

Migrating QNX[®] Momentics[®] Development Suite 6.3.0 Board Support Packages (BSPs) to QNX Momentics 6.3.0 SP1 or SP2

Document Type:
Document:

Technical Note
Content Extracted from QMS8402

Introduction

This document describes how to install, compile, and run a BSP originally designed for QNX Momentics 6.3.0 into a QNX Momentics 6.3.0 SP1 or SP2 environment.

Overview

Binaries included in a 6.3.0 BSP are fully compatible with a 6.3.0 SP1 or SP2 environment, which means that you need only rebuild your bootable image (i.e. OS image) on a 6.3.0 SP1 or SP2 host to run on your target.

The only changes that you need to make to migrate the 6.3.0 BSP source code to 6.3.0 SP1 or SP2 are in the serial driver source code.

The remainder of this document explains:

- How to create the OS image based on existing 6.3.0 BSP binaries.
- How to modify the serial driver source code to be compatible with SP1 and SP2.

Note: The Character DDK for 6.3.0 SP1 includes a fully modified serial driver (i.e. *devc-ser8250*) as an example.

Note: Once you've migrated a 6.3.0 BSP to 6.3.0 SP1, it's forward compatible with 6.3.0 SP2 and requires no further source code changes for SP2. However, BSPs designated as being qualified only against 6.3.0 SP2 are not necessarily backward compatible to previous service pack levels, as they may be dependent on other SP2 components.

Migrating an unmodified BSP from QNX Momentics 6.3.0 to 6.3.0 SP1 or SP2

This section explains how to install, compile, and run an untouched 6.3.0 BSP in a 6.3.0 SP1 or SP2 environment.

1. Using the precompiled binaries provided with the BSP:

Binaries provided in a 6.3.0 BSP are fully compatible with the 6.3.0 SP1 or SP2 environment. Therefore if you have existing 6.3.0 BSP binaries (including the serial driver binary), you can use them to create an 6.3.0 SP1 or SP2 OS image on your SP1 or SP2 host environment, as follows:

- Install the 6.3.0 BSP binary package on a 6.3.0 SP1 or SP2 host as you would on a 6.3.0 system.
- Generate the OS Image:

a- From the command line:

Run `mkifs` to generate the OS image based on the buildfile found in `$QNX_TARGET/CPU/boot/build/*.build`.

For example, on a MS Windows XP host:

```
cd $QNX_TARGET\ppcbe\boot\build
mkifs -v 85x0ads.8540.build 85x0ads.8540.ifs
```

b- From the QNX Momentics IDE:

Import the buildfile using the QNX System Builder:

- Choose **File->New->Project**.
- Select **QNX System Builder Project** and click **Next**.
- Enter a project name (e.g. buildfile) and click **Next**.
- Select **Import Existing Buildfile** and click **Next**.
- Enter the buildfile name (e.g. `$QNX_TARGET/ppcbe/boot/build/85x0ads.8540.build`) or click **Browse** to find it.
- Click **Finish**.
- In the Navigator view of the QNX System Builder perspective, expand the project (i.e. buildfile). The OS image (i.e. `85x0ads.8540.ifs`) is now built and located in the Images directory.

Note: On a MS Windows XP host, for example, the OS image could also be in:

```
$HOME\workspace\buildfile\Images\85x0ads.8540.ifs
```

- Download the resulting OS image to the target hardware, as you would do from a 6.3.0 host. See the BSP documentation for details.

2. Modifying the BSP source code:

If you've written a character-device driver that links against the `libio-char.a` library, note that there has been a change in the API to accommodate power management. You need to update any driver source code originally written for 6.3.0 and earlier if you plan to compile it against the new library in 6.3.0 SP1 or SP2.

Note: All serial drivers available in the 6.3.0 SP1 or SP2 core product, the 6.3.0 SP1 Character DDK, and the 6.3.0 SP1 or SP2 BSPs have already been modified.

Here's how to modify the serial driver source:

- Install the 6.3.0 BSP package on a 6.3.0 SP1 or SP2 host, as you would on a 6.3.0 system.

a- Using the command-line tools:

If you're using the command line tools, you'll need to extract the BSP source code contained in a ZIP file in order to make the following changes.

In this document we'll assume that you've installed the Freescale MPC85x0 ADS BSP, and then extracted:

```
$QNX_TARGET\usr\src\archives\qnx\85x0ads.zip
```

into:

```
$QNX_TARGET\usr\src\archives\qnx\85x0ads\
```

b- Using the IDE:

If you're using the QNX Momentics IDE, simply import the BSP, using the QNX System Builder Project (refer to the *IDE User's Guide* for more details).

- Modify the serial driver source code:
The `ttc()` function call with the `TTC_INIT_ATTACH` parameter originally created the device's name, based on the unit number passed in and then attached the driver's resource-manager interface:

```
ttc(TTC_INIT_ATTACH, &tty, unit);
```

This call has been split up into:

```
ttc(TTC_INIT_TTYNAME, &tty, unit);
ttc(TTC_INIT_POWER, &tty, 0);
ttc(TTC_INIT_ATTACH, &tty, 0);
```

You need to call `ttc()` with these arguments in this order, but not necessarily in direct sequence:

- `TTC_INIT_TTYNAME` to create the device's name, based on the unit number passed in.
- `TTC_INIT_POWER` to initialize the data structures related to power management. Even if your driver doesn't implement

power management, you must still make the call with `TTC_INIT_POWER`.

- `TTY_INIT_ATTACH` to attach the driver's resource-manager interface.

Note: You must make the calls with `TTC_INIT_TTYNAME` and `TTC_INIT_POWER` before you attach any interrupt handlers.

Here's the code for the 8250 serial driver (i.e. `devc-ser8250`) before you change `init.c`:

```
create_device()
{
    ...
    // Initialize termios cc codes to an ANSI terminal.
    ttc(TTC_INIT_CC, &dev->tty, 0);

    ...

    // Set up the IRQ handler
    ser_stty(dev);
    ser_attach_intr(dev);

    ...

    // Assume that the basename is set in the device name.
    // This will attach to the path assigned by the unit
    // number/minor number combination
    unit = SET_NAME_NUMBER(unit) | NUMBER_DEV_FROM_USER;
    ttc(TTC_INIT_ATTACH, &dev->tty, unit);
}
```

Here's the code after the changes to `init.c`:

```
create_device()
{
    ...
    // Initialize termios cc codes to an ANSI terminal.
    ttc(TTC_INIT_CC, &dev->tty, 0);

    // Initialize the device's name.
    // Assume that the basename is set in device
    // name. This will attach to the path assigned by the
    // unit number/minor number combination
    unit = SET_NAME_NUMBER(unit) | NUMBER_DEV_FROM_USER;
    ttc(TTC_INIT_TTYNAME, &dev->tty, unit);

    // Initialize power management structures before
    // attaching ISR
    ttc(TTC_INIT_POWER, &dev->tty, 0);

    ...

    // Set up the IRQ handler
    ser_stty(dev);
    ser_attach_intr(dev);
```

```

...
// Attach the resource manager
ttc(TTC_INIT_ATTACH, &dev->tty, 0);
}

```

To take advantage of the software flow control updates (Ref# 21272), you must update your custom serial driver's `tto.c` file to check if the `OSW_PAGED_OVERRIDE` flag is asserted. If this flag is asserted, allow the transmission of one (and only one) character, and then clear this flag. This allows the transmission of software flow control characters when `io-char` is in a suspended state.

For example:

```

if(dev->flags & (OHW_PAGED|OSW_PAGED) && !(dev->xflags
& OSW_PAGED_OVERRIDE))
return(0);
.
.
.
write_8250(port[REG_TX], c);
/* Clear the OSW_PAGED_OVERRIDE flag as we only want
* one character to be transmitted in this case.
*/
if (dev->tty.xflags & OSW_PAGED_OVERRIDE)
atomic_clr(&dev->tty.xflags, OSW_PAGED_OVERRIDE);

```

For more information, see the description of the `TTYDEV` structure in the Character DDK documentation.

- Modify the serial driver's Makefile.

You must also edit the `common.mk` file from the root of the serial driver source directory. Using the previous BSP example:

a- From the command line:

- `cd $QNX_TARGET\usr\src\archives\qnx\85x0ads\`
- `cd src\hardware\devc`
- Edit the `common.mk` from this directory.
- Change the setting of `LIBS` to this:

```
LIBS=io-char pm ps
```

- Save and close `common.mk`.

c- From the QNX Momentics IDE:

- From the Navigator view of the QNX System Builder perspective, expand the 8250 serial driver source code (i.e. `bsp-mpc85x0ads_devc-ser8250`).
 - Double-click on `common.mk` in order to open it in the editor.
 - Change the setting of `LIBS` to this:
- ```
LIBS=io-char pm ps
```
- Save and close `common.mk`.

**Note:** The `common.mk` could also be found in:

`$HOME\workspace\bsp-mpc85x0ads_devc-ser8250\`

- Recompile the serial driver source code.
- Rebuild the entire BSP in order to include the new serial driver (see the BSP documentation for more details).
- Download the new OS image to the board (see BSP documentation for more details).