Tx flow level

SmartDSP OS 3.2.2 declares the default value of `out_msg_param.flow_level` (`SRIO_OMDATR_DTFLOWLVL_MEDIUM`) as deprecated. Users are encouraged to set the proper flow level values as future releases will not default.

tvpe_pointers

SmartDSP OS 3.2.2 declares the typedef `tvpe_pointers` (defined in `SmartDSP\drivers\maple\rev1\include\maple_tvpe.h`) as deprecated. Users are encouraged to move from `tvpe_pointers` to `maple_tvpe_pointers`. `tvpe_pointers` support will be discontinued in the future releases.

osFrameSecInfoGet() osFrameSecInfoSet()

SmartDSP OS 2.5.1 declares `osFrameSecInfoGet()` and `osFrameSecInfoSet()` (defined in `SmartDSP\include\common\os_frame.h`) as deprecated. These functions are no longer used as of 2.4.0 when the SEC driver was ported to the COP API.

osHwiGpioMap - replaces osHwiGetGPIO. defined in os_hwi.h

SmartDSP OS 2.4.2 declares the function `osHwiGetGPIO` (defined in `SmartDSP\include\common\os_hwi.h`) as deprecated. This function has been replaced by `osHwiGpioMap`.

msc814x_dma_attributes_config_t.dont_update

SmartDSP OS 2.4.0 declares the field `dont_update` in the structure `msc814x_dma_attributes_config_t` (defined in `SmartDSP/include/arch/starcore/msc814x/msc814x_dma.h`) as deprecated. The feature has been declared as not-supported by the hardware design team.

Interrupt Support

SmartDSP OS 2.4.0 declares the interrupts `OS_INT_M3_S_ECC_ERR`, `OS_INT_M3_D_ECC_ERR` and `OS_INT_L2_IC_ECC` (defined in `SmartDSP/include/arch/starcore/msc814x/msc814x_hwi.h`) as deprecated. These interrupts have been declared as not-supported by the hardware design team. Updated `SmartDSP\demos\starcore\msc814x\ecc_demo`.

MSC814x SEC driver BIO API

SmartDSP OS 2.4.0 ported the MSC814x SEC driver from the BIO to the COP API. The BIO SEC driver is deprecated as of this release.

Some e_address_type enumerations

SmartDSP OS 2.4.0 declares the `e_address_type` enumerations `OCEAN_36_BIT_ADDR` and `SRIO_34_BIT_ADDR_ID` (defined in `SmartDSP/include/arch/starcore/msc814x/msc814x_srio_dma.h`) as deprecated.

Users are encouraged to move to using the new enumerations (`OCN_DMA_ADDR_ATMU` and `OCN_DMA_ADDR_BYPASS`) respectively. Support for the old enumerations will be discontinued in the future releases.

Backward compatibility, in 2.4.0 is maintained by:

```
#define OCEAN_36_BIT_ADDR    OCN_DMA_ADDR_ATMU
#define SRIO_34_BIT_ADDR_ID OCN_DMA_ADDR_BYPASS
```

MSC814x.h Header File

SmartDSP OS 2.4.0 declares the msc814x.h file as deprecated. User are encouraged to move to including smartdsp_os_device.h file instead.

e_dma2ocn_win_sz

SmartDSP OS 2.3.0 declared the typedef e_dma2ocn_win_sz as deprecated. Users are encouraged to move from e_dma2ocn_win_sz to dma2ocn_win_sz_t. e_dma2ocn_win_sz support will be discontinued in the future releases.

msc814xDmaChannellsFrozen()

SmartDSP OS 2.3.0 declared msc814xDmaChannelIsFrozen as deprecated. Users are encouraged to move from msc814xDmaChannellsFrozen to msc814xDmaChannelFrozenStatus. msc814xDmaChannelIsFrozen() has been defined as:

```
#define msc814xDmaChannelIsFrozen(dma_channel)
(msc814xDmaChannelFrozenStatus(dma_channel) != NO_FREEZE)
```

ifAttachIC()

SmartDSP OS 2.2.1 declared ifAttachIC as deprecated. Users are encouraged to move from ifAttachIC to ifEnableIC. ifAttachIC support will be discontinued in the future releases.

Deprecated non c99 types

SmartDSP OS 2.2 declared non c99 types support as deprecated. Non-c99 support have been discontinued in 2.2.1.

Users who wish to continue the non-c99 types must uncomment the following line in os_datatypes.h:

```
/* #define USING_OLD_DATATYPES */
```

Deprecated udpFrameNew

SmartDSP OS 2.2 declared udpFrameNew API function as deprecated. Users are encouraged to move from udpFrameNew to udpFrameGet. udpFrameNew will be discontinued in the future releases.

Deprecated osFrameNew

SmartDSP OS 2.2 declared osFrameNew API function as deprecated as stated from version 1.3.5. Users are encouraged to move from osFrameNew to osFrameGet. osFrameNew support will be discontinued in future release.

Deprecated osFrameFree/osFrameDelete

SmartDSP OS 2.2 declared osFrameFree/osFrameDelete API function as deprecated as stated from version 1.3.5/2.2. Users are encouraged to move from osFrameFree/osFrameDelete to osFrameRelease. osFrameFree/osFrameDelete support will be discontinued in future release.

10 New/Changed Features Log

A03.10.00

- End of MSC8144 support
- New queue API
- os_accessors.h
- osHwilsCreated()
- Changes in os_frame_t
- COP channel callback
- MSC8157 clocks
- Updated SoC memory map files
- Hardware Timers
- Interrupt optimizations
- Updates to MAPLE-B/B2 driver
- Updates to MSC8157 eMSG driver
- Updates to MSC8157 CPRI driver
- MSC8156 doorbell timeout
- QE UEC IP fragmentation
- MSC8157 Debug Print support
- MSC8157 ADS BSP
- MSC8157 demos
- MSC8156 demos
- CommExpert 3.5.2
- Project Wizard
- MSC8157 EQPE model
- AN4256 Rev B

A03.09.00

- MSC8157 kernel
- Updated memory maps
- New queue API
- os_accessors.h
- osHwilsCreated()
- Hardware Timers
- GCR Interrupts Optimization
- MSC8x51 and MSC8x52 drivers
- MAPLE-B2 driver
 - MAPLE-B2 μ code rev05
 - MAPLE-B2 eTVPE driver
 - MAPLE-B2 eFTPE driver
 - MAPLE-B2 eFTPE data size sets
 - MAPLE-B2 CRCPE driver
 - MAPLE-B2 DEPE driver
 - MAPLE-B2 EQPE driver
 - MAPLE-B2 priority schemes
 - Unified PE initialization code
 - Channel open parameters
- HSSI support
- MSC8157 eMSG driver
- MSC8157 ADS BSP
- Asymmetric memory map
- Basic demo
- DMA demo
- eMSG type 10 demo



- eMSG type 11 demo
- Hardware timers demo
- I2C EEPROM demo
- I2C scan demo
- Intercore messages demo
- Intercore queues demo
- L2 cache demo
- MAPLE-B2 CRCPE demo
- MAPLE-B2 DEPE demo
- MAPLE-B2 EQPE demo
- MAPLE-B2 eTVPE Turbo demo
- MAPLE-B2 eTVPE WiMax demo
- MAPLE-B2 eFTPE demo
- Multitask demo
- Net demo
- OCeaN DMA demo
- SPI EEPROM demo
- SPI flash demo
- SPI flash demo
- UEC multicore demo
- DEPE Golden Model
- EQPE Golden model
- AN4256 Rev A

R03.08.01

- MSC8157 kernel
- Updated memory maps
- New frame API
- OS_TASK_EXPIRED
- Common cache sweep
- Asymmetric linker support
- MSC8157 drivers
- MAPLE-B driver
- MAPLE-B2 driver
- MAPLE-B/B2 driver
- Debug Print Driver
- HSSI digital loopback
- MSC8157 eMSG driver
- IPv6 stack
- etherInput() change
- MAPLE-B2 eTVPE Turbo demo
- MAPLE-B2 eTVPE WiMax demo
- MAPLE-B2 DEPE demo
- MAPLE-B2 EQPE demo
- MAPLE-B2 eFTPE demo
- IPv6 demos
- MSC8156 AMC BSP
- MSC8156 AMC basic demo
- MSC8156 AMC I²C EEPROM demo
- MSC8156 AMC UEC demo
- MSC8156 basic and C++ demos
- MSC8156 QE SPI demo
- MSC8156 asymmetric memory map
- DEPE Golden Model



- EQPE Golden model
- CommExpert changes
- CodeWarrior MSC8156 AMC wizard
- AN3679 Rev D

B03.07.02

- MSC8156 EVM wizard support
- MSC8156 EVM BSP
- MSC8156 EVM leds demo
- MSC8156 EVM I²C demo
- MSC8156 EVM sRIO DMA demo

R03.07.00

- osMallocAligned()
- os_cpp.h in public header files
- Changes to Multi-Buffered-Frames
- osTaskNameGet()
- Interrupts during initialization phase
- MSC815x L2 cache prefetch
- MAPLE TVPE dynamic switching
- Maple CRC polynomial enumerations
- MAPLE µcode rev38
- DMA interrupt API
- UEC MAC address
- RapidIO™ doorbell ISR
- µcode support for Debug Print and RIONET
- PCIe MSI
- IP reassembly
- ICMP error payload
- Debug Hook demo
- MSC815x PEX MSI demo
- net_mbf demos
- MAPLE Viterbi demo
- SmartDSP HEAT server
- CommExpert Kernel Awareness

R03.06.01

- Improved Interrupt Dispatcher
- Unification of MSC81/25x heaps
- MSC825x derivatives
- CORESUB_10
- MSC815x UEC Scheduler API
- MAPLE µcode rev37
- TVPE driver updates
- Debug Print Changes
- MSC815x DMA DREQ
- RIONET Fixed Queue and Size
- User RIONET Tx structure
- QE_ENET10
- SRIO7
- MSC815x Basic/C++ demos
- MSC815x net_tftp demo
- MSC815x DMA demo
- MSC8144 MJPEG demo
- MSC8156 MJPEG demo
- MSC815x RIONET demo



- New Project Wizard
- AN3679 Rec C

R03.05.00

- Updated patch layout
- bool typedef change
- Removal of implicit conversions
- OS_WAIT macro
- Spinlock algorithm
- C++ Support
- Virtual Interrupt API
- New MSC815x heap
- MSC815x L2 cache initialization
- MSC815x asynchronous cache API
- MSC815x Optimized cache sweep
- MSC815x updated memory map
- OCeaN DMA continuous channel mode
- OCeaN DMA dynamic channel allocation
- Support MSC815x user RIONET transfer callbacks
- MSC815x changes to sRIO Alternate ID and Accept All
- Updated MAPLE memory map
- MAPLE µcode directory structuring
- MAPLE memory allocation change
- Updated initialization structures
- QE_ENET21 erratum support
- MAPLE µcode rev37
- New µcode package for Debug Prints
- IP multicast support
- os_config.h template
- C++ demo
- Net Multicast demo
- MSBA8100 TVPE doorbell demo
- MSC815x sRIO BSP
- MSC815x basic demo
- MSC815x RIONET demo
- Updated MSC815x sRIO demos
- SmartDSP HEAT client
- SmartDSP HEAT server
- CommExpert
- Documentation plugins

R03.04.00

- Added os_spinlock_handle typedef
- MSC815x Rev2 kernel support
- Support for different M2/L2 splits across cores
- Interrupt support in single OCeaN DMA transfers
- Configurable number of QUICC Engine™ filters
- New µcode package for Debug Prints
- MSC815x hardware timer demo
- Updated demo README files
- SmartDSP HEAT server to enable reopening an open file handle
- SmartDSP OS Concepts Guide revision
- SmartDSP OS API Reference Manual revision

R03.03.00

- osSysInfoGet() provides software context
- Fast spinlock API
- osHwiMultipleDelete()
- PEX Root Compex Driver
- Ethernet Lossless Flow Control
- MAPLE TVPE new features
- MAPLE μ code rev35
- Added option to modify all attributes of DMA buffer
- Change to PEX inbound ATMU window configuration
- New μ code package for Debug Prints
- MSC815x SRTP/SRTCP
- MSC815x Net Pattern demo
- MSC815x SRTP demo
- MSC815x Debug Print μ code demo
- MSC815x PEX RC demo
- sc3000-ld support

B03.02.02

- MSC815x interrupt dispatcher performance improvement
- Cache performance improvements
- COP API performance improvements
- MSC815x local non-cacheable heaps in M3/DDR0/DDR1
- osTickTime()
- New cache API
- MSC8154 kernel libraries
- MAPLE performance improvement
- MAPLE profiling
- MAPLE CRCPE driver
- MAPLE μ code rev33
- New RIONET μ code
- UEC SGMII port enumeration
- MSC8154 driver libraries
- Security API
- MSC815x IPSec
- MSC815x eCLI
- MSC815x intercore queue demo
- MSC815x MAPLE CRCPE demo
- MSC815x MAPLE TVPE performance demo
- MSC815x IPSec demo
- MSC815x Shell (eCLI) demo

B03.02.00

- Wrapper for reading the Device ID
- NMI executed on precise interrupt dispatcher
- RIONET flexible heap for Tx
- Improved performance in OCeaN
- MAPLE μ code rev30
- eCLI
- MSC8144 eCLI

B03.01.01



- MSC815x full kernel
- Kernel Awareness
- Class profiling
- MSC815x full driver suite
- MSC815x ADS BSP
- MSC815x demos

A03.01.00

- MMU and cache features
- QE support
- DMA support
- I2C support
- Intercore messages support
- MAPLE support
- MSC815x ADS BSP

B02.05.01

- COP device control options
- MAPLE µcode rev29
- MAPLE direct driver
- sRIO DMA driver optimization
- UDP/IP forwarding
- IP header options support
- MAPLE TVPE demo
- MAPLE FFTPE demo

R02.05.00

- COP statistics
- Individual task PID/DID
- MAPLE µcode rev26
- SmartDSP HEAT

R02.04.02

- Debug Hooks
- Print osMalloc memory consumption
- Better support for Ethernet Jumbo Frames
- Reduced queue size for MSC814x UEC driver
- MAPLE µcode rev25
- MAPLE PE hardware semaphore lock ID
- MAPLE sRIO controller base address
- TVPE CRC polynom from register to BD
- New sRIO error recovery code
- New RIONET µcode
- MAPLE driver header file relocation
- Doxygen documentaiton clearer
- New MAPLE doorbell support
- New MAPLE iFFT gaurd bands demo

R02.04.00

- Changed os_mem_type enumeration
- osMalloc changed to osSpinLockIrqXxx from osSpinLockXxx
- COP API
- Support for BIO underrun.
- Added local M3 DDR heaps
- Multi-Core Scheduler (MCS) for MSC814x
- Support for MSBA8100 with Rev13 of MAPLE u-code
- Hierachial sRIO bringup
- MSC814x flexible heaps in channel open
- Better flow control support
- MSBA8100ADS BSP
- QE timestamp support
- Support for static entries in the ARP table
- MSBA8100 Demos
- MSC814x MSC demo

R02.03.00 + R02.03.01

- Hardware timers ISR argument
- CIO channel Tx returns value
- Frame reference counter
- Kernel awareness stack
- MSC814x improvement in MMU address translation performance
- Removed mandatory TCP callback
- Change in SRTP context creation API
- MSC814x SEC driver fixes
- MSC814x DMA driver updates
- MSC814x sRIO DMA driver updates
- MSC814x sRIO doorbell initialization updates
- MSC814x RIONET priority support
- Dynamic Kernel Awareness Stack demo
- SRTP demo
- MSC814x support for RTLIB file I/O

B02.02.01

- Intercore message queues support
- Clear pending edge hardware interrupt
- Support for hardware timer interval update
- MSC814x shared cacheable M2 heap
- Queue enhancements
- Software timer enhancements
- Task event enhancements
- Sharing NET information between cores (ARP)
- RTCP Support
- SRTP/SRTCP Support
- MSC814x CLASS profiling support
- MSC814x Performance Monitor support
- MSC814x Kernel Awareness, System Performance
- Shell support
- MJPEG demo
- RTP demo

B02.02.00

- ROM and BSS section support
- MSC814x multiple shared heap support
- TCP support
- IPSec support
- MSC814x SEC support
- MSC814x SRIO DMA support
- MSC814x UEC multi-buffered frame support
- MSC8144ADS and Tundra STX board connectivity

B02.01.01

- New Buffer/Frame LLD offset support scheme
- MSC814x SRIO doorbell support
- MSC814x Ethernet over SRIO support
- MSC814x ADS BSP, Pilot revision

B02.01.00

- MSC814x QE added support ATM - AAL0,AAL2,AAL5
- MSC814x QE added support Ethernet - SGMII mode
- MSC814x added RIONET with QE uCode module
- MSC814x ECC support added
- MSC814xADS support added STMicroelectronics SPI flash
- MSC814xADS support added Marvell SGMII ethernet switch
- Kernel Awareness on 814x/812x/8101/711x

B02.00.00

- MSC8144 MMU and Cache support
- MSC8144 QE support (Ethernet and SPI)
- MSC8144 DMA Support
- MSC8144 SRIO and RIONET for single core
- MSC8144 TDM
- MSC8144 I2C and UART
- MSC8144ADS BSP : DS26521 E1/T1 framer, Vetesse VSC7380 Ethernet switch, Marvell PHY, Codec driver, Leds driver, BCSR driver
- Generic SPI and I2C drivers
- Note on OS_SYSTEM_CLOCK definition: The OS_SYSTEM_CLOCK is not taken into account by MSC8144 SmartDSP.
- Note on ISR Stack size : ISR stack size is defined in os_config.h and not in the linker file

B01.04.02

- VLAN support added to NET module. VLAN_SUPPORT is defined in the file net_config_.h.
- The default setting is ON for the Debug version of the NET module, and OFF for the Release version of the NET module.
- MSC711x memory map updated.

B01.04.01

- Watchdog timers support added to MSC711X

- MSC711X EVM support added
- TFTP example added.
- MIB and Statistics support added
- Multicore ARP (ARP Proxy agent) added
- MSC8122 DMA driver - msc8122_dma_init_params_t device initialization structure was added to determine arbitration mode. Default scheme is round robin which did not exist in MSC8102.
- MSC711x DMA driver - removed the link_ch parameter from structure msc711x_dma_transfer_config_t.

B01.04.00

- DHCP and RTP modules added to NET.
- Configuration Tool added.
- Flash Boot Loader Tool added.
- SIO module - rx_callback_parameter and tx_callback_parameter fields were added to sio_dev_open_params_t. Now if these parameters are not NULL, SIO calls the application once per SIO device.
- MSC711x TDM - only_interleave was added to msc711x_tdm_init_params_t structure. If TRUE, no de-interleaving is performed.
- MSC711x TDM - DE_INTERLEAVING_WITH_DMA_GROUP_1 added as optional compilation flag to os_msc711x_drivers.mcp, to enable placing de-interleaving DMA channels in group 1. This flag is recommended only if you want to use more than 16 DMA channels and need to use group priority. It is introduced because of the errata that prevents using channel fixed priority with group fixed priority.
- DMA module - osDmaChainTransferAddEx function was added to enable access to a specific transfer in a DMA chain.
- DMA module - osDmaChainReset function was added to enable deleting all transfers from a chain without deleting the chain.
- DMA module - osDmaTransferDestinationSet was added to enable setting a destination of a transfer that was added by osDmaChainTransferAddEx.
- DMA module - osDmaTransferSourceSet was added to enable setting a source of a transfer that was added by osDmaChainTransferAddEx.
- DMA module - osDmaTransferSizeSet was added to enable setting a size of a transfer that was added by osDmaChainTransferAddEx.
- MSC711x DMA – Bandwidth control field (bwc) of type dma_bwc_t was added to msc711x_dma_transfer_config_t. possible values:
- DMA_NO_STALLS, DMA_DYNAMIC_PRIORITY_ELEVATION, DMA_4_STALLS, DMA_8_STALLS. Note that using
- DMA_DYNAMIC_PRIORITY_ELEVATION might degrade the core performance.
- MSC8122 TSEC driver - removed the channel_num parameter from struct tsec_channel_params_t and the relevant demos: tsec, net and test_net.
- MSC711x FEC driver - removed the channel_num parameter from struct fec_channel_params_t and the relevant demos: fec, net, net_rtp and test_net.
- MSC711x UART driver - Added interrupt priority parameter to struct msc711x_uart_init_params_t and the relevant demos and tests : uart demo, uart test and system test.

R01.03.05

- Full MSC711x support.
- TDM, UART, DMA, HDI and Ethernet (FEC) drivers for MSC711x.

- SIO module introduced, for synchronized I/O (TDM). TDM over CIO is still included, but will be removed in future versions.
- Some OS objects and functions were deprecated. Please see the “Deprecated List” in the manual, for details about changed/removed objects and functions.
- Error codes (in `os_error.h`) were changed, to provide more information than before.
- User’s hooks are supported through the OS Log functionality.
- New demonstration applications were added for MSC8101 and MSC8102.
- SmartDSP OS Reference Manual was updated.
- Documents for the various drivers and demos were updated. The documents can be found in the “doc” directory.
- Important: ISR vector resides at address 0x0. Linker files were changed appropriately.
- Important: All StarCore applications must define the compiler flag
- STARCORE for correct compilation of the code.
- Please read the notes above, explaining how to connect to MSC711x ADS and MSC8122 ADS.

B01.03.00

- MSC8102 libraries now support MSC8122.
- Added Ethernet (TSEC) driver for MSC8122.
- Added a network library, containing a UDP/IP stack (including ARP and ICMP) and an MII module.
- The BIO layer handles Tx/Rx error status from LLD.
- The Frames API contains header/offset manipulation.
- Basic datatypes were changed to the C99 standard. Old datatypes are still defined, for backward compatibility only. To prevent old types from clashing with external datatype definitions, you can uncomment the line that defines `USING_OLD_DATATYPES` in `os_datatypes.h`.
- ISR vector was moved to address 0x0, (unlike address 0x1000 in previous versions).
- Optimization levels are now `-O3` for runtime code and `-O3 -Os` for initialization code.
- SmartDSP OS Reference Manual was updated.
- Important: All StarCore applications need to define the compiler flag STARCORE for correct compilation of the code.

R01.02.00

- BIO module added.
- Frames added.
- Queues added.
- DSI BIO driver added.
- `osHwiSwiftEnable` and `osHwiSwiftDisable` API introduced.
- `osSharedResourceLock` and `osSharedResourceUnlock` API introduced.
- `osMemBlockUnsafeGet` and `osMemBlockUnsafeFree` API introduced.
- `OS_ASSERT_COND`, `VAR_UNUSED` macros added.
- Logging capability was added to MSC8101, with appropriate libraries.
- Refined the Memory Management documentation, to contain some needed macros.
- Important: All StarCore applications need to define the compiler flag STARCORE for correct compilation of the code.

R01.01.00

- Initialization method was changed.
- Some types’ names were changed (`_t` was added):

```

os_version      --> os_version_t
os_mem_part     --> os_mem_part_t
os_barrier      --> os_barrier_t

```

- Some functions' names were changed:

```

osHwiInstallVector --> osHwiVectorInstall
osHwiSetPriority   --> osHwiPrioritySet
osSwiSetPriority   --> osSwiPrioritySet
osSwiGetPriority   --> osSwiPriorityGet
osSwiSetCount     --> osSwiCountSet
osSwiGetCount     --> osSwiCountGet
osSwiIncCount     --> osSwiCountInc
osSwiDecCount     --> osSwiCountDec
osSwiGetStatus    --> osSwiStatus

```

- Directory structure was changed (see new directory structure in SmartDSP OS Reference Manual chapter 2).
- CodeWarrior default stack is used instead of a user defined stack.
- CIO and DMA modules introduced.
- TDM, UART, DMA and DSI drivers for MSC8102.
- Logging capability was added to MSC8102.
- Documentation updated (added documentation for UART and TDM).
- CodeWarrior Stationery projects were added. In order to use it:

1. Copy the contents of cw_stationary directory to your CodeWarrior installation under Stationery\StarCore\SmartDSP_OS.
2. Copy SmartDSP directory to your CodeWarrior installation under "(CodeWarrior_Examples)\StarCore_Examples".
3. Choose File->New->StarCore Stationery->SmartDSP_OS.