



Austin Design Center
11000 North Mopac Expressway
Stonelake Bldg. 6 Suite 500
Austin, Texas 78759

USBDM User Guide ***V2.1.2.1***

Updated 7/27/08

The information contained herein is confidential and proprietary to SMSC, shall be used solely in accordance with the agreement pursuant to which it is provided, and shall not be reproduced or disclosed to others without the prior written consent of SMSC. Although the information is believed to be accurate, no responsibility is assumed for inaccuracies. SMSC reserves the right to make changes to this document and to specifications and product descriptions at any time without notice. Neither the provision of this information nor the sale of the described semiconductor devices conveys any licenses under any patent rights or other intellectual property rights of SMSC or others. The product may contain design defects or errors known as anomalies, including but not necessarily limited to any which may be identified in this document, which may cause the product to deviate from published specifications. SMSC products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an officer of SMSC will be fully at the risk of the customer. SMSC is a registered trademark of Standard Microsystems Corporation ("SMSC").

SMSC DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND AGAINST INFRINGEMENT AND THE LIKE, AND ANY AND ALL WARRANTIES ARISING FROM ANY COURSE OF DEALING OR USAGE OF TRADE. IN NO EVENT SHALL SMSC BE LIABLE FOR ANY DIRECT, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES; OR FOR LOST DATA, PROFITS, SAVINGS OR REVENUES OF ANY KIND; REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF SMSC OR OTHERS; STRICT LIABILITY; BREACH OF WARRANTY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY OF BUYER IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER OR NOT SMSC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Table of Contents

Revision History	3
Using the USB Drive Manager Application (for Windows Vista/XP only)	3
The Info Tab	3
The Branding Tab	4
Using .dat files with USBDM	4
The Configuration Tab	4
Attribute Bit Definitions	5
Programming the NVStore Data.....	5
Using Device Firmware Upgrade (DFU)	6
Overview	6
Choosing a Flash Eeprom for Your Device.....	6
Setting up the Hardware	6
Using the USBDM Application to Perform Device Firmware Upgrade (DFU).....	7
Console Mode	7
USBDM Revision History	9

Revision History

1.0: Initial Revision

Using the USB Drive Manager Application (for Windows Vista/XP only)

The USB Drive Manager (USBDM) application can be used to create the eeprom.dat file and program the a device via USB, plus some additional functions such as creating end-user firmware updates contained within a single, easily distributable exe, and having the ability to instantly read the NVStore data from the device without the need for a driver swap.

Note: USBDM will not work for updating a SMSC standalone hub.

Note: The USBDM Application is supported in Windows Vista/XP only.

Getting Started:

To start the USB Drive Manager application, run the USBDM_Setup.msi Windows installer file. The application can be launched by clicking on the Start\Programs\USBDM\USBDM icon. Once the application opens you will see the screen showing the device information. If there is no device present, a virtual device will be listed instead of the USB MSC Device information. This virtual device allows a .dat file to be edited without the need for a device to be attached to the host computer.

The USBDM Toolbar

The toolbar buttons are displayed at the top left hand side of the application. Starting from left to right, they perform the following functions:

Button 1: Refresh Drive List
Button 2: Load .dat file
Button 3: Save .dat file
Button 4: Upload Firmware
Button 5: Copy
Button 6: Paste

*If you do not see these buttons displayed, go to “View” in menu bar and make sure there is a check next to the “Toolbar” option.

*Clicking on the “Help” option above the toolbar and selecting “About Drive Manager” will display the version of the USBDM application.

The Info Tab

The info tab is displayed whenever a USB mass storage class device is attached to the host while USBDM is running. This tab displays the key fields in the NVStore data for the device. Note: Unless the device contains the SMSC USBDM firmware extensions, most of the data fields will display INVALID.

Attach a device containing the USBDM firmware extensions to the PC via a USB cable. The USB Drive Manager application will read the NVStore data for this device if valid data exists. It will display information for each drive that is available on the device. You can toggle between the information for each of these drives by single clicking on the Drive entry under the “USB MSC Device” folder on the left side of the application.

Note: The detach button seen on this tab will momentarily detach the target device from the system.

The Branding Tab

The Branding tab is used to write vendor specific data to the NVStore. Programmable fields include: Vendor ID, Product ID, Language ID, Product String, Manufacturing String, and Serial Number String. Any of this information can be changed on the device. Once you have entered the information for your device, click on the “Update Now” button to program the NVStore.

Vendor ID: Unique for every vendor. Assigned by the USB Implementers Forum.

Product ID: Unique to product. Assigned by vendor.

Language ID: 0409 is the Language Code for English. Other Language Codes may be found in the USB specification.

Product String: 28 characters max. Used to identify the product. This string will be used during the USB enumeration process in Windows.

Manufacturing String: 28 characters max. Used to identify the manufacturer.

Serial Number: 12 hex digits max. Must be unique to each device.

Inquiry Manufacturer (8 Bytes) and Product (5 Bytes) ID Strings: If bit 4 of the 1st attribute byte is set, the device will use these strings in response to a USB inquiry command, instead of the USB Descriptor Manufacturer and Product ID Strings.

Using .dat files with USBDM

The Load .dat file button can be used to populate these fields from a valid .dat file. After clicking the Load .dat file button, you will be prompted to specify a .dat file. Once the .dat file has loaded, the text fields will be updated to reflect the data in the .dat file. Any changes made to the text fields can also be saved into a .dat format using the Save .dat file button at the top of the application.

The Configuration Tab

The Configuration tab contains all of the other NVStore programmable fields not found in the Branding Tab.

The Configuration Tab is where you set:

- 1) The NVStore signature
- 2) The attribute bits
- 3) The LUN assignments
- 4) The LUN IDs
- 5) Miscellaneous settings such as the USB descriptors bMaxPower and bmAttribute

These user programmable fields are described in detail in the following paragraphs.

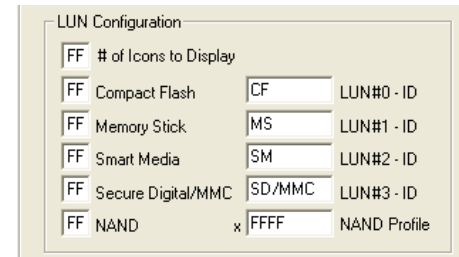
Attribute Bytes: This field should only be used in development phase to modify attribute bits that have not yet been implemented into check boxes and dropdown choices on this tab. There needs to be full understanding of what effect altering a specific bit will have on the device before changing this field. All released features will be able to be selected without utilizing this field.

Attribute Bits: The attribute bits are used to customize the functionality of the firmware. They are organized by which particular media type they pertain to. Attribute bit checkboxes that are not specific to a media type are contained in the Misc. Settings section. A complete list of all programmable attribute bits and their function is listed in the section of this document entitled “Attribute Bit Definitions and NVStore Editable Values.” In the image shown above “Reverse SD Card Write Protect Sense” is the only option selected. Placing a check to the left of an option sets an attribute bit. If the box is unchecked, the attribute bit will be cleared. Any of these options may be checked or unchecked depending on the various needs for the product being programmed. There are also dropdown options in the “MMC-4” section.

LUN Configuration:

LUN ID Strings (7 bytes each)—There are four LUN ID strings corresponding to LUN# 0,1,2, and 3.

Number of Icons to Display, CF Lun #, MS Lun #, NAND Lun #, SD/MMC Lun #, SM Lun #- These bytes are used to specify the number of LUNs the device exposes to the host. These bytes are also used for icon sharing- Assigning more than one LUN to a single icon. This is used in applications where the device utilizes a combo socket and the OEM wishes to have only a single icon displayed for one or more interfaces. For more information, see the section of this document entitled “LUN Configuration and Icon Sharing.” If this field is set to “FF”, the program assumes that you are using the default value of “04” and will display icons for CF, MS, SM, and SD. If this field is any other value besides “FF”, you must specify the LUN# assignments in the boxes starting with LUN 00 and going to (# of Icons to Display -1).



LUN Configuration			
FF	# of Icons to Display		
FF	Compact Flash	CF	LUN#0 - ID
FF	Memory Stick	MS	LUN#1 - ID
FF	Smart Media	SM	LUN#2 - ID
FF	Secure Digital/MMC	SD/MMC	LUN#3 - ID
FF	NAND	x FFFF	NAND Profile

Note that more than one interface (CF, MS, SM, or SD) can share a LUN. Remember **LUN numbering always starts at 00**.

The configuration to the right directs the firmware to show three LUNs in the order of CF, SD/MMC, and SM. Note that Memory Stick is not enabled in this configuration.

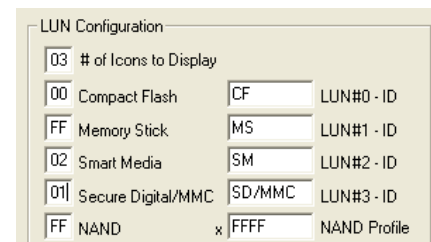
Of Icons to Display: 03

Compact Flash (1st LUN): 00

Memory Stick (will not display): FF

Smart Media (2nd LUN): 02

Secure Digital/MMC (3rd LUN):01



LUN Configuration			
03	# of Icons to Display		
00	Compact Flash	CF	LUN#0 - ID
FF	Memory Stick	MS	LUN#1 - ID
02	Smart Media	SM	LUN#2 - ID
01	Secure Digital/MMC	SD/MMC	LUN#3 - ID
FF	NAND	x FFFF	NAND Profile

Misc. Settings: The Misc. Settings section is used to program the other miscellaneous NVStore editable values. They are:

- 1) **bMaxPower** (1 byte): Per USB specification. Do not set this value greater than 100mA
- 2) **Blink Rate** (1 byte): Programmable in 10ms intervals. Hi bit indicates idle state: 0–Off, 1–On. The remaining bits are used to determine the blink interval up to a max of 128 x 10 ms.
- 3) **Blink Duration** (1 byte): This byte is used to designate the number of seconds that the GPIO 0 LED will continue to blink after a drive access. Setting this byte to “05” will cause the GPIO 0 LED to blink for 5 seconds after a drive access.
- 4) **bmAttribute** (1 byte): Per USB Specification.
 - 80 – Device is Bus Powered
 - C0 – Device is Self Powered

Attribute Bit Definitions

Attributes (4 bytes): The attribute value for your device is determined by the options selecting in the USBDM utility provided by SMSC. Changing the checkboxes and dropdown boxes and updating the device can update this information. In the USBDM GUI, these bits are organized by which media type/feature they affect. The majority of these bits are displayed as checkboxes in the USBDM GUI.

Programming the NVStore Data

Once the eeprom.dat file has been created and loaded into USBDM, you are ready to program the NVStore data into your device.

Press the “Update Now” button on either the Branding or Configuration Tab of the USBDM application. Both buttons will update all of the information displayed on any tab in USBDM. The operation will report that the Update completed successfully once the data has been programmed.

Using Device Firmware Upgrade (DFU)

Overview

Device Firmware Upgrade (DFU) is the process by which device firmware is updated through a standard USB cable, eliminating the need to remove, reprogram, and replace flash memory. This operation is accomplished by special code in the internal ROM. This code can then later be called upon to essentially change the USB device into a flash programmable device. Then new firmware can then be uploaded to the device and reprogrammed into the flash. Once the operation is complete, the device configures itself back to a normal USB device and begins utilizing the new firmware.

SMSC's Device Firmware Upgrade (DFU) package gives manufacturers the ability to easily utilize DFU to dynamically update the firmware and descriptor information in their devices. This allows for in circuit programming of new device firmware both on the assembly line, and by the end user in the field. This affords both the manufacturer and the end user a great opportunity to utilize the feature enhancements and bug fixes of new code immediately once it becomes available.

Choosing a Flash Eeprom for Your Device

SMSC provides customers the "dfu.hex" file that supports only the **following** flash eeproms. SST39XF010, SST29SF010, SST29SF020, SST29SF040, SST29VF020, SST29VF040, SST39XF020, SST39XF040, STM29f010b, M29W010B, M29W200BT, M29W200BB, M29W400BT, M29W400BB, M29W040B, MX29F00LT, MBM29LV400TC, MBM29LV200TC, STM29W010B, PM39LV010, PM39LV020, PM39LV040, AM29LV001BT, AM29LV001BB, AM29LV002BT, AM29LV002BB, AM29LV200BT, AM29LV200BB, S29AL004DT, S29AL004DB, S29AL008DT, S29AL008DB AM29LV010B, AM29LV004BT, AM29LV004BB, AMLV040B.

Note that for all flash devices > 1Mbit that support a "bottom boot sector", the flash should be configured so that the boot sector addresses appear in the lower 64KB and are configured for "BYTE WIDE" access if applicable.

Note that for all flash devices > 1Mbit that support a "top boot sector", the flash should be configured so that the top boot sector addresses does NOT appear in the lower 128KB and are configured for "BYTE WIDE" access if applicable.

While all of these flash support DFU firmware uploads, only the SST39xx010 supports NO EEPROM operation.

If you wish to use another flash in your device, it would most likely require some modification to the existing DFU code by SMSC to support the electrical characteristics of the new chip. If this is the case, please contact SMSC sales to have the project scheduled.

If you do decide to use another flash eeprom, there are a few requirements to look for to make sure it will work with DFU. First of all it should be 128KB and byte writable. It needs to have an access time of less than 66 nanoseconds in order for the external firmware to operate properly. Also, it should have equivalent programming characteristics as the supported chips, i.e., block size, erase size, read/write/erase speed, command set, and command address. Provided the chip meets all of the above requirements, there is a good chance that it will support DFU.

Setting up the Hardware

Either a USB 1.1 or 2.0 controller may be used for the DFU operation, however some USB 2.0 host controller drivers such as OMI's have been found to have defects which prevent DFU from performing normally. If you are going to use a USB 2.0 host controller, it is recommended that you use Microsoft's host controller drivers in order to achieve the best results. Once the board is attached and powered up, it should enumerate as a normal USB flash media controller. When you see the drive icon(s) appear, the device is ready. The following section describes the next step in the process, which is setting up the software application to perform the DFU.

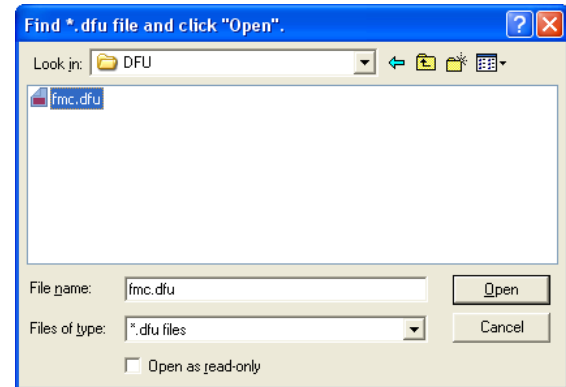
Using the USBDM Application to Perform Device Firmware Upgrade (DFU)

Updating the Firmware:

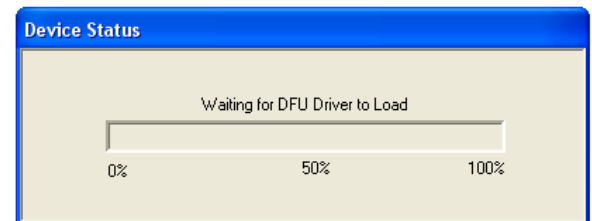
To perform a firmware update, click on the “Upload Firmware” button at the top of the application.



You will then be prompted to select the .dfu file that you wish to upload to your device. Navigate to the .dfu file (if it is not already listed in the current folder) and click open.



You will see a pop up box on your screen that displays the status of the firmware upload. This status will cycle through “Waiting for DFU Driver to Load,” “Switching to DFU Mode,” “Uploading New Firmware,” “Validating New Firmware,” and “Firmware Upload Successful.” Once the loading is complete you will be prompted to unplug the device and reattach it to continue (or to restart the host if the device is internally mounted). Once the device is reattached, the device will enumerate and the information for the updated firmware will be loaded into the USB Drive Manager application.



Note: The first time USBDM is used for DFU on a Windows XP host, the found new hardware wizard will be seen when the DFU driver is used during the firmware update process. This will only happen the first time a DFU is performed on a host. When this comes up, choose to have windows automatically install the driver. Choose to continue loading the SMSC DFU driver even though it is unsigned. While this is occurring, you may receive a message from USBDM asking you if you wish to continue waiting for the device to respond. Select yes to continue waiting.

Console Mode

When executed with command line parameters the application will perform the requested function then terminate. The only two acceptable parameters are dfu and nvstore.

At a command prompt, type the following:

```
executable <parameter one> <parameter two> <vid> <pid>
```

```
executable:      usbdm
parameter one:   operation (dfu or nvstore)
parameter two:   file
parameter three: direction (an i for input or an o for output – only for nvstore)
vid:             vendor ID (hex) (i.e. 035c)
pid:             product ID (hex) (i.e. 053b)
```

Examples:

```
usbdm dfu c:\SMSC\USB2256_ext.dfu 035c 053b
usbdm nvstore c:\SMSC\USB2256.dat o 035c 053b
```

The VID/PIDs are optional, but highly recommended.

The application will return a value of 0 to indicate success. Otherwise, one of the following values will be returned:

ERROR_FORMAT_FAILED	1
ERROR_HANDLE_INVALID	2
ERROR_IN_INQUIRY_DATA	3
ERROR_IN_READ_CAPACITY_DATA	4
ERROR_IN_DEVICE_GEOMETRY	5
ERROR_IN_MEDIA_ERASE	6
ERROR_DRIVE_WRITE_PROTECTED	7
ERROR_WRITING_TO_DEVICE	8
ERROR_INVALID_PARAMS_FOR_FORMAT	9
ERROR_INSUFFICIENT_RESOURCES	10
ERROR_IN_REQUEST_SENSE_DATA	11
ERROR_DATA_TRANSFER_TOO_BIG	12
ERROR_READING_FROM_DEVICE	13
ERROR_SYSTEM_FILES_NOT_FOUND	14
ERROR_CANNOT_COPY_SYSTEM_FILES	15
ERROR_CANNOT_LOCK_VOLUME	16
ERR_NO_MATCHING_DEVICES_CONNECTED	17
ERR_TOO_MANY_MATCHING_DEVICES	18
ERR_NOT_ENOUGH_MEMORY	19
ERR_DOWNLOAD_FAILED_RESTORE_FAILED	20
ERR_FIRMWARE_FILE_NOT_FOUND	21
ERR_READING_FIRMWARE_FILE	22
ERR_DEVICE_ID_FIRMWARE_MISMATCH	23
ERR_PRODUCT_ID_FIRMWARE_MISMATCH	24
ERR_VENDOR_ID_FIRMWARE_MISMATCH	25
ERR_FIRMWARE_INVALID_OR_MISSING_SUFFIX	26
ERR_CANT_ISSUE_USB_RESET	27
ERR_DFU_DETACH_FAILED	28
ERR_DFU_DOWNLOAD_FAILED	29
ERR_CANT_GET_DFU_DESCRIPTOR	30
ERR_DFU_STATE_ERROR	31
ERR_DFU_GET_STATUS_FAILED	32
ERR_DFU_UPLOAD_FAILED	33
ERR_OPENING_DFU_DRIVER	34
ERR_PNP_DRIVER_UPDATE_FAILED	35
ERR_REBOOT_REQUIRED	36
ERR_DO_MANUAL_SWAP	37
ERR_EEPROM_BUFFER_TOO_SMALL	38
ERR_RECEIVING_EEPROM_DATA	39
ERR_WRITING_EEPROM_DATA	40
ERR_DFU_RECOVERY_FAILED	41
ERR_LOAD_DAT_FILE	42
ERR_WRITE_NVRAM	43

Note: The first time a dfu is done, the found new hardware wizard will be seen. When this comes up, choose the location of the SMSC dfu driver and load it even though it is unsigned. While this is occurring, you may receive a message asking you if you wish to continue waiting for the device to respond. Select 'yes' to continue waiting.

USBDM Revision History

- 2.1.1.9: - **DFU and NVStore console mode for Sony.**
 - Enabled console mode dfu and nvstore functionality.
- 2.1.2.0 - Added ability to enter VID/PID for console mode.
- 2.1.2.1 - Added capability to read MS data bus width in console mode.