AC426
Application Note
Implementing Production Release Mode Programming for SmartFusion2

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

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

Figures

Table

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: • 12 V adapter (provided along with the kit) • FlashPro4 programmer • USB A to Mini-B cable	Rev D or later
Host PC or Laptop	
Software	
FlashPro Express	**Note:** Refer to the `readme.txt` 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:

1. 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

2. For demo design files download link:

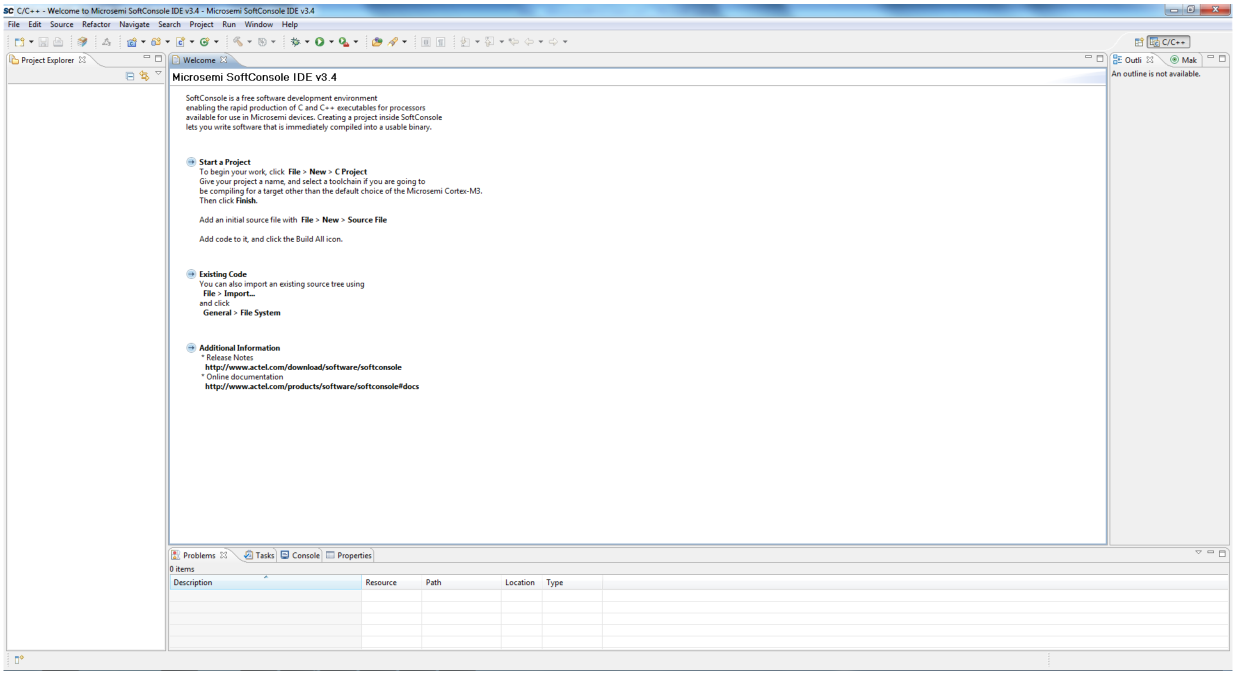
   http://soc.microsemi.com/download/rsc/?f=m2s_ac426_df

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**


SC C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
C/C++
Project Explorer
Welcome
Outli
Mak
An outline is not available.
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
Location
Type

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment
enabling the rapid production of C and C++ executables for processors
available for use in Microsemi devices. Creating a project inside SoftConsole
lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to
be compiling for a target other than the default choice of the Microsemi Cortex-M3.
Then click **Finish.**

Add an initial source file with **File** > **New** > **Source File**

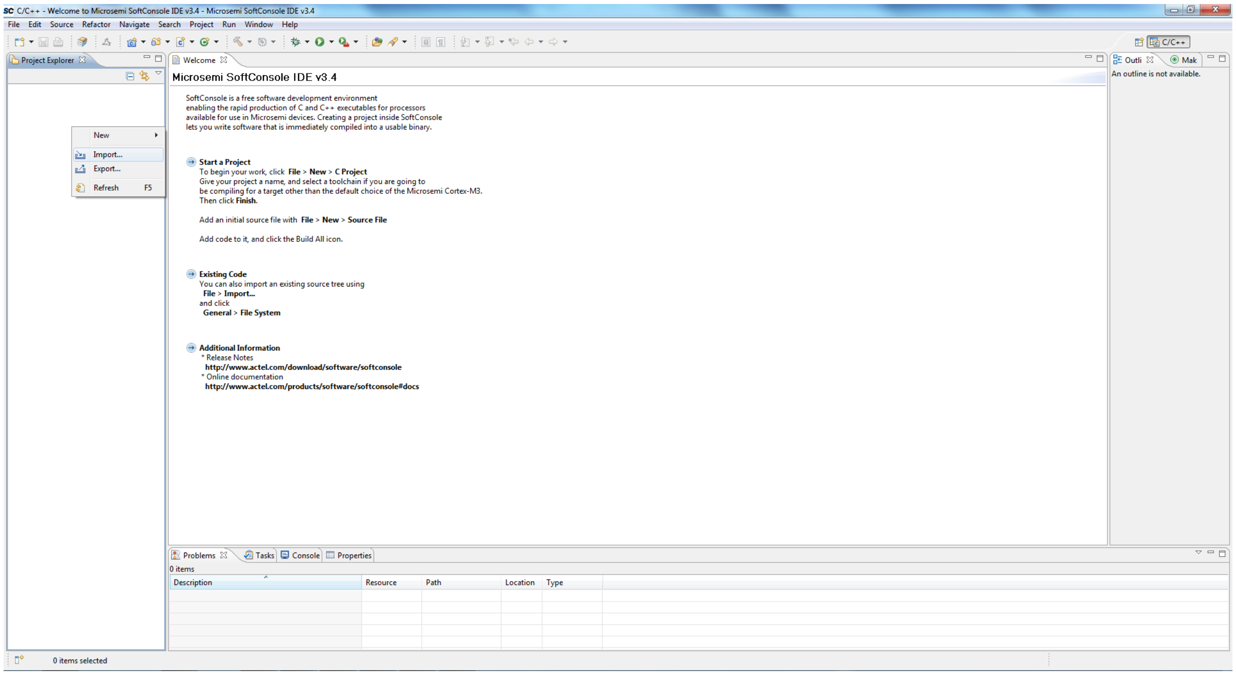
Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

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

Figure 2 • **SoftConsole IDE - Import**


SC C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
C/C++
Project Explorer
Welcome
New
Import...
Export...
Refresh F5
Outli
Mak
An outline is not available.
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
Location
Type
0 items selected

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment
enabling the rapid production of C and C++ executables for processors
available for use in Microsemi devices. Creating a project inside SoftConsole
lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to
be compiling for a target other than the default choice of the Microsemi Cortex-M3.
Then click **Finish.**

Add an initial source file with **File** > **New** > **Source File**

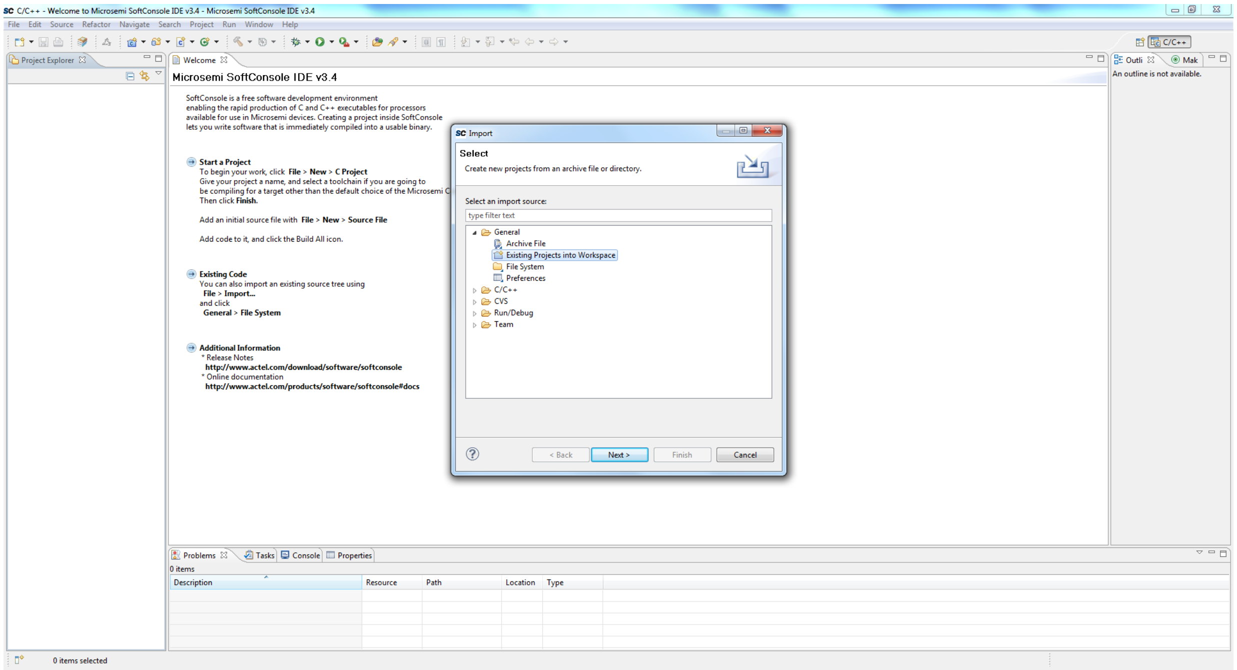
Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

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

Figure 3 • **SoftConsole IDE - Importing Existing Projects**



SC C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Welcome
Outli
Mak
C/C++
Problems
Tasks
Console
Properties
Description
Resource
Path
Location
Type
Import
< Back
Next >
Finish
Cancel

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Microsemi devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to be compiling for a target other than the default choice of the Microsemi C
Then click **Finish**.

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

An outline is not available.

0 items

0 items selected

Select

Create new projects from an archive file or directory.

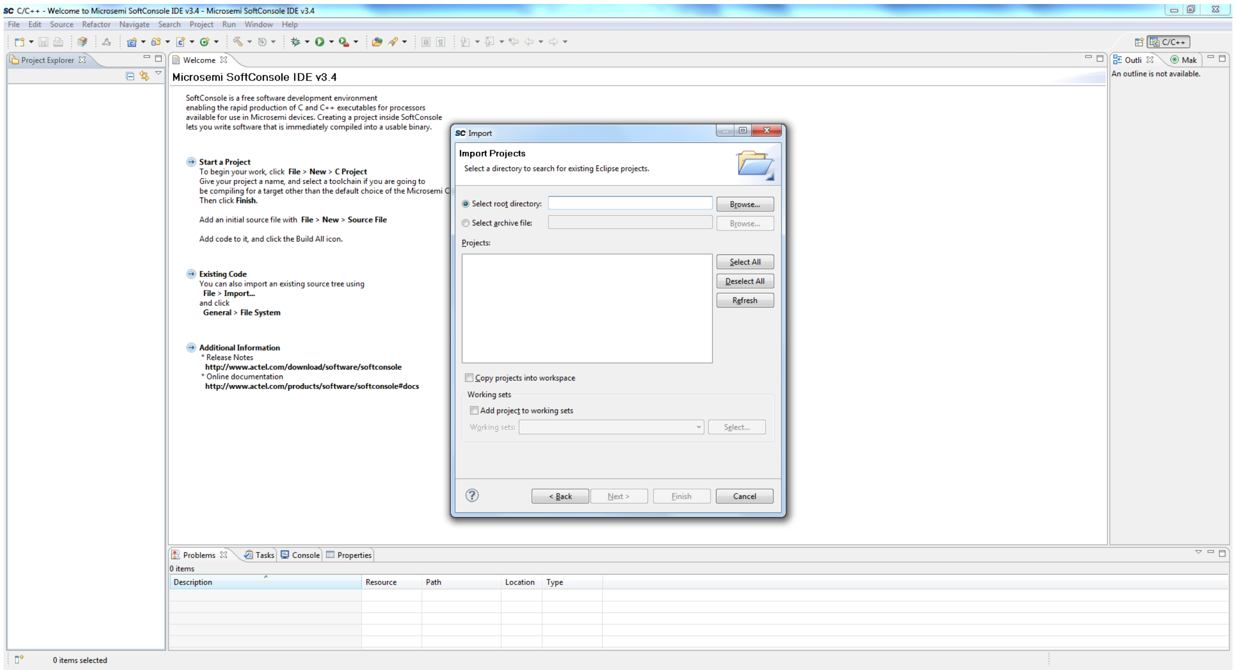
Select an import source:

type filter text

- General
  - Archive File
  - Existing Projects into Workspace
  - File System
  - Preferences
- C/C++
- CVS
- Run/Debug
- Team

Import window is displayed, as shown in Figure 4.

Figure 4 • **SoftConsole IDE - Browse**



Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Microsemi devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to be compiling for a target other than the default choice of the Microsemi C
Then click **Finish**.

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

An outline is not available.

0 items

0 items selected

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory:

○ Select archive file:

Projects:

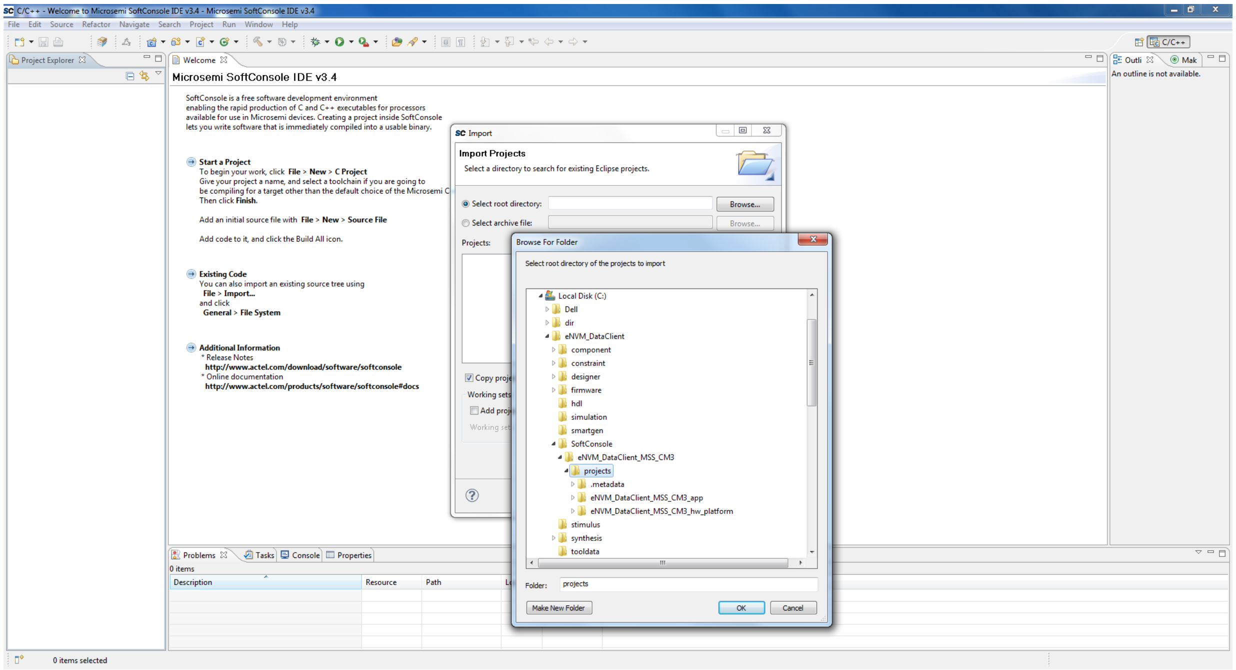
□ Copy projects into workspace

Working sets

□ Add project to working sets

Working sets:

Figure 5 • **SoftConsole IDE - Browsing for Projects Folder**



C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Welcome
C/C++
Outli
Mak
An outline is not available.
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
0 items selected

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment
enabling the rapid production of C and C++ executables for processors
available for use in Microsemi devices. Creating a project inside SoftConsole
lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to
be compiling for a target other than the default choice of the Microsemi C
Then click **Finish.**

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

Import

Import Projects
Select a directory to search for existing Eclipse projects.

● Select root directory: Browse...
○ Select archive file: Browse...
Projects:
☑ Copy proj
Working sets
□ Add proj
Working set

Browse For Folder

Select root directory of the projects to import

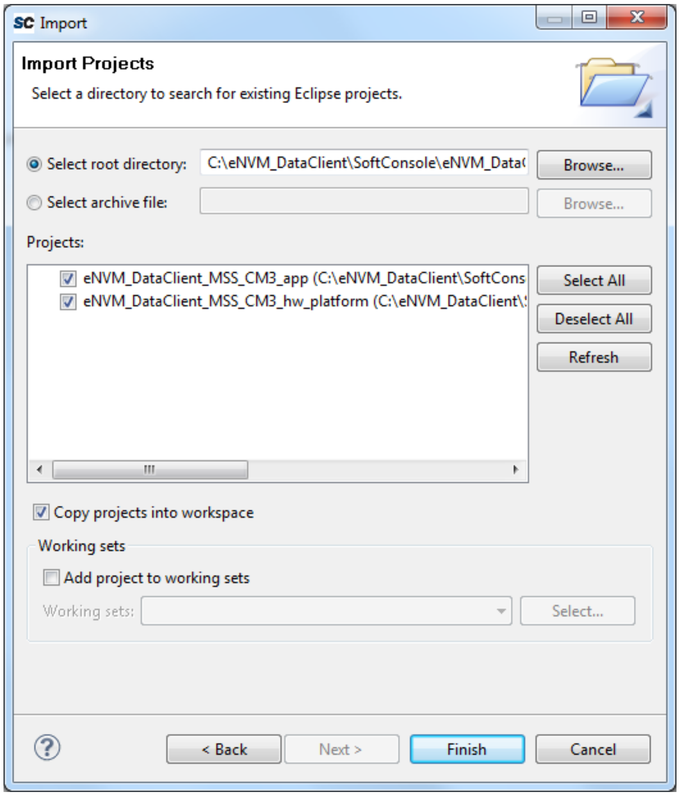
```
Local Disk (C:)
  Dell
  dir
  eNVM_DataClient
    component
    constraint
    designer
    firmware
    hdl
    simulation
    smartgen
    SoftConsole
      eNVM_DataClient_MSS_CM3
        projects
          .metadata
          eNVM_DataClient_MSS_CM3_app
          eNVM_DataClient_MSS_CM3_hw_platform
    stimulus
    synthesis
    tooldata
```

Folder: projects

Make New Folder | OK | Cancel

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**



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: C:\eNVM_DataClient\SoftConsole\eNVM_Data(

○ Select archive file:

Projects:

☑ eNVM_DataClient_MSS_CM3_app (C:\eNVM_DataClient\SoftCons

☑ eNVM_DataClient_MSS_CM3_hw_platform (C:\eNVM_DataClient\

☑ Copy projects into workspace

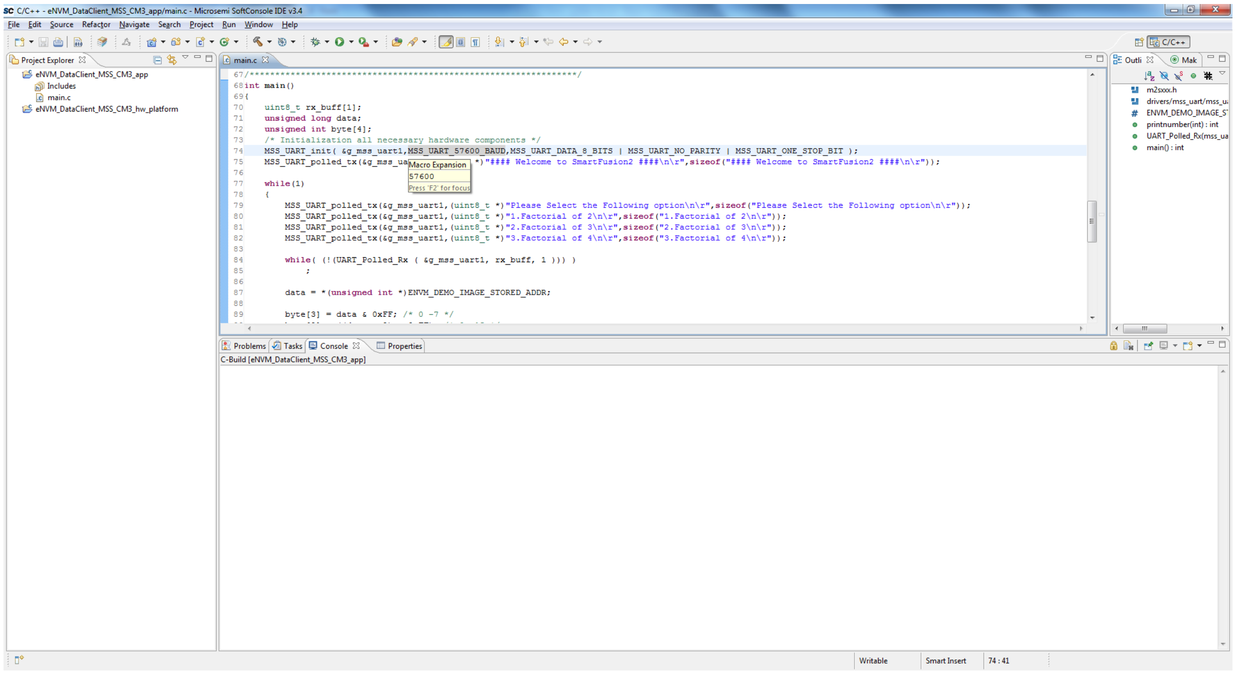
Working sets

☐ Add project to working sets

Working sets:

The SoftConsole perspective is shown in Figure 7.

Figure 7 • **SoftConsole Workspace**



C/C++ - eNVM_DataClient_MSS_CM3_app/main.c - Microsemi SoftConsole IDE v3.4

Project Explorer:

- eNVM_DataClient_MSS_CM3_app
  - Includes
  - main.c
- eNVM_DataClient_MSS_CM3_hw_platform

main.c:

```
/*******************************************************************/
int main()
{
    uint8_t rx_buff[1];
    unsigned long data;
    unsigned int byte[4];
    /* Initialization all necessary hardware components */
    MSS_UART_init( &g_mss_uart1,MSS_UART_57600_BAUD,MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT );
    MSS_UART_polled_tx(&g_mss_ua         *)"#### Welcome to SmartFusion2 ####\n\r",sizeof("#### Welcome to SmartFusion2 ####\n\r"));

    while(1)
    {
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"Please Select the Following option\n\r",sizeof("Please Select the Following option\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"1.Factorial of 2\n\r",sizeof("1.Factorial of 2\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"2.Factorial of 3\n\r",sizeof("2.Factorial of 3\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"3.Factorial of 4\n\r",sizeof("3.Factorial of 4\n\r"));

        while( (!(UART_Polled_Rx ( &g_mss_uart1, rx_buff, 1 ))) )
            ;

        data = *(unsigned int *)ENVM_DEMO_IMAGE_STORED_ADDR;

        byte[3] = data & 0xFF; /* 0 -7 */
```

Macro Expansion
57600
Press 'F2' for focus

Outline:

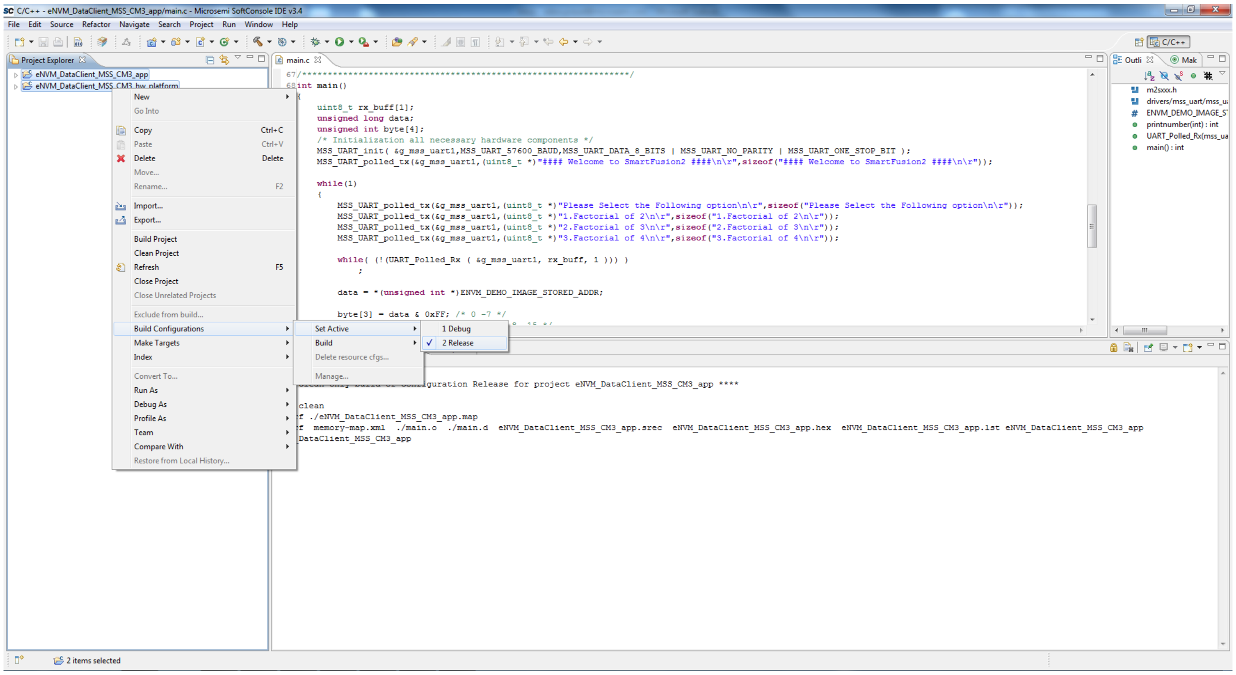
- m2sxxx.h
- drivers/mss_uart/mss_u
- ENVM_DEMO_IMAGE_S
- printnumber(int) : int
- UART_Polled_Rx(mss_ua
- main() : int

Console:

C-Build [eNVM_DataClient_MSS_CM3_app]

5. 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**

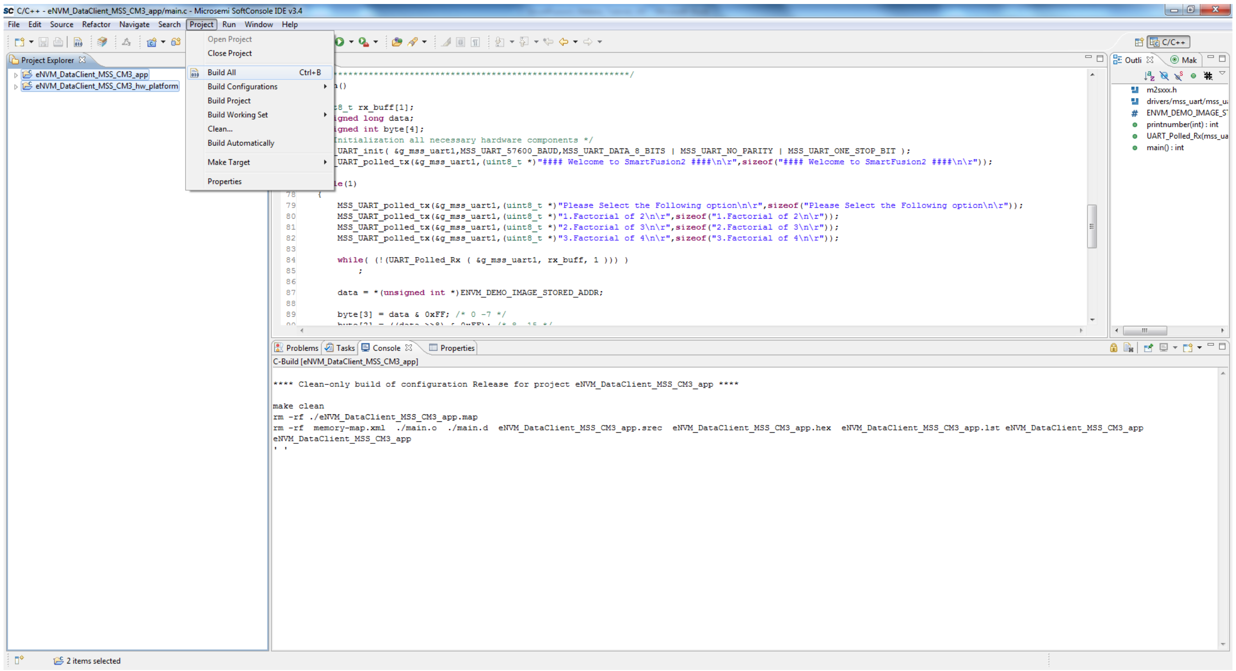


```
67/*****************************************************************/
68int main()
{
    uint8_t rx_buff[1];
    unsigned long data;
    unsigned int byte[4];
    /* Initialization all necessary hardware components */
    MSS_UART_init( &g_mss_uart1,MSS_UART_57600_BAUD,MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT );
    MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"#### Welcome to SmartFusion2 ####\n\r",sizeof("#### Welcome to SmartFusion2 ####\n\r"));

    while(1)
    {
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"Please Select the Following option\n\r",sizeof("Please Select the Following option\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"1.Factorial of 2\n\r",sizeof("1.Factorial of 2\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"2.Factorial of 3\n\r",sizeof("2.Factorial of 3\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"3.Factorial of 4\n\r",sizeof("3.Factorial of 4\n\r"));

        while( (!(UART_Polled_Rx ( &g_mss_uart1, rx_buff, 1 ))) )
            ;

        data = *(unsigned int *)ENVM_DEMO_IMAGE_STORED_ADDR;

        byte[3] = data & 0xFF; /* 0 -7 */
```

```
guration Release for project eNVM_DataClient_MSS_CM3_app ****

clean
f ./eNVM_DataClient_MSS_CM3_app.map
f  memory-map.xml  ./main.o  ./main.d  eNVM_DataClient_MSS_CM3_app.srec  eNVM_DataClient_MSS_CM3_app.hex  eNVM_DataClient_MSS_CM3_app.lst eNVM_DataClient_MSS_CM3_app
DataClient_MSS_CM3_app
```

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

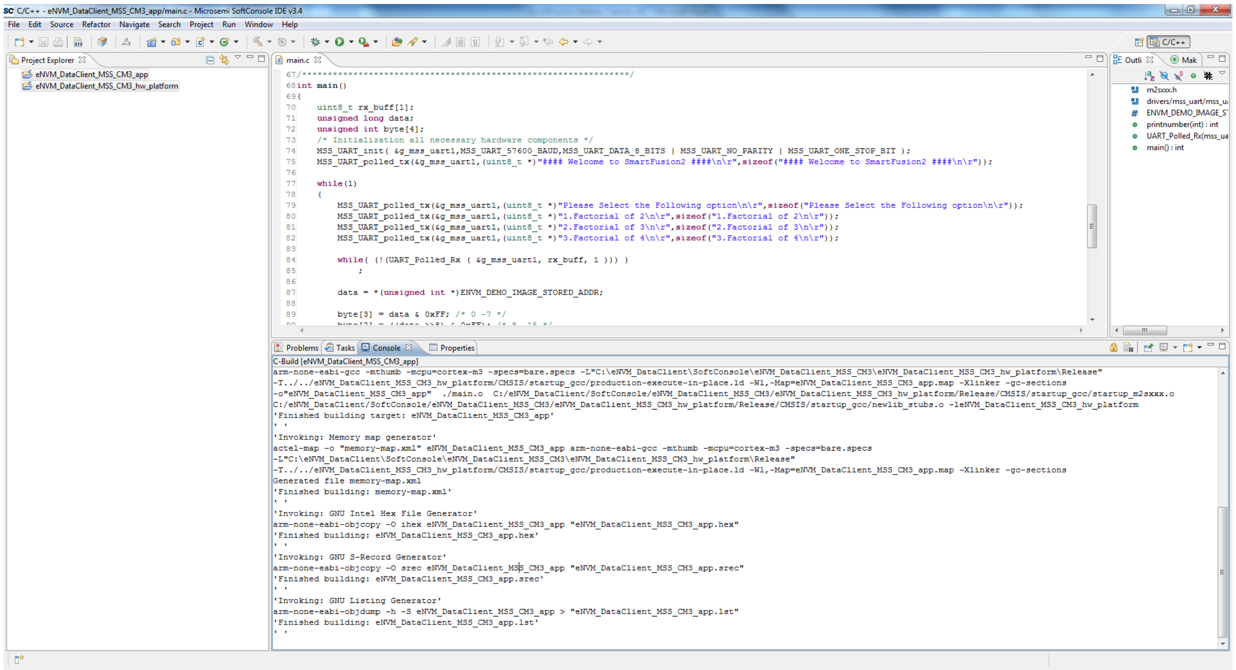
Figure 9 • **Build All**



```
    {
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"Please Select the Following option\n\r",sizeof("Please Select the Following option\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"1.Factorial of 2\n\r",sizeof("1.Factorial of 2\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"2.Factorial of 3\n\r",sizeof("2.Factorial of 3\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"3.Factorial of 4\n\r",sizeof("3.Factorial of 4\n\r"));

        while( (!(UART_Polled_Rx ( &g_mss_uart1, rx_buff, 1 ))) )
            ;

        data = *(unsigned int *)ENVM_DEMO_IMAGE_STORED_ADDR;

        byte[3] = data & 0xFF; /* 0 -7 */
```

```
C-Build [eNVM_DataClient_MSS_CM3_app]

**** Clean-only build of configuration Release for project eNVM_DataClient_MSS_CM3_app ****

make clean
rm -rf ./eNVM_DataClient_MSS_CM3_app.map
rm -rf  memory-map.xml  ./main.o  ./main.d  eNVM_DataClient_MSS_CM3_app.srec  eNVM_DataClient_MSS_CM3_app.hex  eNVM_DataClient_MSS_CM3_app.lst eNVM_DataClient_MSS_CM3_app
eNVM_DataClient_MSS_CM3_app
' '
```

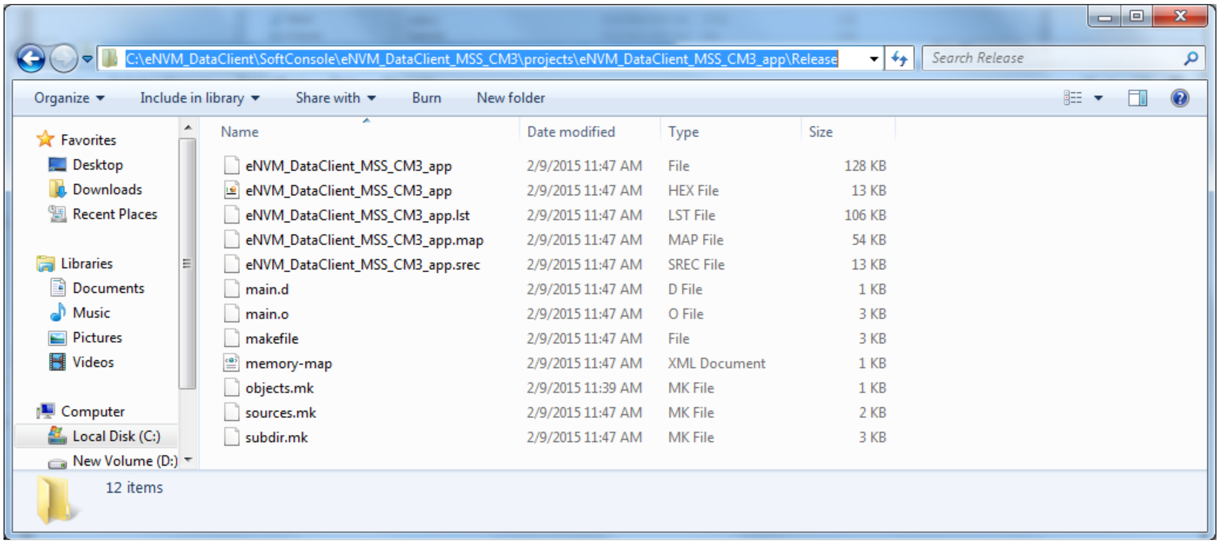
Figure 10 • **Generating Hex File**


C/C++ - eNVM_DataClient_MSS_CM3_app/main.c - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
eNVM_DataClient_MSS_CM3_app
eNVM_DataClient_MSS_CM3_hw_platform
main.c
Outli
Mak
m2sxxx.h
drivers/mss_uart/mss_u
ENVM_DEMO_IMAGE_S
printnumber(int) : int
UART_Polled_Rx(mss_ua
main() : int
Problems
Tasks
Console
Properties
C/C++

```
/*******************************************************************/
int main()
{
    uint8_t rx_buff[1];
    unsigned long data;
    unsigned int byte[4];
    /* Initialization all necessary hardware components */
    MSS_UART_init( &g_mss_uart1,MSS_UART_57600_BAUD,MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT );
    MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"#### Welcome to SmartFusion2 ####\n\r",sizeof("#### Welcome to SmartFusion2 ####\n\r"));

    while(1)
    {
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"Please Select the Following option\n\r",sizeof("Please Select the Following option\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"1.Factorial of 2\n\r",sizeof("1.Factorial of 2\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"2.Factorial of 3\n\r",sizeof("2.Factorial of 3\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"3.Factorial of 4\n\r",sizeof("3.Factorial of 4\n\r"));

        while( (!(UART_Polled_Rx ( &g_mss_uart1, rx_buff, 1 ))) )
            ;

        data = *(unsigned int *)ENVM_DEMO_IMAGE_STORED_ADDR;

        byte[3] = data & 0xFF; /* 0 -7 */
```

```
C-Build [eNVM_DataClient_MSS_CM3_app]
arm-none-eabi-gcc -mthumb -mcpu=cortex-m3 -specs=bare.specs -L"C:\eNVM_DataClient\SoftConsole\eNVM_DataClient_MSS_CM3\eNVM_DataClient_MSS_CM3_hw_platform\Release"
-T../../eNVM_DataClient_MSS_CM3_hw_platform/CMSIS/startup_gcc/production-execute-in-place.ld -Wl,-Map=eNVM_DataClient_MSS_CM3_app.map -Xlinker -gc-sections
-o"eNVM_DataClient_MSS_CM3_app"  ./main.o  C:/eNVM_DataClient/SoftConsole/eNVM_DataClient_MSS_CM3/eNVM_DataClient_MSS_CM3_hw_platform/Release/CMSIS/startup_gcc/startup_m2sxxx.o
C:/eNVM_DataClient/SoftConsole/eNVM_DataClient_MSS_CM3/eNVM_DataClient_MSS_CM3_hw_platform/Release/CMSIS/startup_gcc/newlib_stubs.o -leNVM_DataClient_MSS_CM3_hw_platform
'Finished building target: eNVM_DataClient_MSS_CM3_app'
' '
'Invoking: Memory map generator'
actel-map -o "memory-map.xml" eNVM_DataClient_MSS_CM3_app arm-none-eabi-gcc -mthumb -mcpu=cortex-m3 -specs=bare.specs
-L"C:\eNVM_DataClient\SoftConsole\eNVM_DataClient_MSS_CM3\eNVM_DataClient_MSS_CM3_hw_platform\Release"
-T../../eNVM_DataClient_MSS_CM3_hw_platform/CMSIS/startup_gcc/production-execute-in-place.ld -Wl,-Map=eNVM_DataClient_MSS_CM3_app.map -Xlinker -gc-sections
Generated file memory-map.xml
'Finished building: memory-map.xml'
' '
'Invoking: GNU Intel Hex File Generator'
arm-none-eabi-objcopy -O ihex eNVM_DataClient_MSS_CM3_app "eNVM_DataClient_MSS_CM3_app.hex"
'Finished building: eNVM_DataClient_MSS_CM3_app.hex'
' '
'Invoking: GNU S-Record Generator'
arm-none-eabi-objcopy -O srec eNVM_DataClient_MSS_CM3_app "eNVM_DataClient_MSS_CM3_app.srec"
'Finished building: eNVM_DataClient_MSS_CM3_app.srec'
' '
'Invoking: GNU Listing Generator'
arm-none-eabi-objdump -h -S eNVM_DataClient_MSS_CM3_app > "eNVM_DataClient_MSS_CM3_app.lst"
'Finished building: eNVM_DataClient_MSS_CM3_app.lst'
' '
```

Figure 11 • **Hex File in Release Folder**



C:\eNVM_DataClient\SoftConsole\eNVM_DataClient_MSS_CM3\projects\eNVM_DataClient_MSS_CM3_app\Release

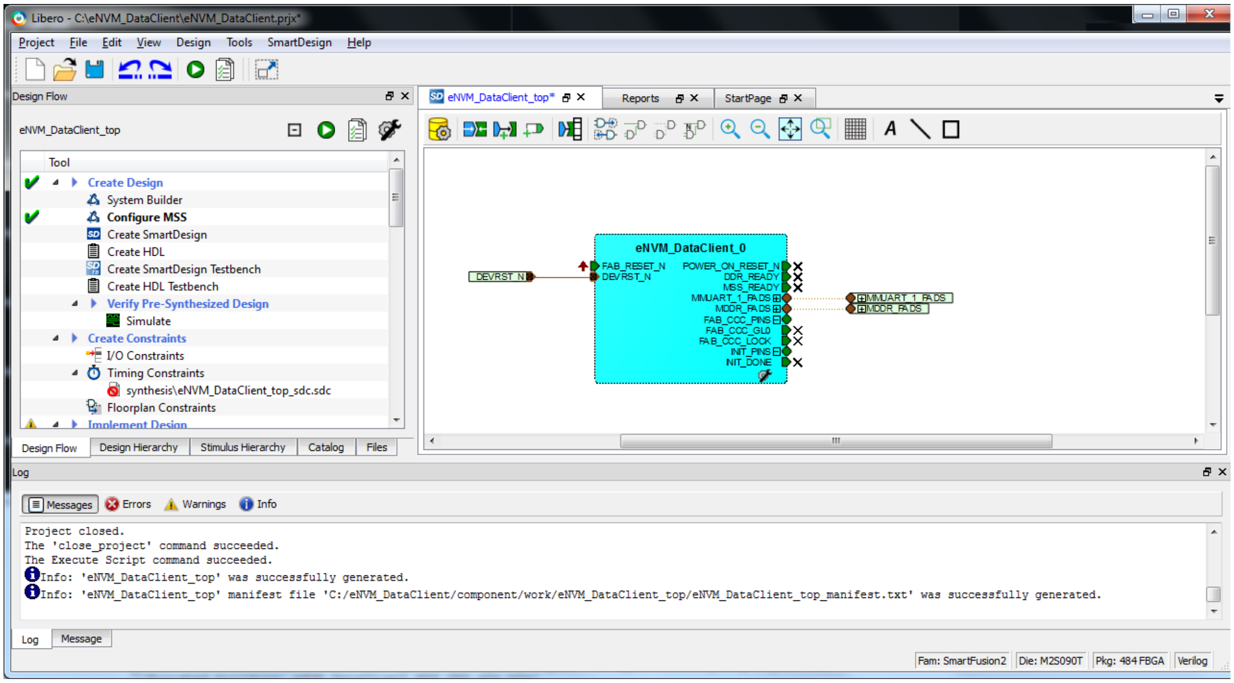
Name	Date modified	Type	Size
eNVM_DataClient_MSS_CM3_app	2/9/2015 11:47 AM	File	128 KB
eNVM_DataClient_MSS_CM3_app	2/9/2015 11:47 AM	HEX File	13 KB
eNVM_DataClient_MSS_CM3_app.lst	2/9/2015 11:47 AM	LST File	106 KB
eNVM_DataClient_MSS_CM3_app.map	2/9/2015 11:47 AM	MAP File	54 KB
eNVM_DataClient_MSS_CM3_app.srec	2/9/2015 11:47 AM	SREC File	13 KB
main.d	2/9/2015 11:47 AM	D File	1 KB
main.o	2/9/2015 11:47 AM	O File	3 KB
makefile	2/9/2015 11:47 AM	File	3 KB
memory-map	2/9/2015 11:47 AM	XML Document	1 KB
objects.mk	2/9/2015 11:39 AM	MK File	1 KB
sources.mk	2/9/2015 11:47 AM	MK File	2 KB
subdir.mk	2/9/2015 11:47 AM	MK File	3 KB

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.

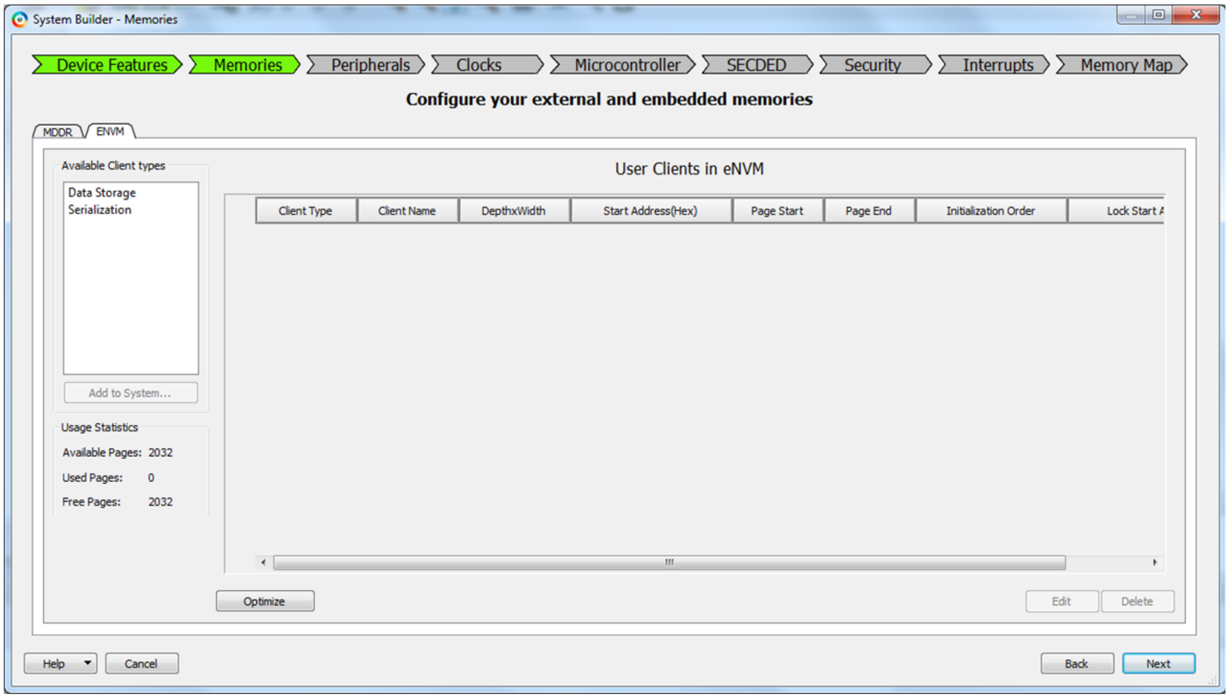
Figure 12 • **eNVM_DataClient_top Tab**



```
Project closed.
The 'close_project' command succeeded.
The Execute Script command succeeded.
Info: 'eNVM_DataClient_top' was successfully generated.
Info: 'eNVM_DataClient_top' manifest file 'C:/eNVM_DataClient/component/work/eNVM_DataClient_top/eNVM_DataClient_top_manifest.txt' was successfully generated.
```

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**



Configure your external and embedded memories

Available Client types

- Data Storage
- Serialization

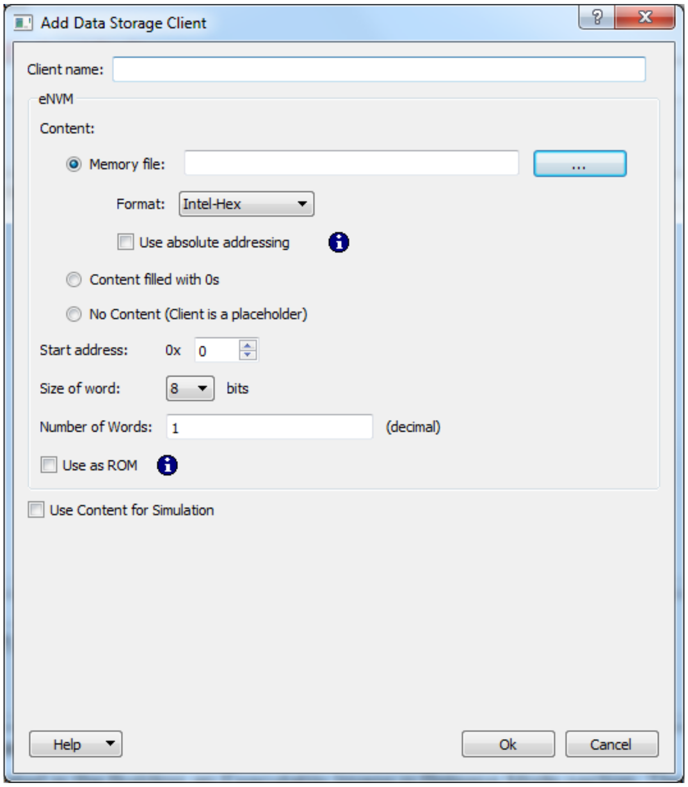
User Clients in eNVM

Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order	Lock Start A

Usage Statistics

Available Pages:	2032
Used Pages:	0
Free Pages:	2032

Figure 14 • **Add Data Storage Client Window**



Client name:

eNVM

Content:

● Memory file:

Format: Intel-Hex

□ Use absolute addressing

○ Content filled with 0s

○ No Content (Client is a placeholder)

Start address: 0x 0

Size of word: 8 bits

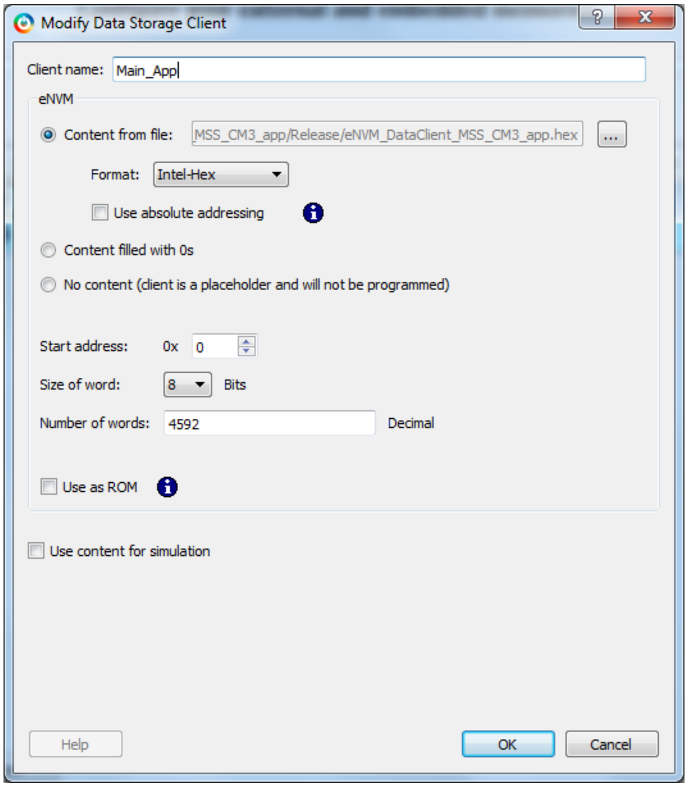
Number of Words: 1 (decimal)

□ Use as ROM

□ Use Content for Simulation

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

Figure 15 • **Add Data Storage Client Window**



Modify Data Storage Client

Client name: Main_App

eNVM

● Content from file: MSS_CM3_app/Release/eNVM_DataClient_MSS_CM3_app.hex

Format: Intel-Hex

□ Use absolute addressing

○ Content filled with 0s

○ No content (client is a placeholder and will not be programmed)

Start address: 0x 0

Size of word: 8 Bits

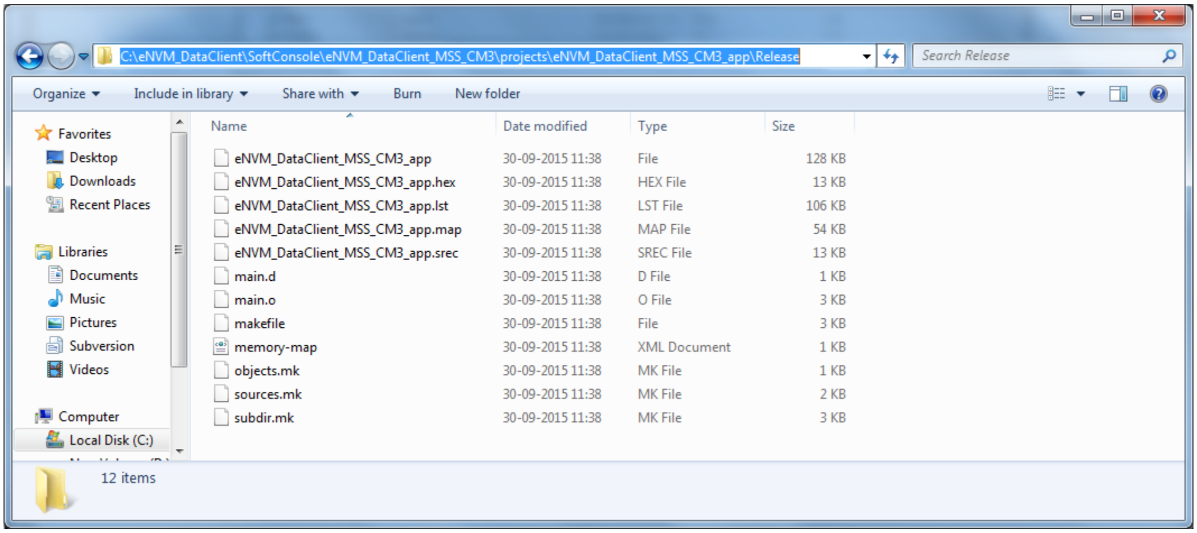
Number of words: 4592 Decimal

□ Use as ROM

□ Use content for simulation

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**

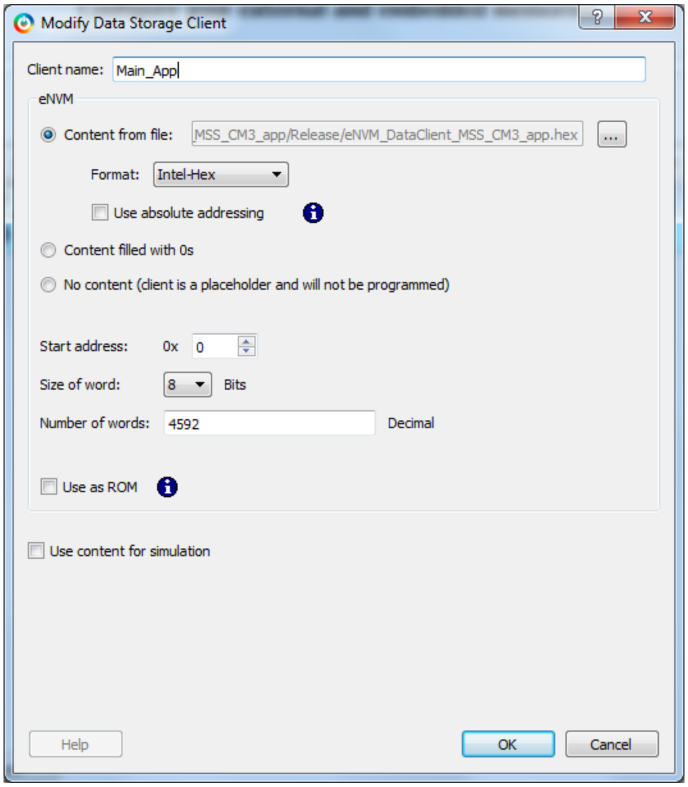


C:\eNVM_DataClient\SoftConsole\eNVM_DataClient_MSS_CM3\projects\eNVM_DataClient_MSS_CM3_app\Release

Name	Date modified	Type	Size
eNVM_DataClient_MSS_CM3_app	30-09-2015 11:38	File	128 KB
eNVM_DataClient_MSS_CM3_app.hex	30-09-2015 11:38	HEX File	13 KB
eNVM_DataClient_MSS_CM3_app.lst	30-09-2015 11:38	LST File	106 KB
eNVM_DataClient_MSS_CM3_app.map	30-09-2015 11:38	MAP File	54 KB
eNVM_DataClient_MSS_CM3_app.srec	30-09-2015 11:38	SREC File	13 KB
main.d	30-09-2015 11:38	D File	1 KB
main.o	30-09-2015 11:38	O File	3 KB
makefile	30-09-2015 11:38	File	3 KB
memory-map	30-09-2015 11:38	XML Document	1 KB
objects.mk	30-09-2015 11:38	MK File	1 KB
sources.mk	30-09-2015 11:38	MK File	2 KB
subdir.mk	30-09-2015 11:38	MK File	3 KB

12 items

Figure 17 • **Add Data Storage Client Window**



Modify Data Storage Client

Client name: Main_App

eNVM

● Content from file: MSS_CM3_app/Release/eNVM_DataClient_MSS_CM3_app.hex

Format: Intel-Hex

□ Use absolute addressing

○ Content filled with 0s

○ No content (client is a placeholder and will not be programmed)

Start address: 0x 0

Size of word: 8 Bits

Number of words: 4592 Decimal

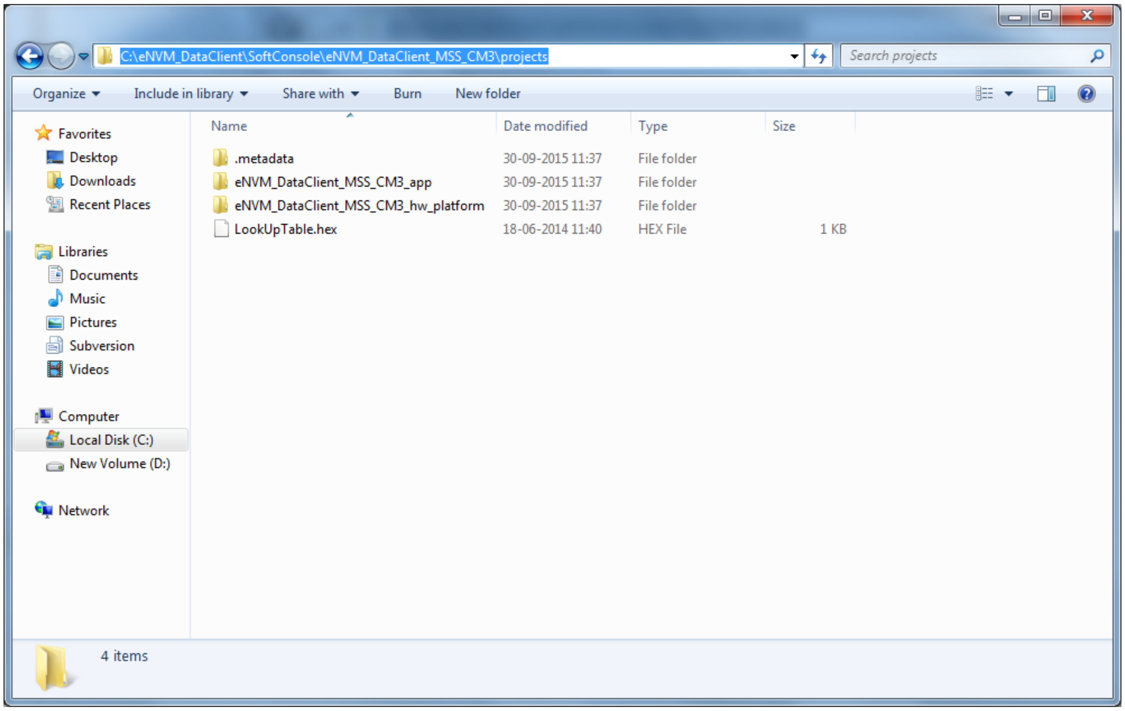
□ Use as ROM

□ Use content for simulation

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**



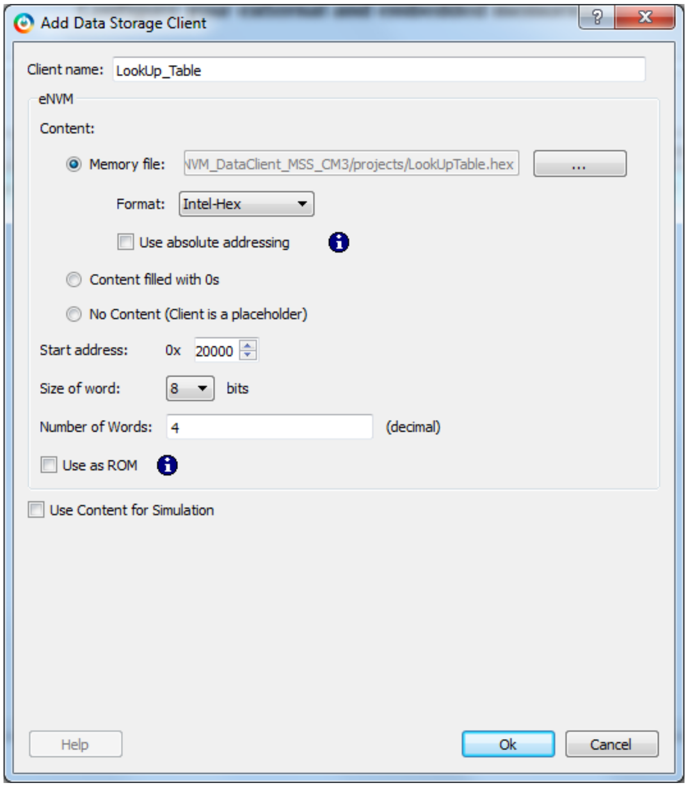
C:\eNVM_DataClient\SoftConsole\eNVM_DataClient_MSS_CM3\projects

Name	Date modified	Type	Size
.metadata	30-09-2015 11:37	File folder	
eNVM_DataClient_MSS_CM3_app	30-09-2015 11:37	File folder	
eNVM_DataClient_MSS_CM3_hw_platform	30-09-2015 11:37	File folder	
LookUpTable.hex	18-06-2014 11:40	HEX File	1 KB

4 items

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**



Add Data Storage Client

Client name: LookUp_Table

eNVM

Content:

● Memory file: NVM_DataClient_MSS_CM3/projects/LookUpTable.hex

Format: Intel-Hex

□ Use absolute addressing

○ Content filled with 0s

○ No Content (Client is a placeholder)

Start address: 0x 20000

Size of word: 8 bits

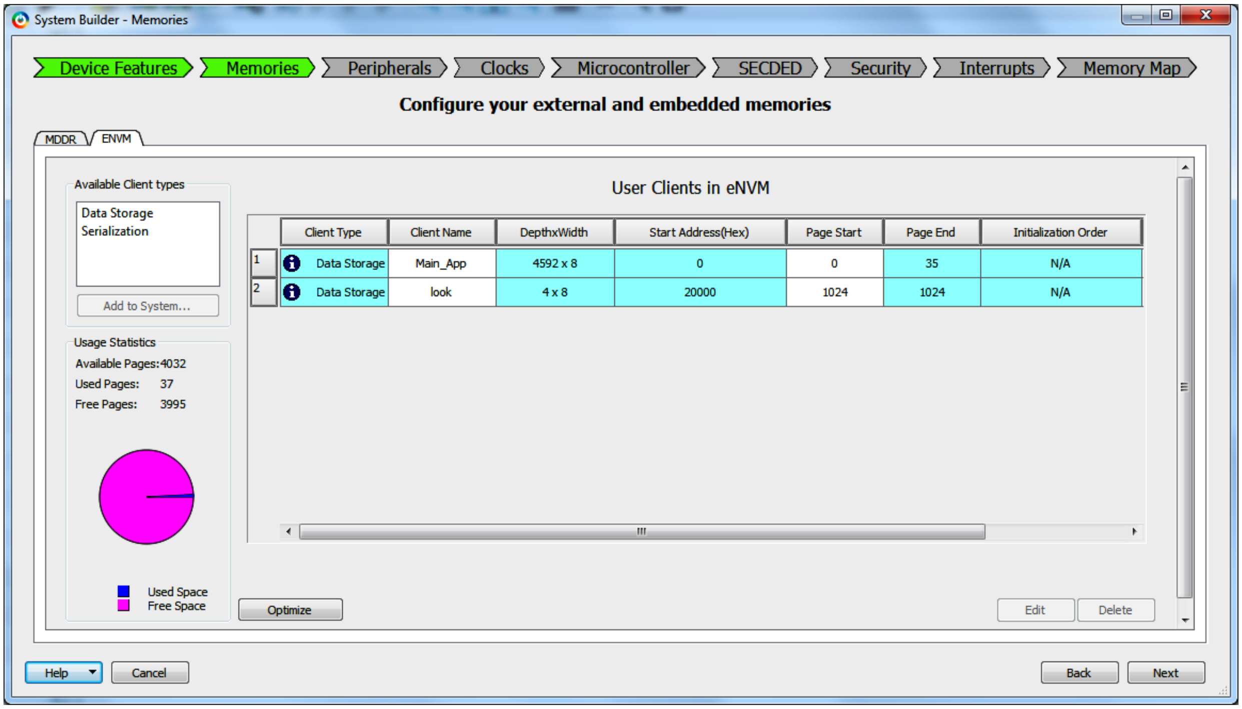
Number of Words: 4 (decimal)

□ Use as ROM

□ Use Content for Simulation

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**



Configure your external and embedded memories

Available Client types

Data Storage
Serialization

Usage Statistics

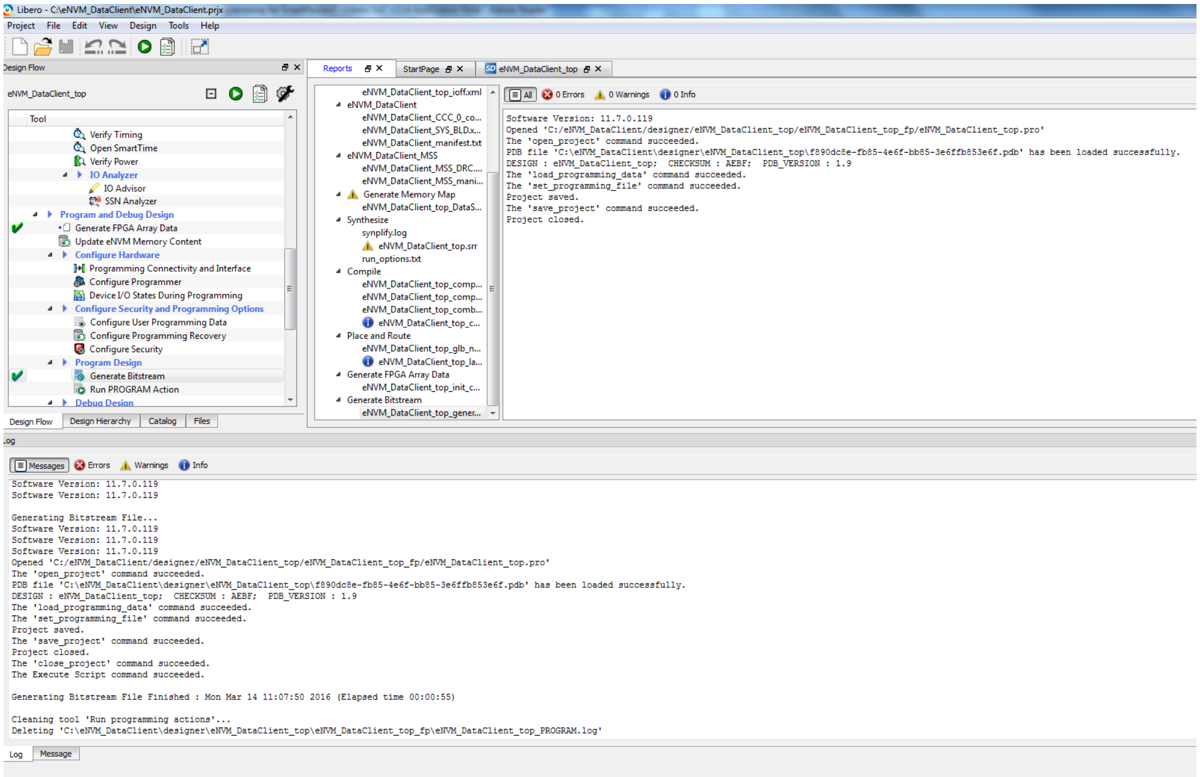
Available Pages:	4032
Used Pages:	37
Free Pages:	3995

User Clients in eNVM

	Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order
1	Data Storage	Main_App	4592 x 8	0	0	35	N/A
2	Data Storage	look	4 x 8	20000	1024	1024	N/A

10. Save **eNVM_DataClient_top** and regenerate the **eNVM_DataClient_to**p 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.

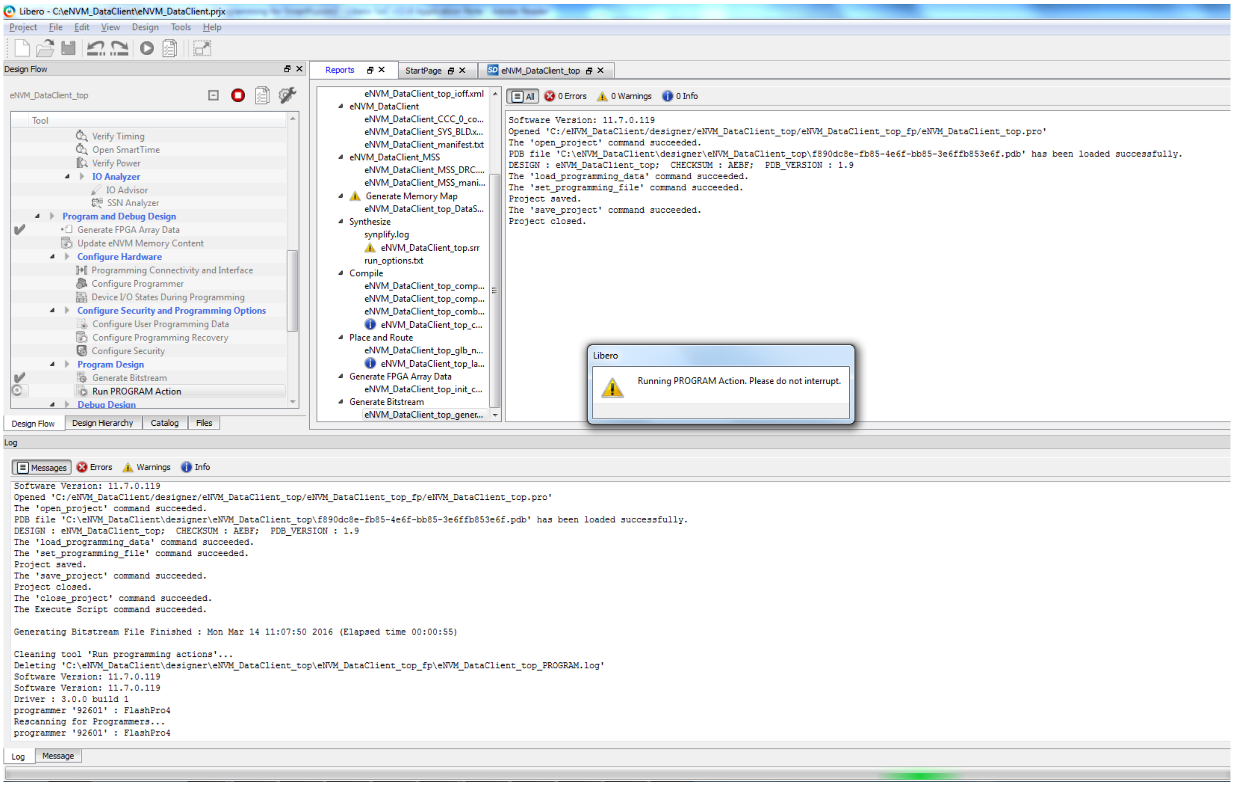
Figure 21 • Selecting Run PROGRAM Action



```
Software Version: 11.7.0.119
Opened 'C:/eNVM_DataClient/designer/eNVM_DataClient_top/eNVM_DataClient_top_fp/eNVM_DataClient_top.pro'
The 'open_project' command succeeded.
PDB file 'C:\eNVM_DataClient\designer\eNVM_DataClient_top\f890dc8e-fb85-4e6f-bb85-3e6ffb853e6f.pdb' has been loaded successfully.
DESIGN : eNVM_DataClient_top;  CHECKSUM : AEBF;  PDB_VERSION : 1.9
The 'load_programming_data' command succeeded.
The 'set_programming_file' command succeeded.
Project saved.
The 'save_project' command succeeded.
Project closed.
```

```
Software Version: 11.7.0.119
Software Version: 11.7.0.119

Generating Bitstream File...
Software Version: 11.7.0.119
Software Version: 11.7.0.119
Software Version: 11.7.0.119
Opened 'C:/eNVM_DataClient/designer/eNVM_DataClient_top/eNVM_DataClient_top_fp/eNVM_DataClient_top.pro'
The 'open_project' command succeeded.
PDB file 'C:\eNVM_DataClient\designer\eNVM_DataClient_top\f890dc8e-fb85-4e6f-bb85-3e6ffb853e6f.pdb' has been loaded successfully.
DESIGN : eNVM_DataClient_top;  CHECKSUM : AEBF;  PDB_VERSION : 1.9
The 'load_programming_data' command succeeded.
The 'set_programming_file' command succeeded.
Project saved.
The 'save_project' command succeeded.
Project closed.
The 'close_project' command succeeded.
The Execute Script command succeeded.

Generating Bitstream File Finished : Mon Mar 14 11:07:50 2016 (Elapsed time 00:00:55)

Cleaning tool 'Run programming actions'...
Deleting 'C:\eNVM_DataClient\designer\eNVM_DataClient_top\eNVM_DataClient_top_fp\eNVM_DataClient_top_PROGRAM.log'
```

Figure 22 • Programming SmartFusion2 Security Evaluation Kit



```
Software Version: 11.7.0.119
Opened 'C:/eNVM_DataClient/designer/eNVM_DataClient_top/eNVM_DataClient_top_fp/eNVM_DataClient_top.pro'
The 'open_project' command succeeded.
PDB file 'C:\eNVM_DataClient\designer\eNVM_DataClient_top\f890dc8e-fb85-4e6f-bb85-3e6ffb853e6f.pdb' has been loaded successfully.
DESIGN : eNVM_DataClient_top;  CHECKSUM : AEBF;  PDB_VERSION : 1.9
The 'load_programming_data' command succeeded.
The 'set_programming_file' command succeeded.
Project saved.
The 'save_project' command succeeded.
Project closed.
```

Libero

Running PROGRAM Action. Please do not interrupt.

```
Software Version: 11.7.0.119
Opened 'C:/eNVM_DataClient/designer/eNVM_DataClient_top/eNVM_DataClient_top_fp/eNVM_DataClient_top.pro'
The 'open_project' command succeeded.
PDB file 'C:\eNVM_DataClient\designer\eNVM_DataClient_top\f890dc8e-fb85-4e6f-bb85-3e6ffb853e6f.pdb' has been loaded successfully.
DESIGN : eNVM_DataClient_top;  CHECKSUM : AEBF;  PDB_VERSION : 1.9
The 'load_programming_data' command succeeded.
The 'set_programming_file' command succeeded.
Project saved.
The 'save_project' command succeeded.
Project closed.
The 'close_project' command succeeded.
The Execute Script command succeeded.

Generating Bitstream File Finished : Mon Mar 14 11:07:50 2016 (Elapsed time 00:00:55)

Cleaning tool 'Run programming actions'...
Deleting 'C:\eNVM_DataClient\designer\eNVM_DataClient_top\eNVM_DataClient_top_fp\eNVM_DataClient_top_PROGRAM.log'
Software Version: 11.7.0.119
Software Version: 11.7.0.119
Driver : 3.0.0 build 1
programmer '92601' : FlashPro4
Rescanning for Programmers...
programmer '92601' : FlashPro4
```

2.8 Updating eNVM Memory Content

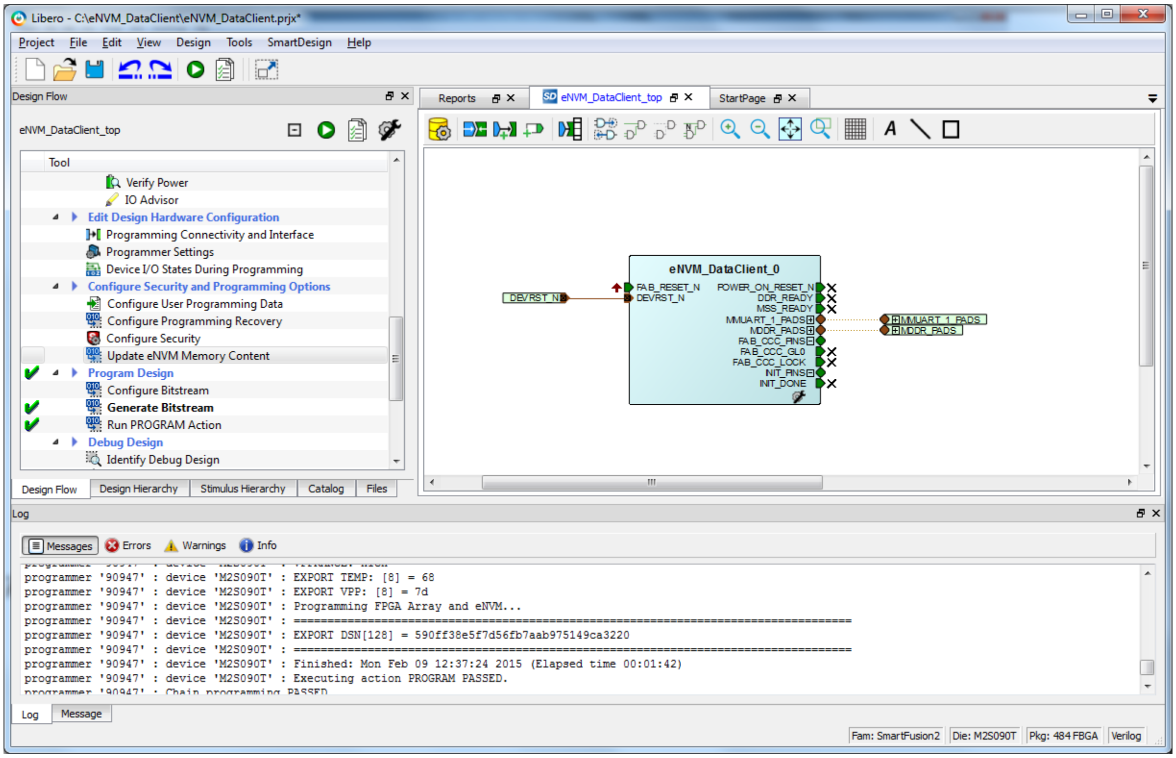
This option is available in Libero design flow to load the updated `.hex` files after compiling the application code in SoftConsole. Without this option, every time the SoftConsole application is built, System Builder eNVM Configurator is required for updating the eNVM client with the new `.hex` file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

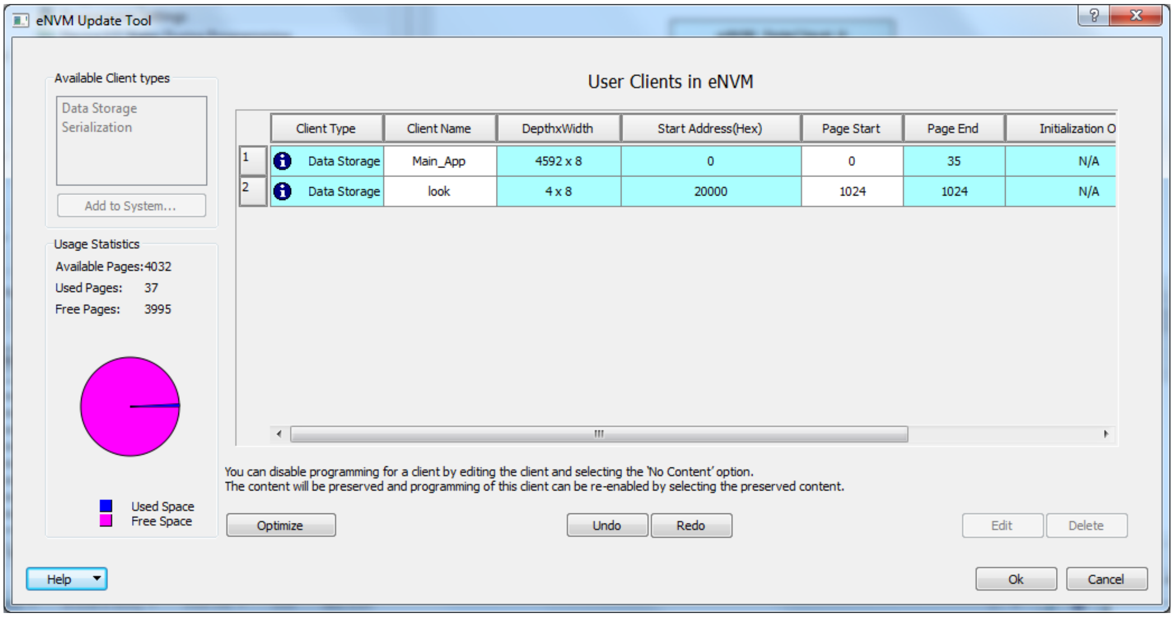
Figure 23 • **Choosing Update eNVM Memory Content Option from Libero Design Flow**



```
programmer '90947' : device 'M2S090T' : EXPORT TEMP: [8] = 68
programmer '90947' : device 'M2S090T' : EXPORT VPP: [8] = 7d
programmer '90947' : device 'M2S090T' : Programming FPGA Array and eNVM...
programmer '90947' : device 'M2S090T' : ===============================================================================
programmer '90947' : device 'M2S090T' : EXPORT DSN[128] = 590ff38e5f7d56fb7aab975149ca3220
programmer '90947' : device 'M2S090T' : ===============================================================================
programmer '90947' : device 'M2S090T' : Finished: Mon Feb 09 12:37:24 2015 (Elapsed time 00:01:42)
programmer '90947' : device 'M2S090T' : Executing action PROGRAM PASSED.
programmer '90947' : Chain programming PASSED.
```

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

Figure 24 • eNVM Memory Content Window



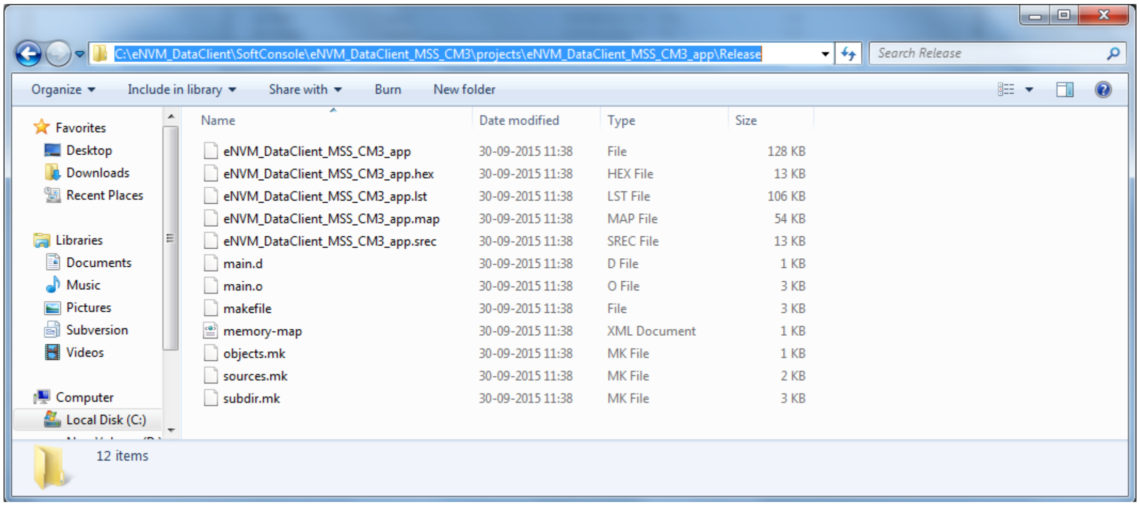
Usage Statistics
Available Pages: 4032
Used Pages: 37
Free Pages: 3995

	Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization O
1	Data Storage	Main_App	4592 x 8	0	0	35	N/A
2	Data Storage	look	4 x 8	20000	1024	1024	N/A

You can disable programming for a client by editing the client and selecting the 'No Content' option.
The content will be preserved and programming of this client can be re-enabled by selecting the preserved content.

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



C:\eNVM_DataClient\SoftConsole\eNVM_DataClient_MSS_CM3\projects\eNVM_DataClient_MSS_CM3_app\Release

Name	Date modified	Type	Size
eNVM_DataClient_MSS_CM3_app	30-09-2015 11:38	File	128 KB
eNVM_DataClient_MSS_CM3_app.hex	30-09-2015 11:38	HEX File	13 KB
eNVM_DataClient_MSS_CM3_app.lst	30-09-2015 11:38	LST File	106 KB
eNVM_DataClient_MSS_CM3_app.map	30-09-2015 11:38	MAP File	54 KB
eNVM_DataClient_MSS_CM3_app.srec	30-09-2015 11:38	SREC File	13 KB
main.d	30-09-2015 11:38	D File	1 KB
main.o	30-09-2015 11:38	O File	3 KB
makefile	30-09-2015 11:38	File	3 KB
memory-map	30-09-2015 11:38	XML Document	1 KB
objects.mk	30-09-2015 11:38	MK File	1 KB
sources.mk	30-09-2015 11:38	MK File	2 KB
subdir.mk	30-09-2015 11:38	MK File	3 KB

12 items

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.

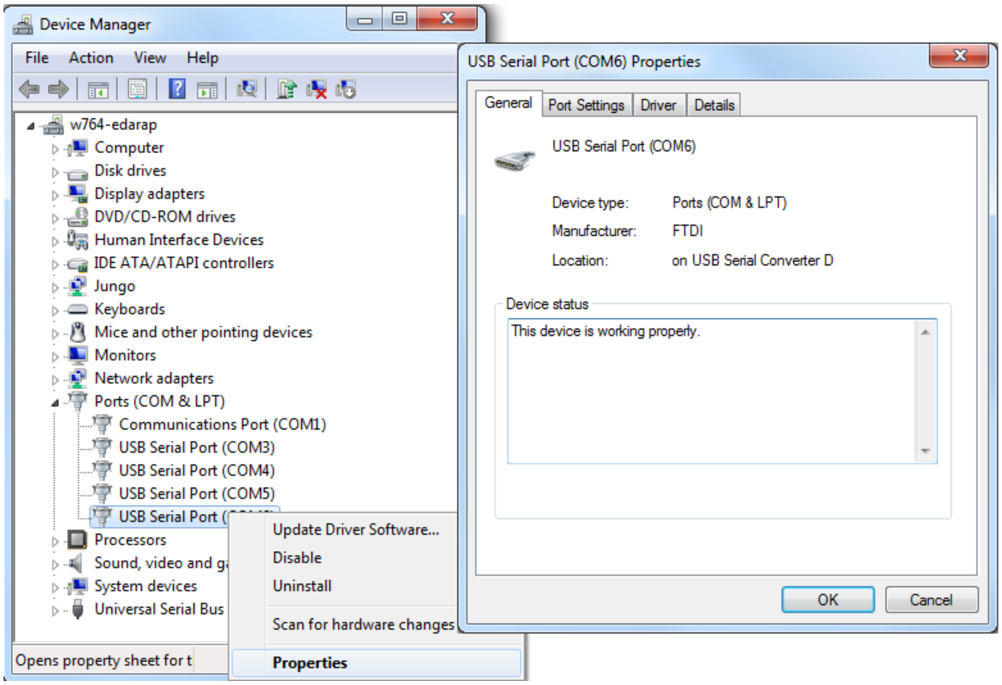
2.9 Running the Design on SmartFusion2 Security Evaluation Kit

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**



USB Serial Port (COM6)

Device type: Ports (COM & LPT)

Manufacturer: FTDI

Location: on USB Serial Converter D

Device status

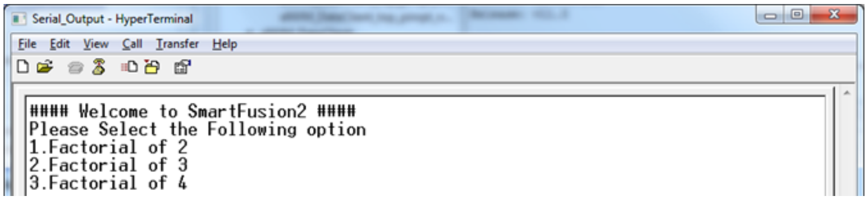
This device is working properly.

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.

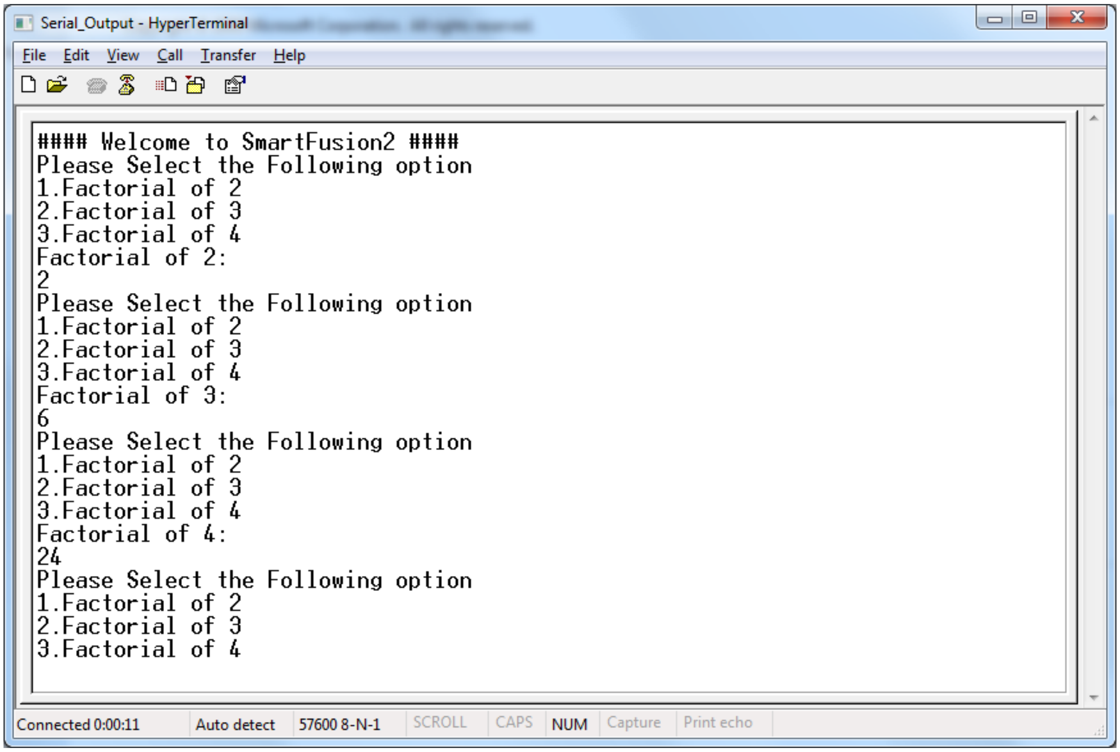
Figure 27 • **Selecting Factorial of a Number - User Options**


Serial_Output - HyperTerminal
File Edit View Call Transfer Help

```
#### Welcome to SmartFusion2 ####
Please Select the Following option
1.Factorial of 2
2.Factorial of 3
3.Factorial of 4
```

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**



```
#### Welcome to SmartFusion2 ####
Please Select the Following option
1.Factorial of 2
2.Factorial of 3
3.Factorial of 4
Factorial of 2:
2
Please Select the Following option
1.Factorial of 2
2.Factorial of 3
3.Factorial of 4
Factorial of 3:
6
Please Select the Following option
1.Factorial of 2
2.Factorial of 3
3.Factorial of 4
Factorial of 4:
24
Please Select the Following option
1.Factorial of 2
2.Factorial of 3
3.Factorial of 4
```

2.10 Building Multiple Executable Images in Release Mode

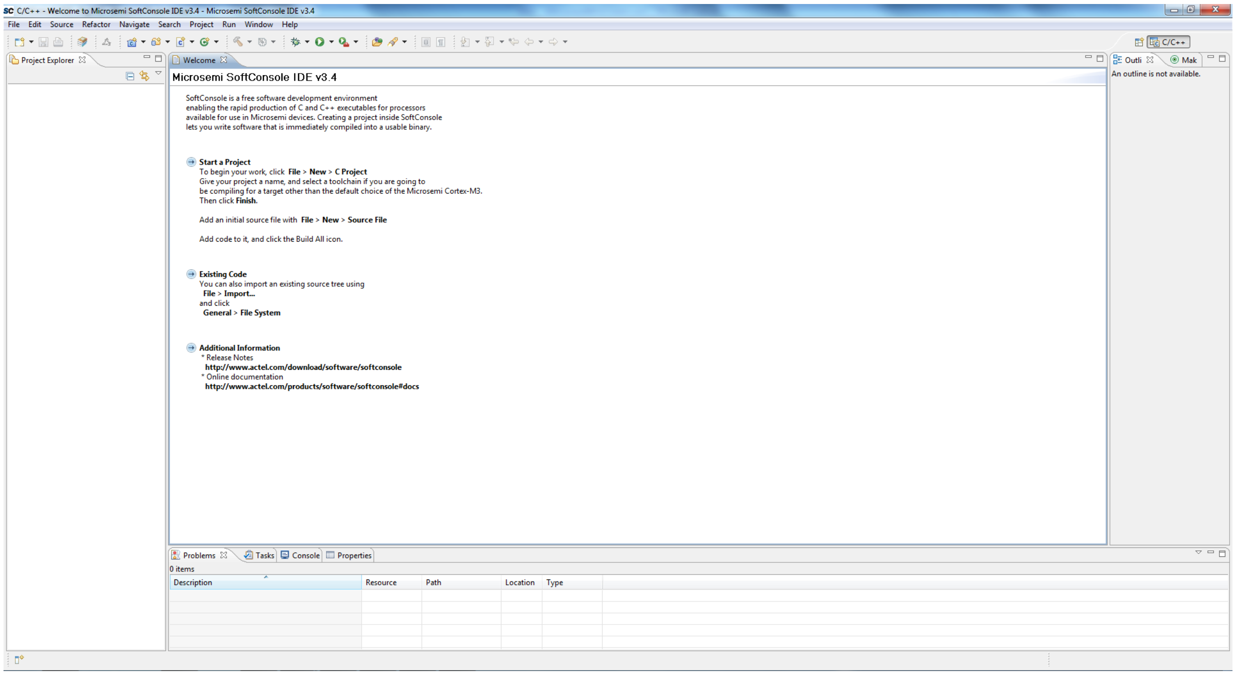
The design files are available to download from the Microsemi website. For design files, refer to: Prerequisites, page 3.

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**


SC C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Welcome
C/C++
Outli
Mak
Problems
Tasks
Console
Properties

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment
enabling the rapid production of C and C++ executables for processors
available for use in Microsemi devices. Creating a project inside SoftConsole
lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to
be compiling for a target other than the default choice of the Microsemi Cortex-M3.
Then click **Finish**.

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

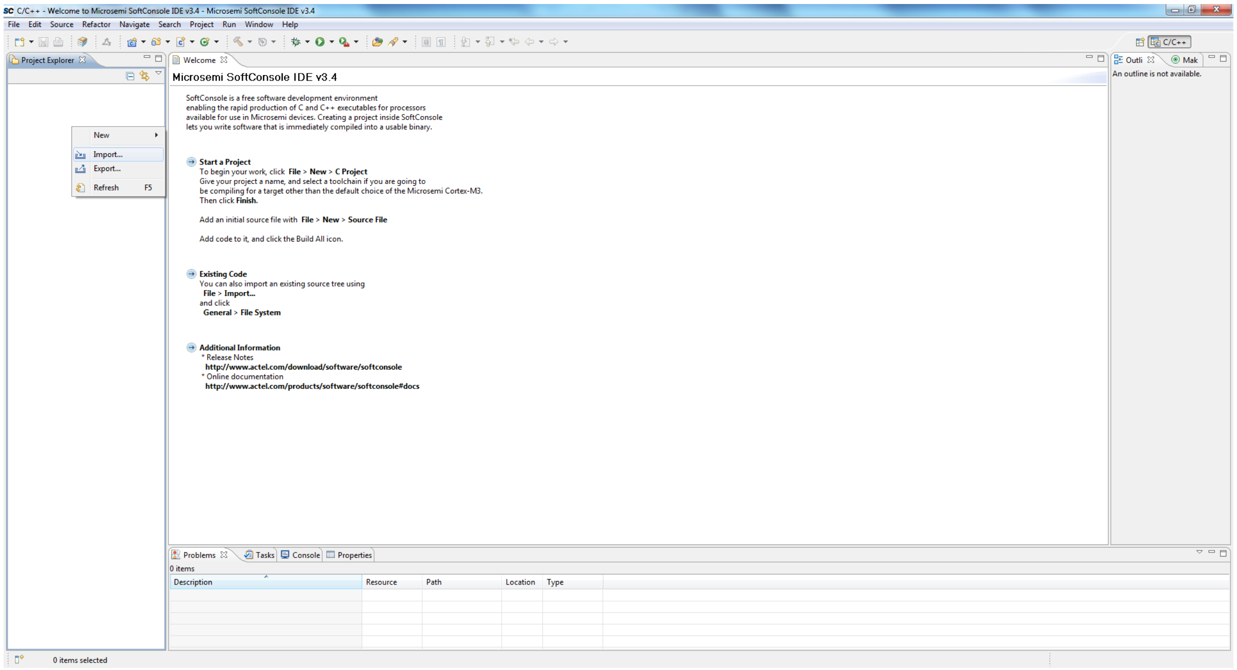
An outline is not available.

0 items

Description	Resource	Path	Location	Type

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

Figure 30 • **SoftConsole IDE - Import**


SC C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
New
Import...
Export...
Refresh F5
Welcome
C/C++
Outli
Mak
Problems
Tasks
Console
Properties

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment
enabling the rapid production of C and C++ executables for processors
available for use in Microsemi devices. Creating a project inside SoftConsole
lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to
be compiling for a target other than the default choice of the Microsemi Cortex-M3.
Then click **Finish**.

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

An outline is not available.

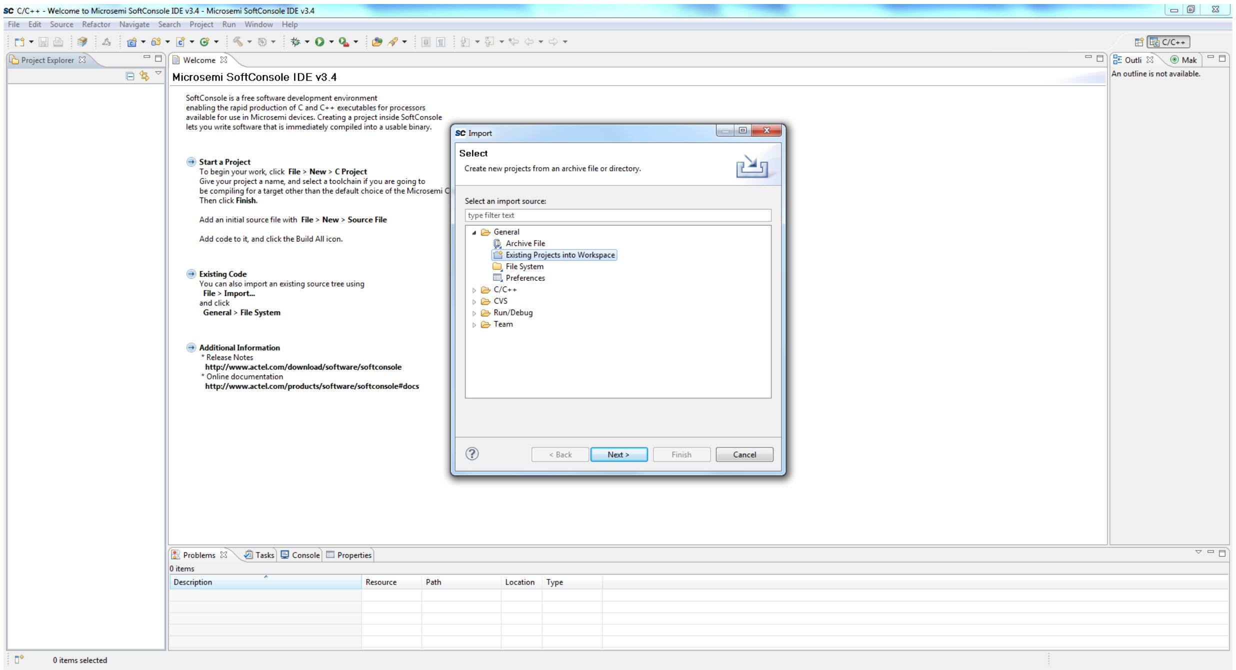
0 items

Description	Resource	Path	Location	Type

0 items selected

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

Figure 31 • **SoftConsole IDE - General**


C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Welcome
C/C++
Outli
Mak
An outline is not available.
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
Location
Type
0 items selected
Import
? < Back Next > Finish Cancel

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Microsemi devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to be compiling for a target other than the default choice of the Microsemi C
Then click **Finish**.

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

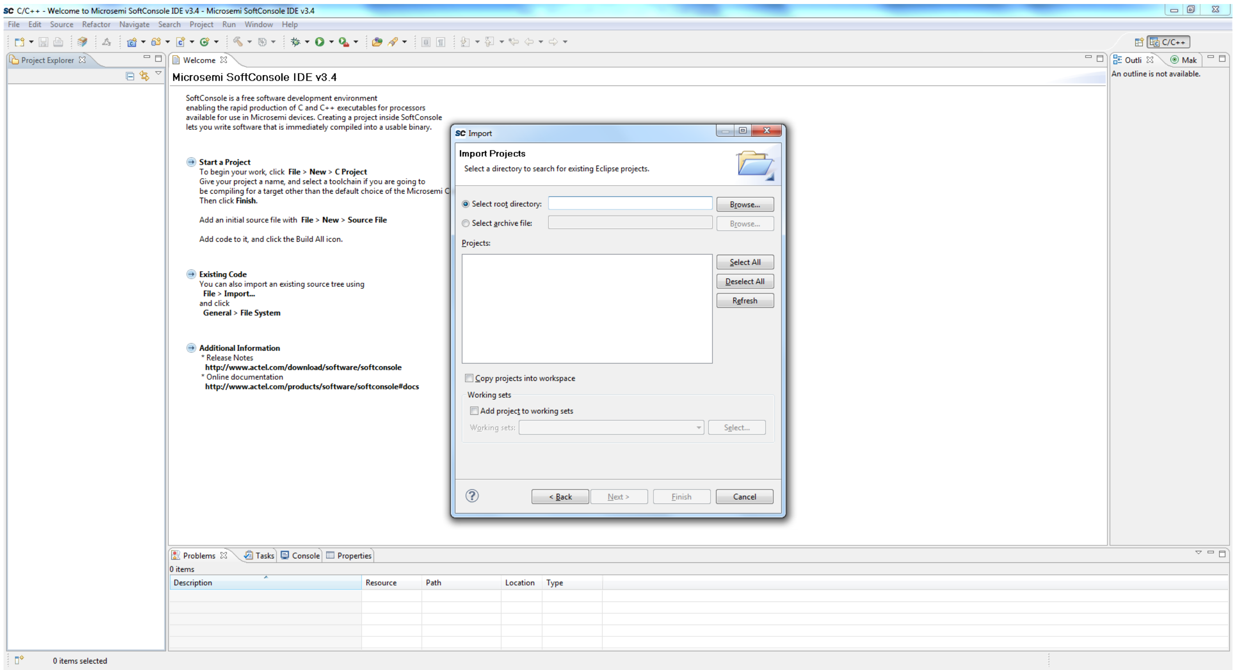
Select
Create new projects from an archive file or directory.

Select an import source:
type filter text

- General
  - Archive File
  - Existing Projects into Workspace
  - File System
  - Preferences
- C/C++
- CVS
- Run/Debug
- Team

Import window is shown in Figure 32.

Figure 32 • **SoftConsole IDE - Browse**


C/C++ - Welcome to Microsemi SoftConsole IDE v3.4 - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Welcome
C/C++
Outli
Mak
An outline is not available.
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
Location
Type
0 items selected
Import
Browse...
Browse...
Select All
Deselect All
Refresh
Select...
? < Back Next > Finish Cancel

Microsemi SoftConsole IDE v3.4

SoftConsole is a free software development environment enabling the rapid production of C and C++ executables for processors available for use in Microsemi devices. Creating a project inside SoftConsole lets you write software that is immediately compiled into a usable binary.

Start a Project
To begin your work, click **File** > **New** > **C Project**
Give your project a name, and select a toolchain if you are going to be compiling for a target other than the default choice of the Microsemi C
Then click **Finish**.

Add an initial source file with **File** > **New** > **Source File**

Add code to it, and click the Build All icon.

Existing Code
You can also import an existing source tree using
File > **Import...**
and click
General > **File System**

Additional Information
* Release Notes
http://www.actel.com/download/software/softconsole
* Online documentation
http://www.actel.com/products/software/softconsole#docs

Import Projects
Select a directory to search for existing Eclipse projects.

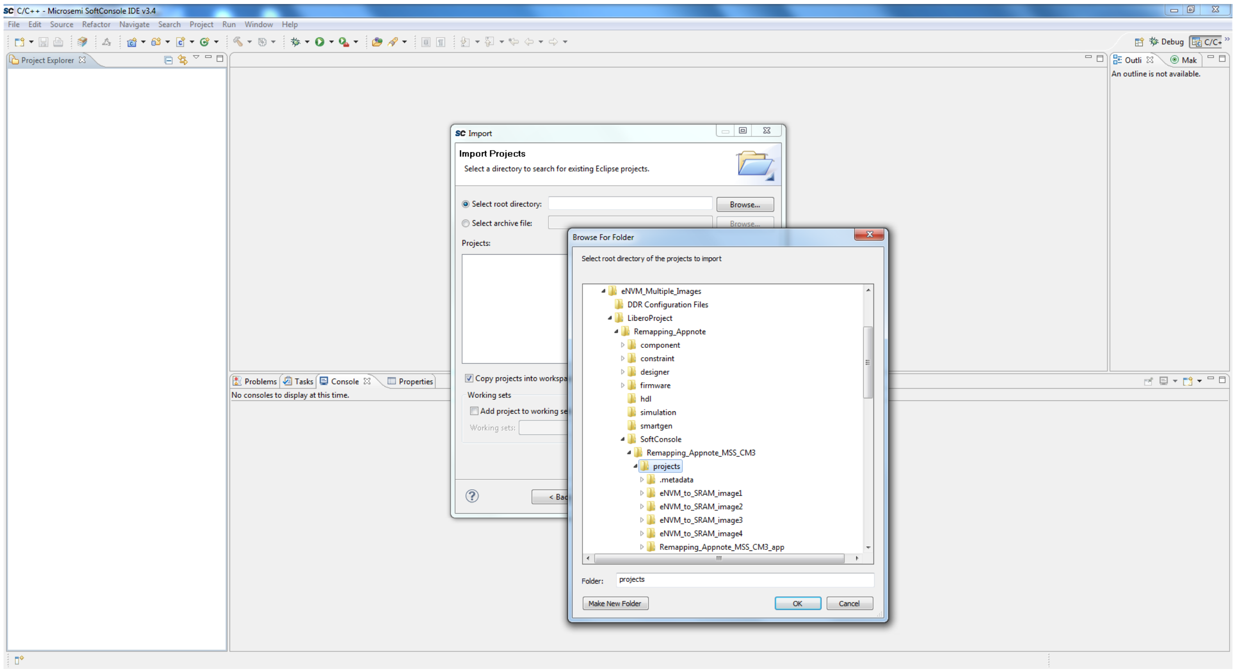
● Select root directory:
○ Select archive file:

Projects:

□ Copy projects into workspace

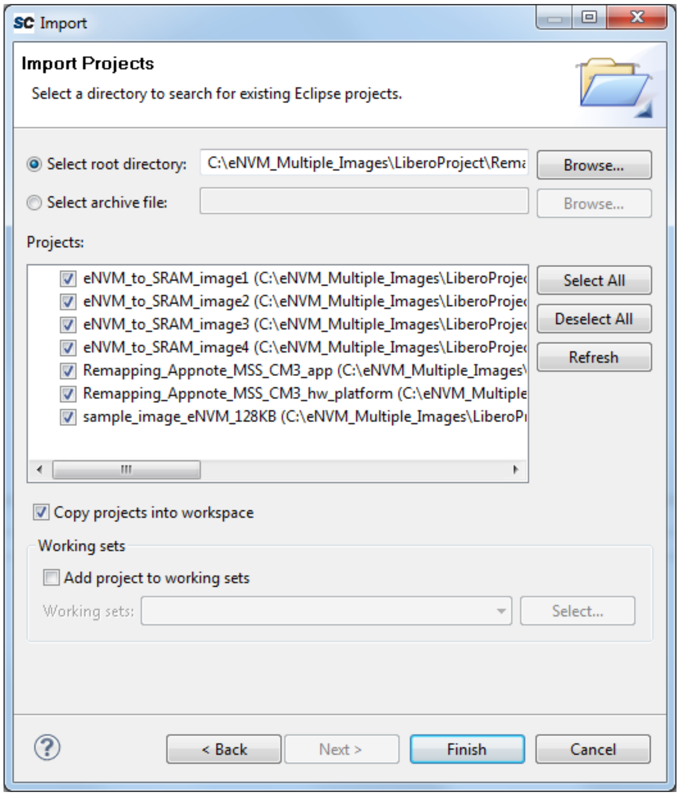
Working sets
□ Add project to working sets
Working sets:

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**



● Select root directory: C:\eNVM_Multiple_Images\LiberoProject\Rema

○ Select archive file:

☑ eNVM_to_SRAM_image1 (C:\eNVM_Multiple_Images\LiberoProje
☑ eNVM_to_SRAM_image2 (C:\eNVM_Multiple_Images\LiberoProje
☑ eNVM_to_SRAM_image3 (C:\eNVM_Multiple_Images\LiberoProje
☑ eNVM_to_SRAM_image4 (C:\eNVM_Multiple_Images\LiberoProje
☑ Remapping_Appnote_MSS_CM3_app (C:\eNVM_Multiple_Images\
☑ Remapping_Appnote_MSS_CM3_hw_platform (C:\eNVM_Multiple
☑ sample_image_eNVM_128KB (C:\eNVM_Multiple_Images\LiberoP

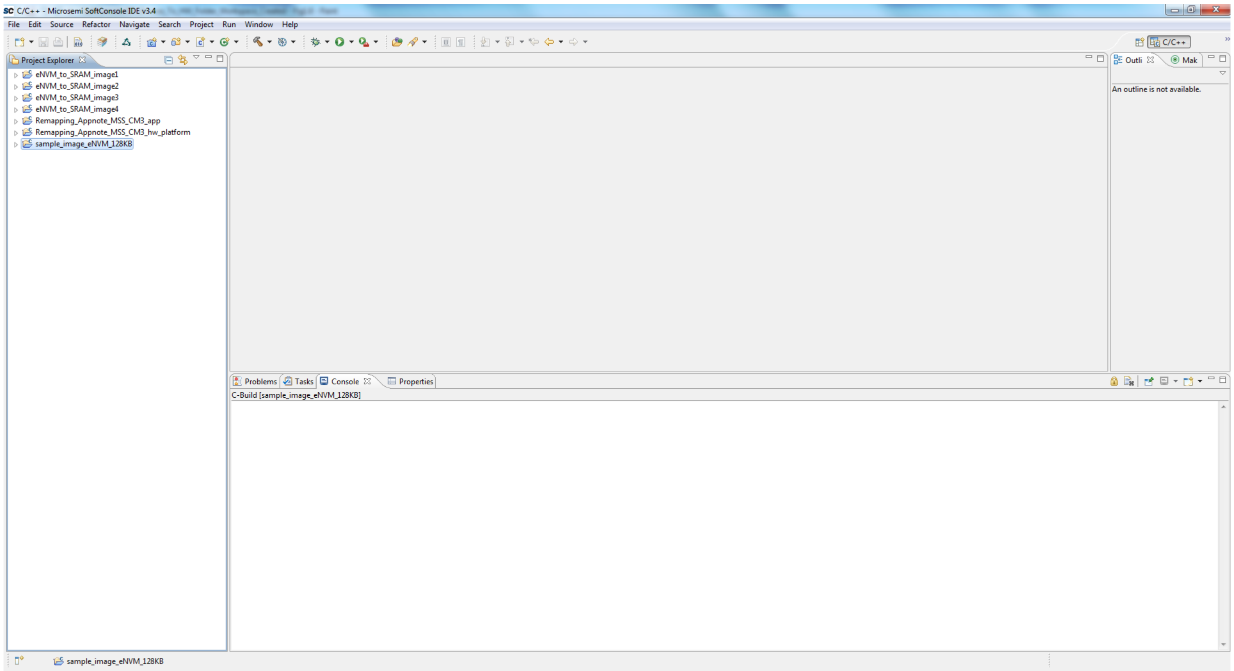
☑ Copy projects into workspace

□ Add project to working sets

Working sets:

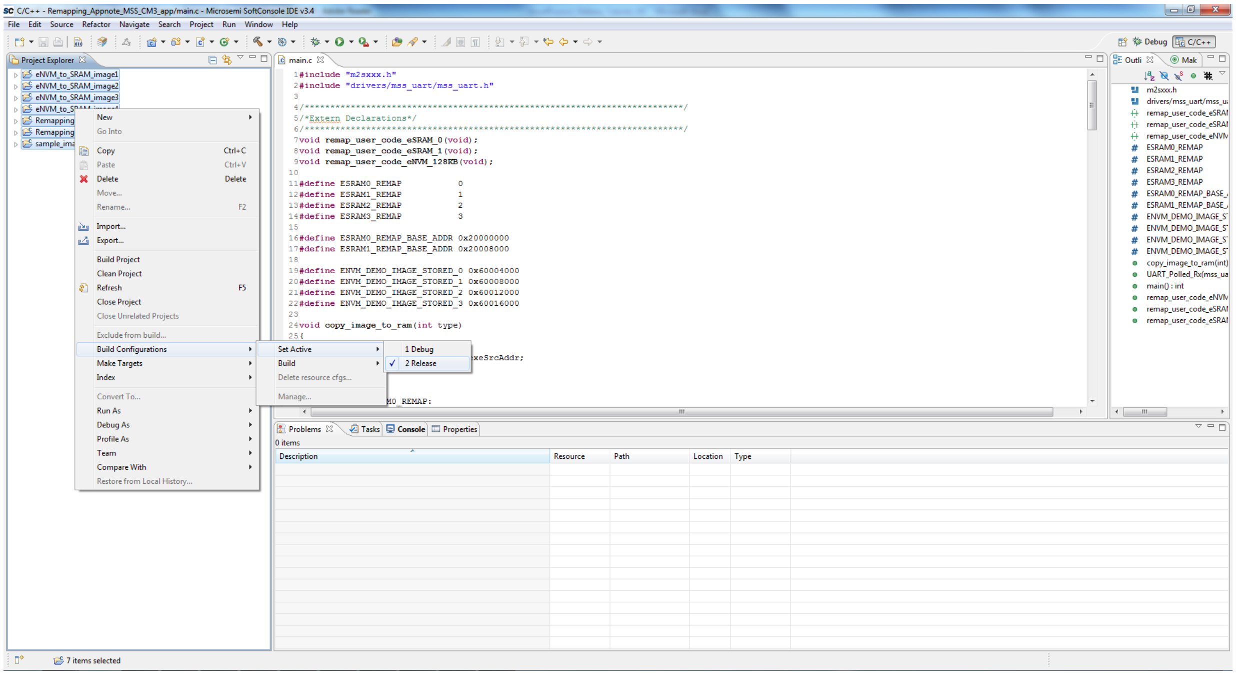
The SoftConsole perspective is shown in Figure 35.

Figure 35 • **SoftConsole Workspace**


C/C++ - Microsemi SoftConsole IDE v3.4
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
eNVM_to_SRAM_image1
eNVM_to_SRAM_image2
eNVM_to_SRAM_image3
eNVM_to_SRAM_image4
Remapping_Appnote_MSS_CM3_app
Remapping_Appnote_MSS_CM3_hw_platform
sample_image_eNVM_128KB
Outli
Mak
C/C++
An outline is not available.
Problems
Tasks
Console
Properties
C-Build [sample_image_eNVM_128KB]
sample_image_eNVM_128KB

5. 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**



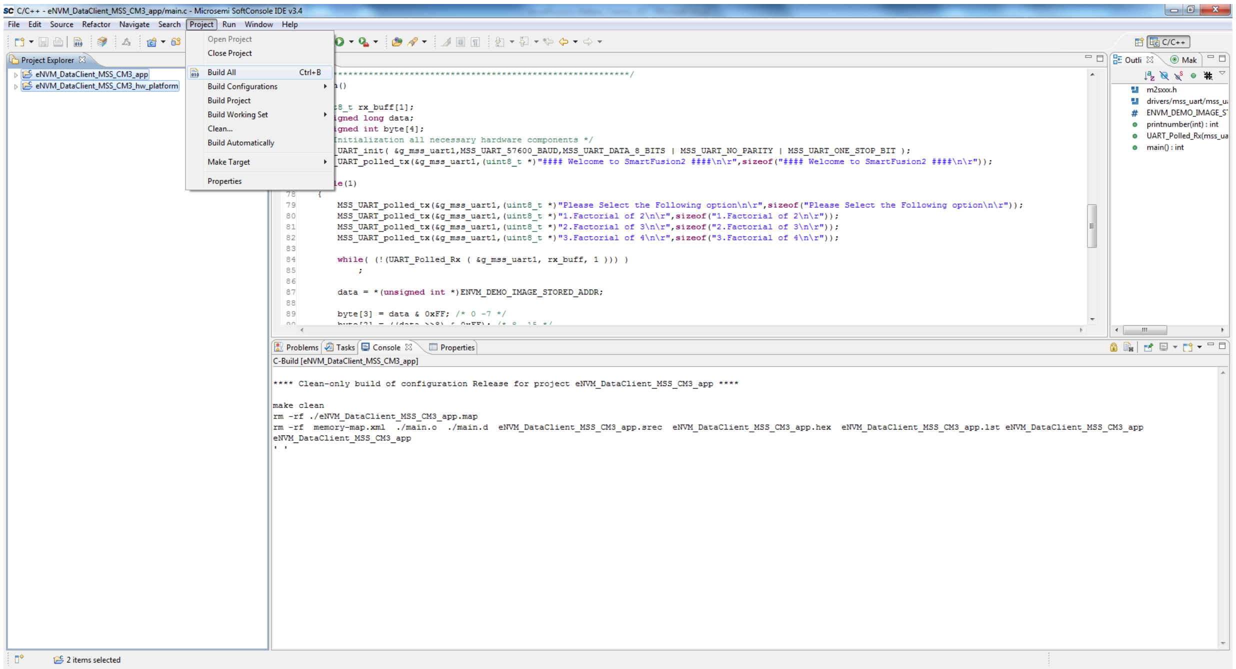
```
#include "m2sxxx.h"
#include "drivers/mss_uart/mss_uart.h"

/**********************************************************************/
/*Extern Declarations*/
/**********************************************************************/
void remap_user_code_eSRAM_0(void);
void remap_user_code_eSRAM_1(void);
void remap_user_code_eNVM_128KB(void);

#define ESRAM0_REMAP          0
#define ESRAM1_REMAP          1
#define ESRAM2_REMAP          2
#define ESRAM3_REMAP          3

#define ESRAM0_REMAP_BASE_ADDR 0x20000000
#define ESRAM1_REMAP_BASE_ADDR 0x20008000

#define ENVM_DEMO_IMAGE_STORED_0 0x60004000
#define ENVM_DEMO_IMAGE_STORED_1 0x60008000
#define ENVM_DEMO_IMAGE_STORED_2 0x60012000
#define ENVM_DEMO_IMAGE_STORED_3 0x60016000

void copy_image_to_ram(int type)
{
                   xeSrcAddr;

                   M0_REMAP:
```

0 items

Description	Resource	Path	Location	Type

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**



```
**************************************************************/
n()

t8_t rx_buff[1];
igned long data;
igned int byte[4];
Initialization all necessary hardware components */
_UART_init( &g_mss_uart1,MSS_UART_57600_BAUD,MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT );
_UART_polled_tx(&g_mss_uart1,(uint8_t *)"#### Welcome to SmartFusion2 ####\n\r",sizeof("#### Welcome to SmartFusion2 ####\n\r"));

le(1)
    {
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"Please Select the Following option\n\r",sizeof("Please Select the Following option\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"1.Factorial of 2\n\r",sizeof("1.Factorial of 2\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"2.Factorial of 3\n\r",sizeof("2.Factorial of 3\n\r"));
        MSS_UART_polled_tx(&g_mss_uart1,(uint8_t *)"3.Factorial of 4\n\r",sizeof("3.Factorial of 4\n\r"));

        while( (!(UART_Polled_Rx ( &g_mss_uart1, rx_buff, 1 ))) )
            ;

        data = *(unsigned int *)ENVM_DEMO_IMAGE_STORED_ADDR;

        byte[3] = data & 0xFF; /* 0 -7 */
```

C-Build [eNVM_DataClient_MSS_CM3_app]

```
**** Clean-only build of configuration Release for project eNVM_DataClient_MSS_CM3_app ****

make clean
rm -rf ./eNVM_DataClient_MSS_CM3_app.map
rm -rf  memory-map.xml  ./main.o  ./main.d  eNVM_DataClient_MSS_CM3_app.srec  eNVM_DataClient_MSS_CM3_app.hex  eNVM_DataClient_MSS_CM3_app.lst eNVM_DataClient_MSS_CM3_app
eNVM_DataClient_MSS_CM3_app
' '
```

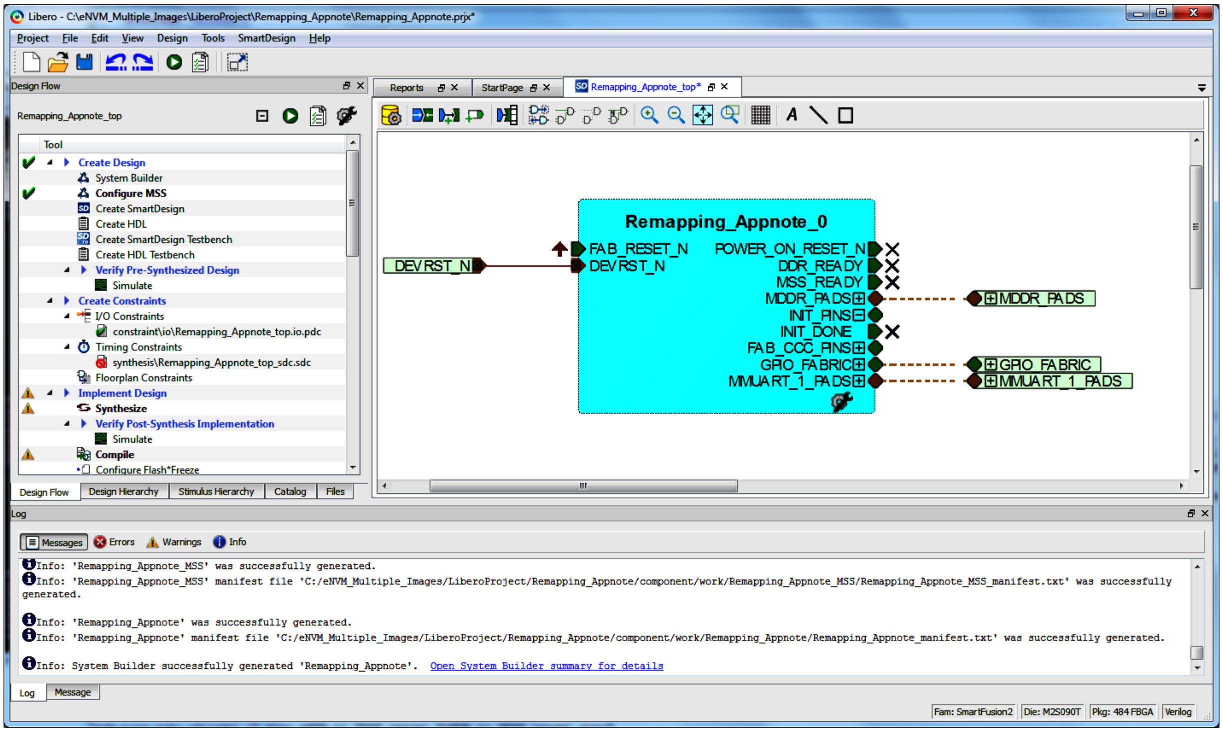
2.11 Loading Multiple Executable Images into eNVM

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.

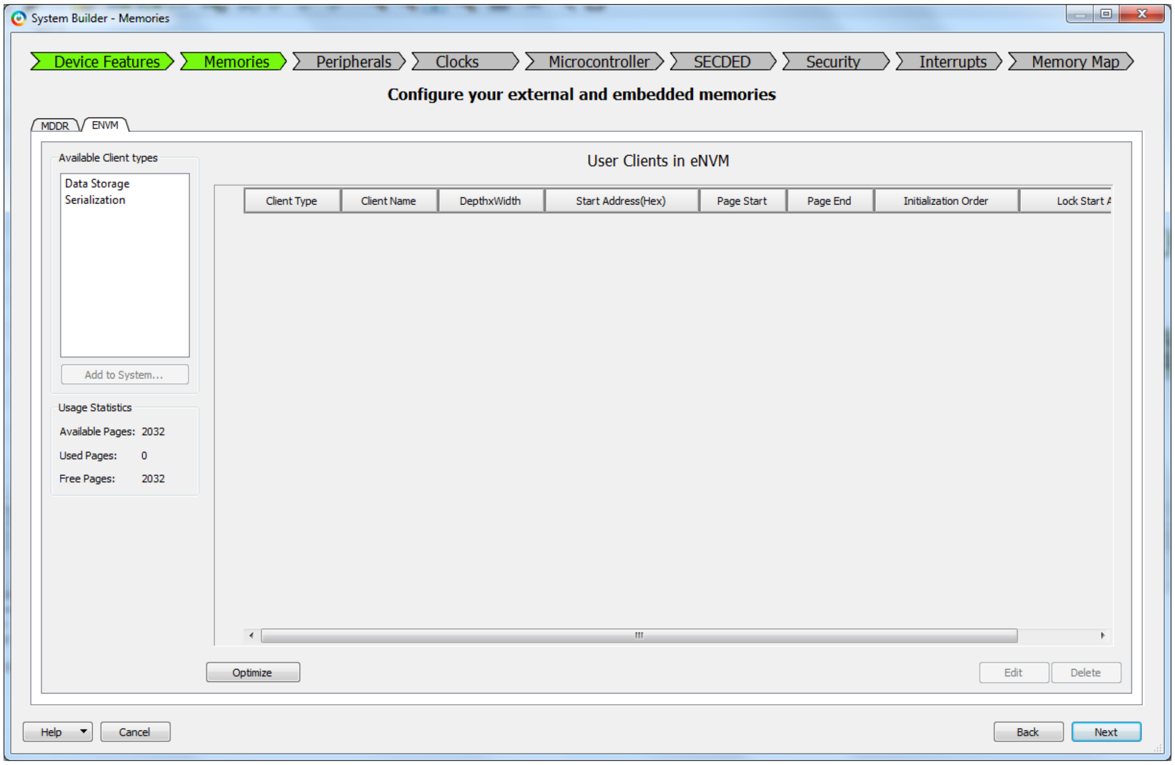
Figure 38 • **Remapping_Appnote_top Tab**



```
Info: 'Remapping_Appnote_MSS' was successfully generated.
Info: 'Remapping_Appnote_MSS' manifest file 'C:/eNVM_Multiple_Images/LiberoProject/Remapping_Appnote/component/work/Remapping_Appnote_MSS/Remapping_Appnote_MSS_manifest.txt' was successfully
generated.

Info: 'Remapping_Appnote' was successfully generated.
Info: 'Remapping_Appnote' manifest file 'C:/eNVM_Multiple_Images/LiberoProject/Remapping_Appnote/component/work/Remapping_Appnote/Remapping_Appnote_manifest.txt' was successfully generated.

Info: System Builder successfully generated 'Remapping_Appnote'.  Open System Builder summary for details
```

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**



Configure your external and embedded memories

Available Client types

Data Storage
Serialization

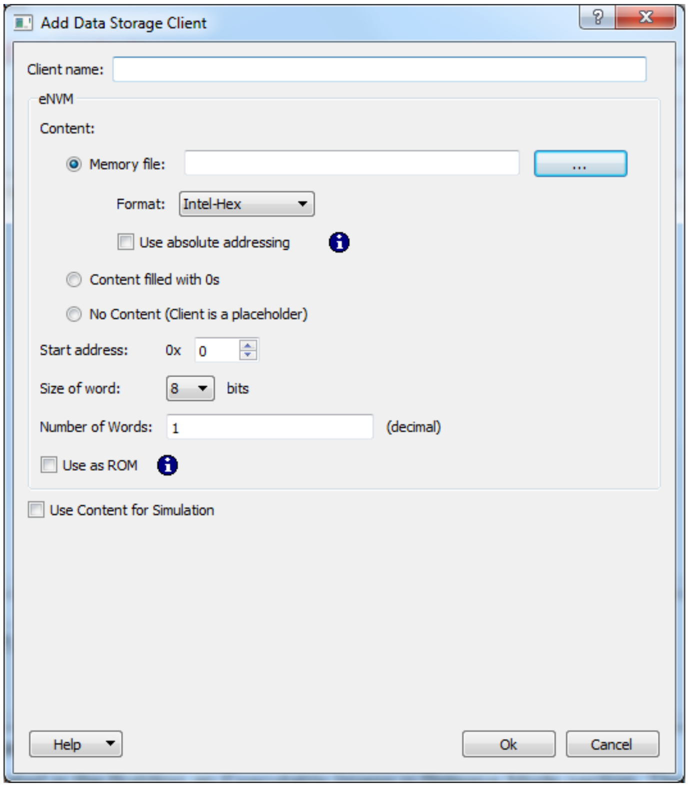
User Clients in eNVM

Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order	Lock Start A

Usage Statistics

Available Pages: 2032
Used Pages: 0
Free Pages: 2032

Figure 40 • **Add to Data Storage Client Window**



Client name:

eNVM

Content:

● Memory file:

Format: Intel-Hex

□ Use absolute addressing

○ Content filled with 0s

○ No Content (Client is a placeholder)

Start address: 0x 0

Size of word: 8 bits

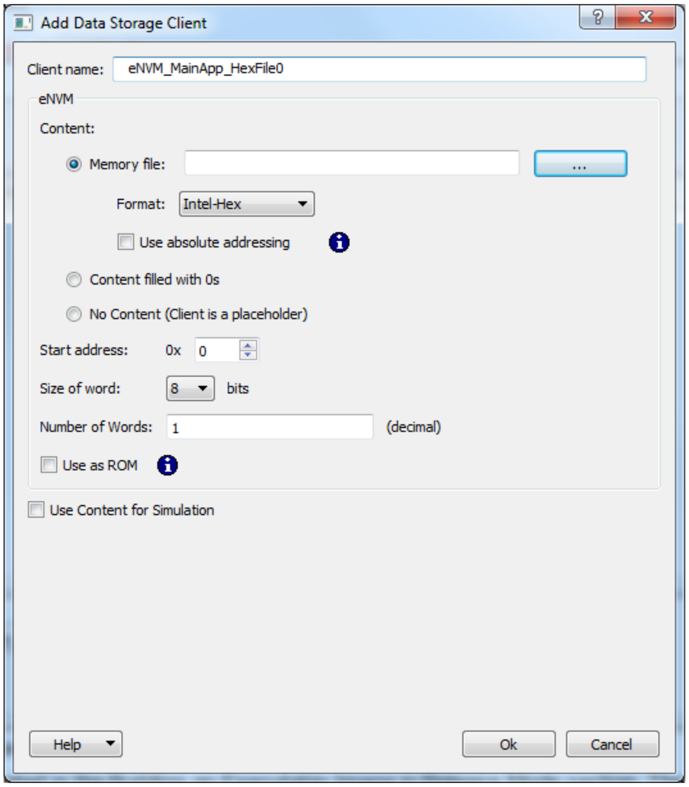
Number of Words: 1 (decimal)

□ Use as ROM

□ Use Content for Simulation

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

Figure 41 • **Add to Data Storage Client Window**



Client name: eNVM_MainApp_HexFile0

eNVM

Content:

● Memory file:

Format: Intel-Hex

□ Use absolute addressing

○ Content filled with 0s

○ No Content (Client is a placeholder)

Start address: 0x 0

Size of word: 8 bits

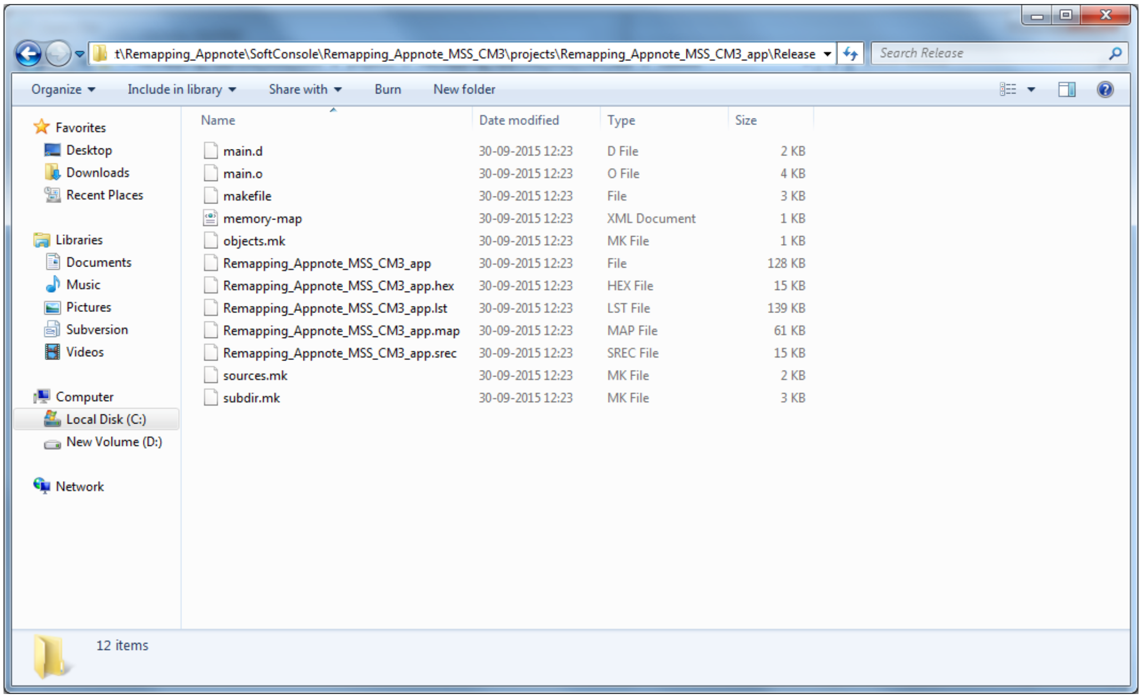
Number of Words: 1 (decimal)

□ Use as ROM

□ Use Content for Simulation

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.

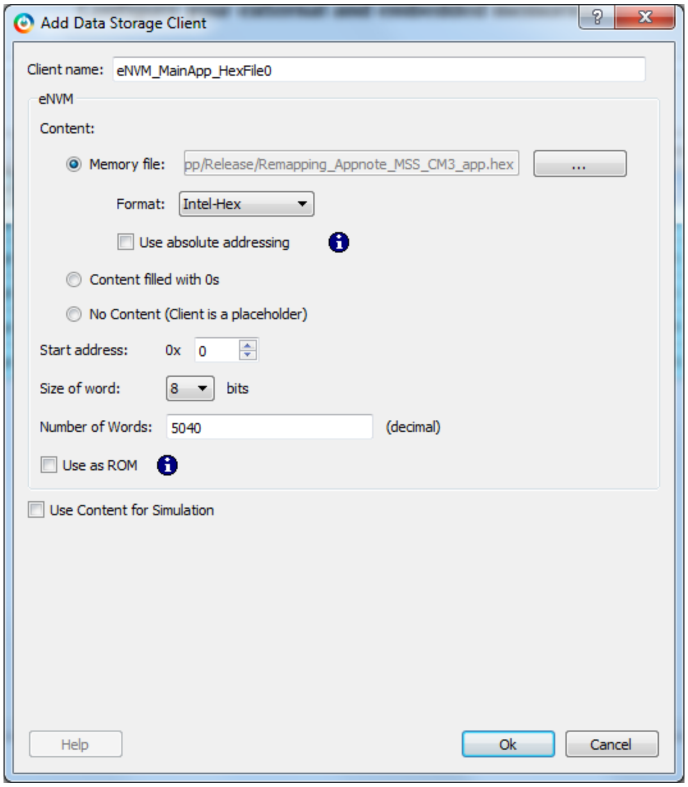
Figure 42 • **Executable Image in Release Folder**



t\Remapping_Appnote\SoftConsole\Remapping_Appnote_MSS_CM3\projects\Remapping_Appnote_MSS_CM3_app\Release

Name	Date modified	Type	Size
main.d	30-09-2015 12:23	D File	2 KB
main.o	30-09-2015 12:23	O File	4 KB
makefile	30-09-2015 12:23	File	3 KB
memory-map	30-09-2015 12:23	XML Document	1 KB
objects.mk	30-09-2015 12:23	MK File	1 KB
Remapping_Appnote_MSS_CM3_app	30-09-2015 12:23	File	128 KB
Remapping_Appnote_MSS_CM3_app.hex	30-09-2015 12:23	HEX File	15 KB
Remapping_Appnote_MSS_CM3_app.lst	30-09-2015 12:23	LST File	139 KB
Remapping_Appnote_MSS_CM3_app.map	30-09-2015 12:23	MAP File	61 KB
Remapping_Appnote_MSS_CM3_app.srec	30-09-2015 12:23	SREC File	15 KB
sources.mk	30-09-2015 12:23	MK File	2 KB
subdir.mk	30-09-2015 12:23	MK File	3 KB

Figure 43 • **Add Data Storage Client Window**



Add Data Storage Client

Client name: eNVM_MainApp_HexFile0

eNVM

Content:

● Memory file: pp/Release/Remapping_Appnote_MSS_CM3_app.hex

Format: Intel-Hex

□ Use absolute addressing

○ Content filled with 0s

○ No Content (Client is a placeholder)

Start address: 0x 0

Size of word: 8 bits

Number of Words: 5040 (decimal)

□ Use as ROM

□ Use Content for Simulation

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.

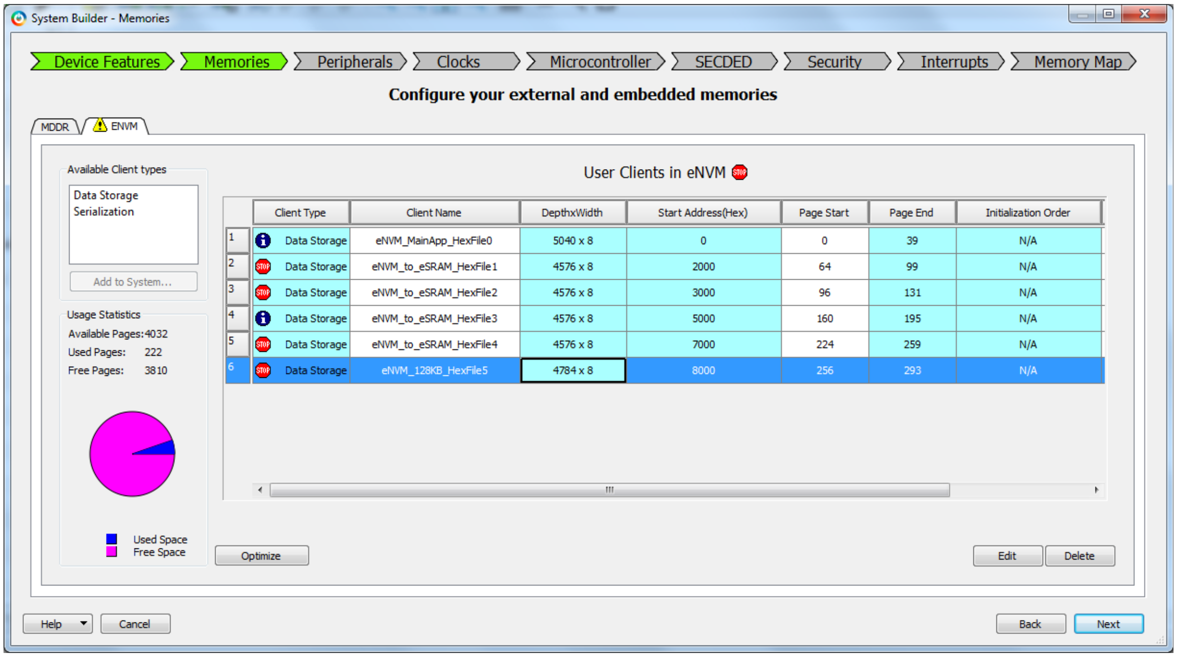
Table 2 lists the SoftConsole application projects in the Release mode, their `.hex` files, and their mapping to the eNVM Client Name.

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**



Configure your external and embedded memories

Available Client types

Data Storage
Serialization

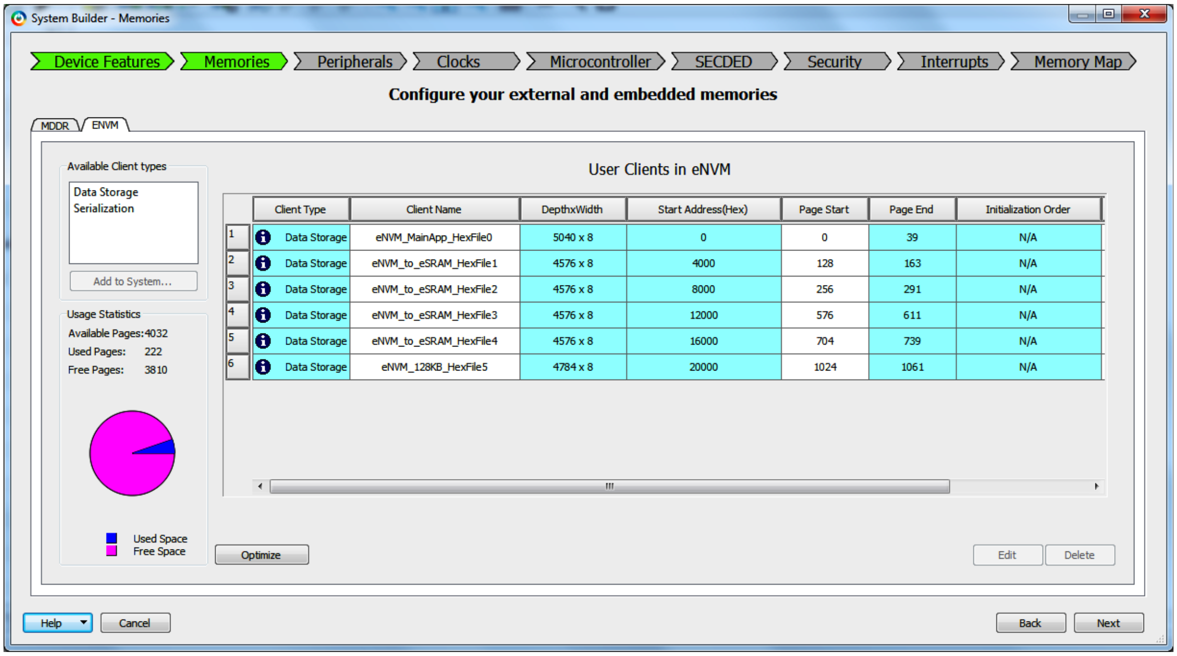
Usage Statistics
Available Pages: 4032
Used Pages: 222
Free Pages: 3810

User Clients in eNVM 🛑

		Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order
1	ℹ	Data Storage	eNVM_MainApp_HexFile0	5040 x 8	0	0	39	N/A
2	🛑	Data Storage	eNVM_to_eSRAM_HexFile1	4576 x 8	2000	64	99	N/A
3	🛑	Data Storage	eNVM_to_eSRAM_HexFile2	4576 x 8	3000	96	131	N/A
4	ℹ	Data Storage	eNVM_to_eSRAM_HexFile3	4576 x 8	5000	160	195	N/A
5	🛑	Data Storage	eNVM_to_eSRAM_HexFile4	4576 x 8	7000	224	259	N/A
6	🛑	Data Storage	eNVM_128KB_HexFile5	4784 x 8	8000	256	293	N/A

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

Figure 45 • **Hex Files with Correct Start Addresses**



Configure your external and embedded memories

Available Client types

Data Storage
Serialization

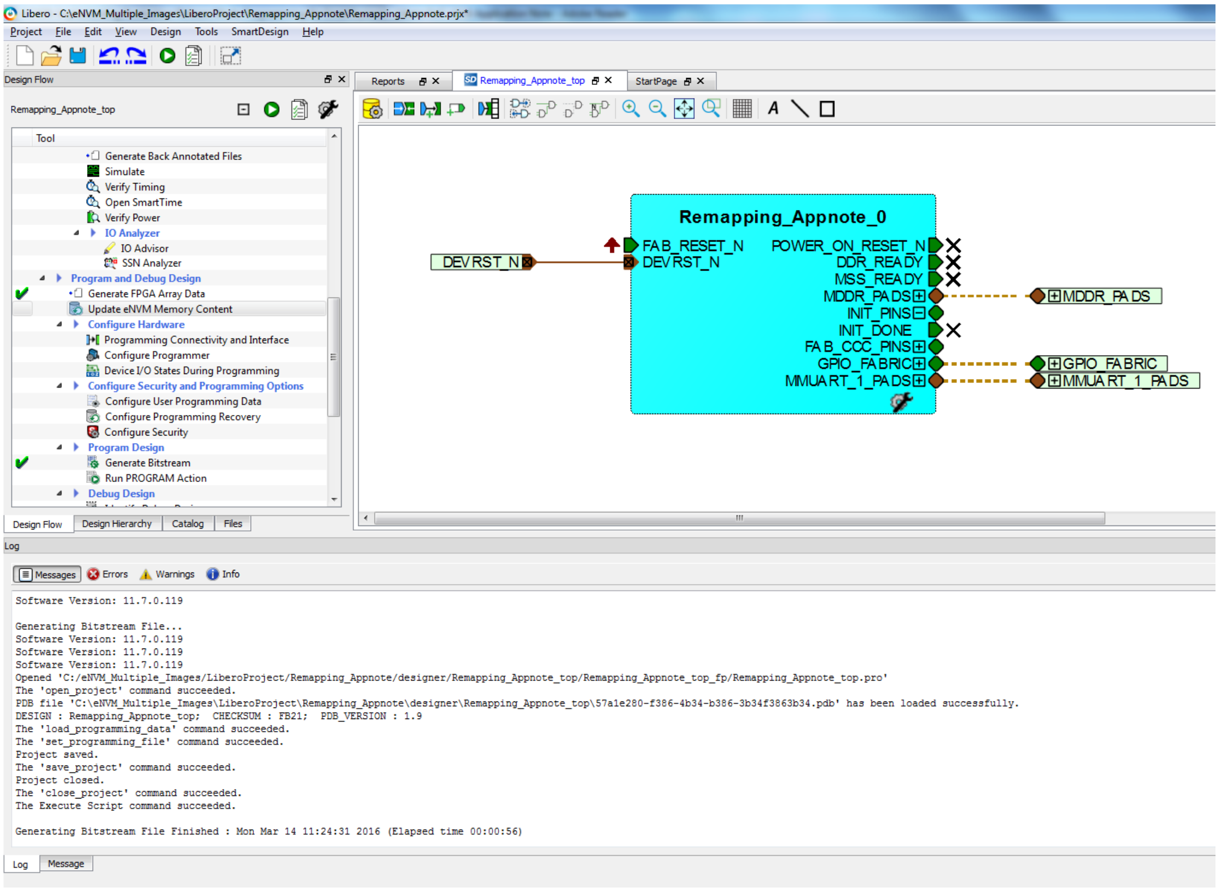
Usage Statistics
Available Pages: 4032
Used Pages: 222
Free Pages: 3810

User Clients in eNVM

		Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	Initialization Order
1	ℹ	Data Storage	eNVM_MainApp_HexFile0	5040 x 8	0	0	39	N/A
2	ℹ	Data Storage	eNVM_to_eSRAM_HexFile1	4576 x 8	4000	128	163	N/A
3	ℹ	Data Storage	eNVM_to_eSRAM_HexFile2	4576 x 8	8000	256	291	N/A
4	ℹ	Data Storage	eNVM_to_eSRAM_HexFile3	4576 x 8	12000	576	611	N/A
5	ℹ	Data Storage	eNVM_to_eSRAM_HexFile4	4576 x 8	16000	704	739	N/A
6	ℹ	Data Storage	eNVM_128KB_HexFile5	4784 x 8	20000	1024	1061	N/A

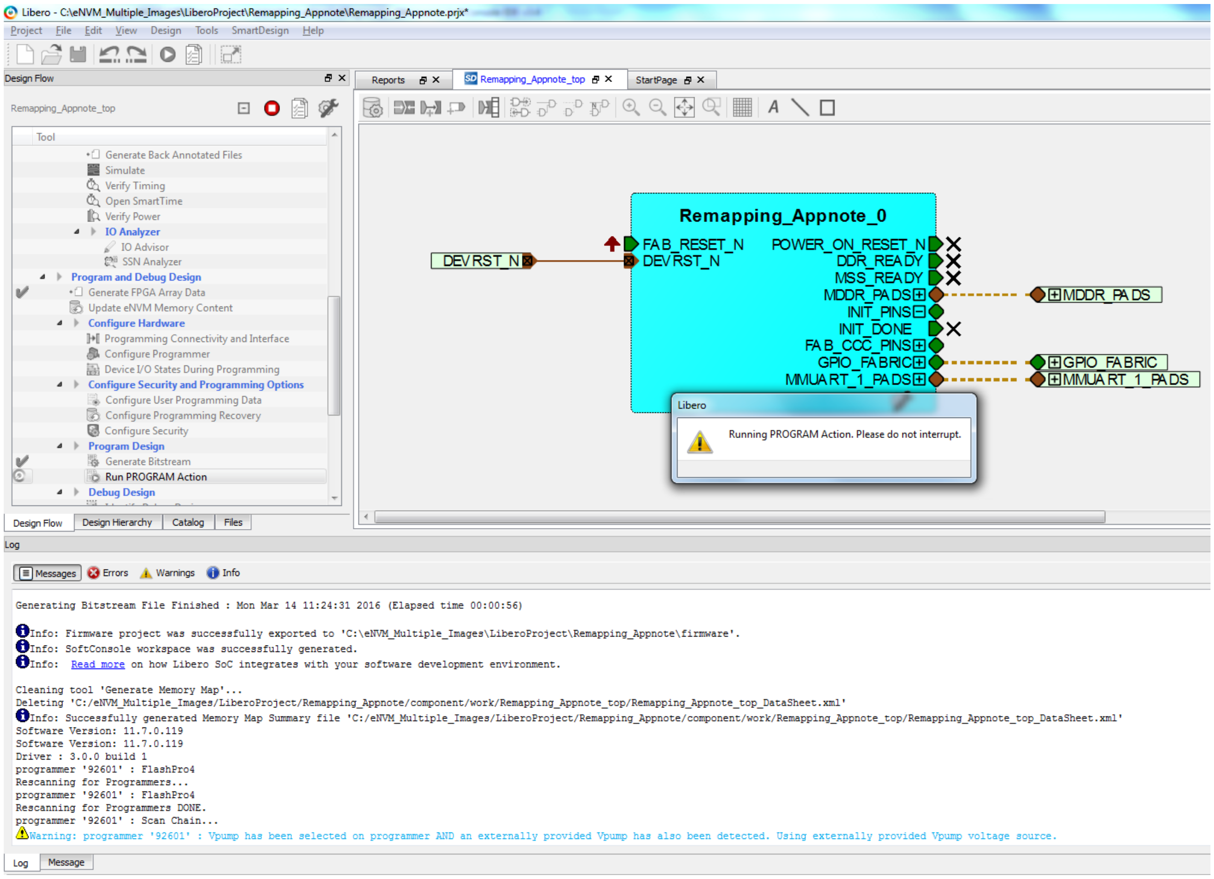
8. In the **System Builder - Memories** page, keep the other settings to default and click **Next**, refer to Figure 45.
9. Save **Remapping_Appnote_top** and regenerate the **Remapping_Appnote_0** component by clicking Generate Component in SmartDesign.
10. Click **Generate Bitstream** to complete the 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**



```
Software Version: 11.7.0.119

Generating Bitstream File...
Software Version: 11.7.0.119
Software Version: 11.7.0.119
Software Version: 11.7.0.119
Opened 'C:/eNVM_Multiple_Images/LiberoProject/Remapping_Appnote/designer/Remapping_Appnote_top/Remapping_Appnote_top_fp/Remapping_Appnote_top.pro'
The 'open_project' command succeeded.
PDB file 'C:\eNVM_Multiple_Images\LiberoProject\Remapping_Appnote\designer\Remapping_Appnote_top\57a1e280-f386-4b34-b386-3b34f3863b34.pdb' has been loaded successfully.
DESIGN : Remapping_Appnote_top;  CHECKSUM : FB21;  PDB_VERSION : 1.9
The 'load_programming_data' command succeeded.
The 'set_programming_file' command succeeded.
Project saved.
The 'save_project' command succeeded.
Project closed.
The 'close_project' command succeeded.
The Execute Script command succeeded.

Generating Bitstream File Finished : Mon Mar 14 11:24:31 2016 (Elapsed time 00:00:56)
```

Figure 47 • **Programming SmartFusion2 Security Evaluation Kit**



Running PROGRAM Action. Please do not interrupt.

```
Generating Bitstream File Finished : Mon Mar 14 11:24:31 2016 (Elapsed time 00:00:56)

Info: Firmware project was successfully exported to 'C:\eNVM_Multiple_Images\LiberoProject\Remapping_Appnote\firmware'.
Info: SoftConsole workspace was successfully generated.
Info:  Read more on how Libero SoC integrates with your software development environment.

Cleaning tool 'Generate Memory Map'...
Deleting 'C:/eNVM_Multiple_Images/LiberoProject/Remapping_Appnote/component/work/Remapping_Appnote_top/Remapping_Appnote_top_DataSheet.xml'
Info: Successfully generated Memory Map Summary file 'C:/eNVM_Multiple_Images/LiberoProject/Remapping_Appnote/component/work/Remapping_Appnote_top/Remapping_Appnote_top_DataSheet.xml'
Software Version: 11.7.0.119
Software Version: 11.7.0.119
Driver : 3.0.0 build 1
programmer '92601' : FlashPro4
Rescanning for Programmers...
programmer '92601' : FlashPro4
Rescanning for Programmers DONE.
programmer '92601' : Scan Chain...
Warning: programmer '92601' : Vpump has been selected on programmer AND an externally provided Vpump has also been detected. Using externally provided Vpump voltage source.
```

2.12 Updating eNVM Memory Content

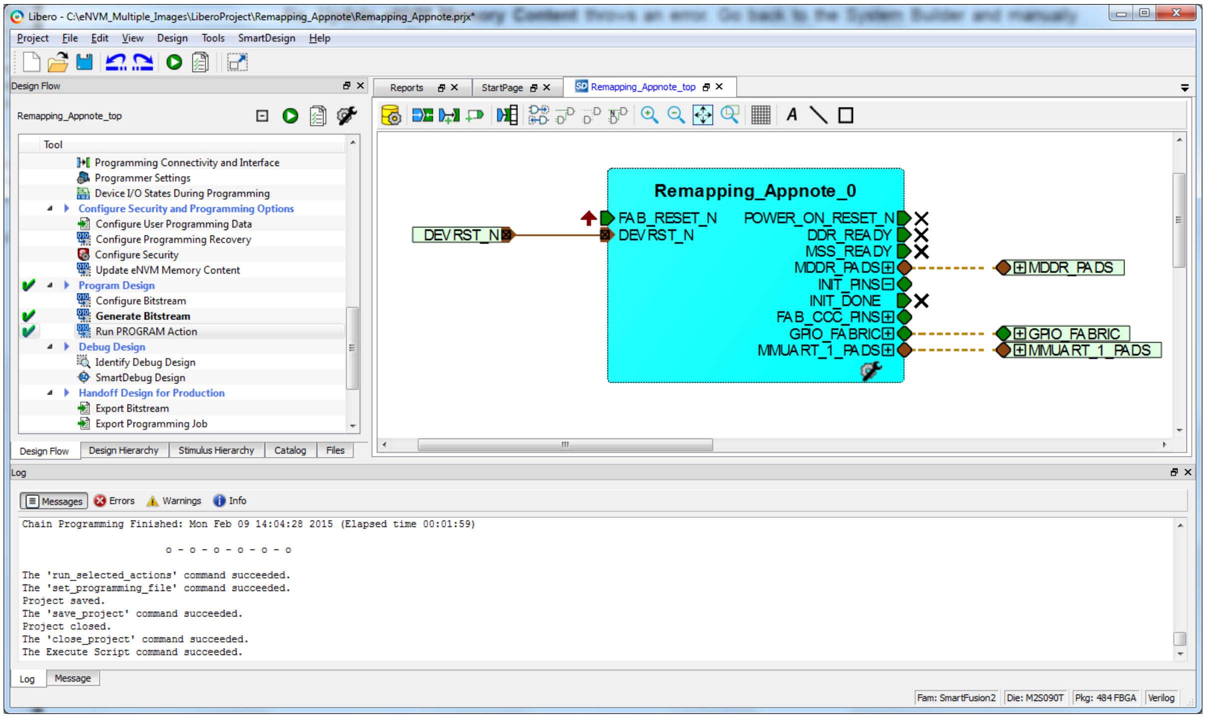
This option is available in Libero design flow to load the updated `.hex` files after compiling the application code in SoftConsole. Without this option, every time the SoftConsole application is built, System Builder eNVM Configurator is required to update the eNVM client with the new `.hex` file generated,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

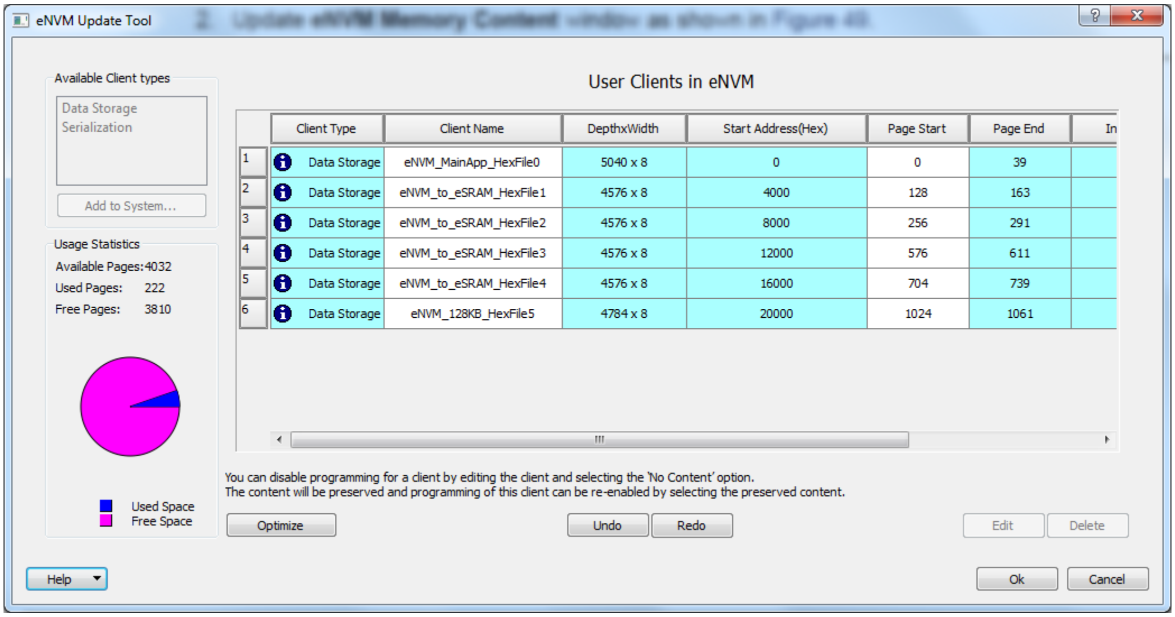
Figure 48 • **Update eNVM Memory Content Option**



```
Chain Programming Finished: Mon Feb 09 14:04:28 2015 (Elapsed time 00:01:59)

                        o - o - o - o - o - o

The 'run_selected_actions' command succeeded.
The 'set_programming_file' command succeeded.
Project saved.
The 'save_project' command succeeded.
Project closed.
The 'close_project' command succeeded.
The Execute Script command succeeded.
```

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

Figure 49 • **Update eNVM Memory Content Window**



Usage Statistics
Available Pages: 4032
Used Pages: 222
Free Pages: 3810

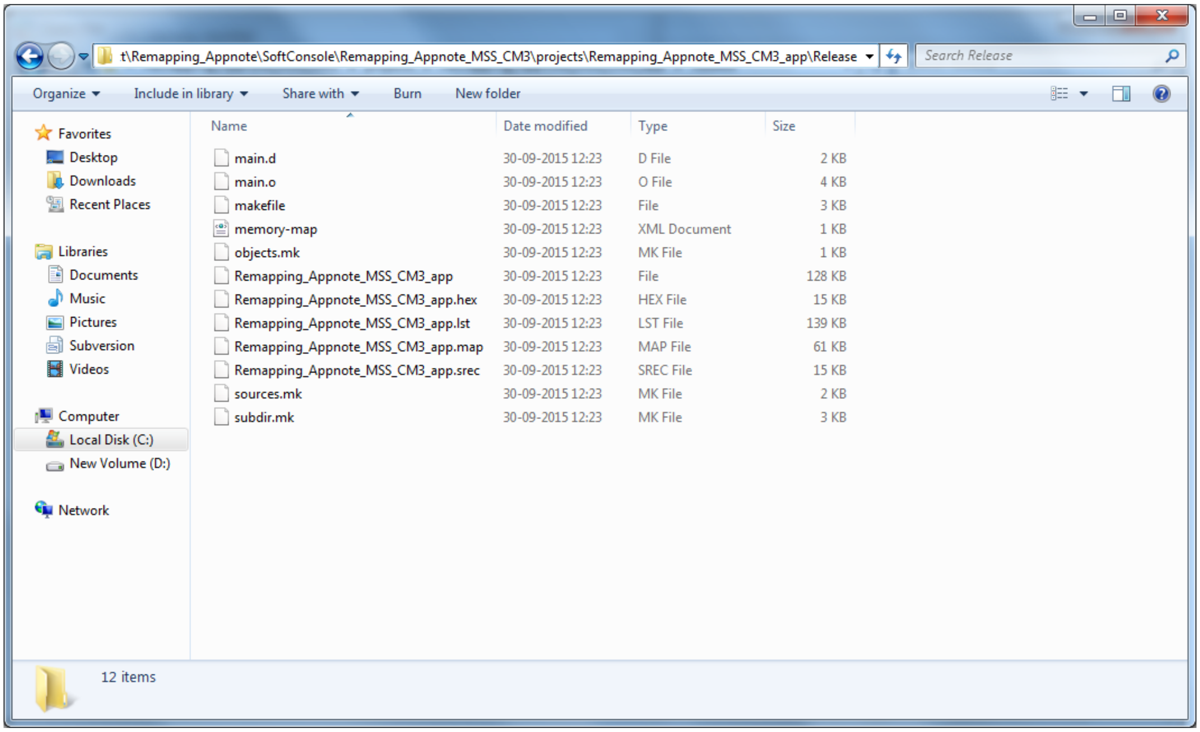
User Clients in eNVM

	Client Type	Client Name	DepthxWidth	Start Address(Hex)	Page Start	Page End	In
1	Data Storage	eNVM_MainApp_HexFile0	5040 x 8	0	0	39	
2	Data Storage	eNVM_to_eSRAM_HexFile1	4576 x 8	4000	128	163	
3	Data Storage	eNVM_to_eSRAM_HexFile2	4576 x 8	8000	256	291	
4	Data Storage	eNVM_to_eSRAM_HexFile3	4576 x 8	12000	576	611	
5	Data Storage	eNVM_to_eSRAM_HexFile4	4576 x 8	16000	704	739	
6	Data Storage	eNVM_128KB_HexFile5	4784 x 8	20000	1024	1061	

You can disable programming for a client by editing the client and selecting the 'No Content' option.
The content will be preserved and programming of this client can be re-enabled by selecting the preserved content.

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

Figure 50 • **Browsing through Updated Hex File**



Name	Date modified	Type	Size
main.d	30-09-2015 12:23	D File	2 KB
main.o	30-09-2015 12:23	O File	4 KB
makefile	30-09-2015 12:23	File	3 KB
memory-map	30-09-2015 12:23	XML Document	1 KB
objects.mk	30-09-2015 12:23	MK File	1 KB
Remapping_Appnote_MSS_CM3_app	30-09-2015 12:23	File	128 KB
Remapping_Appnote_MSS_CM3_app.hex	30-09-2015 12:23	HEX File	15 KB
Remapping_Appnote_MSS_CM3_app.lst	30-09-2015 12:23	LST File	139 KB
Remapping_Appnote_MSS_CM3_app.map	30-09-2015 12:23	MAP File	61 KB
Remapping_Appnote_MSS_CM3_app.srec	30-09-2015 12:23	SREC File	15 KB
sources.mk	30-09-2015 12:23	MK File	2 KB
subdir.mk	30-09-2015 12:23	MK File	3 KB

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

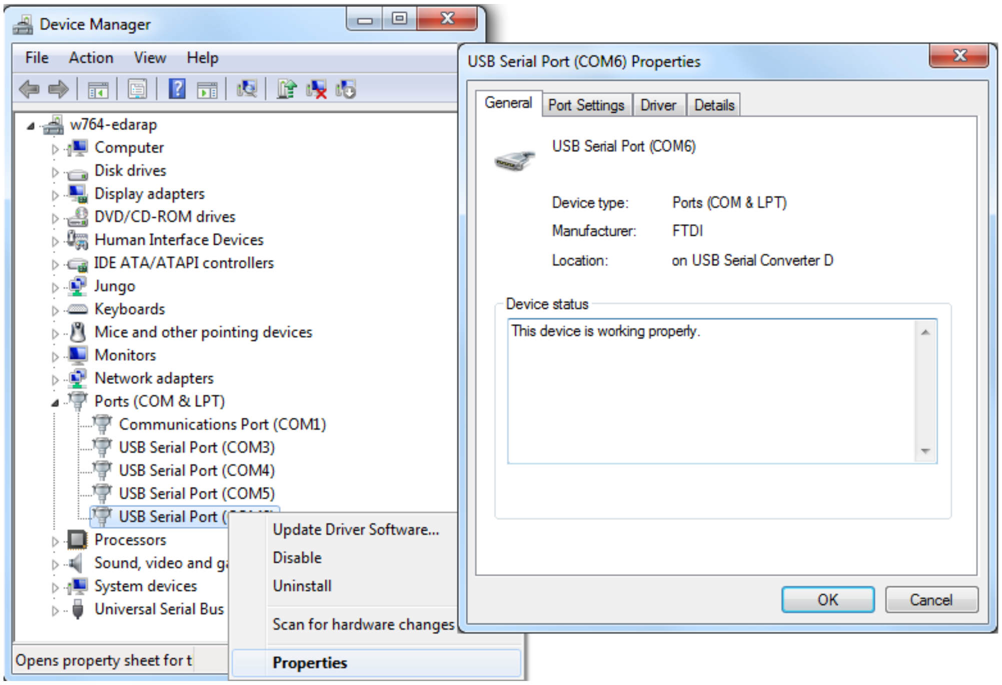
2.13 Running the Design on SmartFusion2 Security Evaluation Kit

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**



USB Serial Port (COM6)

Device type: Ports (COM & LPT)
Manufacturer: FTDI
Location: on USB Serial Converter D

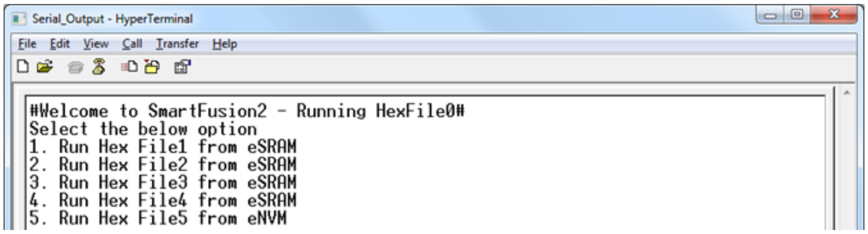
Device status
This device is working properly.

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.

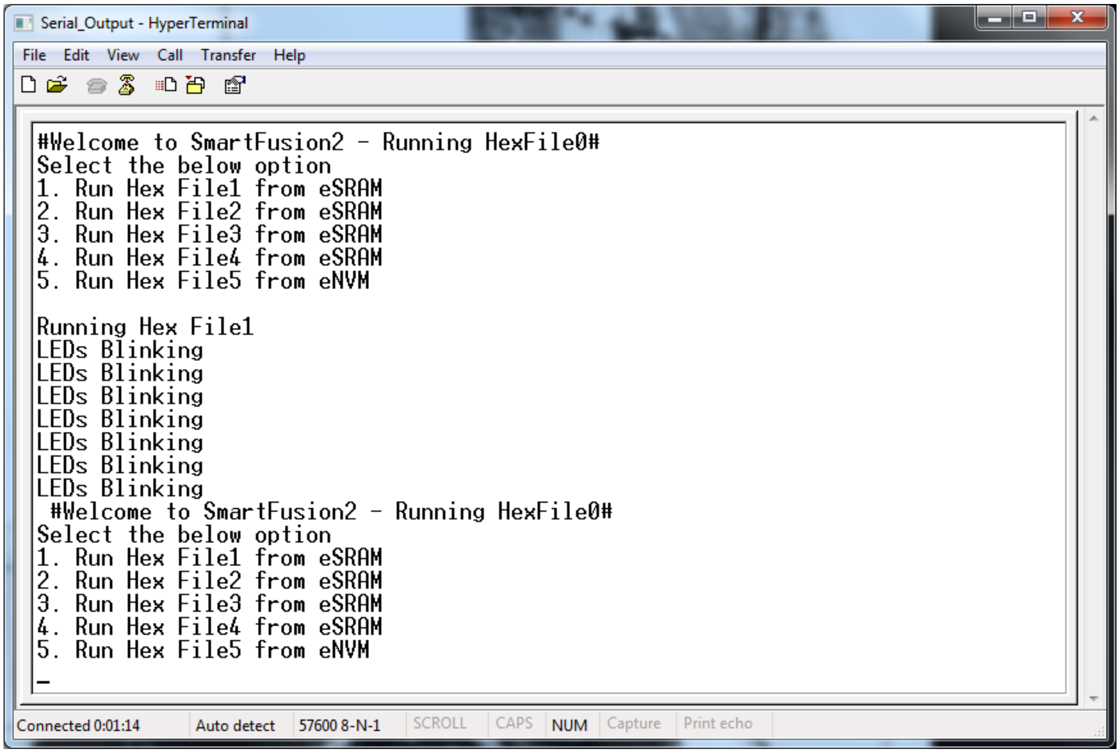
Figure 52 • **Running Hex File - User Options**


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
```

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**



```
#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
_
```

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 option5. 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

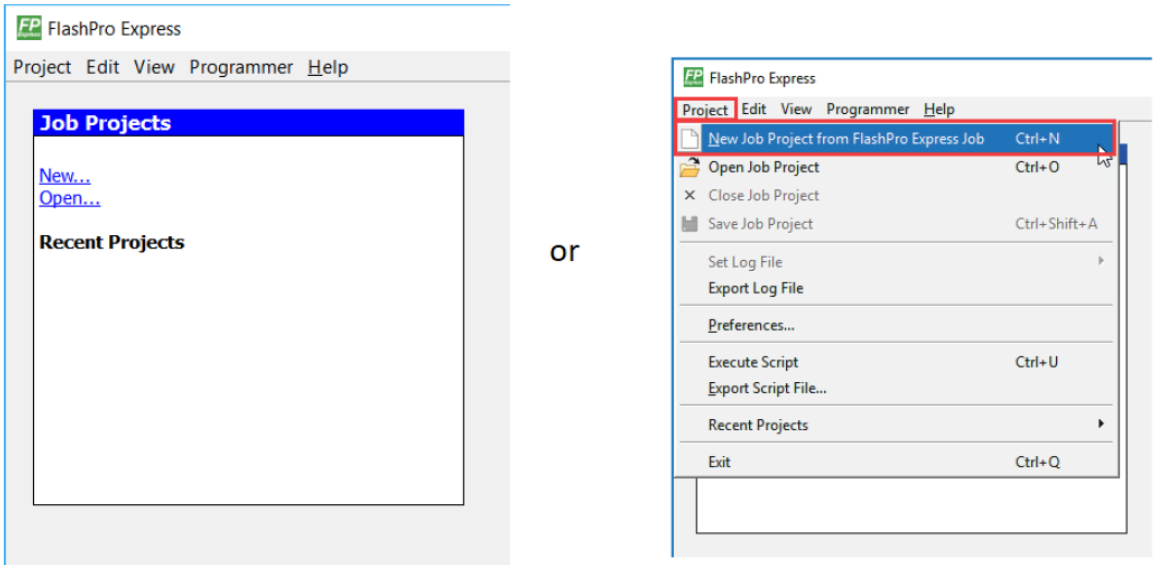
This section describes how to program the SmartFusion2 device with the programming job file using FlashPro Express.

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.

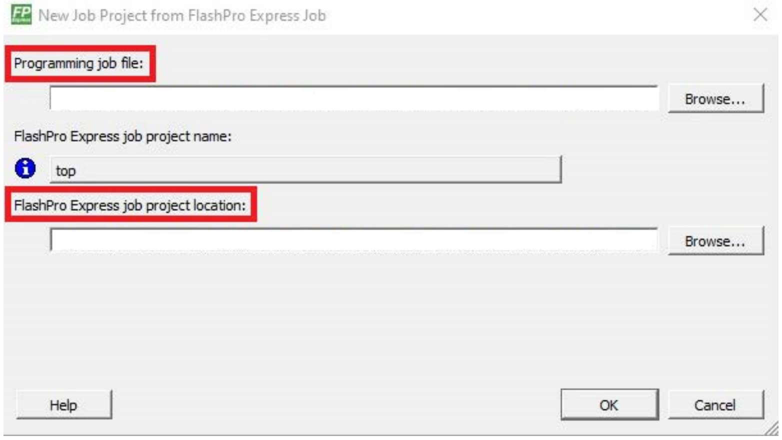
Figure 54 • **FlashPro Express Job Project**



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**

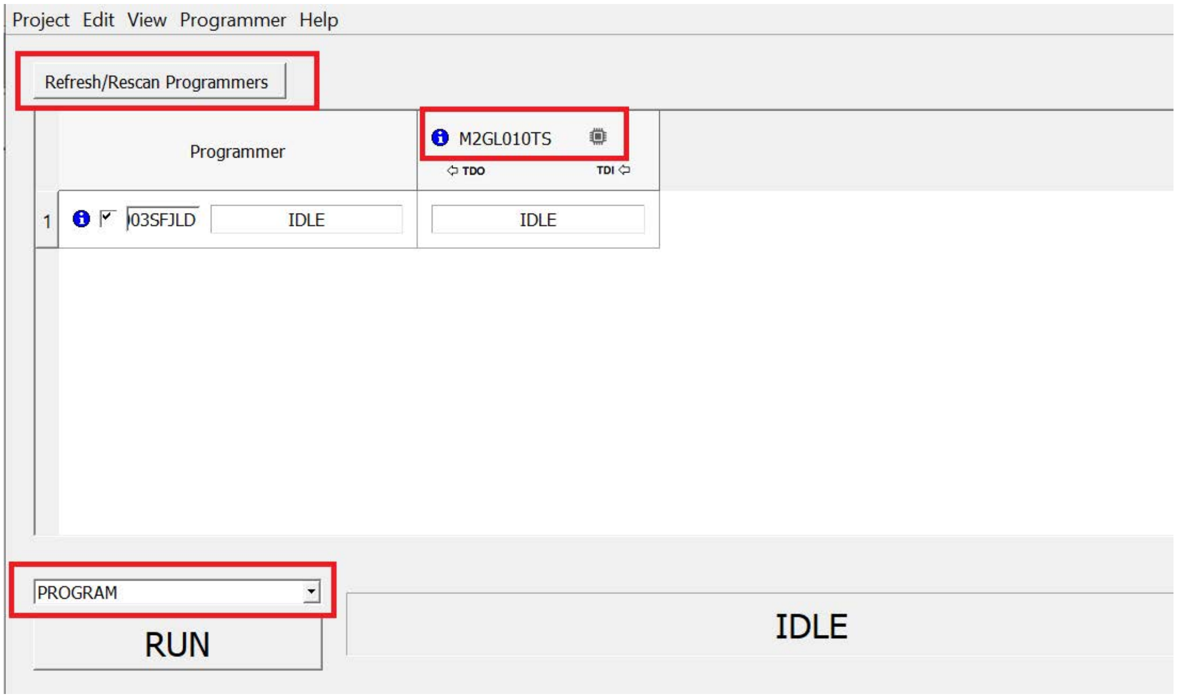


FlashPro Express job project name: top

New Job Project from FlashPro Express Job
Programming job file:
Browse...
FlashPro Express job project location:
Browse...
Help
OK
Cancel

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**

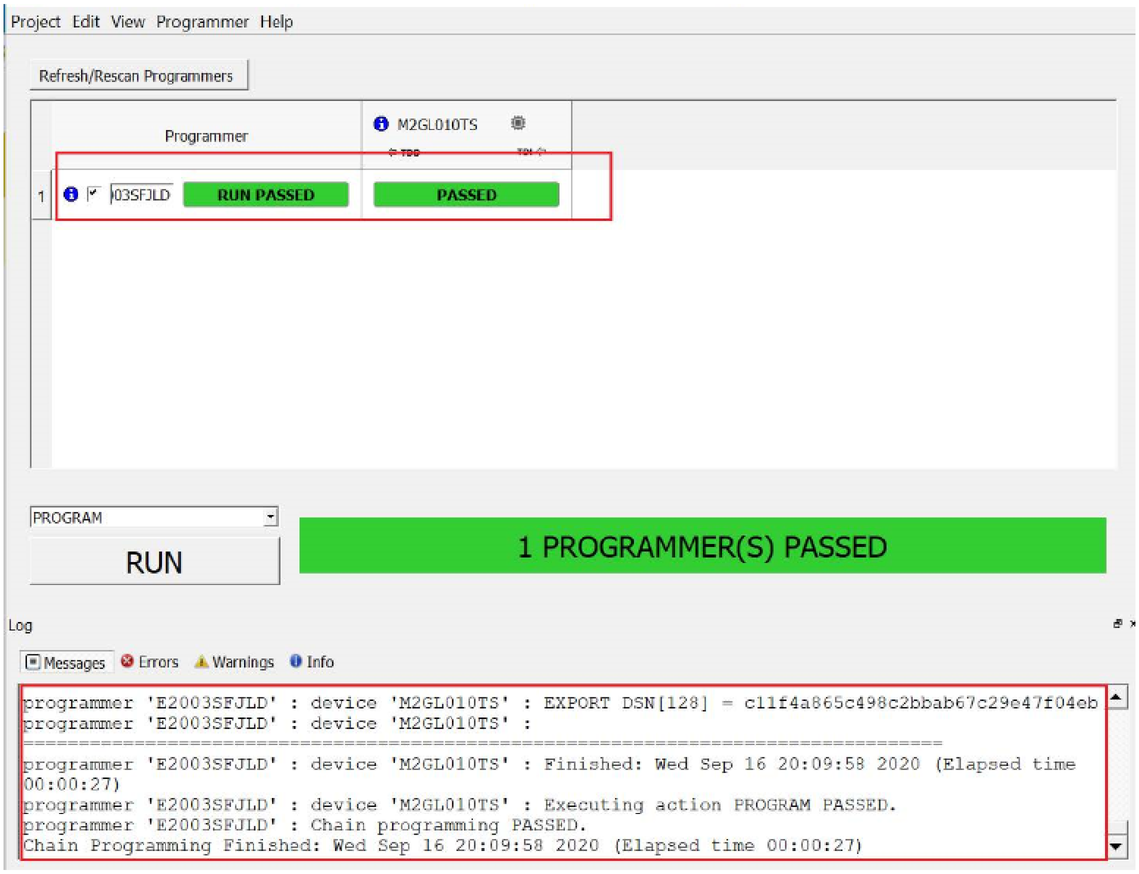


	Programmer		M2GL010TS
1	☑ 03SFJLD	IDLE	IDLE

PROGRAM

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



```
programmer 'E2003SFJLD' : device 'M2GL010TS' : EXPORT DSN[128] = c11f4a865c498c2bbab67c29e47f04eb
programmer 'E2003SFJLD' : device 'M2GL010TS' :
================================================================================
programmer 'E2003SFJLD' : device 'M2GL010TS' : Finished: Wed Sep 16 20:09:58 2020 (Elapsed time
00:00:27)
programmer 'E2003SFJLD' : device 'M2GL010TS' : Executing action PROGRAM PASSED.
programmer 'E2003SFJLD' : Chain programming PASSED.
Chain Programming Finished: Wed Sep 16 20:09:58 2020 (Elapsed time 00:00:27)
```

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

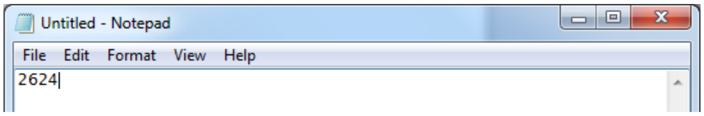
4 Appendix 2: Creating LookUpTable.Hex File

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.

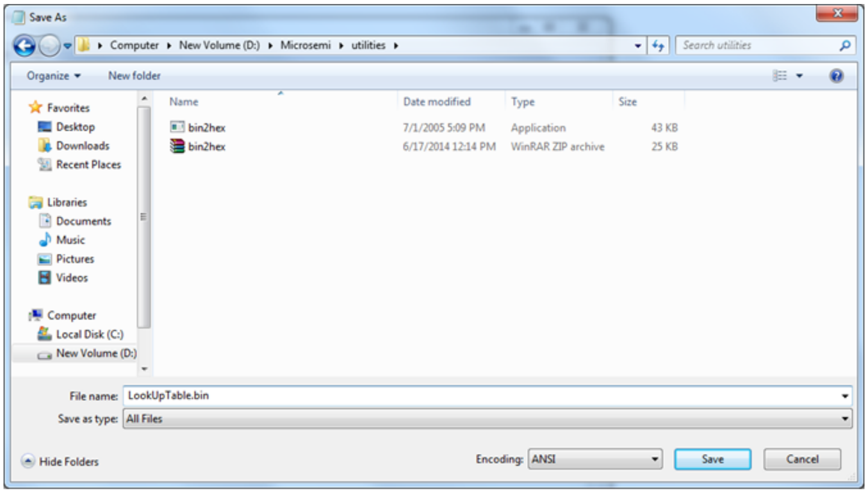
Figure 58 • **Entering Factorial Values in Notepad**



```
2624
```

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

Figure 59 • **Saving Notepad**



Name	Date modified	Type	Size
bin2hex	7/1/2005 5:09 PM	Application	43 KB
bin2hex	6/17/2014 12:14 PM	WinRAR ZIP archive	25 KB

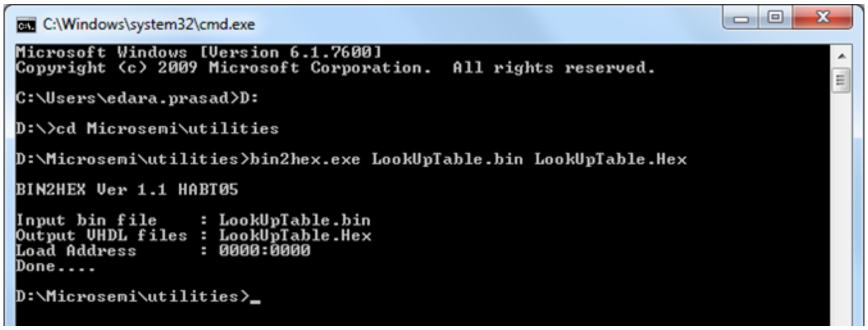
File name: LookUpTable.bin

Save as type: All Files

Encoding: ANSI

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.

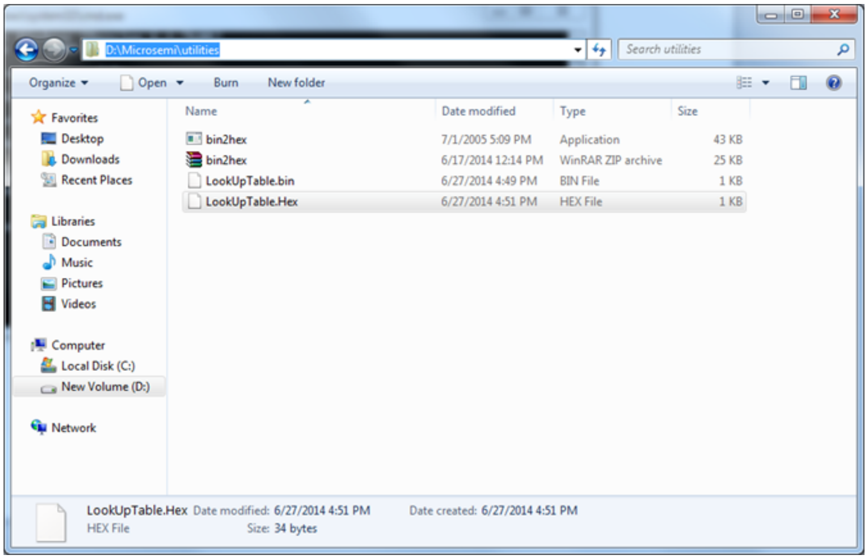
Figure 60 • **Converting Binary File to Hex File**



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\edara.prasad>D:

D:\>cd Microsemi\utilities

D:\Microsemi\utilities>bin2hex.exe LookUpTable.bin LookUpTable.Hex

BIN2HEX Ver 1.1 HABT05

Input bin file     : LookUpTable.bin
Output VHDL files  : LookUpTable.Hex
Load Address       : 0000:0000
Done....

D:\Microsemi\utilities>_
```

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**



Name	Date modified	Type	Size
bin2hex	7/1/2005 5:09 PM	Application	43 KB
bin2hex	6/17/2014 12:14 PM	WinRAR ZIP archive	25 KB
LookUpTable.bin	6/27/2014 4:49 PM	BIN File	1 KB
LookUpTable.Hex	6/27/2014 4:51 PM	HEX File	1 KB

LookUpTable.Hex Date modified: 6/27/2014 4:51 PM Date created: 6/27/2014 4:51 PM
HEX File Size: 34 bytes