

AC426
Application Note
Implementing Production Release Mode Programming
for SmartFusion2



a  **MICROCHIP** company



a  MICROCHIP company

Microsemi Headquarters

One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113

Outside the USA: +1 (949) 380-6100

Sales: +1 (949) 380-6136

Fax: +1 (949) 215-4996

Email: sales.support@microsemi.com

www.microsemi.com

©2021 Microsemi, a wholly owned subsidiary of Microchip Technology Inc. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

Contents

1	Revision History	1
1.1	Revision 6.0	1
1.2	Revision 5.0	1
1.3	Revision 4.0	1
1.4	Revision 3.1	1
1.5	Revision 2.0	1
1.6	Revision 1.0	1
2	Implementing Production Release Mode Programming for SmartFusion2	2
2.1	Purpose	2
2.2	Introduction	2
2.3	References	2
2.4	Design Requirements	3
2.5	Prerequisites	3
2.6	Building Executable Image in Release Mode	4
2.7	Loading the Executable Image into eNVM	10
2.8	Updating eNVM Memory Content	17
2.9	Running the Design on SmartFusion2 Security Evaluation Kit	19
2.10	Building Multiple Executable Images in Release Mode	20
2.11	Loading Multiple Executable Images into eNVM	26
2.12	Updating eNVM Memory Content	32
2.13	Running the Design on SmartFusion2 Security Evaluation Kit	34
2.14	Conclusion	36
3	Appendix 1: Programming the Device Using FlashPro Express	37
4	Appendix 2: Creating LookUpTable.Hex File	40

Figures

Figure 1	SoftConsole IDE	4
Figure 2	SoftConsole IDE - Import	4
Figure 3	SoftConsole IDE - Importing Existing Projects	5
Figure 4	SoftConsole IDE - Browse	5
Figure 5	SoftConsole IDE - Browsing for Projects Folder	6
Figure 6	SoftConsole IDE - Creating Work Space in App Folder	7
Figure 7	SoftConsole Workspace	7
Figure 8	Building Configurations	8
Figure 9	Build All	8
Figure 10	Generating Hex File	9
Figure 11	Hex File in Release Folder	9
Figure 12	eNVM_DataClient_top Tab	10
Figure 13	System Builder - Memories Page	11
Figure 14	Add Data Storage Client Window	11
Figure 15	Add Data Storage Client Window	12
Figure 16	Executable Image in Release Folder	12
Figure 17	Add Data Storage Client Window	13
Figure 18	LookUpTable.hex File	14
Figure 19	Add Data Storage Client Window	14
Figure 20	System Builders - Memories Page	15
Figure 21	Selecting Run PROGRAM Action	16
Figure 22	Programming SmartFusion2 Security Evaluation Kit	16
Figure 23	Choosing Update eNVM Memory Content Option from Libero Design Flow	17
Figure 24	eNVM Memory Content Window	18
Figure 25	Updated Hex File in Release Folder	18
Figure 26	Device Manager Window	19
Figure 27	Selecting Factorial of a Number - User Options	20
Figure 28	Displaying Factorial Value Based on User Option	20
Figure 29	SoftConsole IDE	21
Figure 30	SoftConsole IDE - Import	21
Figure 31	SoftConsole IDE - General	22
Figure 32	SoftConsole IDE - Browse	22
Figure 33	SoftConsole IDE - Browsing for App Folder	23
Figure 34	SoftConsole IDE - Creating Work Space in App Folder	23
Figure 35	SoftConsole Workspace	24
Figure 36	Building Configuration	24
Figure 37	Build All	25
Figure 38	Remapping_Appnote_top Tab	26
Figure 39	System Builder - Memories Page	27
Figure 40	Add to Data Storage Client Window	27
Figure 41	Add to Data Storage Client Window	28
Figure 42	Executable Image in Release Folder	28
Figure 43	Add Data Storage Client Window	29
Figure 44	Error due to Image Overlap	30
Figure 45	Hex Files with Correct Start Addresses	30
Figure 46	Regenerating Remapping_Appnote_0 Component	31
Figure 47	Programming SmartFusion2 Security Evaluation Kit	32
Figure 48	Update eNVM Memory Content Option	33
Figure 49	Update eNVM Memory Content Window	33
Figure 50	Browsing through Updated Hex File	34
Figure 51	Device Manager Window	35
Figure 52	Running Hex File - User Options	35
Figure 53	LEDs Blink Based on User Option	36
Figure 54	FlashPro Express Job Project	37

Figure 55	New Job Project from FlashPro Express Job	38
Figure 56	Programming the Device	38
Figure 57	FlashPro Express—RUN PASSED	39
Figure 58	Entering Factorial Values in Notepad	40
Figure 59	Saving Notepad	40
Figure 60	Converting Binary File to Hex File	40
Figure 61	LookUpTable.Hex File	41

Table

Table 1	Design Requirements	3
Table 2	SoftConsole Application Projects and their Mapping to eNVM Client	29

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 6.0

The following is a summary of the changes made in this revision.

- Updated the document for Libero SoC v2021.1.
- Removed the references to Libero version numbers.

1.2 Revision 5.0

Updated the document for Libero v11.7 software release (SAR 77926).

1.3 Revision 4.0

Updated the document for Libero v11.6 software release (SAR 72112).

1.4 Revision 3.1

Updated the document for Libero v11.5 software release (SAR 65408).

1.5 Revision 2.0

Added the bin2hex.exe file to the design files and added a link to design files in the [Appendix 2: Creating LookUpTable.Hex File](#), page 40 (SAR 63426).

1.6 Revision 1.0

The first publication of this document.

2 Implementing Production Release Mode Programming for SmartFusion2

2.1 Purpose

This application note demonstrates how to build single and multiple SoftConsole application executable images in the production release mode, store them into embedded non-volatile memory (eNVM) using eNVM data storage client, and execute them on SmartFusion[®]2 System-on-Chip (SoC) Field Programmable Gate Array (FPGA) Security Evaluation Kit.

2.2 Introduction

SmartFusion2 devices have a maximum of two on-chip 256 KB flash memories called eNVM. The eNVM is used to store the application code executable image or store data, which is used by the user application. For more information about eNVM, refer to the eNVM chapter of the *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*.

The user application projects are built in Release/Debug mode for execution on the SmartFusion2 Security Evaluation Kit through SoftConsole. In the Debug mode, the complete symbolic debug information is emitted to the debugging applications, and the code is not optimized. In the Release mode, the complete symbolic debug information is not emitted, and the code execution is optimized. The size of the final release mode executable is smaller than the debug executable. It also runs faster than the debug mode.

This application note describes the following:

- How to build a single SoftConsole application executable image in production release mode, store the image into eNVM using eNVM data storage client, and execute the image directly from eNVM of the SmartFusion2 Security Evaluation Kit.

A sample SoftConsole application project fetches factorial of a number from a non-executable image (LookUpTable.hex) stored in eNVM. `LookUpTable.hex` file contains pre-calculated factorial values. For more information about how to create a `LookUpTable.Hex` file, refer to [Appendix 2: Creating LookUpTable.Hex File](#), page 40.

- How to build multiple SoftConsole application executable images in production release mode and store them into eNVM using eNVM data storage client.

The SoftConsole application projects show how the stored executable images in eNVM directly runs from eNVM, and how to remap the images to run from eSRAM. The SoftConsole application projects designed here are to blink LED on the SmartFusion2 Security Evaluation Kit.

To demonstrate the above two scenarios, two design folders with the SoftConsole application projects are provided with this application note, refer to [Appendix 2: Design Files](#), page 40.

2.3 References

The following list of references is used in this document:

- *UG0331: SmartFusion2 Microcontroller Subsystem User Guide*

2.4 Design Requirements

Table 1 lists the hardware and software design requirements for running this demo design.

Table 1 • Design Requirements

Requirement	Version
Operating system	64 bit Windows 7 and 10
Hardware	
SmartFusion2 Security Evaluation Kit:	Rev D or later
12 V adapter (provided along with the kit) - FlashPro4 programmer - USB A to Mini-B cable	
Host PC or Laptop	
Software	
FlashPro Express	Note: Refer to the <code>readme.txt</code> file provided in the design files for the software versions used with this reference design.
Libero [®] System-on-Chip (SoC)	
SoftConsole	
Host PC Drivers (provided along with the design files)	USB to UART drivers
One of the following serial terminal emulation programs: –	
- HyperTerminal - TeraTerm - PuTTY	

Note: Libero SmartDesign and configuration screen shots shown in this guide are for illustration purpose only. Open the Libero design to see the latest updates.

2.5 Prerequisites

Before you begin:

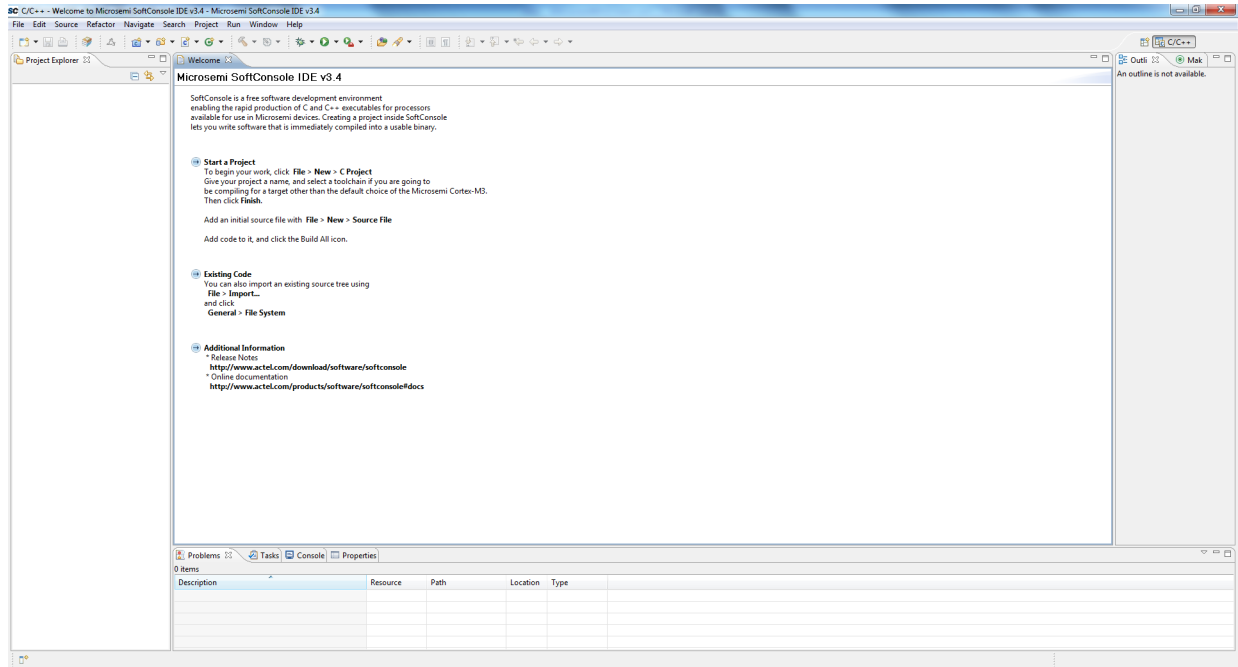
- Download and install Libero SoC (as indicated in the website for this design) on the host PC from the following location.
<https://www.microsemi.com/product-directory/design-resources/1750-libero-soc>
- For demo design files download link:
http://soc.microsemi.com/download/rsc/?f=m2s_ac426_df

2.6 Building Executable Image in Release Mode

The following steps describe how to build an application executable image in the release mode:

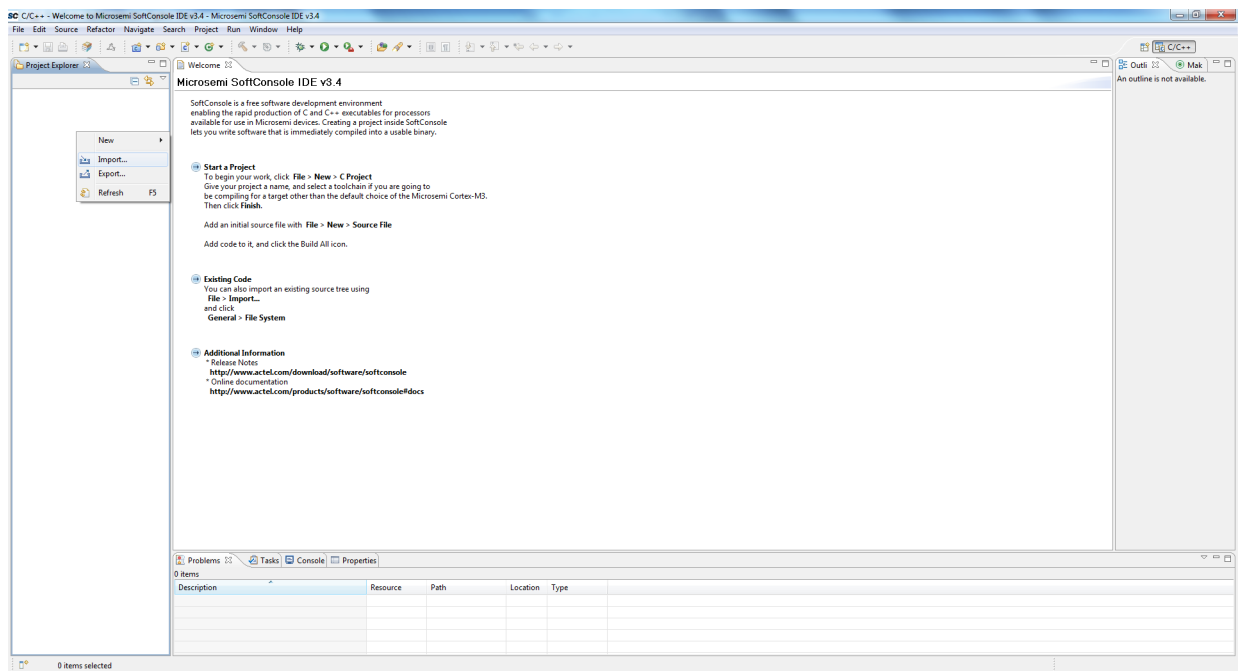
1. Open the standalone SoftConsole IDE.

Figure 1 • SoftConsole IDE



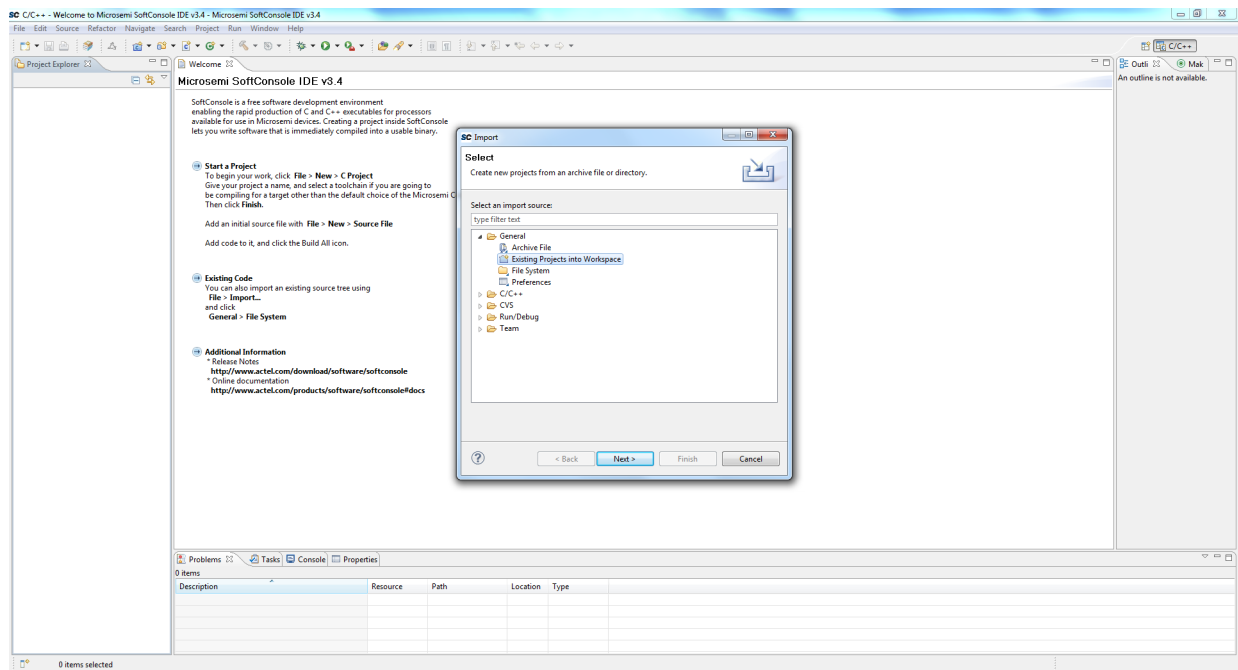
2. Right-click the **Project Explorer** window and choose **Import**, as shown in Figure 2.

Figure 2 • SoftConsole IDE - Import



3. Select **General > Existing Projects into Workspace**, as shown in Figure 3.

Figure 3 • SoftConsole IDE - Importing Existing Projects



Import window is displayed, as shown in Figure 4.

Figure 4 • SoftConsole IDE - Browse

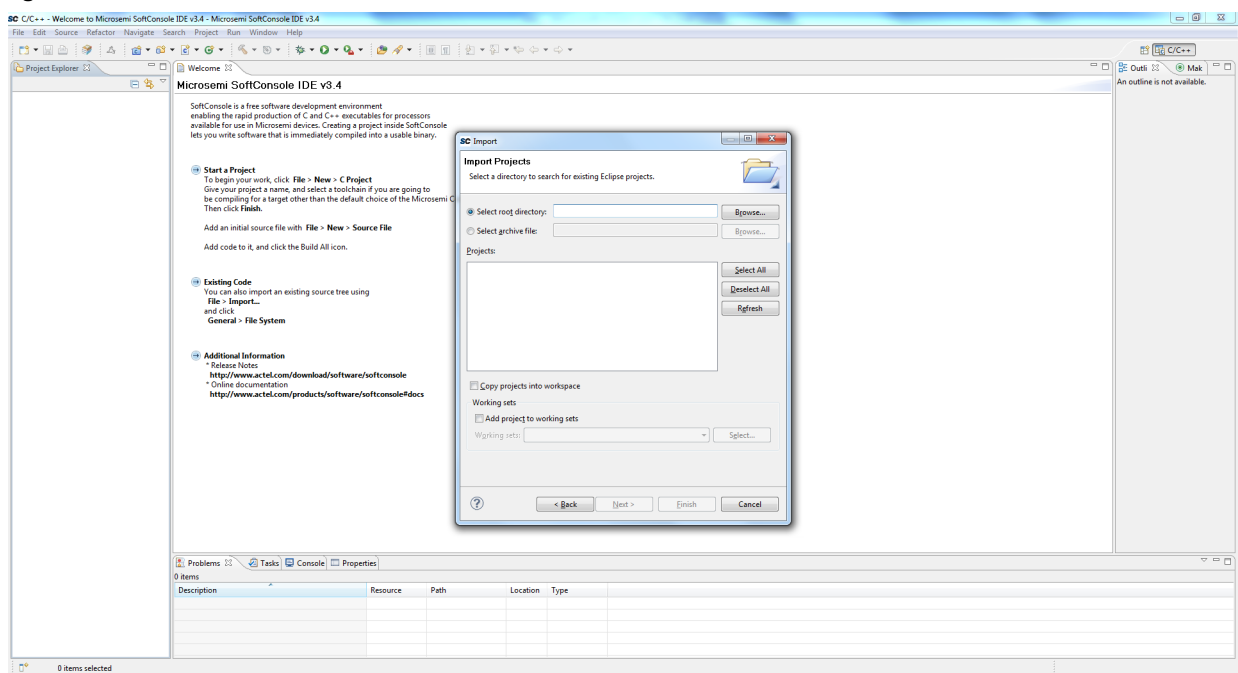
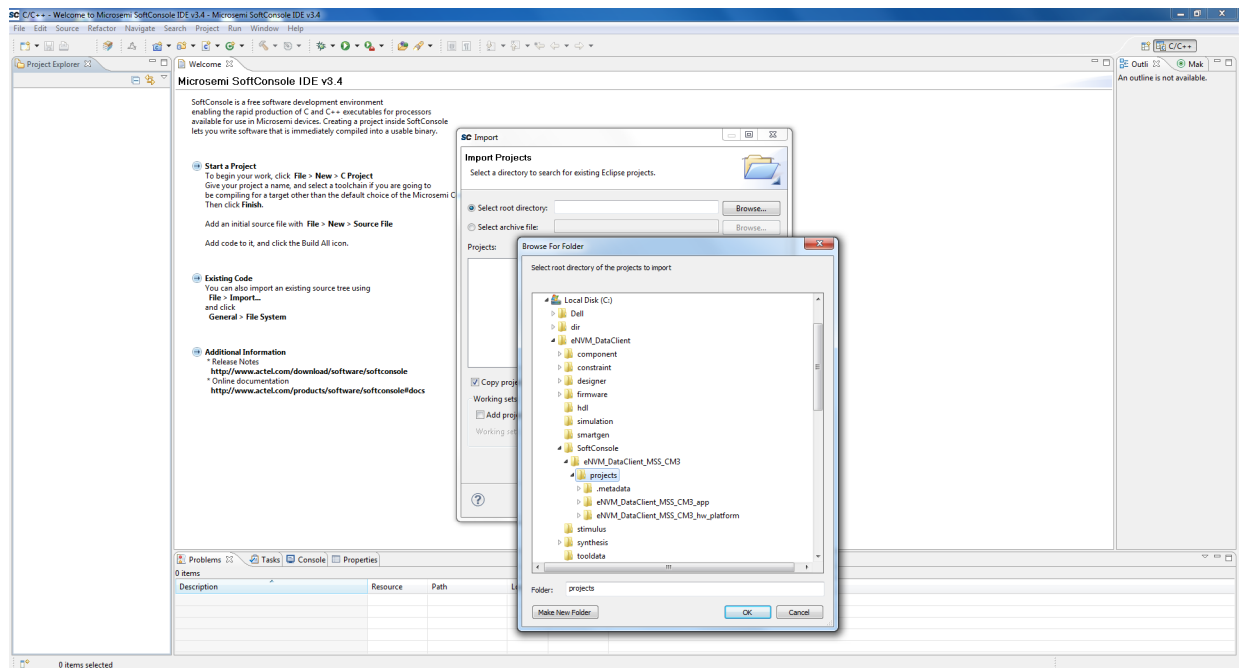
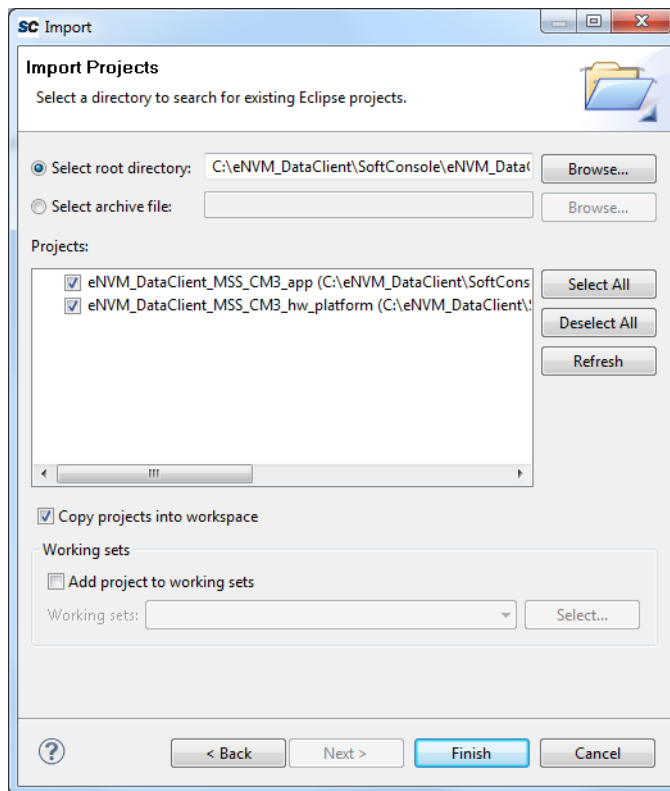


Figure 5 • SoftConsole IDE - Browsing for Projects Folder

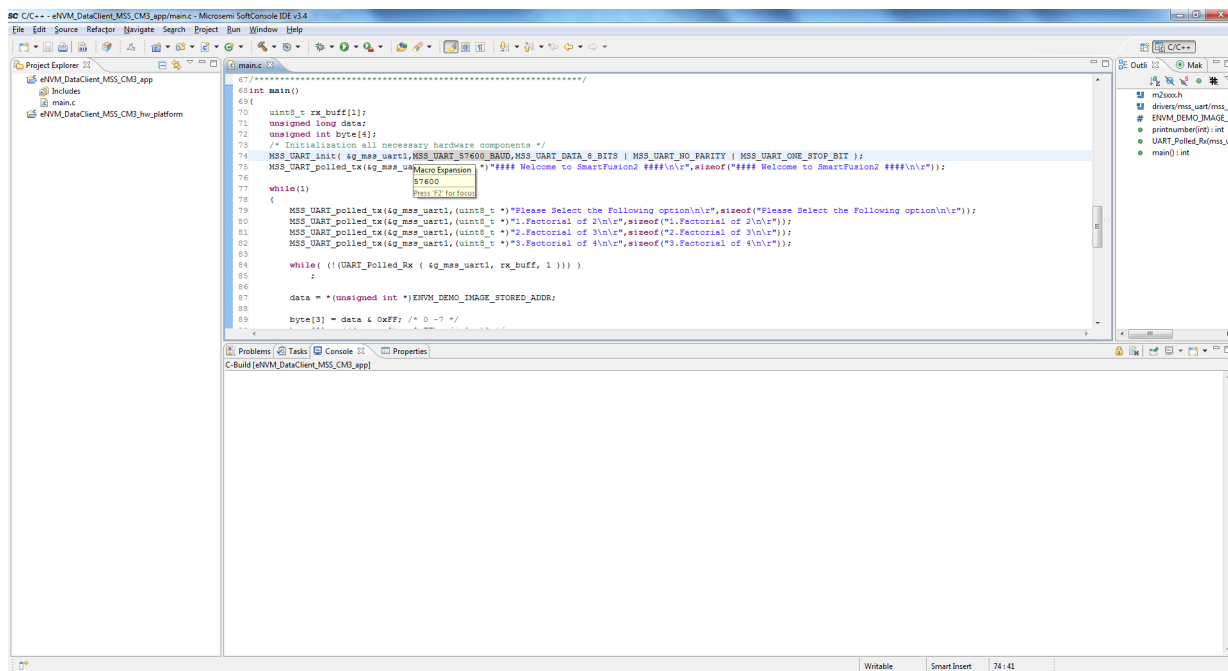
4. Ensure that **Copy projects into workspace** check box is checked, as shown in Figure 6. Click **Finish**.

Figure 6 • SoftConsole IDE - Creating Work Space in App Folder



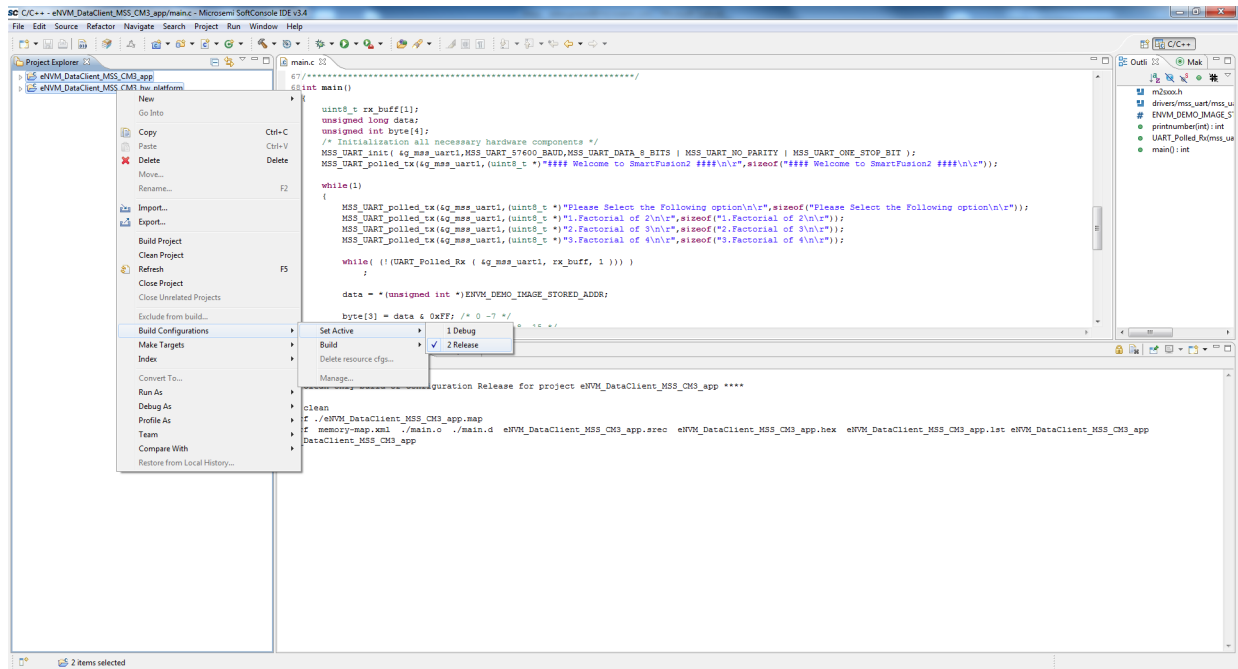
The SoftConsole perspective is shown in Figure 7.

Figure 7 • SoftConsole Workspace



- Right-click both the project names in the **Project Explorer** view and select **Build Configurations > Set Active > Release**, as shown in Figure 8.

Figure 8 • Building Configurations



- Go to **Project > Build All** and build the project, as shown in Figure 9. The .hex file is generated in the **Release** folder (refer to Figure 10, page 9 and Figure 11, page 9).

Figure 9 • Build All

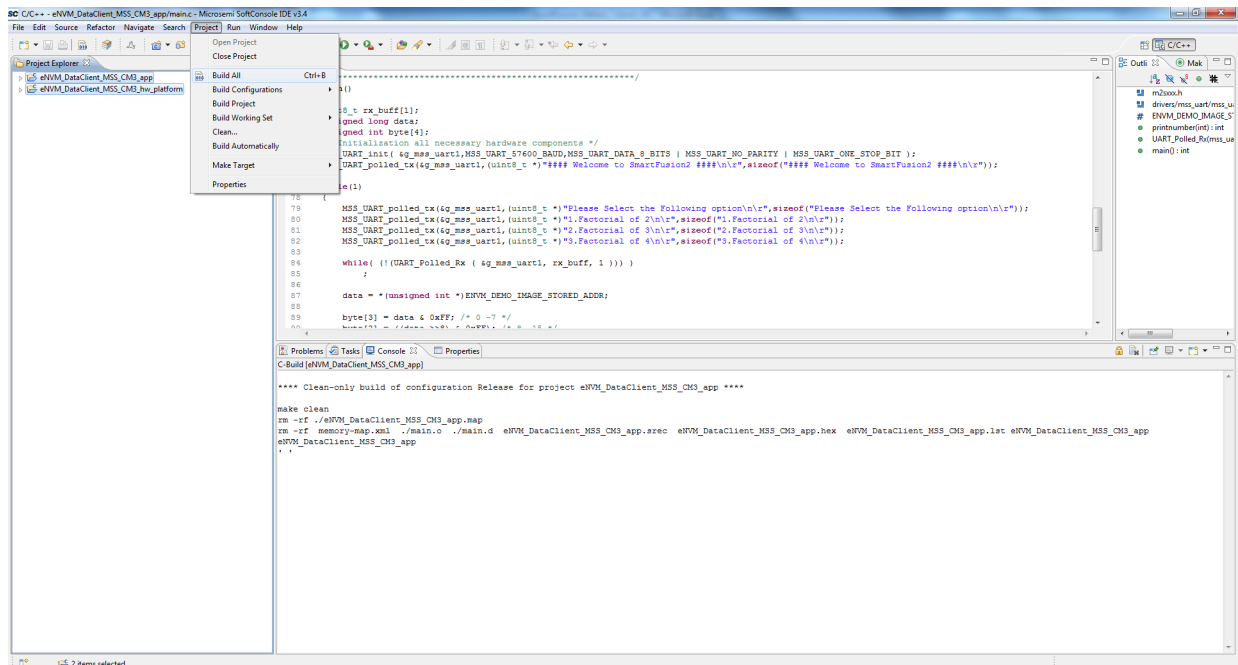


Figure 10 • Generating Hex File

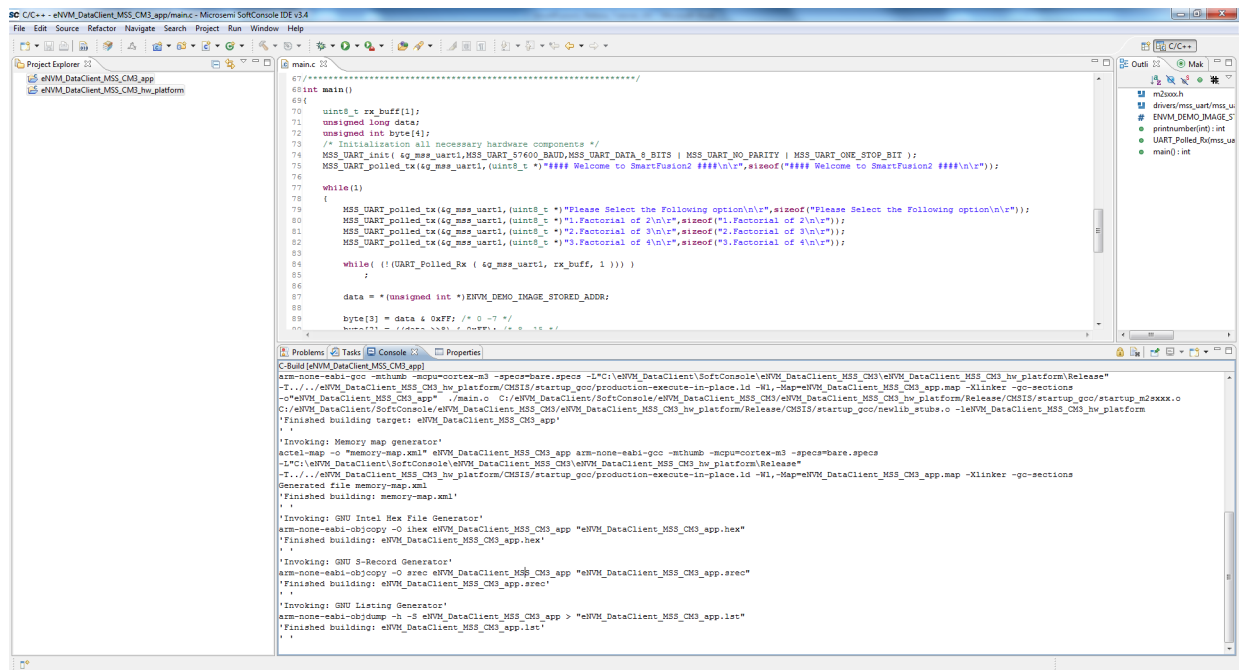
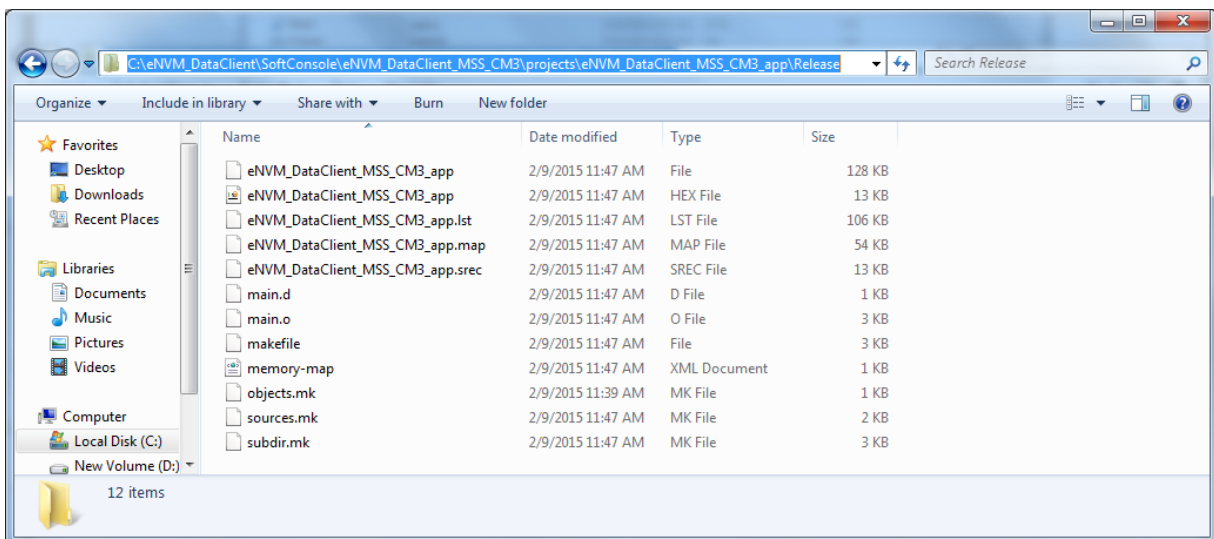


Figure 11 • Hex File in Release Folder

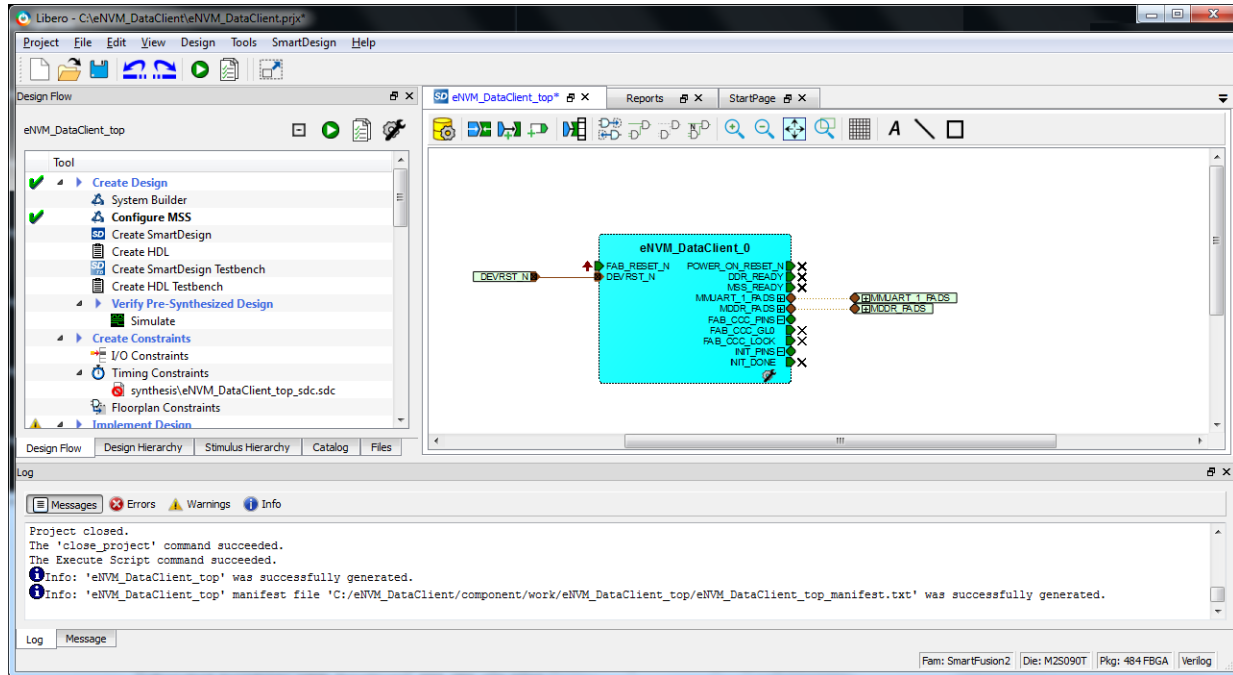


2.7 Loading the Executable Image into eNVM

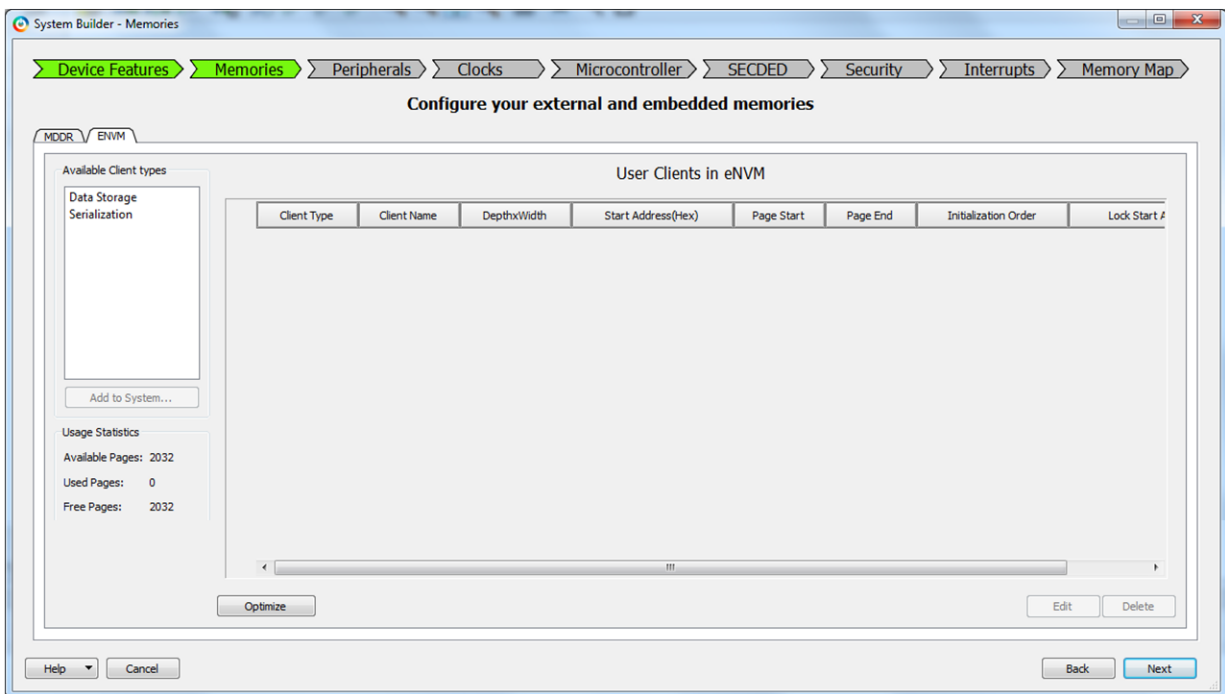
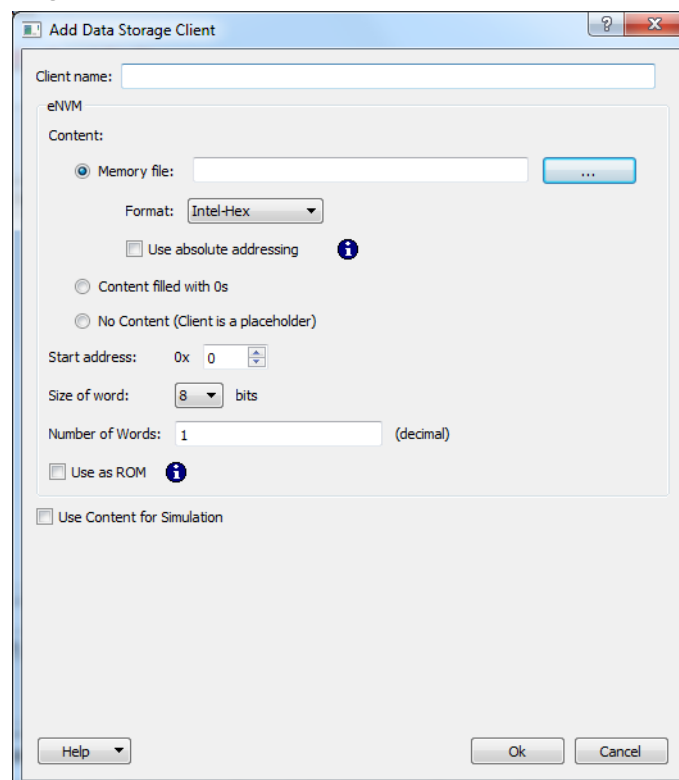
The following steps describe how to load the generated executable image into eNVM of the SmartFusion2 device using the System Builder eNVM Configurator:

1. Open the Libero SoC hardware project **eNVM_DataClient_top** tab, refer to Figure 12.

Figure 12 • eNVM_DataClient_top Tab

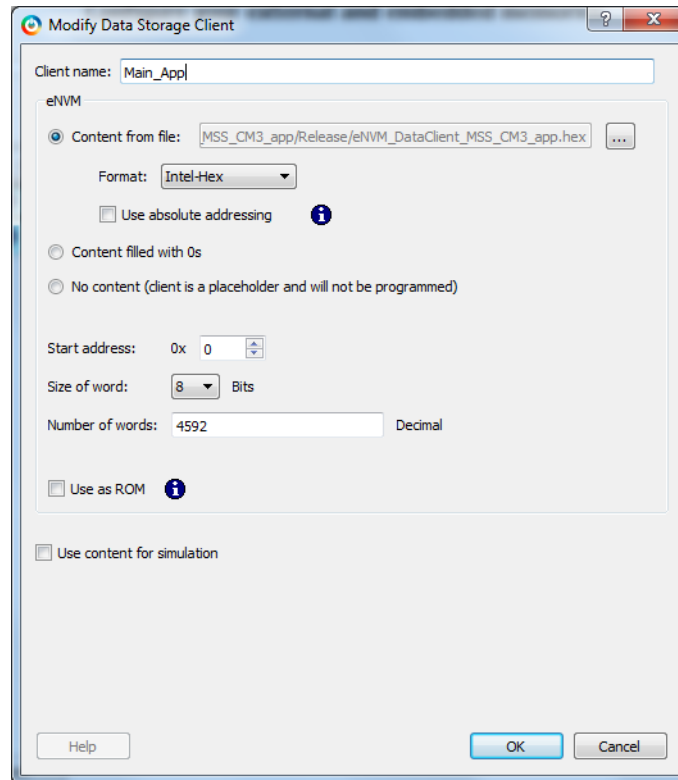


2. Double-click **eNVM_DataClient_0** and go to **System Builder - Memories** page to add the eNVM Data Storage client, refer to Figure 12.
3. Select **Data Storage** under the **Available Client Types** tab and click **Add to System...**, refer to Figure 13. This opens the **Add to Data Storage Client** window, as shown in Figure 14.

Figure 13 • System Builder - Memories Page**Figure 14 • Add Data Storage Client Window**

4. Enter a data storage client name in the **Add to Data Storage Client** window.

Figure 15 • Add Data Storage Client Window



5. Browse for the .hex file generated in the SoftConsole project after building an executable image in the Release Mode. The generated executable image is in the **Release** folder under the SoftConsole project workspace, refer to Figure 16 and Figure 17, page 13.

Figure 16 • Executable Image in Release Folder

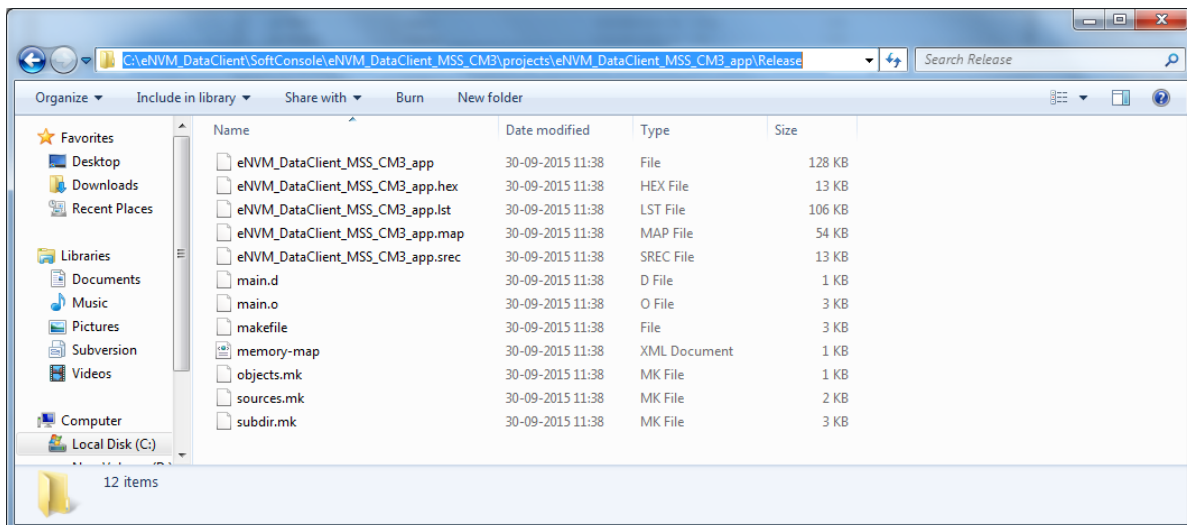
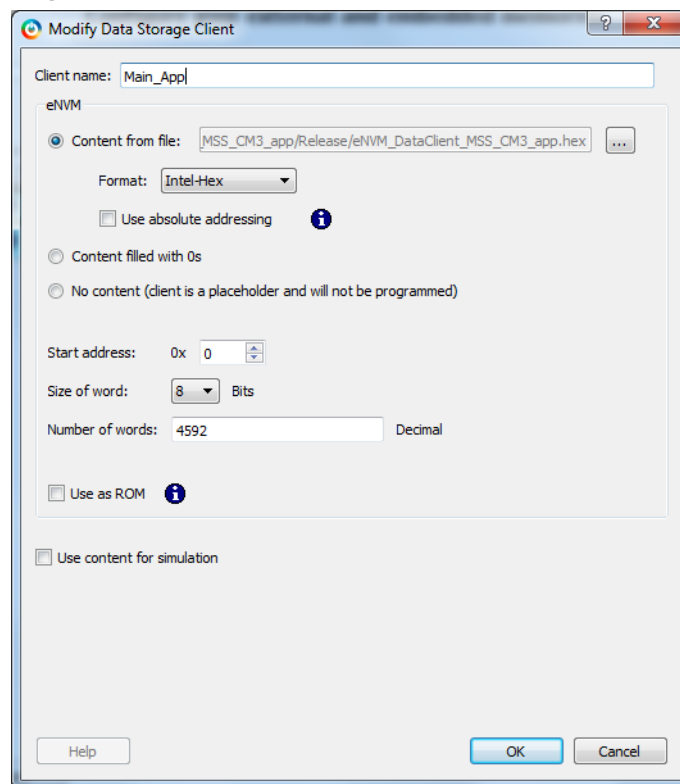
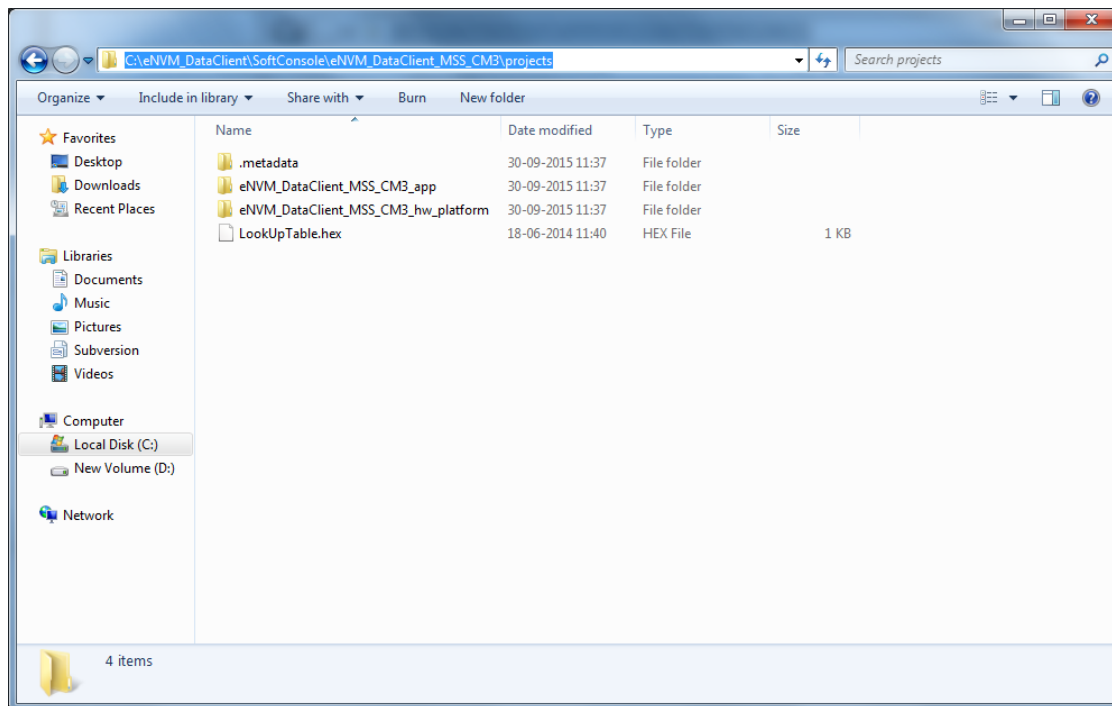
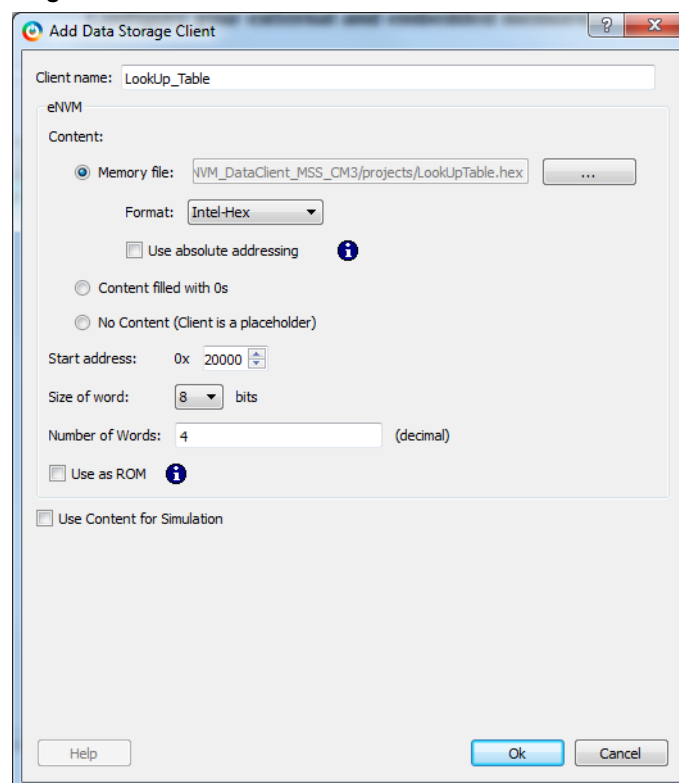


Figure 17 • Add Data Storage Client Window

6. Click **Ok** in the **Add Data Storage Client** window.
 7. Add `LookUpTable.hex` file to eNVM Data Storage Client, as shown in [Figure 18](#), page 14 and [Figure 19](#), page 14.
- Note:** The `LookUpTable.hex` file contains the pre-calculated data of the factorial of numbers 2, 3, and 4 (2, 6, and 24 in Hexa decimal format). This is a non-executable image and is stored in eNVM. The factorial data is fetched by the main application.

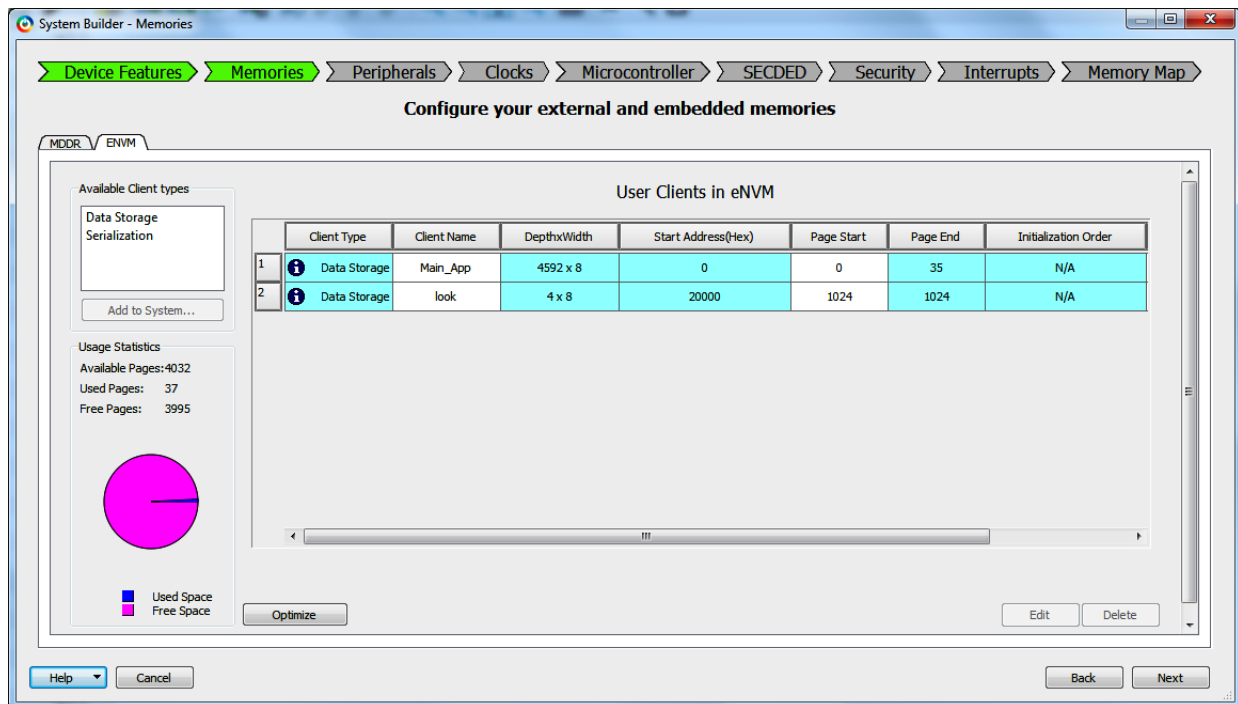
Figure 18 • LookUpTable.hex File

8. Select Start address as 20000 in the Add Data Storage Client window, refer to Figure 19. This is the address in eNVM, where the LookUpTable.hex file is stored. Click **Ok**.

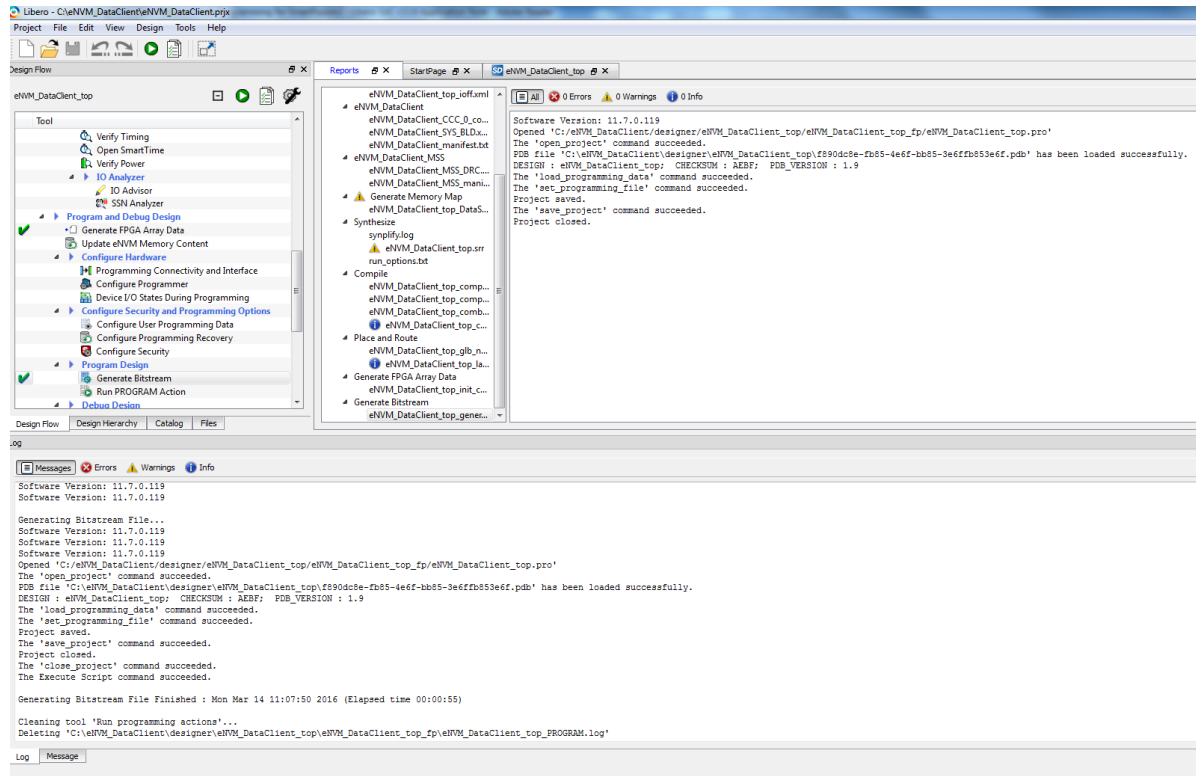
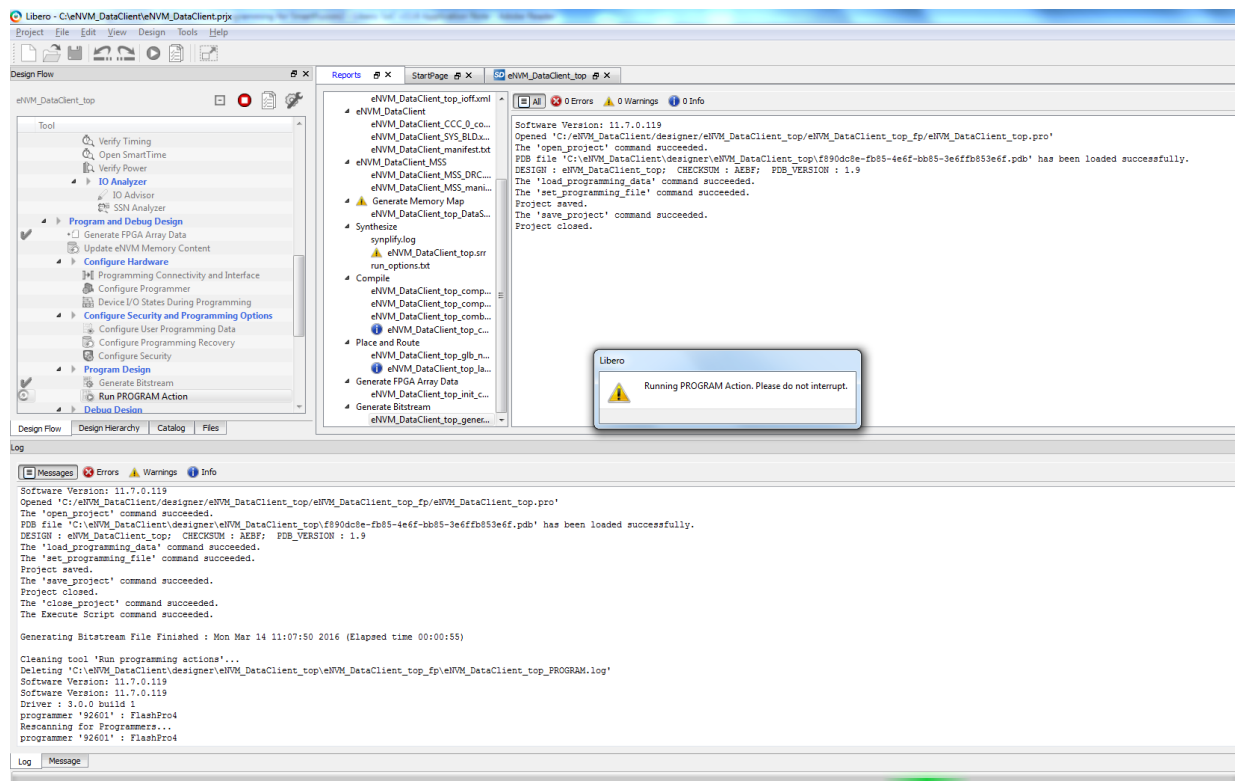
Figure 19 • Add Data Storage Client Window

9. In the **System Builder - Memories** page, keep the other settings to default and click **Next**, refer to Figure 20.

Figure 20 • System Builders - Memories Page



10. Save **eNVM_DataClient_top** and regenerate the **eNVM_DataClient_top** component by clicking **Generate Component** in SmartDesign.
11. Click **Generate Bitstream** to complete the remaining steps to generate fdb file (synthesis, place-and-route).
12. Click **Run PROGRAM Action** to program the SmartFusion2 Security Evaluation Kit, refer to Figure 21, page 16 and Figure 22, page 16.

Figure 21 • Selecting Run PROGRAM Action**Figure 22 • Programming SmartFusion2 Security Evaluation Kit**

2.8 Updating eNVM Memory Content

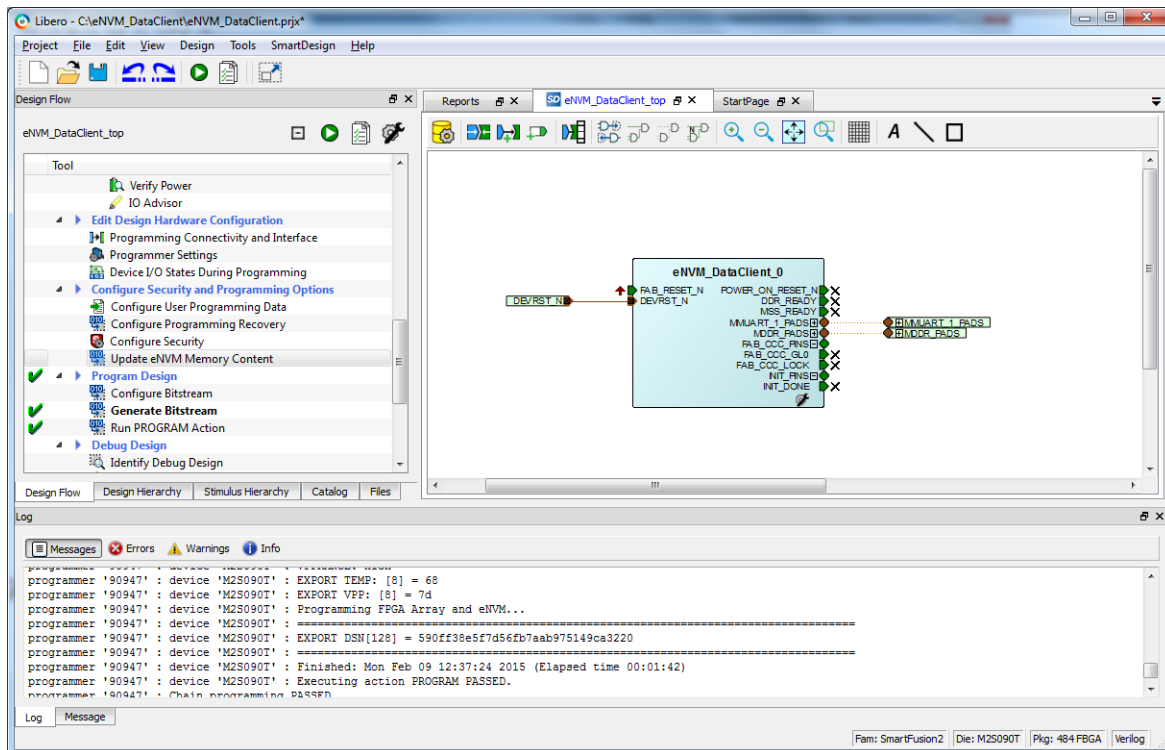
This option is available in Libero design flow to load the updated .hex files after compiling the application code in SoftConsole. Without this option, every time the SoftConsole application is built, System Builder eNVM Configurator is required for updating the eNVM client with the new .hex file generated, which is time consuming.

If the new .hex file generated from the SoftConsole application project is larger than the previous .hex file, **Update eNVM Memory Content** throws an error. Go back to the System Builder and manually update the Hex file.

The following steps describe how to use the **Update eNVM Memory Content** option:

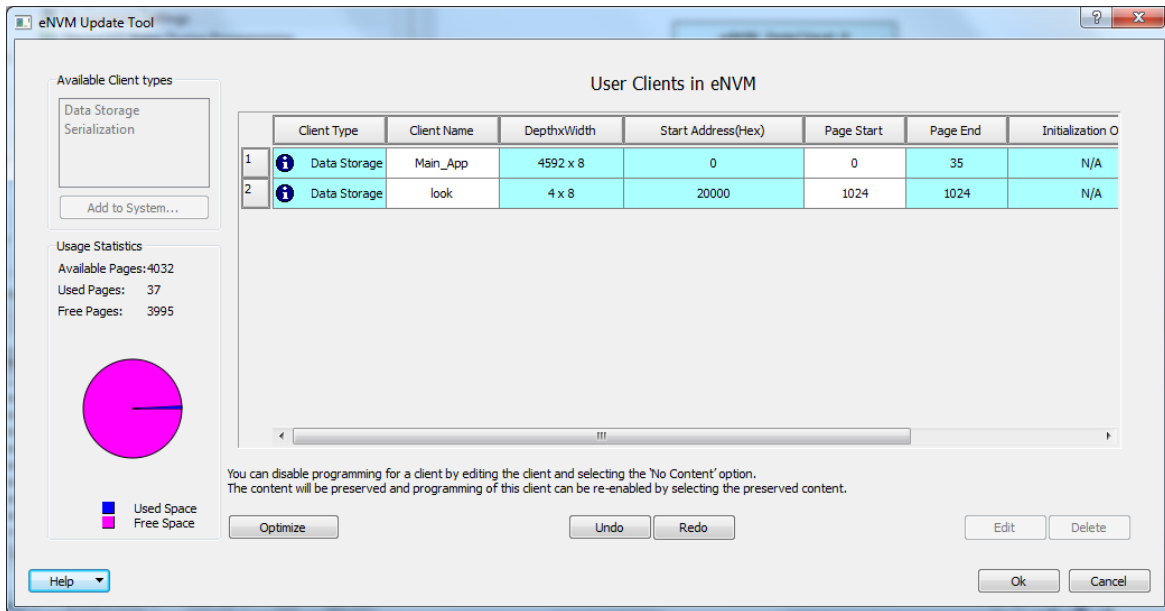
1. Choose **Update eNVM Memory Content** option from Libero design flow.

Figure 23 • Choosing Update eNVM Memory Content Option from Libero Design Flow



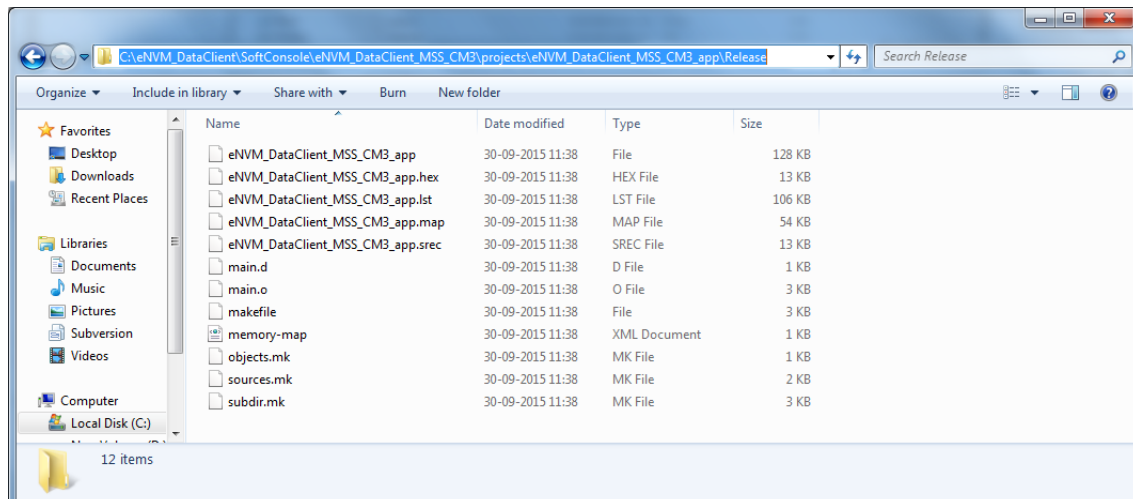
2. Update the **eNVM Memory Content** window, as shown in Figure 24.

Figure 24 • eNVM Memory Content Window



3. Browse through the updated **.hex** file generated in the Release Mode from the SoftConsole application project, refer to Figure 25. Click **Ok**.

Figure 25 • Updated Hex File in Release Folder



4. Click **Generate Bitstream** to complete the remaining steps to generate the fdb file (synthesis, place-and-route).
5. Click **Run PROGRAM Action** to program the SmartFusion2 Security Evaluation Kit.

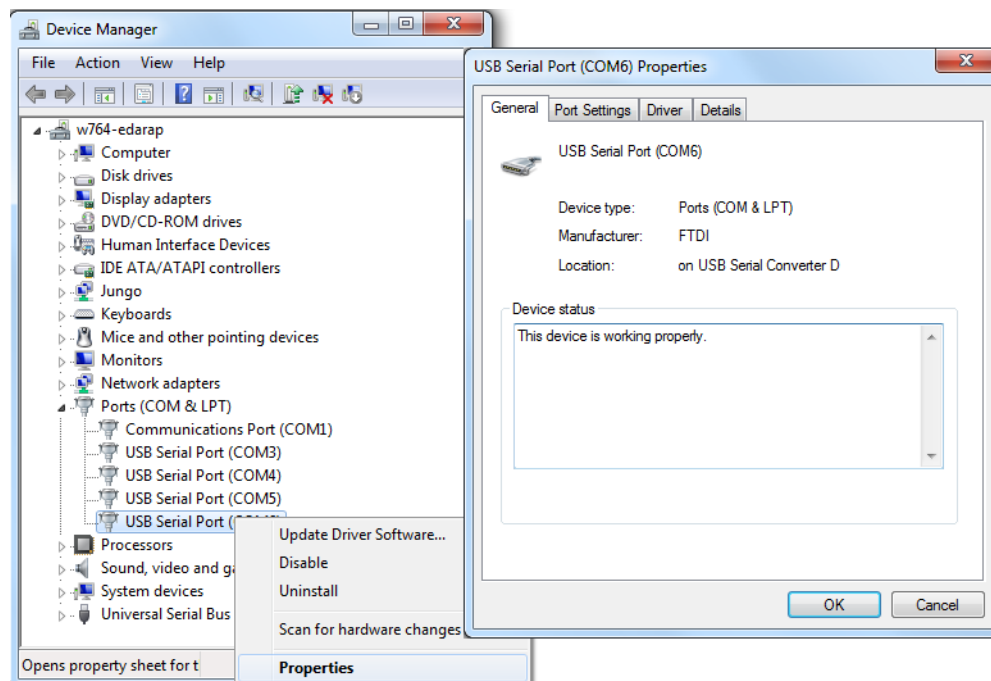
2.9 Running the Design on SmartFusion2 Security Evaluation Kit

The following steps describe how to run the design:

1. Connect the **FlashPro4** programmer to the J5 connector of SmartFusion2 Security Evaluation Kit.
2. Connect the host PC to the J18 connector provided on the SmartFusion2 Security Evaluation Kit using the USB min-B cable. Ensure that the USB to UART bridge drivers are automatically detected (can be verified in the Device Manager as shown in Figure 26).

Note: Copy the COM port number for serial port configuration. Ensure that the COM port location is specified as on **USB Serial Converter D**, as shown in Figure 26.

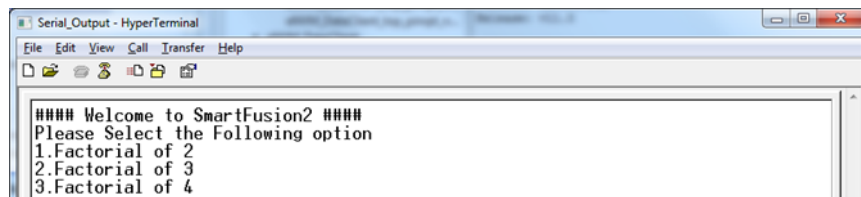
Figure 26 • Device Manager Window



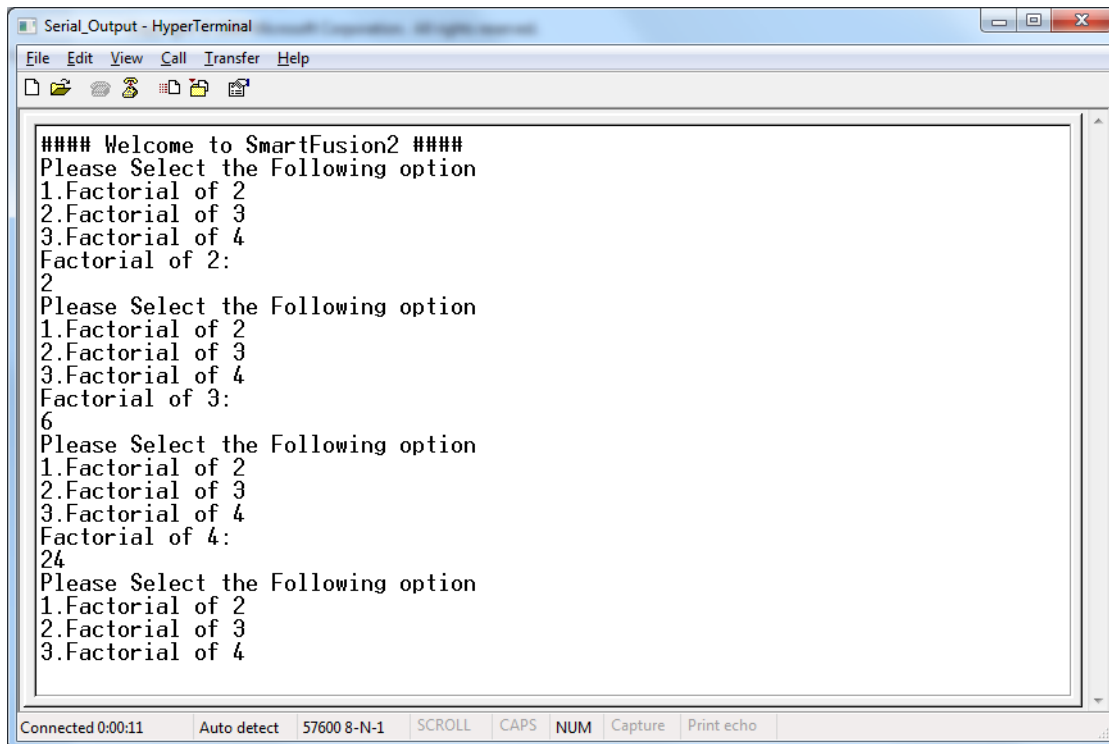
3. If USB to UART bridge drivers are not installed, download and install the drivers from www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.
4. Connect the power supply to the J6 connector and switch ON the power supply switch, SW7.
5. Start the HyperTerminal program with a baud rate of 57600, 8 data bits, 1 stop bit, no parity, and no flow control. If the PC does not have HyperTerminal program, use any free serial terminal emulation program such as PuTTY or TeraTerm. For more information about configuring the HyperTerminal, TeraTerm, and PuTTY, refer to [Configuring Serial Terminal Emulation Programs Tutorial](#).
6. Program the SmartFusion2 Security Evaluation kit board with the job file provided as part of the design files using FlashPro Express software, refer to [Appendix 1: Programming the Device Using FlashPro Express](#), page 37.

Note: This step is required if *.job file in the design folders is used. In Libero design, **Run PROGRAM** Action programs the *.job file to the Board.

7. After successful programming, press SW6 switch to reset the board.
8. The serial terminal program displays the user options, as shown in Figure 27, page 20.
 - Select option 1 to get Factorial of number 2.
 - Select option 2 to get Factorial of number 3.
 - Select option 3 to get Factorial of number 4.

Figure 27 • Selecting Factorial of a Number - User Options

The application image fetches the Factorial Value from the Look up table (Non-executable image) and displays it, as shown in Figure 28.

Figure 28 • Displaying Factorial Value Based on User Option

2.10 Building Multiple Executable Images in Release Mode

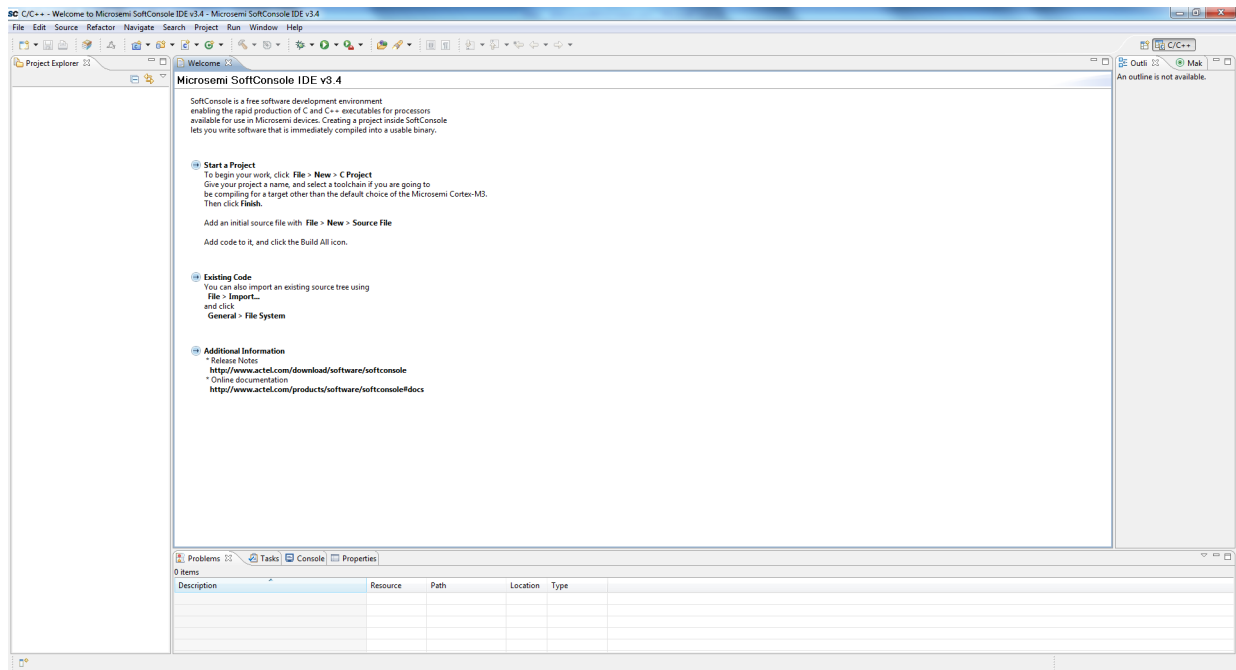
The design files are available to download from the Microsemi website. For design files, refer to: Prerequisites, page 3.

This design contains the eNVM data storage clients for Multiple Hex images. It runs the images in eNVM and eSRAM, and the LEDs blink in various patterns. This design demonstrates how to build multiple images in the release mode and store them in eNVM.

The following steps describe how to build multiple application executable image in the release mode:

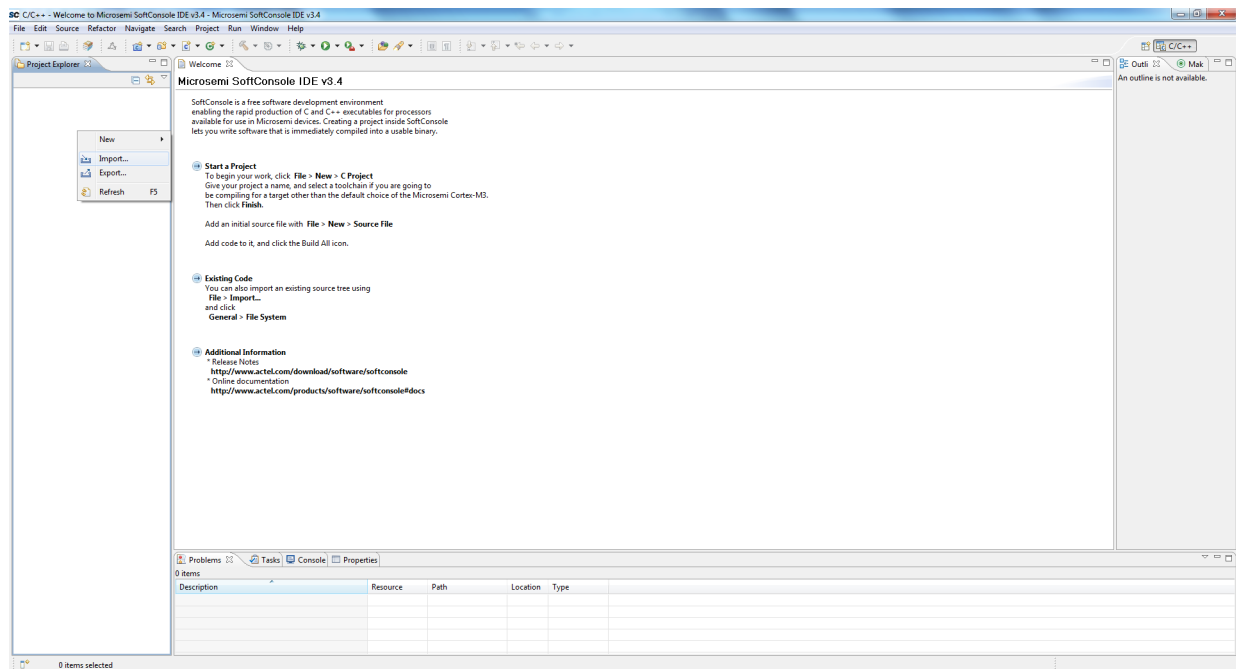
1. Open the standalone SoftConsole IDE.

Figure 29 • SoftConsole IDE



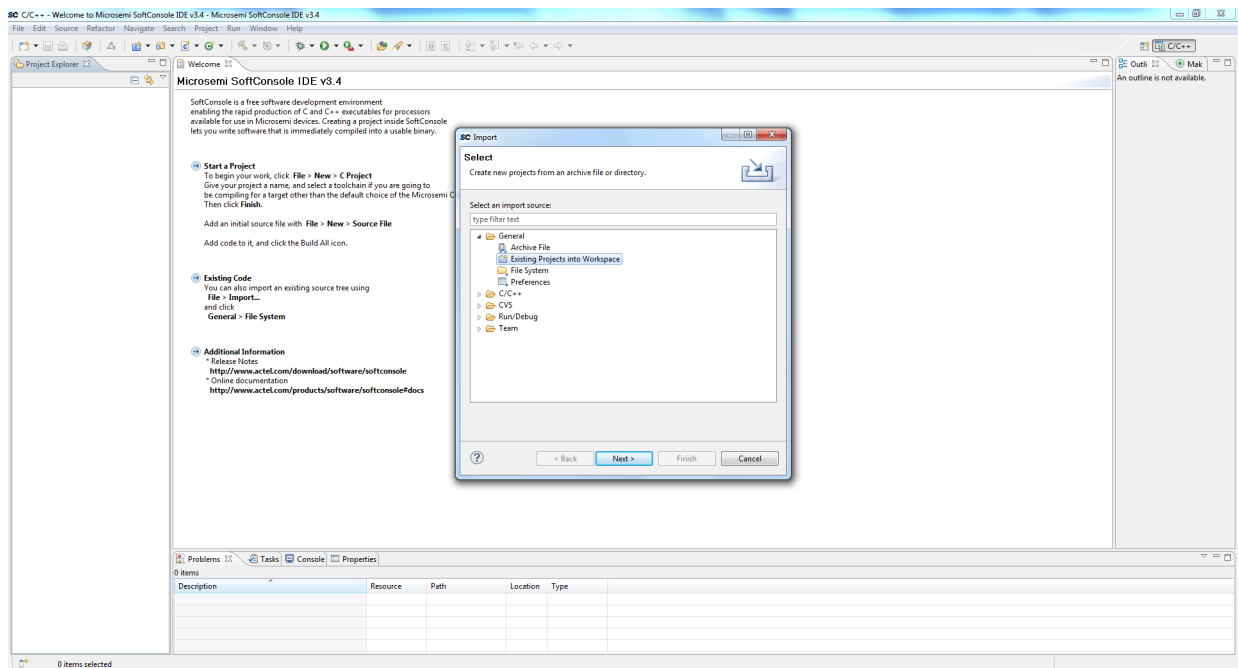
2. Right-click the **Project Explorer** window and choose **Import**, as shown in Figure 30.

Figure 30 • SoftConsole IDE - Import



3. Select **General > Existing Projects into Workspace**, as shown in Figure 31.

Figure 31 • SoftConsole IDE - General



Import window is shown in Figure 32.

Figure 32 • SoftConsole IDE - Browse

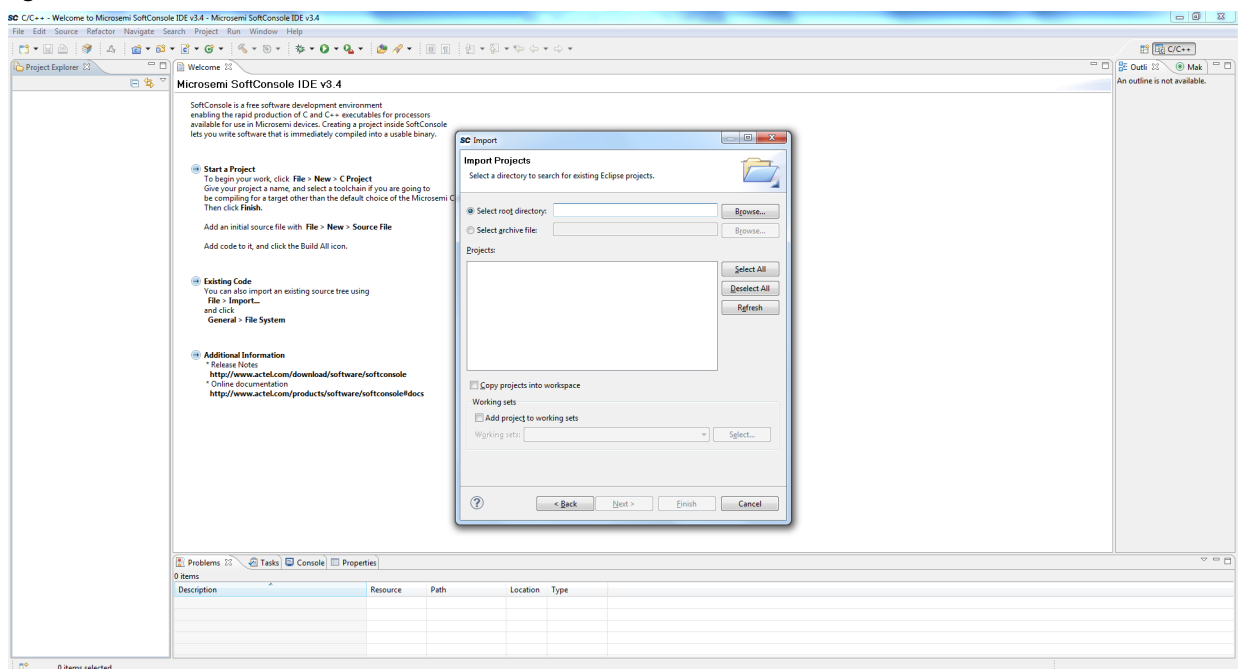
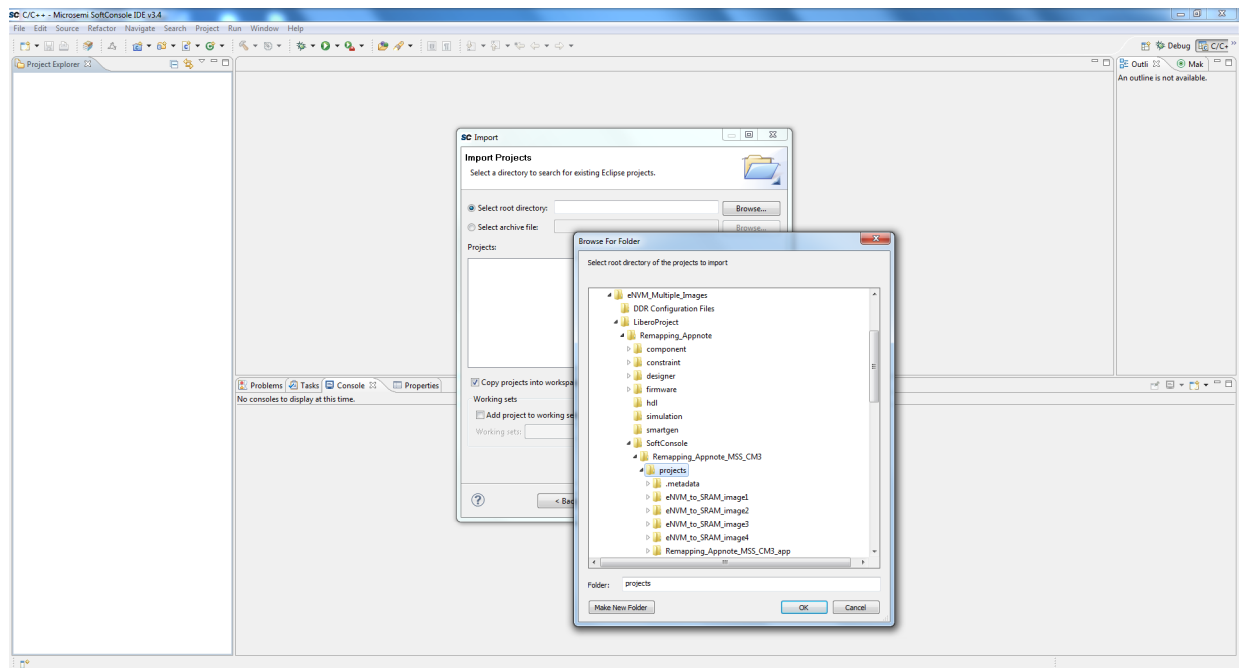
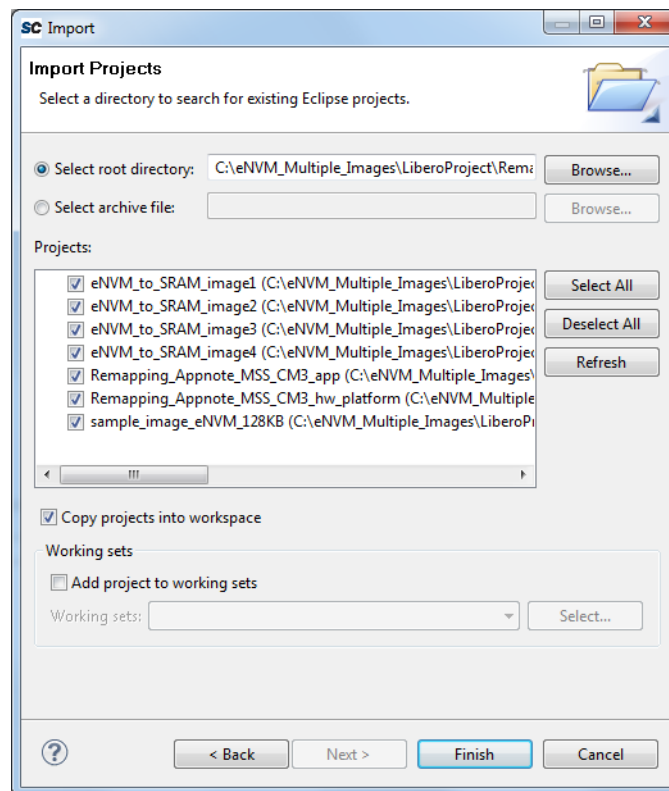


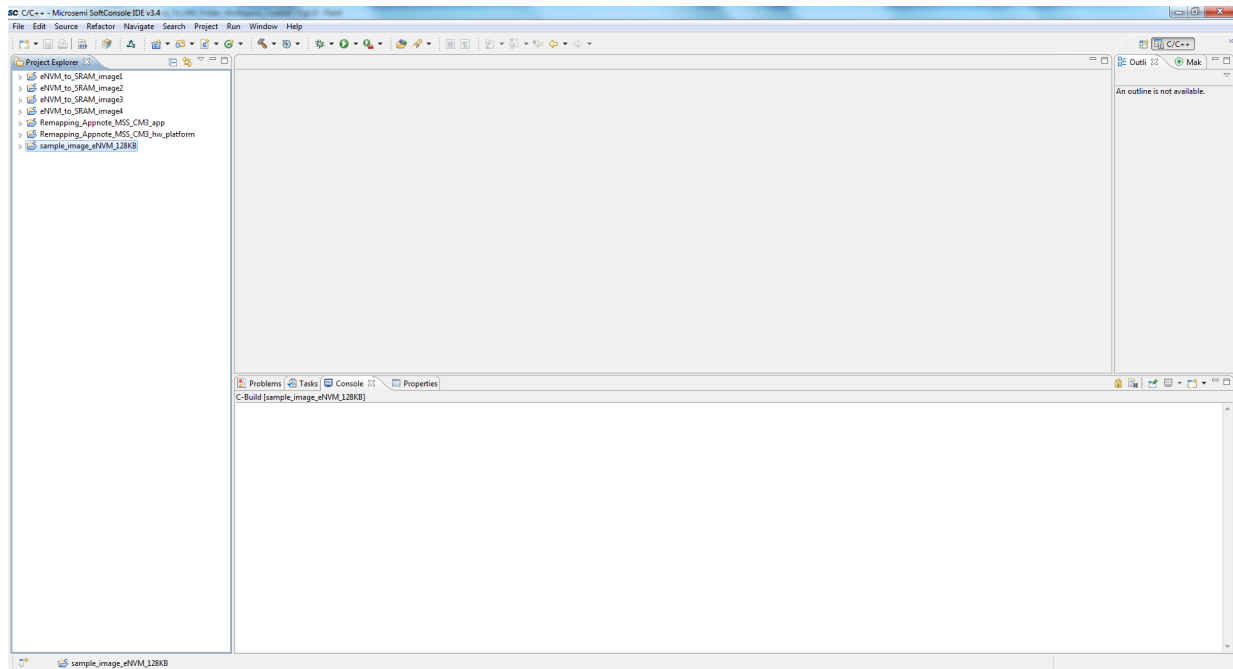
Figure 33 • SoftConsole IDE - Browsing for App Folder

4. Ensure that **Copy projects into workspace** check box is checked, as shown in Figure 34. Click **Finish**.

Figure 34 • SoftConsole IDE - Creating Work Space in App Folder

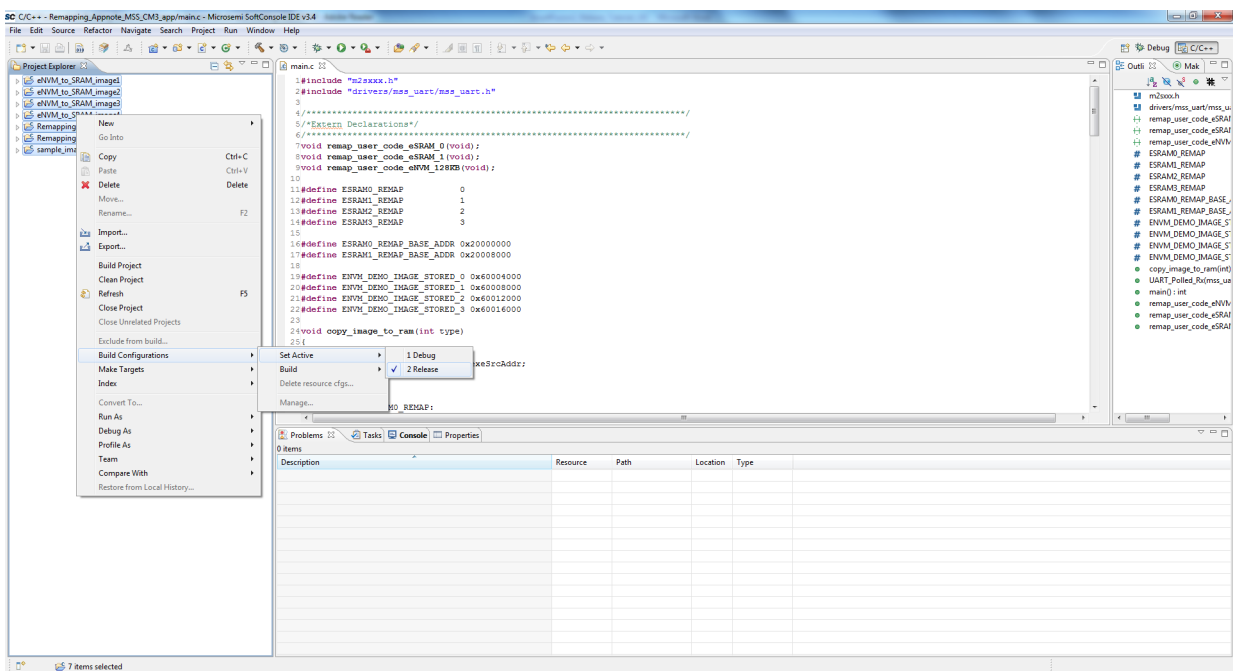
The SoftConsole perspective is shown in Figure 35.

Figure 35 • SoftConsole Workspace



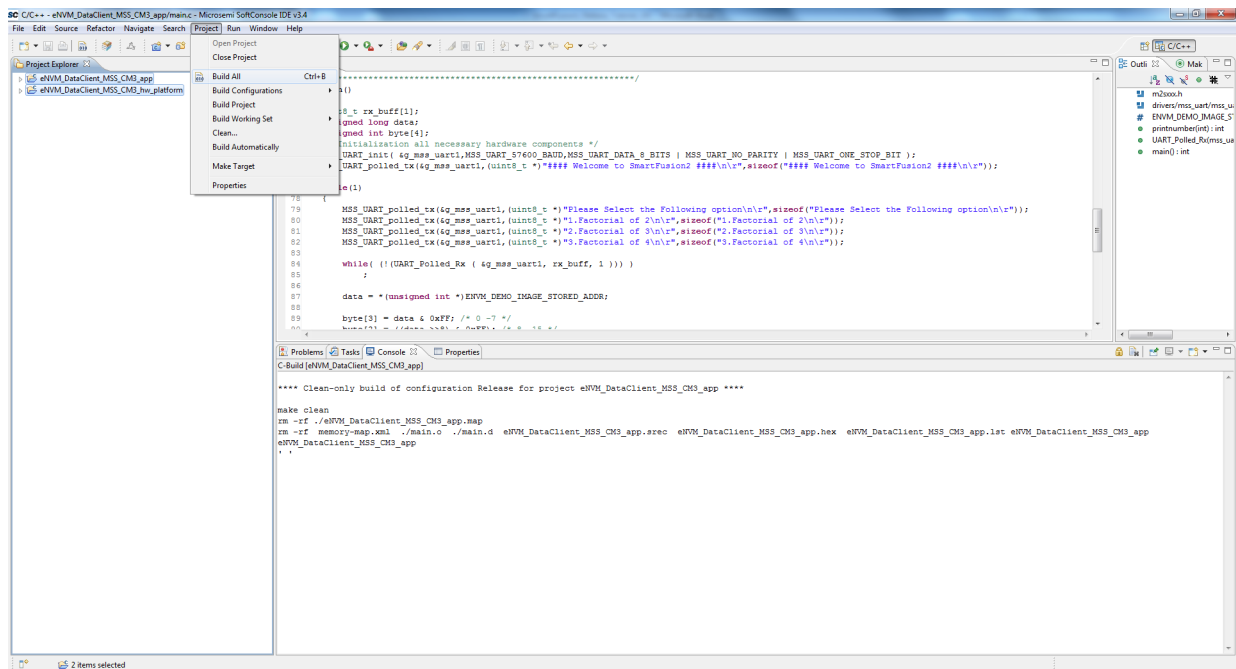
- Right-click both project names in the **Project Explorer** view and select **Build Configurations > Set Active > Release**, as shown in Figure 36.

Figure 36 • Building Configuration



6. Go to **Project > Build All** and build the project, as shown in Figure 37.
 .hex file is generated in the **Release** folder created in the project folder.

Figure 37 • Build All



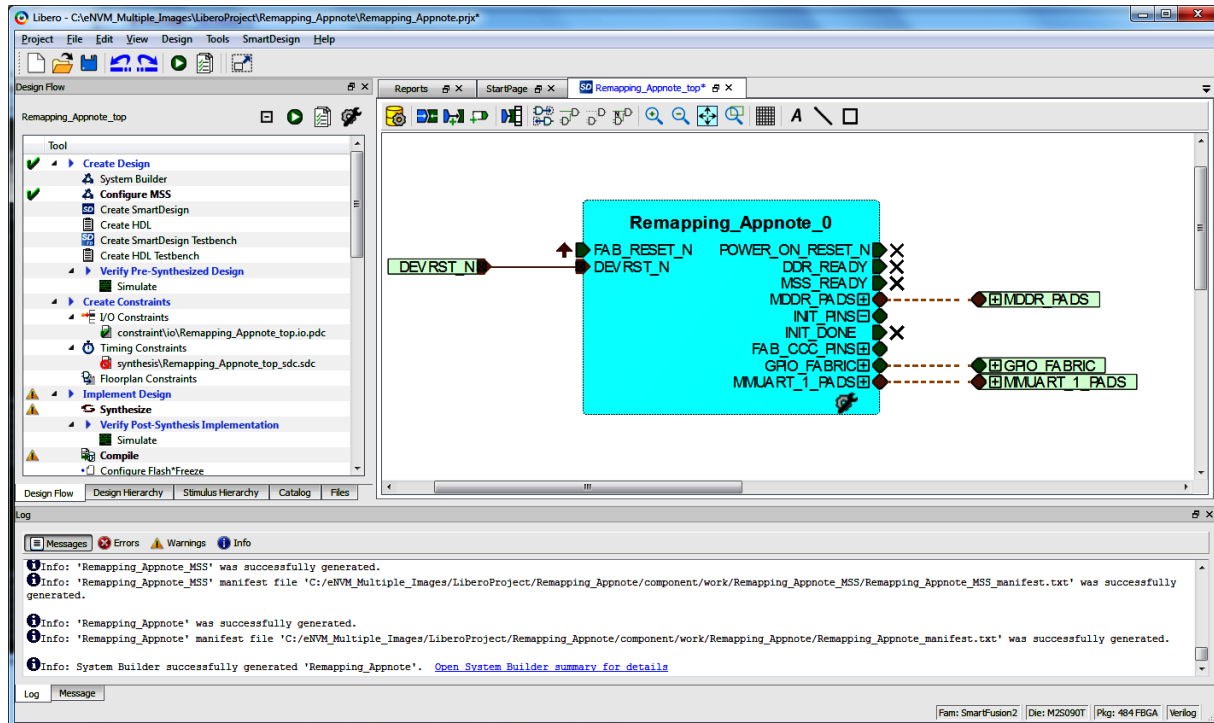
2.11 Loading Multiple Executable Images into eNVM

Multiple SoftConsole application projects generate multiple executable images. These executable images are stored in eNVM.

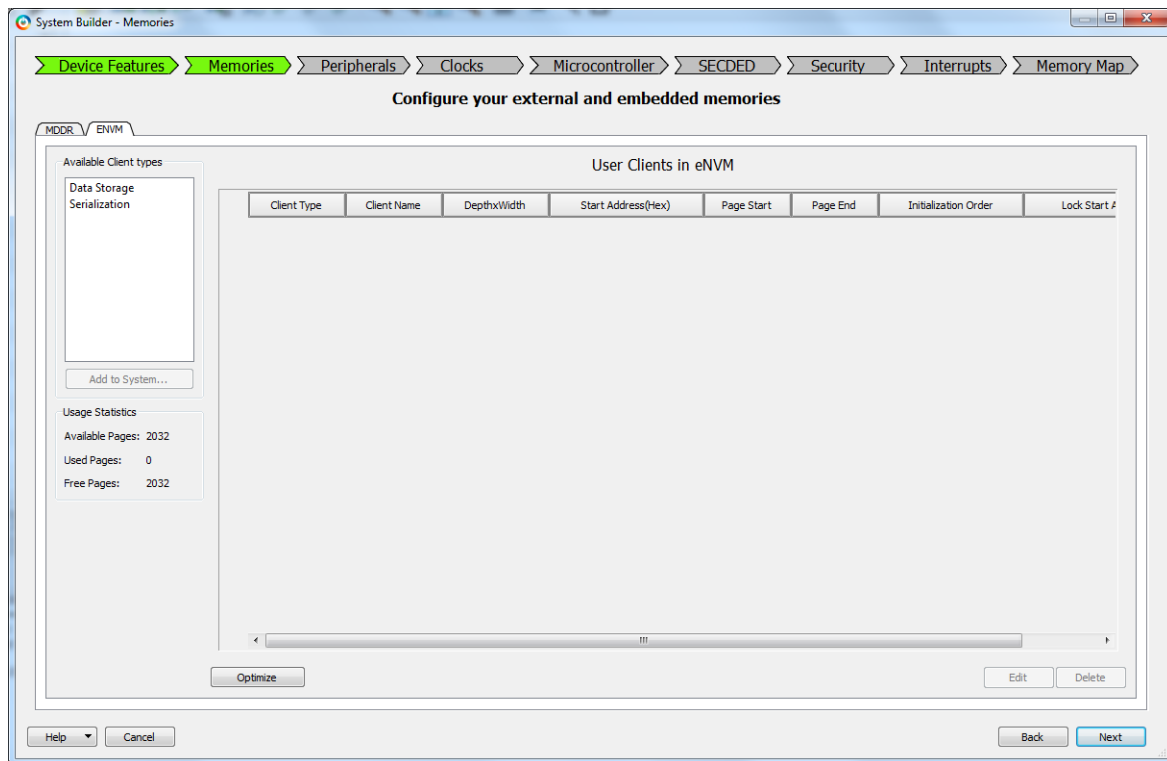
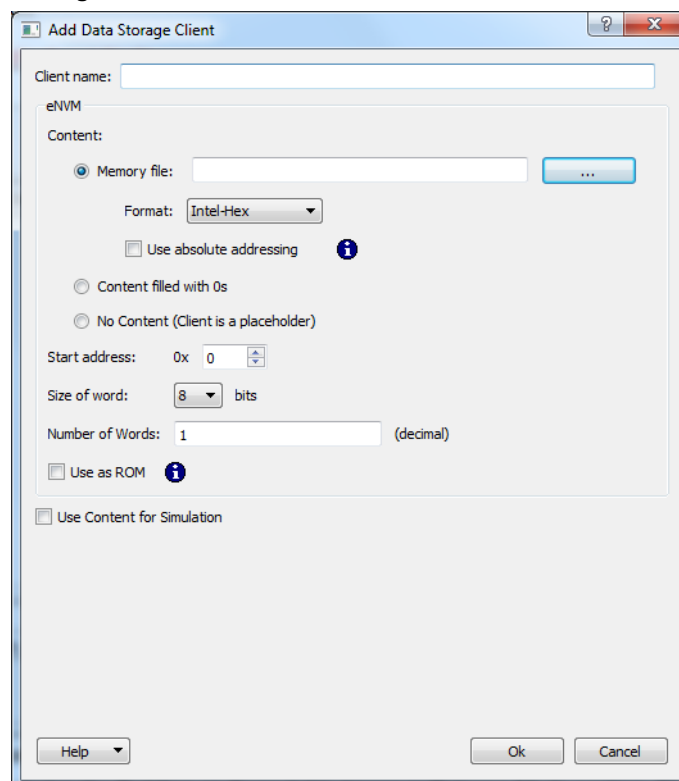
The following steps describe how to load the generated application executable images into eNVM of the SmartFusion2 device using the System Builder eNVM Configurator:

1. Open the Libero SoC hardware project **Remapping_Appnote_top** tab.

Figure 38 • Remapping_Appnote_top Tab

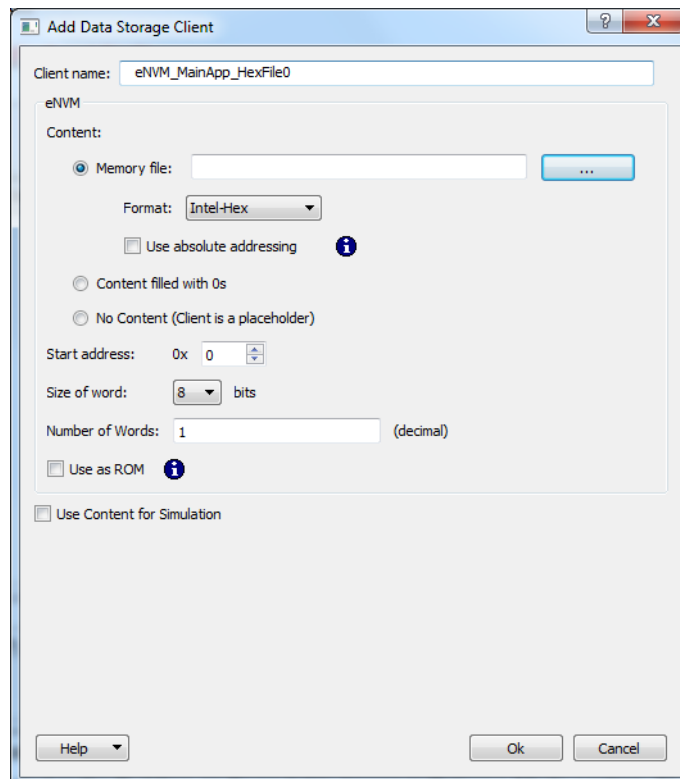


2. Double-click **Remapping_Appnote_0** and go to **System Builder - Memories** page to add the eNVM Data Storage client as shown in Figure 38, page 26 and Figure 39, page 27.
3. Select **Data Storage** under the **Available Client types** and click **Add to System...**, refer to Figure 39, page 27. This opens the **Add to Data Storage Client** window, as shown in Figure 40, page 27.

Figure 39 • System Builder - Memories Page**Figure 40 • Add to Data Storage Client Window**

4. Enter a data storage client name in the **Add to Data Storage Client** window.

Figure 41 • Add to Data Storage Client Window



5. Browse for the .hex file generated after compiling the SoftConsole project in the Release Modes, refer to Figure 43, page 29. The generated executable image is in the **Release** folder under SoftConsole project workspace, refer to Figure 42.

Figure 42 • Executable Image in Release Folder

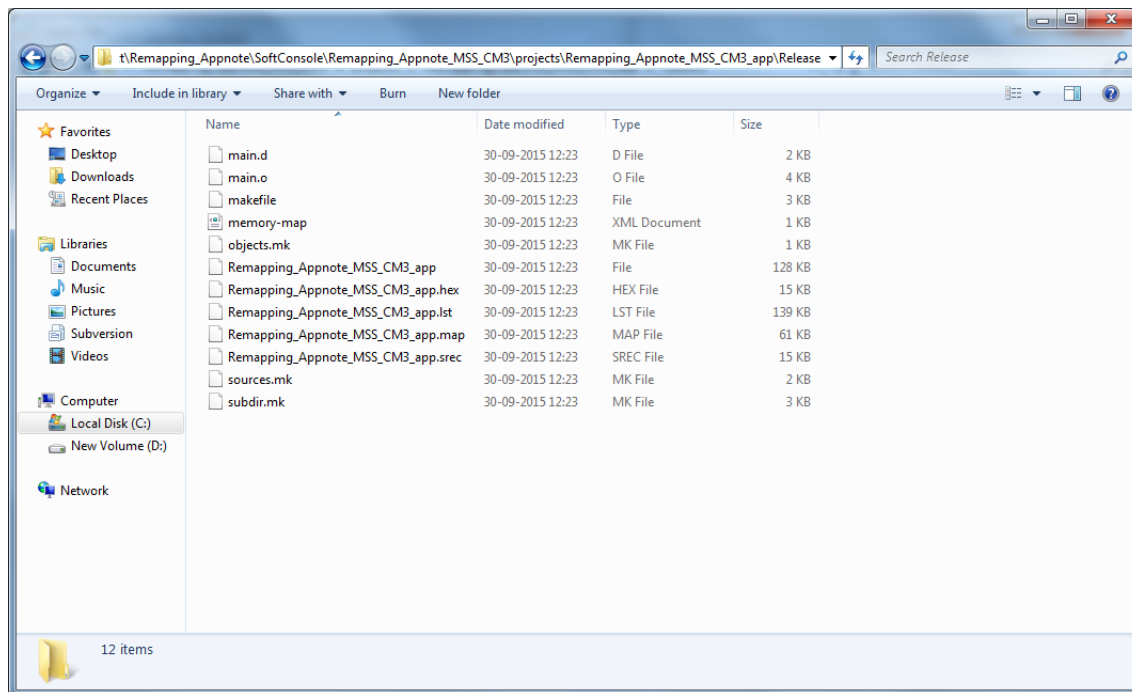
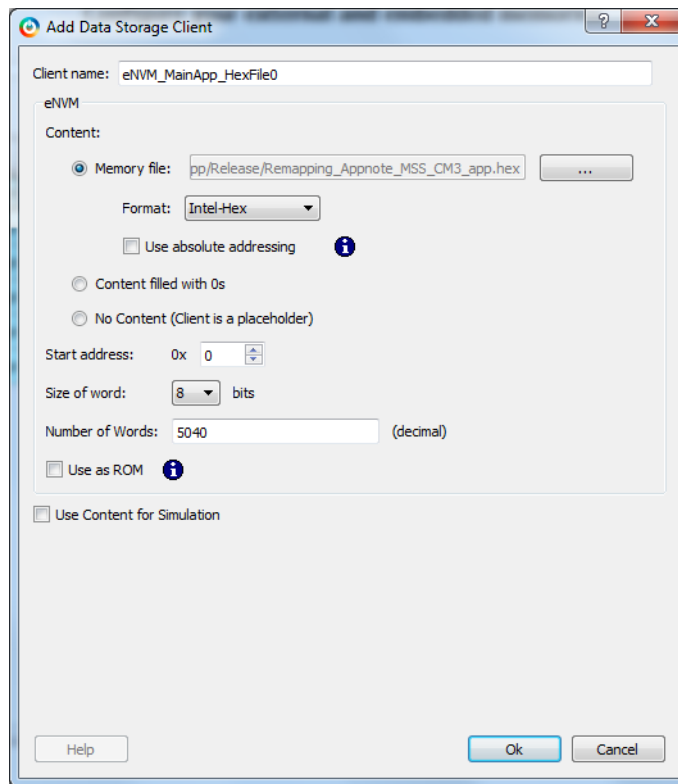


Figure 43 • Add Data Storage Client Window

6. Click **Ok** in the **Add Data Storage Client** window.
7. Add the remaining Hex Files with their corresponding start addresses, as shown in [Figure 45](#), page 30.

[Table 2](#) lists the SoftConsole application projects in the Release mode, their .hex files, and their mapping to the eNVM Client Name.

Table 2 • SoftConsole Application Projects and their Mapping to eNVM Client

ENVM Client Name	SoftConsole Release Mode Path	Hex File Name	Start Address
eNVM_MainApp_HexFile0	Remapping_Appnote_MSS_CM3_app\Release	Remapping_Appnote_MSS_CM3_app.hex	0x0
eNVM_to_SRAM_HexFile1	eNVM_to_SRAM_image1\Release	eNVM_to_SRAM_image1.hex	0x4000
eNVM_to_SRAM_HexFile2	eNVM_to_SRAM_image2\Release	eNVM_to_SRAM_image2.hex	0x8000
eNVM_to_SRAM_HexFile3	eNVM_to_SRAM_image3\Release	eNVM_to_SRAM_image3.hex	0x12000
eNVM_to_SRAM_HexFile4	eNVM_to_SRAM_image4\Release	eNVM_to_SRAM_image4.hex	0x12000
eNVM_128KB_HexFile5	sample_image_eNVM_128KB\Release	sample_image_eNVM_128KB.hex	0x20000

Note: Start address is the address in which the executable image is stored in eNVM. Provide the required memory size based on the total size of the executable image. Otherwise, the images overlap and display error as shown in [Figure 44](#), page 30.

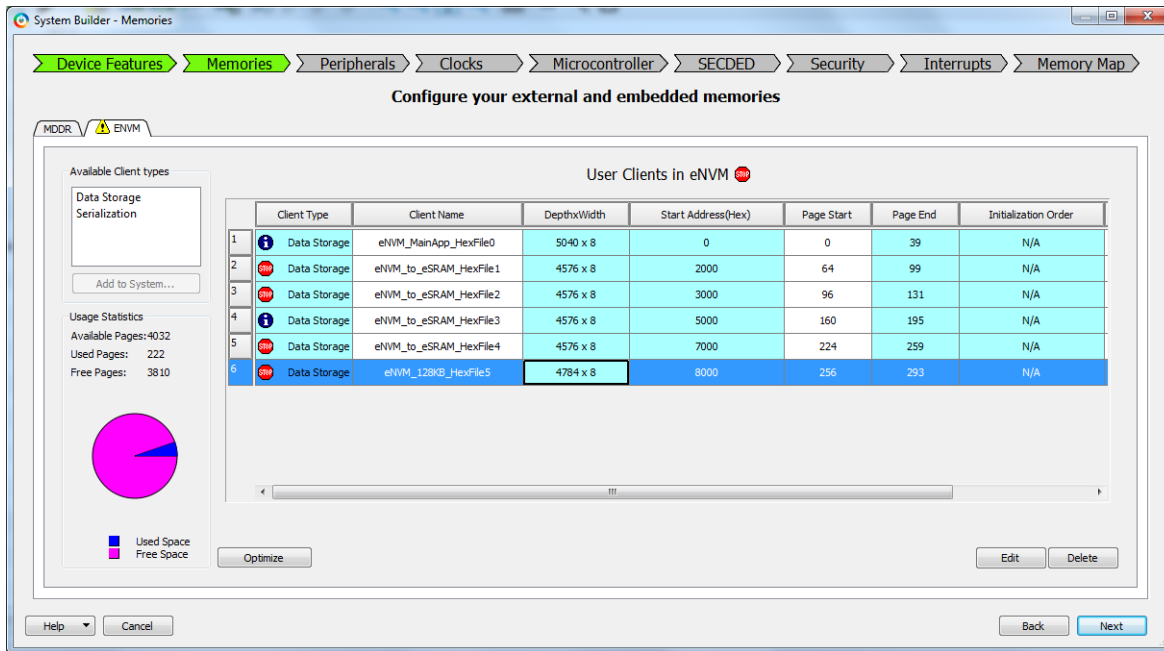
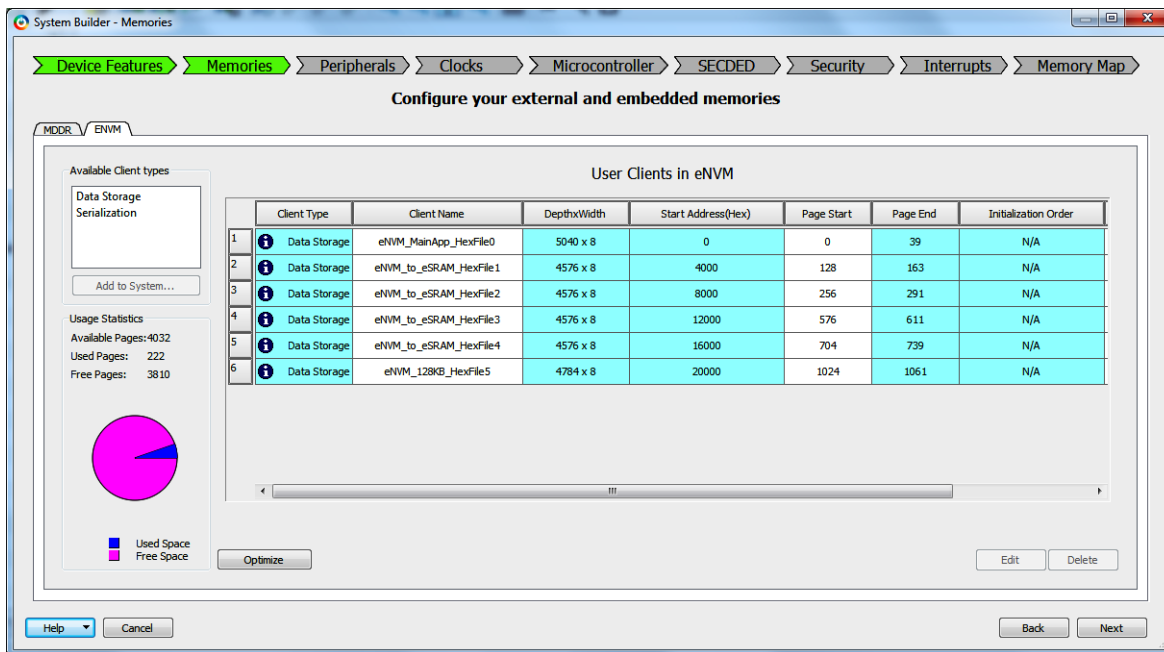
Figure 44 • Error due to Image Overlap

Figure 45 shows the correct memory locations chosen. It does not display any error.

Figure 45 • Hex Files with Correct Start Addresses

8. In the **System Builder - Memories** page, keep the other settings to default and click **Next**, refer to Figure 45.
9. Save **Remapping_Appnote_top** and regenerate the **Remapping_Appnote_0** component by clicking Generate Component in SmartDesign.
10. Click **Generate Bitstream** to complete the remaining steps to generate fdb file (synthesis, place-and-route) as shown in Figure 46, page 31.
11. Click **Run PROGRAM Action** to program the SmartFusion2 device to initialize the eNVM with the memory file as shown in Figure 47, page 32.

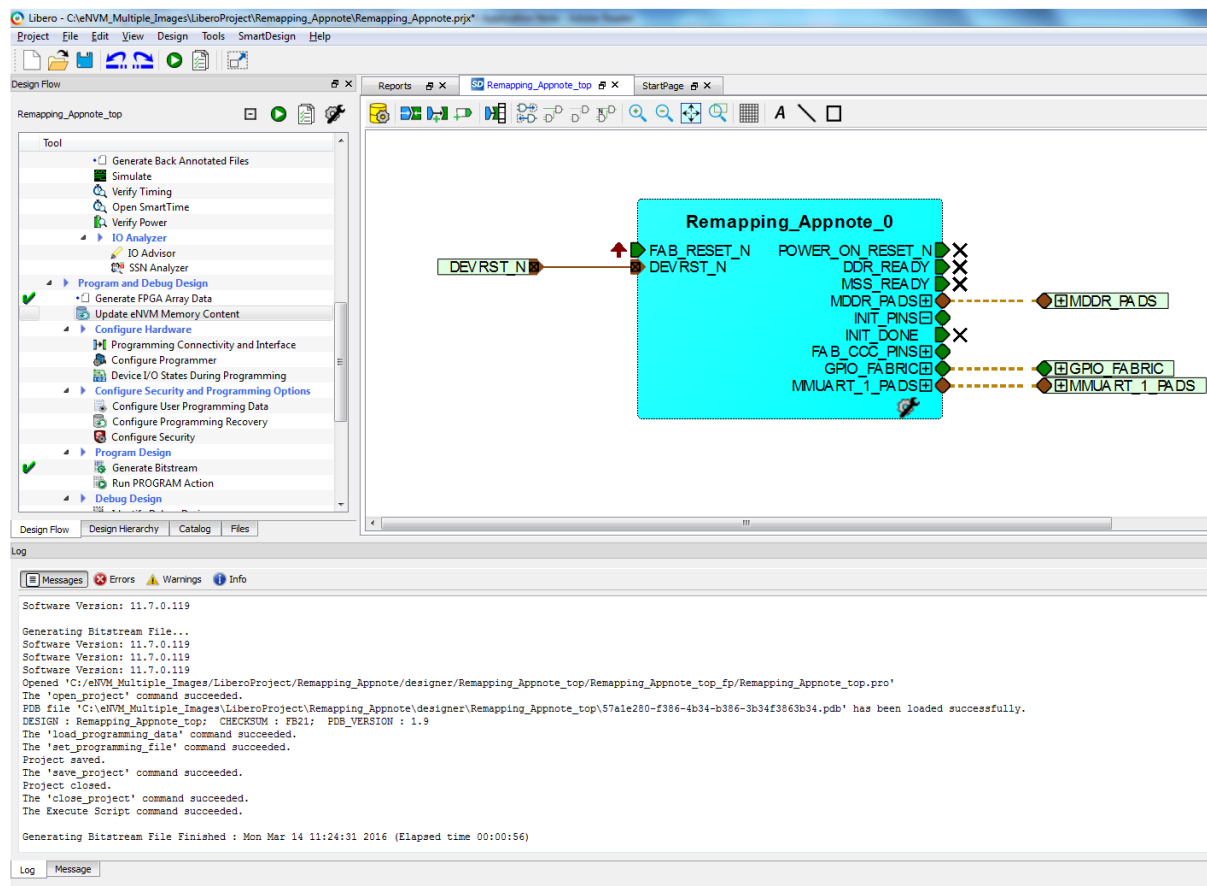
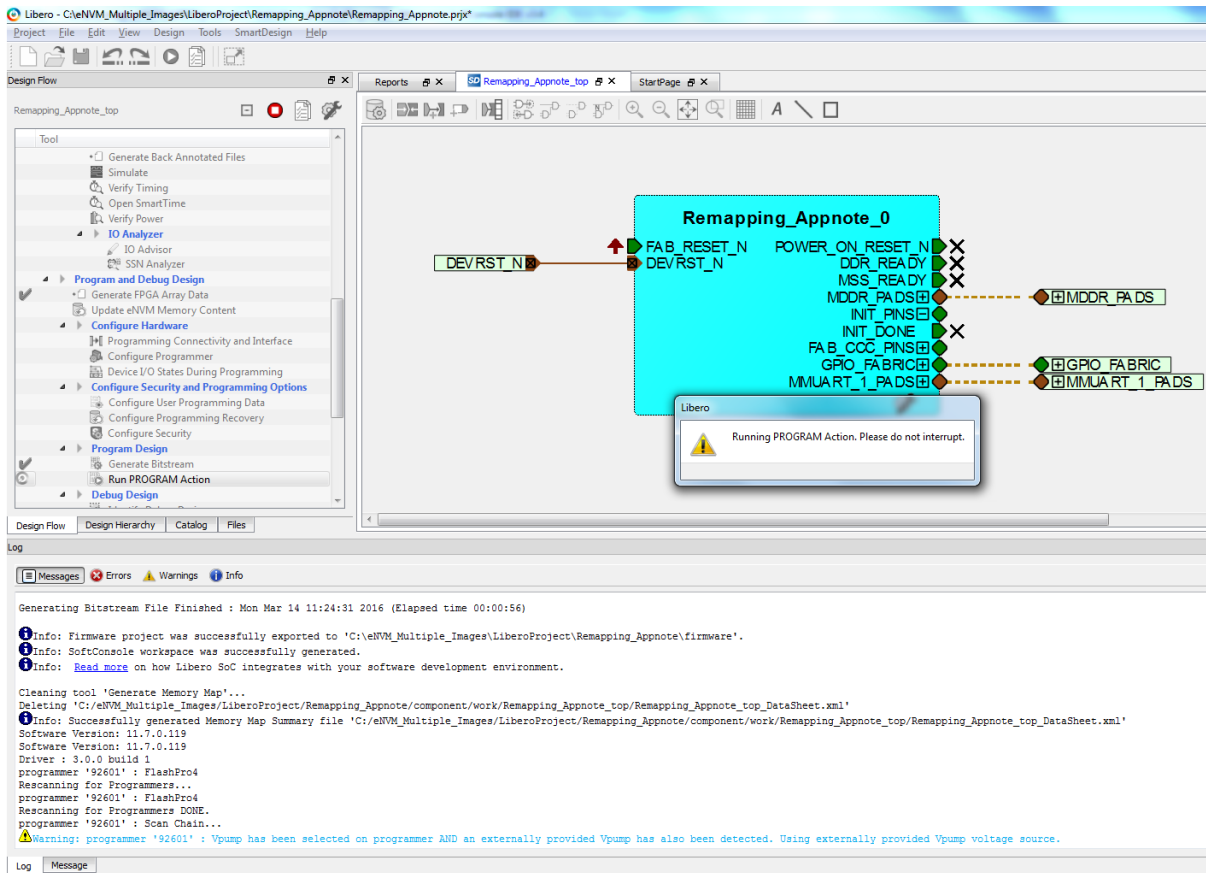
Figure 46 • Regenerating Remapping_Appnote_0 Component

Figure 47 • Programming SmartFusion2 Security Evaluation Kit



2.12 Updating eNVM Memory Content

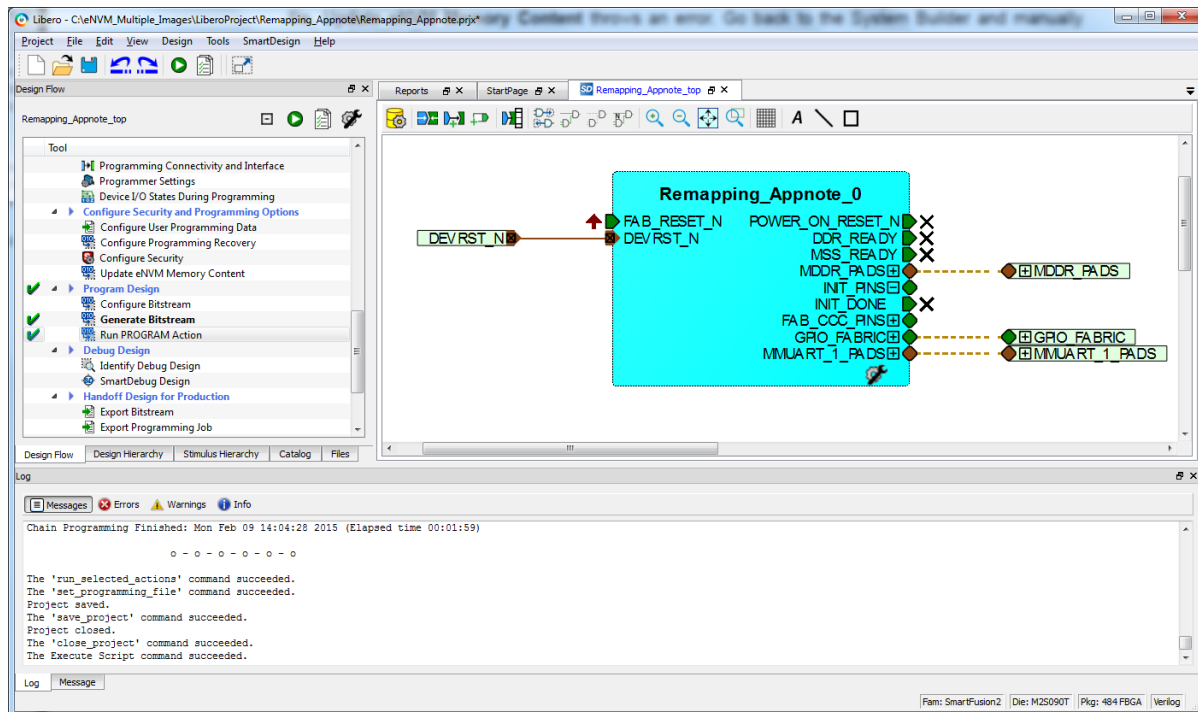
This option is available in Libero design flow to load the updated .hex files after compiling the application code in SoftConsole. Without this option, every time the SoftConsole application is built, System Builder eNVM Configurator is required to update the eNVM client with the new .hex file generated, which may be time consuming.

If the new .hex file generated from the SoftConsole application project is larger than the previous .hex file, **Update eNVM Memory Content** throws an error. Go back to the System Builder and manually update the Hex file.

The following steps describe how to use **Update eNVM Memory Content** option:

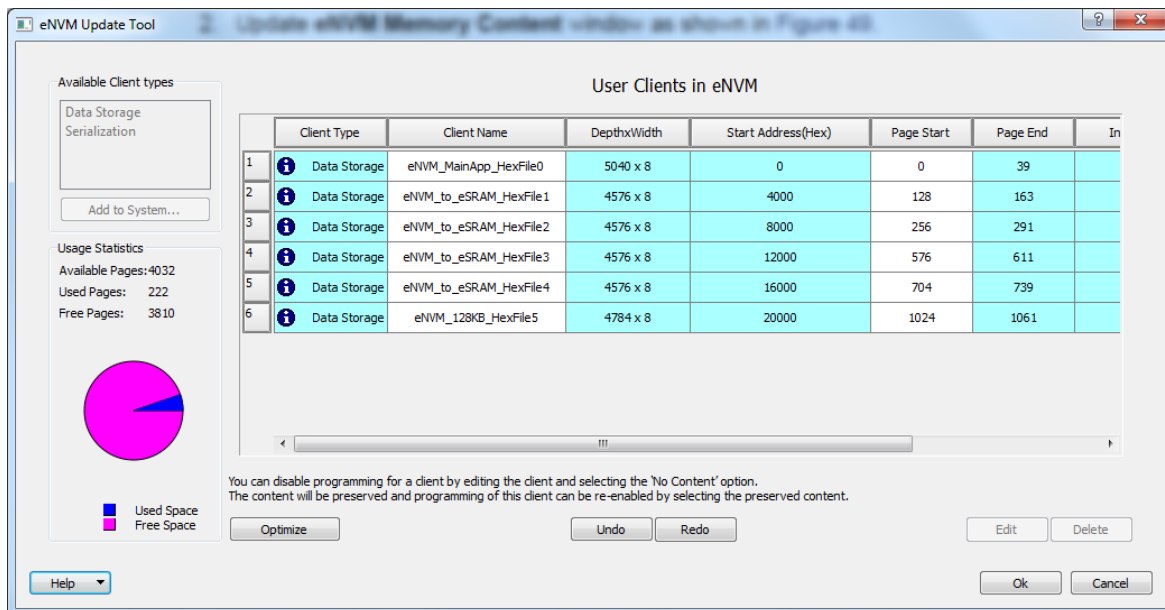
1. Choose **Update eNVM Memory Content** option from Libero design flow.

Figure 48 • Update eNVM Memory Content Option



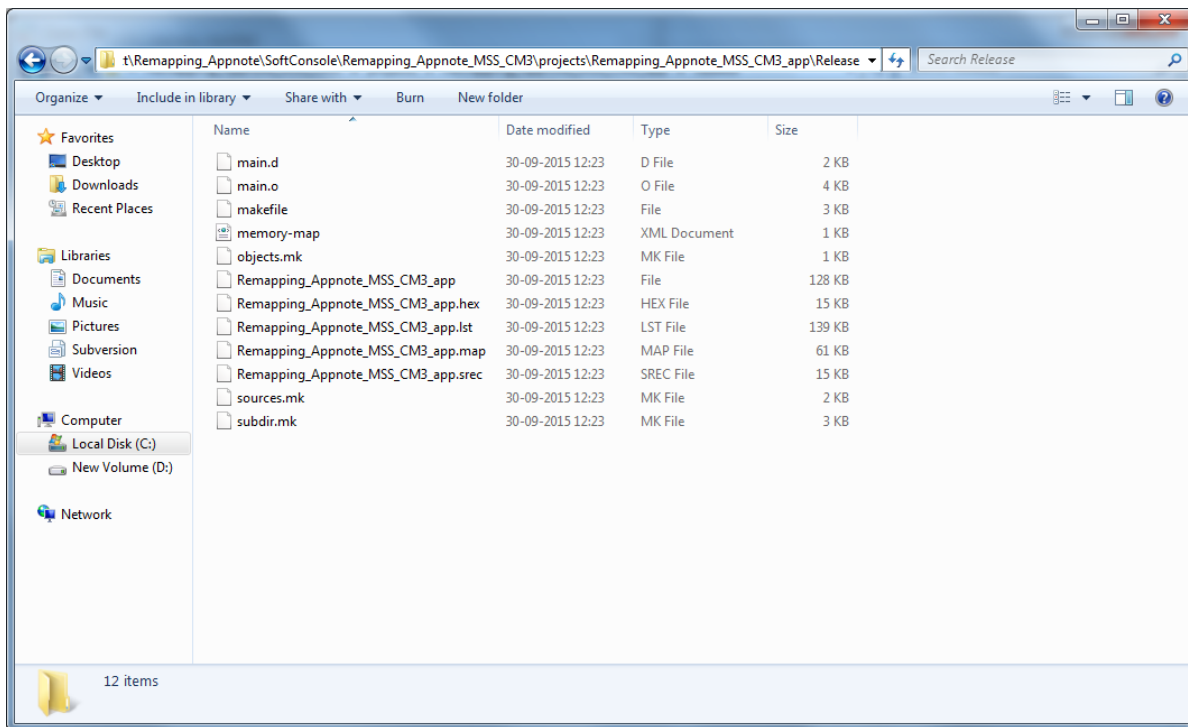
2. Update eNVM Memory Content window, as shown in Figure 49.

Figure 49 • Update eNVM Memory Content Window



3. Browse through the updated .hex file, refer to Figure 50. Click **Ok**.

Figure 50 • Browsing through Updated Hex File



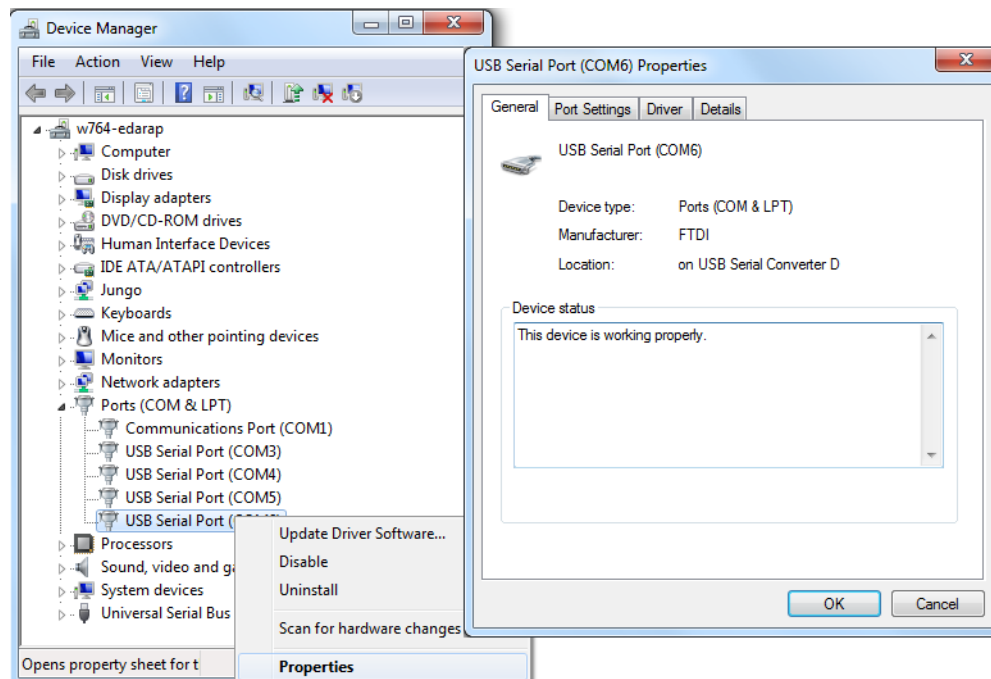
4. Update the .hex files for the remaining data storage clients.
5. Perform **Run PROGRAM Action**.

2.13 Running the Design on SmartFusion2 Security Evaluation Kit

The following steps describe how to run the design:

6. Connect the host PC to the J18 connector provided on the SmartFusion2 Security Evaluation Kit using the USB min-B cable. Ensure that the USB to UART bridge drivers are automatically detected (can be verified in the Device Manager), as shown in Figure 51, page 35.

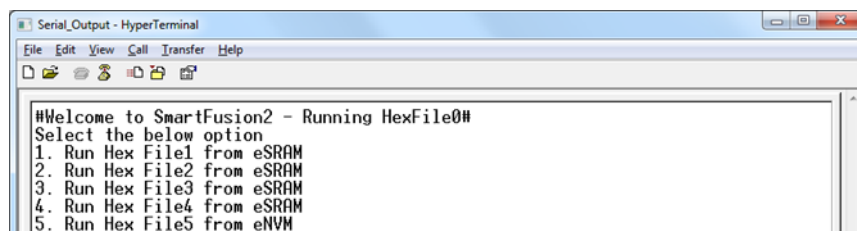
Note: Copy the COM port number for serial port configuration. Ensure that the COM port location is specified as **on USB Serial Converter D**, as shown in Figure 51, page 35.

Figure 51 • Device Manager Window

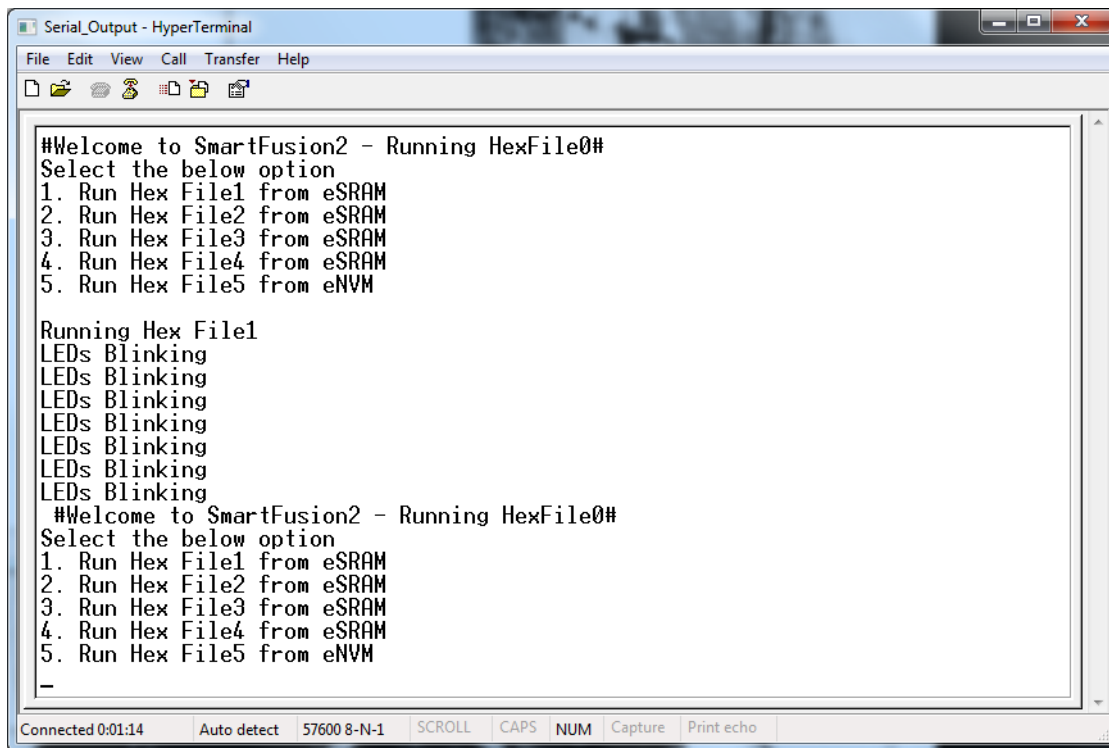
7. If USB to UART bridge drivers are not installed, download and install the drivers from: www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip.
8. Connect the power supply to the J6 connector and switch ON the power supply switch, SW7.
9. Start the HyperTerminal program with a baud rate of 57600, 8 data bits, 1 stop bit, no parity, and no flow control. If the PC does not have HyperTerminal program, use any free serial terminal emulation program such as PuTTY or TeraTerm. For more information about configuring the HyperTerminal, TeraTerm, and PuTTY, refer to [Configuring Serial Terminal Emulation Programs Tutorial](#).
10. Program the SmartFusion2 Security Evaluation kit board with the job file provided as part of the design files using FlashPro Express software, refer to [Appendix 1: Programming the Device Using FlashPro Express](#), page 37.

Note: This step is required if *.job file in the design folders is used. In Libero design, **Run PROGRAM Action** programs the *.job file to the Board.

11. After successful programming, press SW6 switch to reset the board.
12. The serial terminal program displays the user options, as shown in [Figure 52](#).

Figure 52 • Running Hex File - User Options

13. Select option 1 as shown in [Figure 53](#). The application image runs from eSRAM. E1, F4 LEDs blink on the Board.
14. Reset the SW6 pin on SmartFusion2 Security Evaluation Kit, which brings the application to the Main Menu.

Figure 53 • LEDs Blink Based on User Option


```

Serial_Output - HyperTerminal
File Edit View Call Transfer Help

#Welcome to SmartFusion2 - Running HexFile0#
Select the below option
1. Run Hex File1 from eSRAM
2. Run Hex File2 from eSRAM
3. Run Hex File3 from eSRAM
4. Run Hex File4 from eSRAM
5. Run Hex File5 from eNVM

Running Hex File1
LEDs Blinking
LEDs Blinking
LEDs Blinking
LEDs Blinking
LEDs Blinking
LEDs Blinking
LEDs Blinking
#Welcome to SmartFusion2 - Running HexFile0#
Select the below option
1. Run Hex File1 from eSRAM
2. Run Hex File2 from eSRAM
3. Run Hex File3 from eSRAM
4. Run Hex File4 from eSRAM
5. Run Hex File5 from eNVM
-

Connected 0:01:14 Auto detect 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

```

15. Select option 2. The application image runs from eSRAM. F3, G7 LEDs blink on the Board.
16. Reset the SW6 pin on SmartFusion2 Security Evaluation Kit, which brings the application to the Main Menu.
17. Select option 3. The application image runs from eSRAM. H7, J6 LEDs blink on the Board.
18. Reset the SW6 pin on SmartFusion2 Security Evaluation Kit, which brings the application to the Main Menu.
19. Select option 4. The application image runs from eSRAM. H6, H5 LEDs blink on the Board.
20. Reset the SW6 pin on SmartFusion2 Security Evaluation Kit, which brings the application to the Main Menu.
21. Select option 5. The application image runs from eNVM. E1, F4, F3, G7, H7, J6, H6, H5 LEDs blink on the Board.
22. Reset the SW6 pin on SmartFusion2 Security Evaluation Kit, which brings the application to the Main Menu.

2.14 Conclusion

This application note describes the step-by-step approach to build single/multiple SoftConsole application executable images in production release mode, store the executable images into eNVM using eNVM data storage client, and execute them on SmartFusion2 Security Evaluation Kit.

3 Appendix 1: Programming the Device Using FlashPro Express

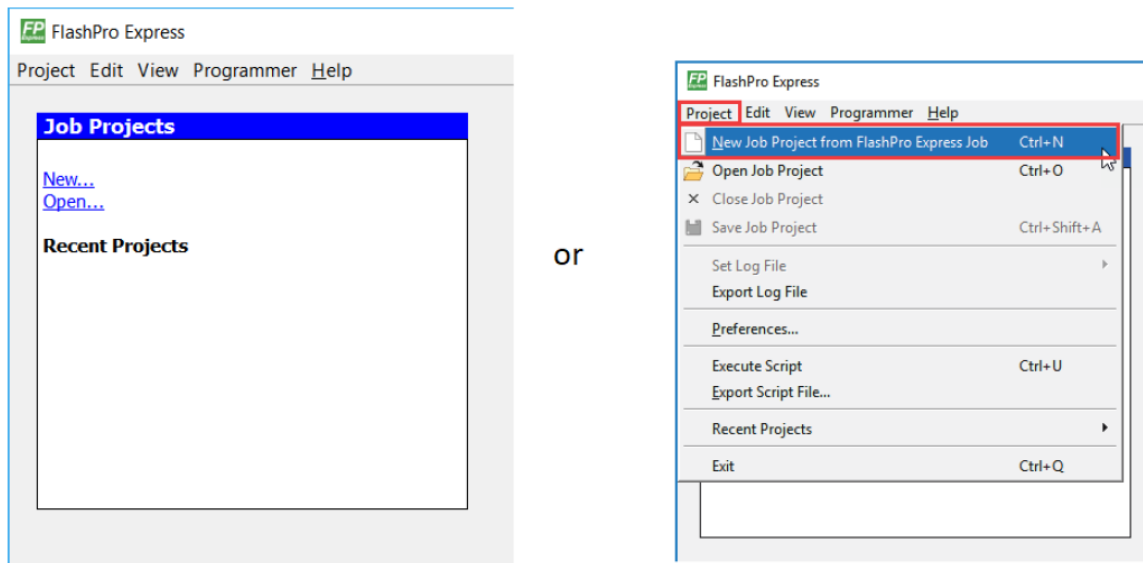
This section describes how to program the SmartFusion2 device with the programming job file using FlashPro Express.

The following steps describe how to program the FlashPro Express:

1. Switch **ON** the power supply switch **SW7**.
2. Launch the FlashPro Express software.
3. Click **New Project**.
4. In the **New Project** window, type the project name as:
 - SmartFusion2: SF2_MDDR_Demo
5. Click **Browse** and navigate to the location where you want to save the project.
6. Select **Single device** as the **Programming mode**.
7. Click **OK**.

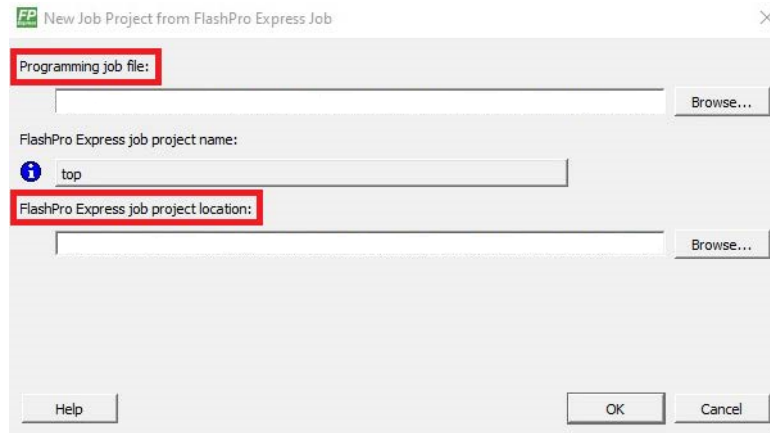
The FlashPro new project is shown in Figure 54.

Figure 54 • FlashPro Express Job Project



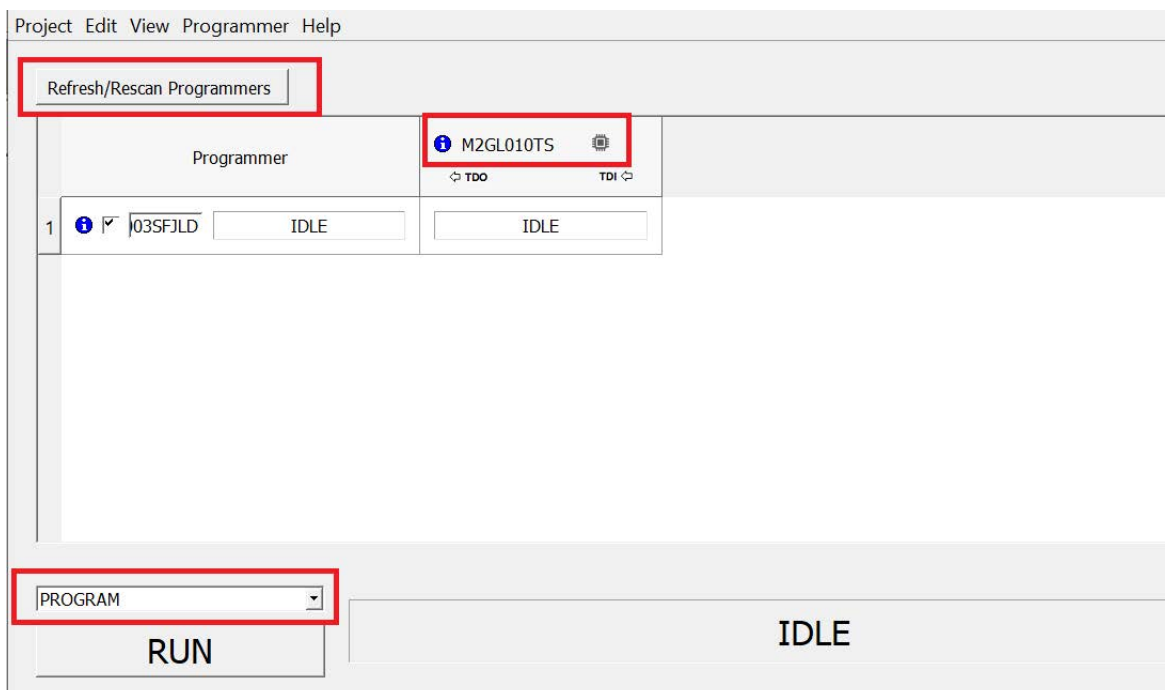
8. Enter the following in the **New Job Project from FlashPro Express Job** dialog box:
 - **Programming job file:** Click **Browse**, and navigate to the location where the .job file is located and select the file. The default location is:
`<download_folder>\m2s_ac426_df\Programming_Job`
 - **FlashPro Express job project name:** Click **Browse** and navigate to the location where you want to save the project.

Figure 55 • New Job Project from FlashPro Express Job

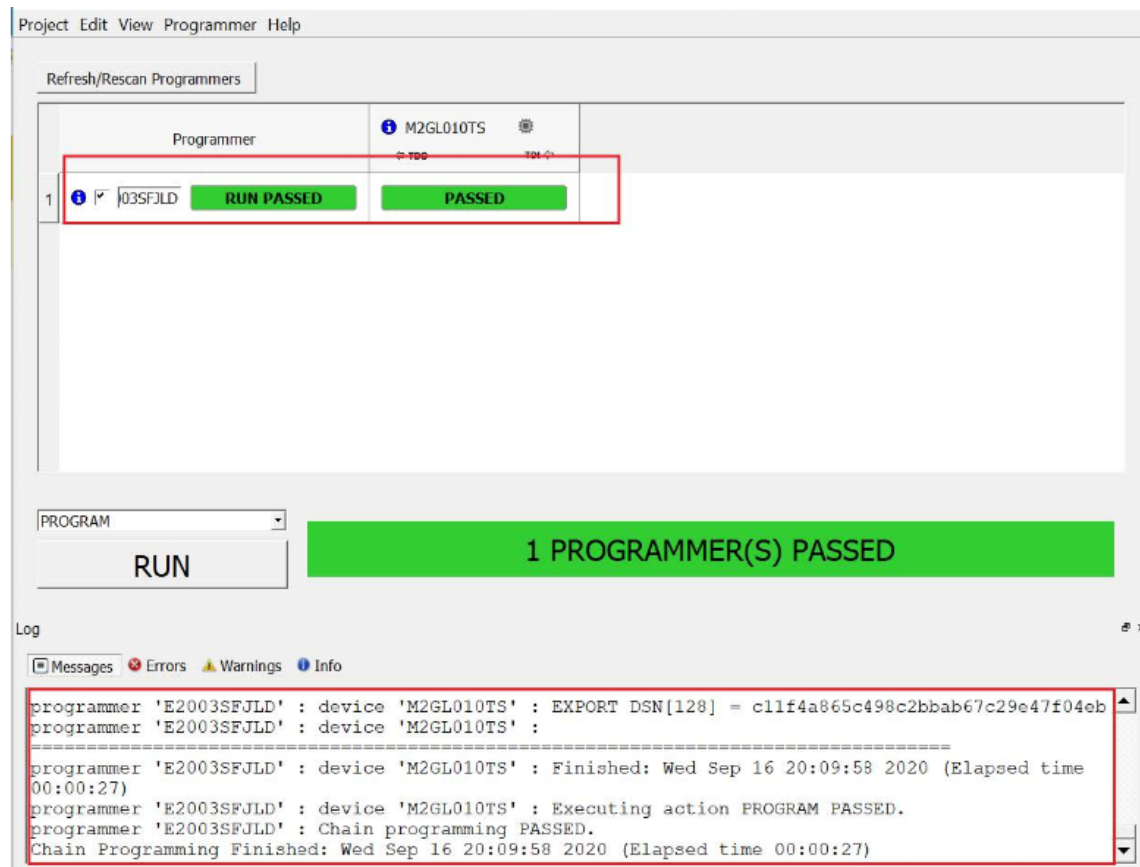


9. Click **OK**. The required programming file is selected and ready to be programmed in the device.
10. The FlashPro Express window appears as shown in Figure 56. Confirm that a programmer number appears in the Programmer field. If it does not, confirm the board connections and click **Refresh/Rescan** Programmers.

Figure 56 • Programming the Device



11. Click **RUN**. When the device is programmed successfully, a **RUN PASSED** status is displayed as shown in Figure 57.

Figure 57 • FlashPro Express—RUN PASSED

12. Close **FlashPro Express** or in the Project tab, click **Exit**.

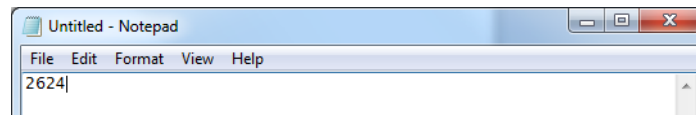
4 Appendix 2: Creating LookUpTable.Hex File

The LookUpTable.Hex file contains the pre-calculated factorial data and this is a non-executable image.

The following steps describe how to create the LookUpTable.Hex file:

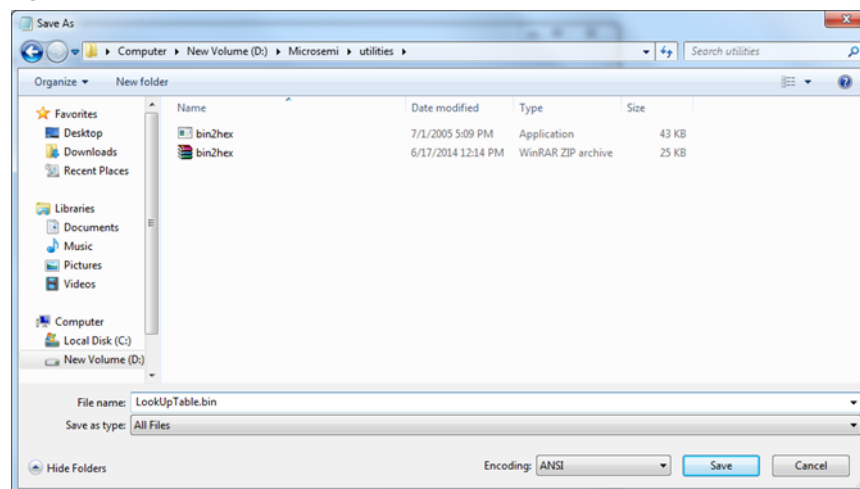
1. Open the notepad and enter the factorial of number 2, 3, 4 as 2, 6, 24 as shown in Figure 58.

Figure 58 • Entering Factorial Values in Notepad



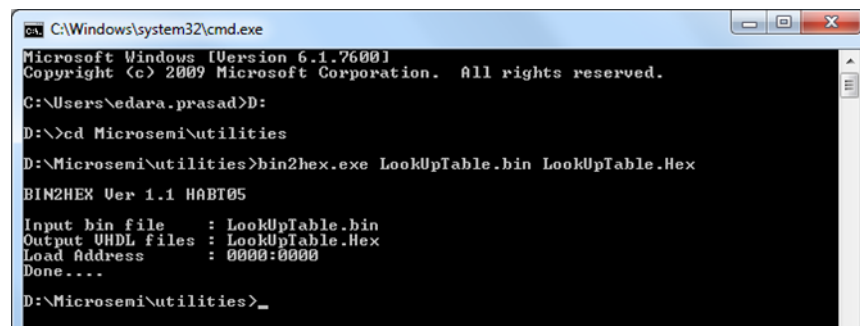
2. Save the notepad as **LookUpTable.bin**, as shown in Figure 59.

Figure 59 • Saving Notepad



3. The executable file, **bin2hex.exe**, converts the LookUpTable.bin, which is in binary format to LookUpTable.Hex in Hex format. This executable file is available in the design files path provided along with this application note. For design files, refer to: [Prerequisites](#), page 3.

Figure 60 • Converting Binary File to Hex File



4. The **LookUpTable.Hex** file is generated, as shown in Figure 61. This file is copied to SoftConsole project folder of the eNVM Data client SoftConsole project.

Figure 61 • LookUpTable.Hex File

