User Manual

Data FLASH Editor

Data FLASH Editor tool for 78K0R and V850 based EEPROM Emulation

Renesas Electronics
www.renesas.com

R01UT0176ED0000, Rev. 1.00
March 31, 2011

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics.

8. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti- crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems;medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

9. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

10. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

11. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.

13. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

    (Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority- owned subsidiaries.

    (Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

Regional Information

Some information contained in this document may vary from country to country. Before using any Renesas Electronics product in your application, please contact the Renesas Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

Visit

http://www.renesas.com

to get in contact with your regional representatives and distributors.

Table of Contents

Chapter 1 Introduction

The Data FLASH Editor is a windows based tool that allows to visualize and modify the EEPROM emulation data of a Data FLASH memory image of a 78K0R or V850 based microcontroller. The Data FLASH Editor does support memory images in the Intel-hex or Motorola-S record file format. The corresponding memory image can be generated by the Data FLASH Converter or it can be uploaded from a 78K0R or V850 based In-Circuit Emulator by using the IAR, GHS or Renesas debugger.

The Data FLASH Editor supports to view, modify and to save EEPROM emulation data of an existing Data FLASH memory image. In addition, the Data FLASH Editor supports to add or remove EEPROM emulation data entries to respectively from a existing Data FLASH memory image.

The Data FLASH Editor does distinguish between three different 78K0R and V850 based device families and FLASH Technologies:

- **78K0R**: MF2 FLASH Technology in combination with the 78K0R/Fx3 EEPROM emulation library EEL (Type T03)
- **V850E/ES**: UX4 and MF2 FLASH Technology in combination with the V850 EEPROM emulation library EEElib (previous Type)
- **V850E2**: UX6LF FLASH Technology in combination with the V850 EEPROM emulation library EEL (Type T05)

Different to V850E/ES device family, the Data FLASH Editor needs to read or to create an XML based raw data management description file first, to support 78K0R and V850E2 devices accordingly. This is caused by the different emulation library concepts of V850E/ES and V850E2, 78K0R. By reading or creating the raw data management description file the Data FLASH Editor gets the corresponding information about the data set IDs and data set length information. Based on this information only the latest valid IDs and matched data sets of a Data FLASH memory image file will be shown as validated data sets within the Data FLASH Editor window. The remaining data sets will be shown as invalidated data sets within the editor window.

1.1 Abbreviations

Abbreviations	Description
Data FLASH	Embedded FLASH where mainly the data of the EEPROM emulation are stored. Beside that also code operation might be possible.
EEPROM emulation	In distinction to a real EEPROM the EEPROM emulation uses some portion of the flash memory to emulate the EEPROM behavior. To gain a similar behavior some side parameters have to be taken in account.
FLASH	“FLASH EPROM” - Electrically erasable and programmable nonvolatile memory. The difference to ROM is, that this type of memory can be re-programmed several times.
FLASH Block	A FLASH block is the smallest erasable unit of the FLASH memory.
EEL	EEPROM Emulation Library
FAL	FLASH Access Library (FLASH access layer)
FDL	FLASH Data Library (Data FLASH access layer)
FSL	FLASH Self-programming Library (accessing code FLASH)
FW	Firmware
FCB	FLASH Control Block
XML	e**X**tensible **M**arkup **L**anguage
MF2	FLASH Technology of 78K0R based MCUs
UX4 / MF2	FLASH Technology of V850E/ES based MCUs
UX6LF	FLASH Technology of V850E2 based MCUs

Table 1 Abbreviations

1.2 Features of Data FLASH Editor

Common features and features for V850E/ES(MF2/UX4) :

- NET Framework 2.0 based application
- User friendly and intuitive windows based graphical user interface.
- Visualization of EEPROM emulation data of a Data FLASH memory image.
- Modification of EEPROM emulation data entries: ADD, MODIFY or REMOVE.
- Loading of Data FLASH memory images generated by the Data FLASH Converter or IAR, GHS, Renesas debuggers.
- Restoring of modified EEPROM emulation data to an existing Data FLASH memory image.

Enhanced features for V850E2(UX6LF) and 78K0R(MF2):

- Reading XML raw data management description file.
- Verifying of XML file based on device parameter file.
- Creation and modification of administration section within XML file.
- Sorting of data set based ID in the Data FLASH editor window.
- Handling data set state between validate and invalidate.
- Creating, saving and modifying of XML file.
- Selecting of data sets from Data FLASH memory image file based on XML file information.

1.3 Trademarks

Chapter 2 Documents / References

For more details on the EEPROM emulation concept of the V850 and 78K0R microcontroller families and the corresponding tools, please refer also to the documents below.

Doc number	Title	Doc Type
U18005EE1V2AN00	32-bit Single-Chip Microcontrollers V850 EEPROM Emulation Library	Application Note
U18042EE1V2AN00	32-Bit Single-Chip Microcontroller V850 Library for the Data Flash Access Layer	Application Note
U19003EE2V0UM00	Data FLASH Converter Conversion tool for data flash memory image generation	Users Manual
U20280EE1V0UM00	V850 Single Voltage Flash devices with Data Flash and based on UX6LF Flash technology EEPROM Emulation Library	Users Manual
R01AN0006ED0102	16 Bit Single-chip Microcomputer 78K0R/Fx3 Series EEPROM Emulation Library	Application Note

Table 2 Documents / References

Chapter 3 Input files / formats

A standard device parameter file, file name extension “*.pr5”, which is used in combination with the PG-FP5 FLASH programmer, is used also by the Data FLASH Editor to gain the corresponding Data FLASH information of the chosen V850 or 78K0R device. The device parameter file has to be loaded first, before any other operation can be performed.

The Data FLASH Editor does support Data FLASH memory images in the Intel-hex and Motorola-S record file format. Additionally for V850E2(UX6LF) and 78K0R(MF2) based devices, a XML raw data description file must be loaded or created before reading a corresponding Data FLASH memory image file in HEX or SREC format.

The predefined tags of an XML based raw data description file are as following:

DataFlash	**Defining the start of the definition file.**
AdminstrativeSection	This section is describing the physical parameters that are needed for the file generation.
FAL_CONFIG_DATAFLASH_SIZE	Number of Data FLASH blocks, accessible by the FAL. Typical number of available Data FLASH blocks.
FAL_CONFIG_BLOCK_SIZE	Data FLASH block size in bytes.
EEL_CONFIG_BLOCK_START	1st block of the EEL pool. ***(Note)***
EEL_CONFIG_BLOCK_CNT	Number of blocks for the EEL pool.
EEL_CONFIG_DF_BASE_ADDRESS	Data FLASH base address.
EEL_CONFIG_BLOCK_CNT_REFRESH_THRESHOLD	Threshold for minimum number of prepared blocks.
EEL_CONFIG_ERASE_SUSPEND_THRESHOLD	Threshold for erase suspend counter. Exceeding this result in warning. ***(Note)***
Record	Defining the start of a ID-L record.
ID	Identifier for the data record (16bit). Format is fixed to four digits in hex.
IDName	Defining the name off the record.
Length	Length definition of the data record.
Data	Any hexadecimal or decimal

***Note*:** These XML values were not used by 78K0R(MF2) based devices. The values were fixed to zero in case of 78K0R(MF2).

Table 3 : Predefined tags of an XML based raw data description file

```
<?xml version="1.0"?>
<DataFlash>
 <AdministrativeSection>
   <FAL_CONFIG_DATAFLASH_SIZE>16</FAL_CONFIG_DATAFLASH_SIZE>
   <FAL_CONFIG_BLOCK_SIZE>0x800</FAL_CONFIG_BLOCK_SIZE>
   <EEL_CONFIG_BLOCK_START>2</EEL_CONFIG_BLOCK_START>
   <EEL_CONFIG_BLOCK_CNT>6</EEL_CONFIG_BLOCK_CNT>
   <EEL_CONFIG_DF_BASE_ADDRESS>0x2000000</EEL_CONFIG_DF_BASE_ADDRESS>
   <EEL_CONFIG_BLOCK_CNT_REFRESH_THRESHOLD>0x2</EEL_CONFIG_BLOCK_CNT_REFRESH_THRESHOLD>
   <EEL_CONFIG_ERASE_SUSPEND_THRESHOLD>10 </EEL_CONFIG_ERASE_SUSPEND_THRESHOLD>
 </AdministrativeSection >
 <Record>
    <ID>0x80</ID>
    <IDName>Name1</IDName>
    <Length>2</Length>
    <Data>0x20 0x30</Data>
 </Record>
</DataFlash>
```

Figure 1 Example of XML based raw data description file

Chapter 4 Output file / formats

The Data FLASH Editor can support Data FLASH memory images in the Intel-hex and Motorola-S record file format. Additionally for the 78K0R(MF2) and V850E2(UX6LF) FLASH Technology, it can output the modified or newly created XML based raw data description file. Especially, if a data set is identified as invalidated it will be saved only to the XML file and it will be omitted in the output HEX or SREC file.

Chapter 5 EEPROM emulation data of V850E/ES (MF2/UX4)

In opposite to a classical EEPROM, where the data is stored on a fixed address and so can always be found on the same location, EEPROM emulation need to store data on changing locations. From the different options to find the data during *Read* operations, the Renesas EEPROM emulation uses the way to bundle a certain set of data with one ID, that is then searched in memory on data *Read*. Writing new data sets is easily done, by appending the data to the data pool.

Furthermore, the write granularity differs between real EEPROM and the Data FLASH used for EEPROM emulation. While the EEPROM can usually be written in 8-bit or 16-bit units, the Data FLASH must be written in 32-bit units (+1Bit for the ID-Tag, see below).

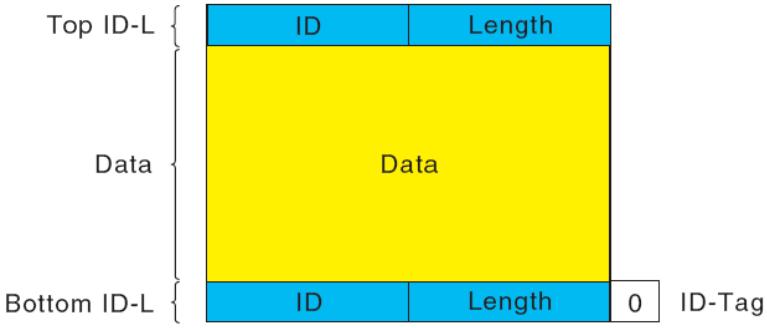


Figure 2 EEPROM Data set representation

The approach of using IDs to identify a set of data is in-line with AUTOSAR, as well as the ID size of 16 bits and length information of 16 bits.

Differing from Code FLASH with 32-bit width, the currently implemented Renesas Data FLASH is 33-bit wide. The 33-rd bit is called the ID-Tag, indicating, that the word is information and not part of the data set itself. The current implementation in the EEPROM emulation layer sets this on the bottom ID-L, while keeping it un-set on the data bits and on the top ID-L. Dedicated hardware on the Data FLASH macro can search for a certain ID by using this ID-Tag.

5.1 Overview

A EEPROM emulation Data FLASH section consists of 3 parts:

- Section Header
- Data Zone
- ID Zone

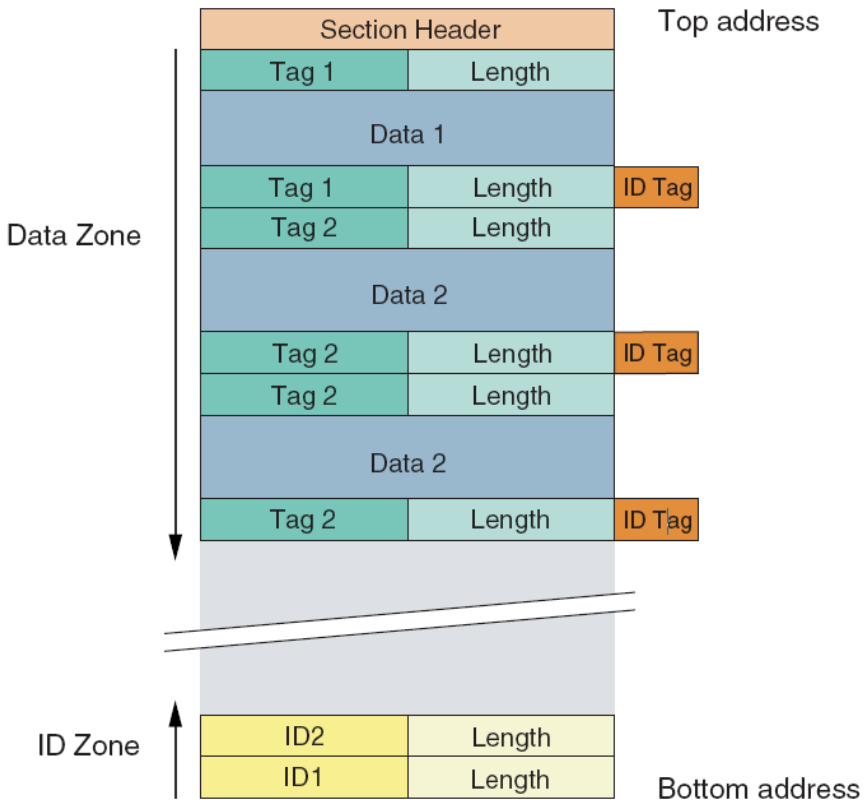


Figure 3 EEPROM Emulation Section Overview

5.2 Section Header

Section header size is 16 Bytes (4 Words), where currently 3 words are used to define the current section status.

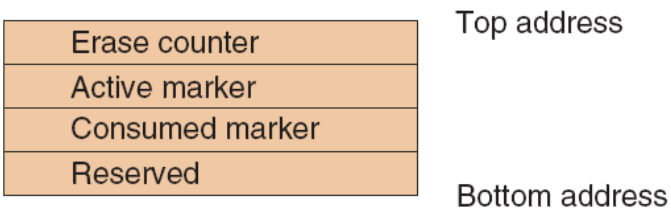


Figure 4 Section Header

The erase counter word consists of the 16-bit counter and a 16-bit inverse value for protection. It describes the number of erase cycles on the data section.

The active marker and / or the consumed marker have the value 0x55555555. The value of a marker which is cleared is 0xFFFFFFFF (Blank Word). The following section states are possible:

The following section states are possible:

- Prepared:

  The section has been erased. In the section header the erase counter is written. Active marker and consumed marker are cleared. This section is ready for activation by a *Refresh* operation.

- Active:

  The section contains the latest data sets. The EEPROM emulation can read and write data to this section. In the section header, additionally to the erase counter, also the active marker is set. Consumed marker is cleared.

- Consumed:

  The section is full, no more space was available to *Write* new data sets. After activating the next section and copying the latest data sets there, the *Refresh* operation marked the full section consumed. In the section header additionally, to the erase counter and active marker, also the consumed marker is set. Next step is to *Prepare* the section.

- Invalid:

  This is no valid state. During operation it can only occur in case of interruption of a FLASH operation like erase to this section. The section header may contain any other data except as described in the three states before. Also a completely erased FLASH (default factory delivery state) is invalid from EEPROM emulation point of view.

The normal section state transitions are:

Prepared --> Active --> Consumed --> Prepared -->...

5.3 ID zone

The ID-zone is a list of IDs of the data sets written to the section so far. The ID-list is dynamically generated during writing new data sets into the section. An ID-List is required in order to simplify and to speed up *Refresh* operations (copying data sets from a full data section to a prepared one). The solution to keep the ID-list dynamically in the EEPROM Emulation FLASH space instead of keeping it statically in the Code FLASH area has been chosen, in order to allow new/added applications to store data sets with new IDs. The ID-zone grows by time in case of writing new data sets with new IDs.

5.4 Data zone

The data zone contains the data sets. New data sets are simply appended after the last written data set. So the data zone grows down in the address space. The section is full and a *Refresh* is required, when there is no more enough erased space between ID zone and data zone for the data set to be written.

Chapter 6 EEPROM emulation data of V850E2(UX6LF)

Beside the same kind of user data management based on data sets (DS) indentified with certain IDs, a new concept which called “Ring buffer style FLASH block management” is implemented in UX6FL EEL. The FLASH blocks are used as a kind of ring buffer, each block passing a complete of life cycle in each ring buffer rotation.

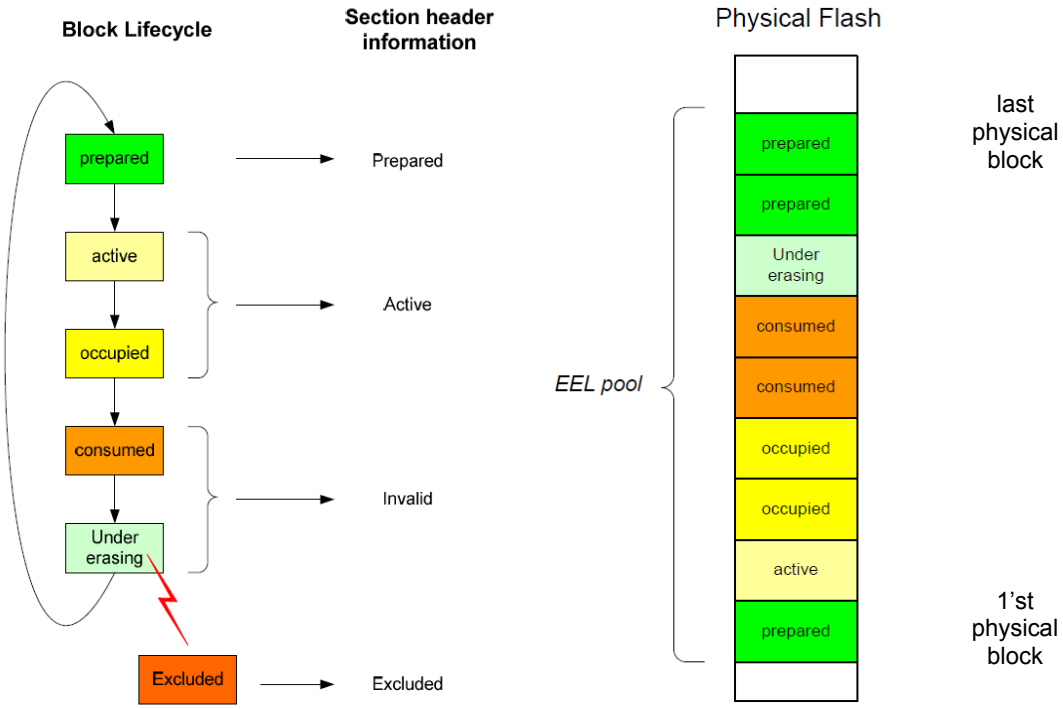


Figure 5 Block Lifecycle & Physical FLASH

6.1 Block structure overview

An EEPROM emulation Data FLASH block contains 3 areas:

- Section Header
- Data Zone
- Reference Zone

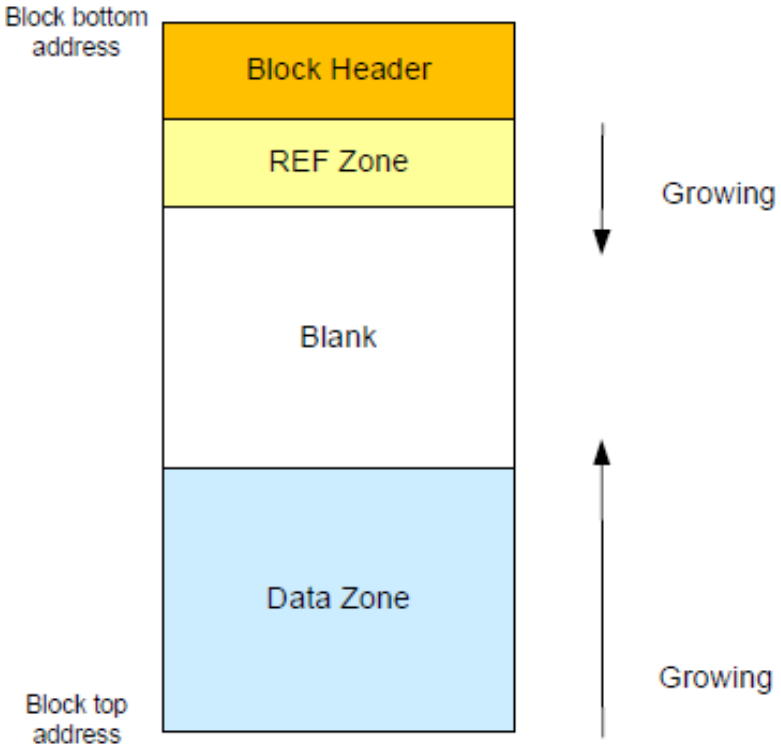


Figure 6 Basic Block structure

6.2 Section Header

The block header size is 32 Bytes (8 Words). It contains the block status words.

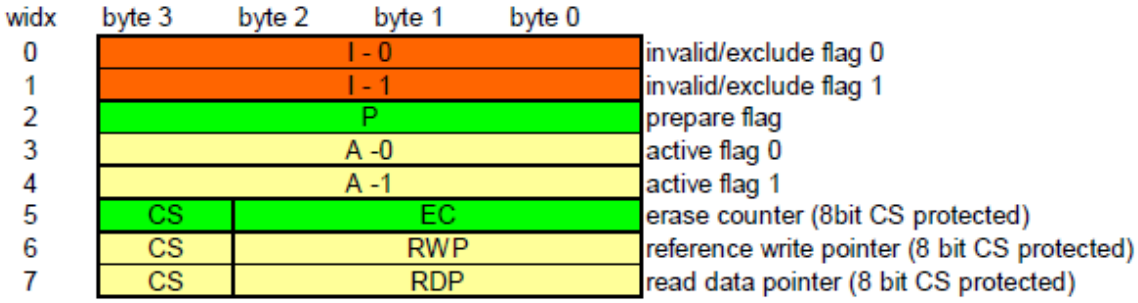


Figure 7 Section Header

The references write pointer (RWP) points to the previous block separator between REF zone and Data zone. By that, the EEL knows for each occupied block the last REF zone entry. The RWP stability is ensured by the A - 1 word, written afterwards in the activation process. If the A - 1 word is valid, the RWP is electrically stable. Additionally, the RWP is checksum protected in order to be robust against accidental overwriting due to application failures

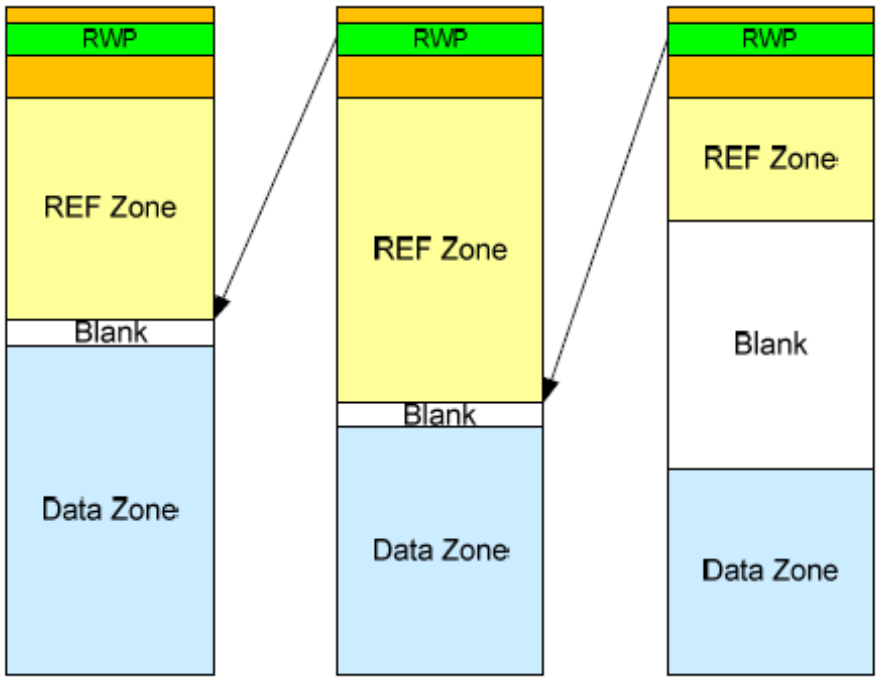


Figure 8 Reference Write Pointer

The EEPROM emulation block header information decides the block status.The following block states are possible:

- Prepared:

```
I-0   = 0xFFFFFFFF
I-1   = 0xFFFFFFFF
P     = 0x55555555
A-0   = 0xFFFFFFFF
A-1   = 0xFFFFFFFF
```

- Active:

```
I-0   = 0xFFFFFFFF
I-1   = 0xFFFFFFFF
P     = 0x55555555
A-0   = 0x55555555
A-1   = 0x55555555
```

- Excluded:

```
I-0   = 0x55555555
I-1   = 0x55555555
```

- Invalid.

```
I-0   = 0
I-1   = 0
```

6.3 Reference zone

The Reference zone contains the Data Sets (DS) management information, basically which are the ID and the pointer to the data. Further information is required in the REF-zone to ensure data consistency in case of write interruption and in order to improve the robustness against user application fails resulting in FLASH words overwrite.

SOR: Start of reference entry (1-bit Tag).

It is written 1st in order to block one REF zone list entry

DRP: Data Reference Pointer

Is written after SOR and contains:

- 16-bit lower half word: ID

- 16-bit upper half word: widx, a pointer to the data.

Especially, if widx is set to zero identifies that the DS is invalidated.

EOP: End of DRP (1-bit Tag)

It is written immediately after the DRP. When written, the read margin of the DRP word is ensured by the write sequence

DCS: Data Check Sum

This is a simple 32bit checksum, calculated over the user data and DRP. It ensures higher robustness (detection) on accidental overwriting of data or DRP. The DRP widx is excluded from the checksum. Additionally, by the write sequence it is ensured, that the read margin of the data is given, when the DCS is available

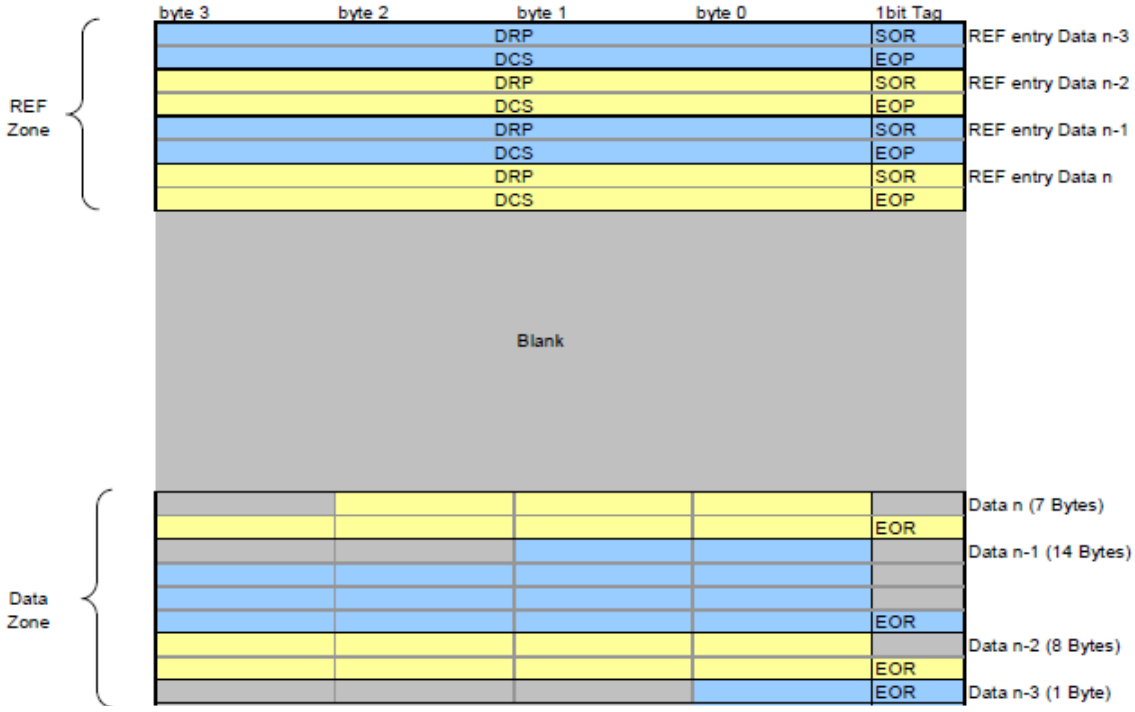


Figure 9 DS Data and management information in FLASH

6.4 Data zone

The data zone contains the pure user data sets. New data sets are simply appended after the last written data set. So the data zone grows up in the address space. The block is full when there is no more enough space between REF zone and data zone for the data set to be written.

Chapter 7 EEPROM emulation data of 78K0R(MF2)

This chapter describes briefly the internal architecture of the EEL – T03 EEPROM Emulation Library. For more details please refer to the corresponding application note of the EEL – T03 EEPROM Emulation Library.

7.1 EEL pool structure

The EEL pool is the virtual storage medium used by the EEL driver for storing data and block management information during its operation. From logical point of view the EEL-pool is organized as a single-linked ring of blocks. “Single-linked ring” means here:

a) the next block to block N is block (N+1)

b) the next block to the last one is the first one.

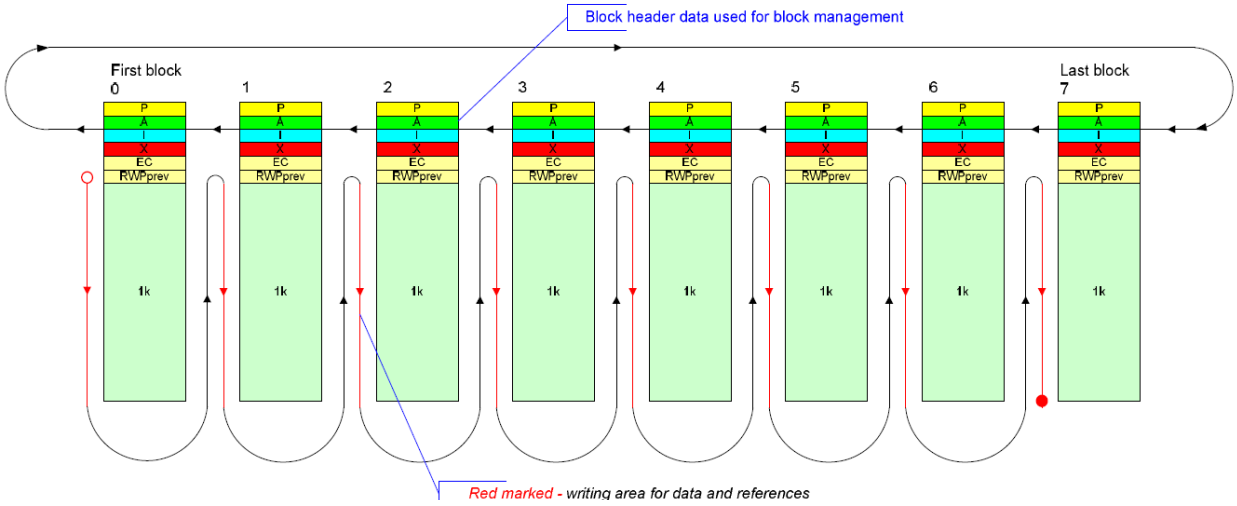


Figure 10 Structure of an empty EEL pool (no data inside)

Each block of the EEL-pool contains a block-header for storing block management information. Because the block indexing within the EEL-pool is based on the homogenous and fixed virtual block numbers 0x0000.... (EEL_POOL_SIZE - 1) it is not necessary to store the neighbors inside the block header.

All flash-blocks of the EEL pool are grouped in three consecutive “regions” indicated by the “block status” in the block header.

“active region” - consists of blocks containing active data
“invalid region” - consists of blocks without active data
“prepared region” - consists only of blocks ready to receive new data

When contemplate EEL-pool blocks clockwise the regions are always in the same fixed chronological order:

“prepared region” is before “active region”
“active region” is before “invalid region”
“invalid region” is before “prepared region”

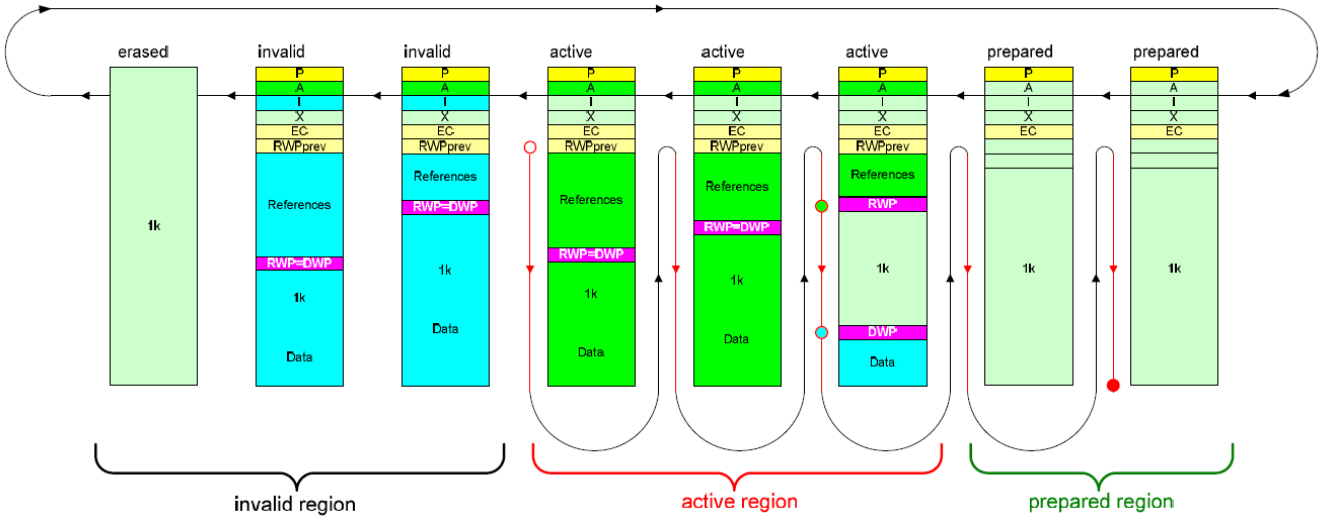


Figure 11 EEL pool regions during normal operation

Block organization scheme based illustrated above offers following advantages:

a) two symmetrical sections (where always 50% of data flash does not contain valid data) are not needed anymore

b) the “active region” can grow and be adapted to the momentary need

c) the reference area is separated from the data inside the same EEL block

d) copy-processes are mostly much faster because reduced to the only last active block has to be released from valid instances.

e) exclude functionality does not reduce performance of the driver

7.2 EEL block structure

Each EEL block belonging to the EEL-pool is basically divided into three areas: the block header, reference area and the data area. The block-header contains information about the actual status of the block which is needed for the block-management within the pool. The reference area contains reference entities off all instances written into this block during its live-cycle. It is necessary for actual data localization after power-on. The data area contains the pure data belonging to the corresponding references in reference area.

widx	byte 0	byte 1	byte 2	byte 3	bidx
0	P				0
1	A				4
2	I				8
3	X				12
4	CS(EC)	EC			16
5	RWPprev		00H	CS8	20
6	CS(XEC)	XEC			24
7	FIP flag				28
8	widx		CS8	ID_1	32
9	CS32(data)				36
10	widx		CS8	ID_2	40
11	CS32(data)				44
12	widx		CS8	ID_3	48
13	CS32(data)				52
14	widx		CS8	ID_4	56
15	CS32(data)				60
16	widx		CS8	ID_5	64
17	CS32(data)				68
18	widx		CS8	ID_6	72
19	CS32(data)				76
20	widx		CS8	ID_7	80
21	CS32(data)				84
22	widx		CS8	ID_8	88
23	CS32(data)				92
24	widx		CS8	ID_9	96
25	CS32(data)				100
26	widx		CS8	ID_10	104
27	CS32(data)				108
28	0xff	0xff	0xff	0xff	112
29	0xff	0xff	0xff	0xff	116
30	0xff	0xff	0xff	0xff	120
31	0xff	0xff	0xff	0xff	124
...					...
...					...
...					...
232	0xff	0xff	0xff	0xff	943
233	0xff	0xff	0xff	0xff	947
234	0xff	0xff	0xff	0xff	951
235	data	data	0xff	0xff	955
236	data	data	data	data	959
237	data	data	data	data	963
238	data	0xff	0xff	0xff	967
239	data	data	data	data	971
240	data	data	data	data	975
241	data	data	data	data	979
242	data	data	data	data	983
243	data	data	data	0xff	987
244	data	data	data	data	991
245	data	data	0xff	0xff	995
246	data	data	data	data	999
247	data	0xff	0xff	0xff	1003
248	data	data	data	data	1007
249	data	data	data	data	1011
250	data	data	data	0xff	1015
251	data	data	0xff	0xff	1019
252	data	0xff	0xff	0xff	1023
widx					bidx

Figure 12 EEL block, general structure

7.2.1 EEL block header

The block header is a small area on the top of each flash block belonging to the EEL pool. It contains all information necessary for block management during EEL operation. The structure of the block header is the same in all blocks of the EEL-pool.

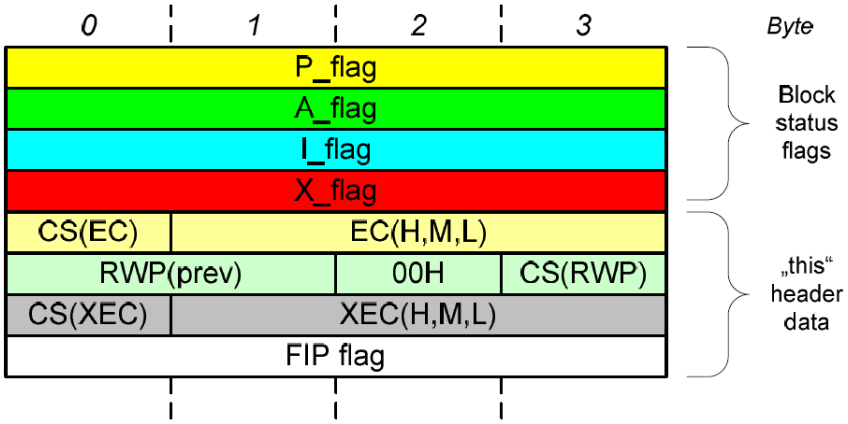


Figure 13 EEL block header structure

EEL block status header:

Each flag within the block header consists of one flash word (4 bytes).

There are two types of block status flags:

- “constructive status flag” used in processes like “activation” and “preparation”
- “destructive status flags” used in processes like “invalidation” and “exclusion”

When reading the exact pattern 0x55555555 a “constructive” flag is TRUE
When reading a pattern other than 0xFFFFFFFF a “destructive” flag is TRUE

When setting “constructive” flag: 0x55555555 is written into the flag-word.
When setting “destructive” flag: 0x00000000 is written into the flag-word.

P_flag: = 0x55555555 marks a “prepared” block that waits for data.
A_flag: = 0x55555555 marks an “active” block that may contain data
I_flag: ≠ 0xFFFFFFFF marks an “invalid” block (without valid data)
X_flag: ≠ 0xFFFFFFFF marks a block “excluded” from block management.

EEL block erase counter:

The block header word four contains the block erase counter. Its consistency is protected by an 8 bit checksum which is used by the EEL internally only.

EEL previous reference write pointer:

Its points the last RWP position of the previous block within the EEL pool. It is used by the EEL internally only.

EEL exclusion erase counter:

Stores the EC value at exclusion time.
It is used by the EEL internally only.

EEL Format In Progress (FIP) indicator:

FIP<>0xFFFFFFFF indicates an FORMAT command discontinued by RESET.
It marks the completely EEL pool as inconsistent and enforces the user to restart the FORMAT command.

7.2.2 Reference area

The “reference area” is located in each EEL block directly behind the block header. It consists of so called reference entries that are used for instance identification, localization and for safeguarding during the read/write process. When writing new data into the EEL a corresponding reference entry is stacked in the reference area.

The reference area is growing upstairs from lower widx to higher.

7.2.3 Data area

The “data area” consists of data-records and is located on the bottom of each EEL pool block. Each data record within the data-area consists of pure data information without any data- frame. The data-frame information exists completely in the corresponding reference-entry in the reference-area.

When writing new data into the EEL the data area is growing downstairs from higher widx to lower.

Chapter 8 Software Installation

The Data FLASH Editor requires a Windows XP, Windows Vista or Windows 7 operating system installed on your personal computer. In additional, the .NET framework 2.0 must be installed on your personal computer to execute the Data FLASH Editor. To start the installation, please run the “setup.exe” installation program. During initialization phase of the setup program a progress bar is shown.

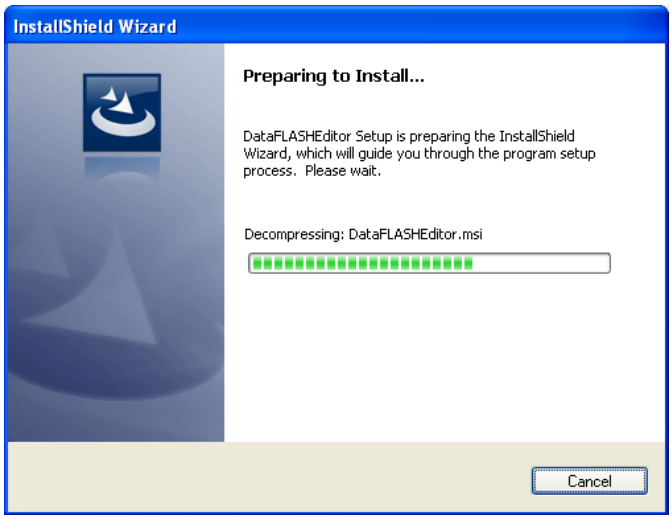


Figure 14 Install preparation

After the setup program has finished initialization a welcome screen appears. Click the Next button to continue the installation.

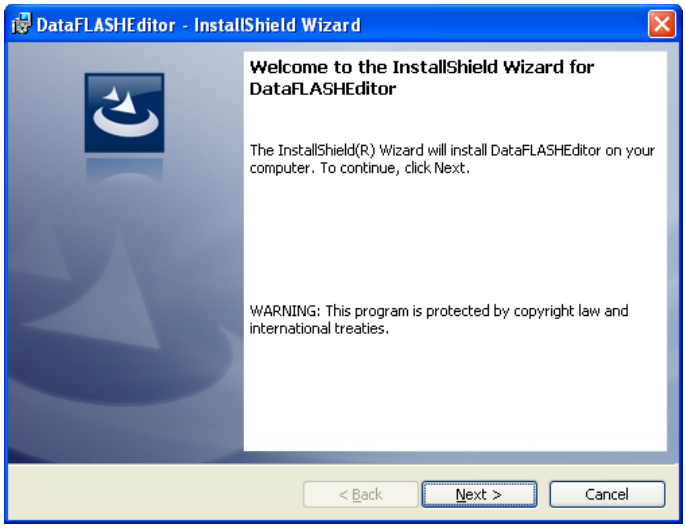


Figure 15 Welcome screen

In the next step of installation, the License Agreement window appears. Please read the Software Program License Agreement carefully. To continue the installation accept the License agreement and click the Next button. If you not accepted the license agreement the installation procedure exits.

Figure 16 License Agreement window

The Destination Folder window appears. You might exit the installation by clicking the ***Cancel*** button. You can click the ***Change*** button to be able to change the installation destination path. Press the ***Next*** button to proceed with the installation.

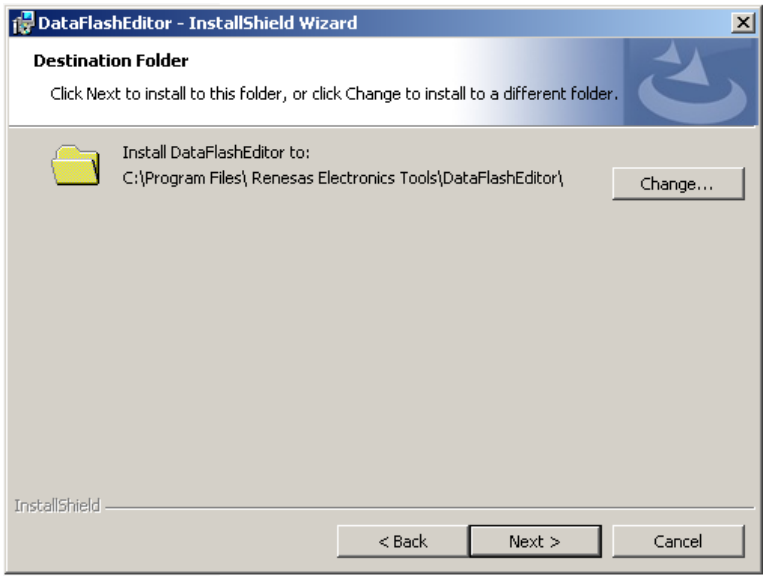


Figure 17 Choose Destination Folder window

After everything has been setup for installation, press the ***Install*** button to start the copy process and to finalize the installation.

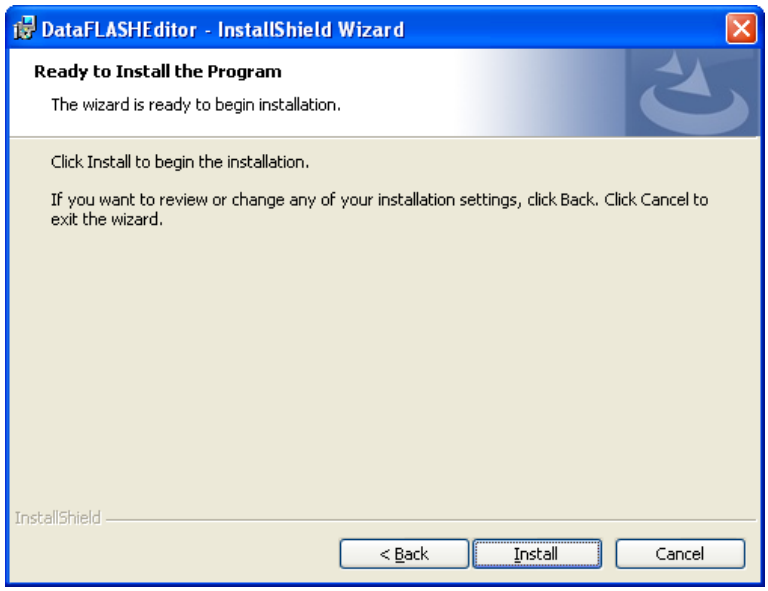


Figure 18 Ready to install window

On the installation progress window you can follow the installation. You can always abort the installation by pressing the ***Cancel*** button.

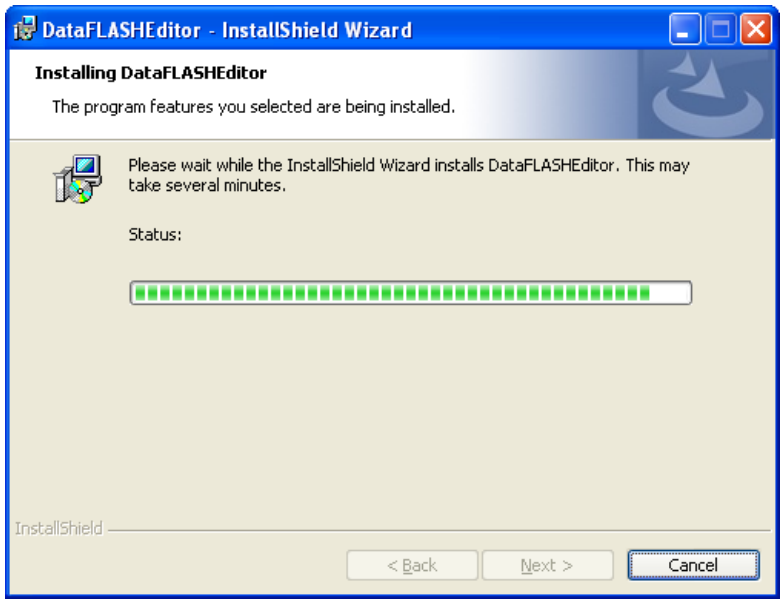


Figure 19 Installation progress window

When the installation is complete, the following message window appears.

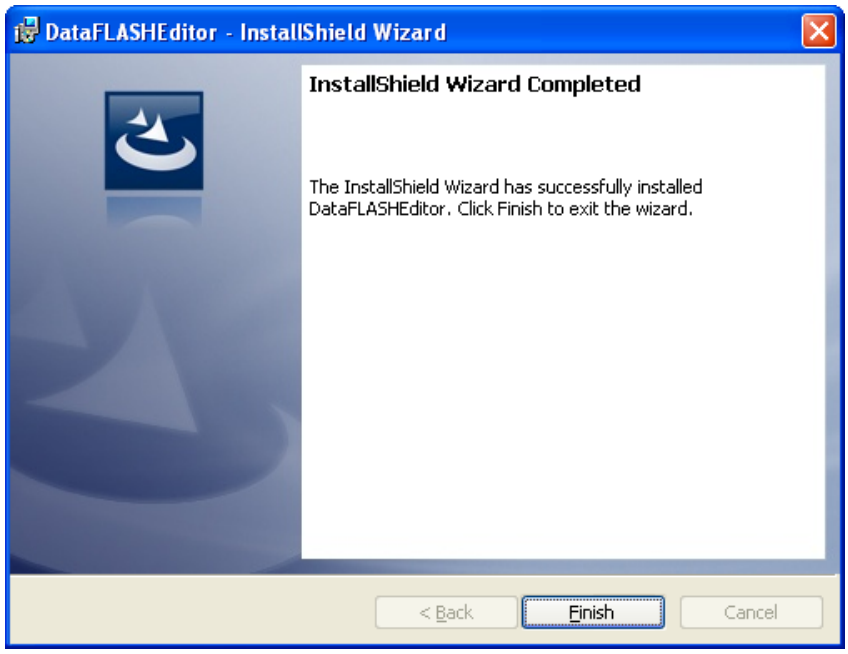


Figure 20 Installation completion window

The program has been installed successfully and an entry has been made into the program menu to start the Data FLASH Editor software.

The installation process is complete now and the setup utility has installed a new program folder which holds the Data FLASH Editor GUI Software and a Data FLASH Editor uninstall icon which allows you to remove the Data FLASH Editor software from your personal computer.

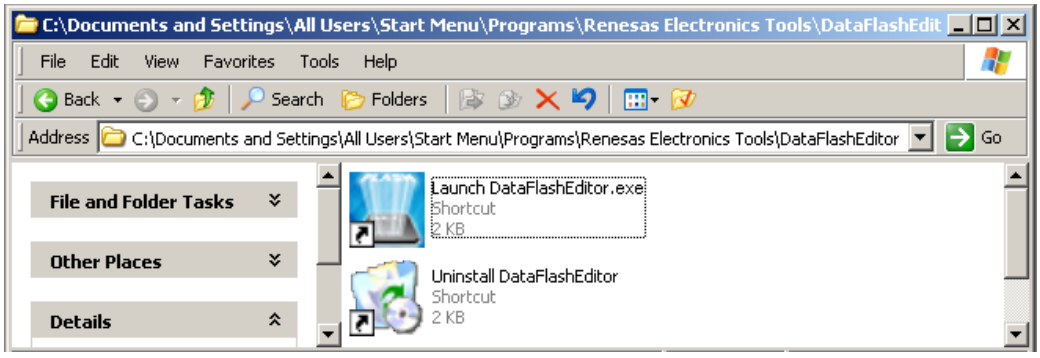


Figure 21 Program folder after installation

Chapter 9 Data FLASH Editor GUI

The graphical user interface of the Data FLASH Editor allows an easy and an intuitive visualization and modification of the EEPROM emulation data of a loaded Data FLASH memory image. When starting the Data FLASH Editor GUI a screen similar to the following one will appear. Some commands and functions may be unavailable when the GUI is started for the first time, depending on Data FLASH area definition or the loaded Data FLASH memory image.

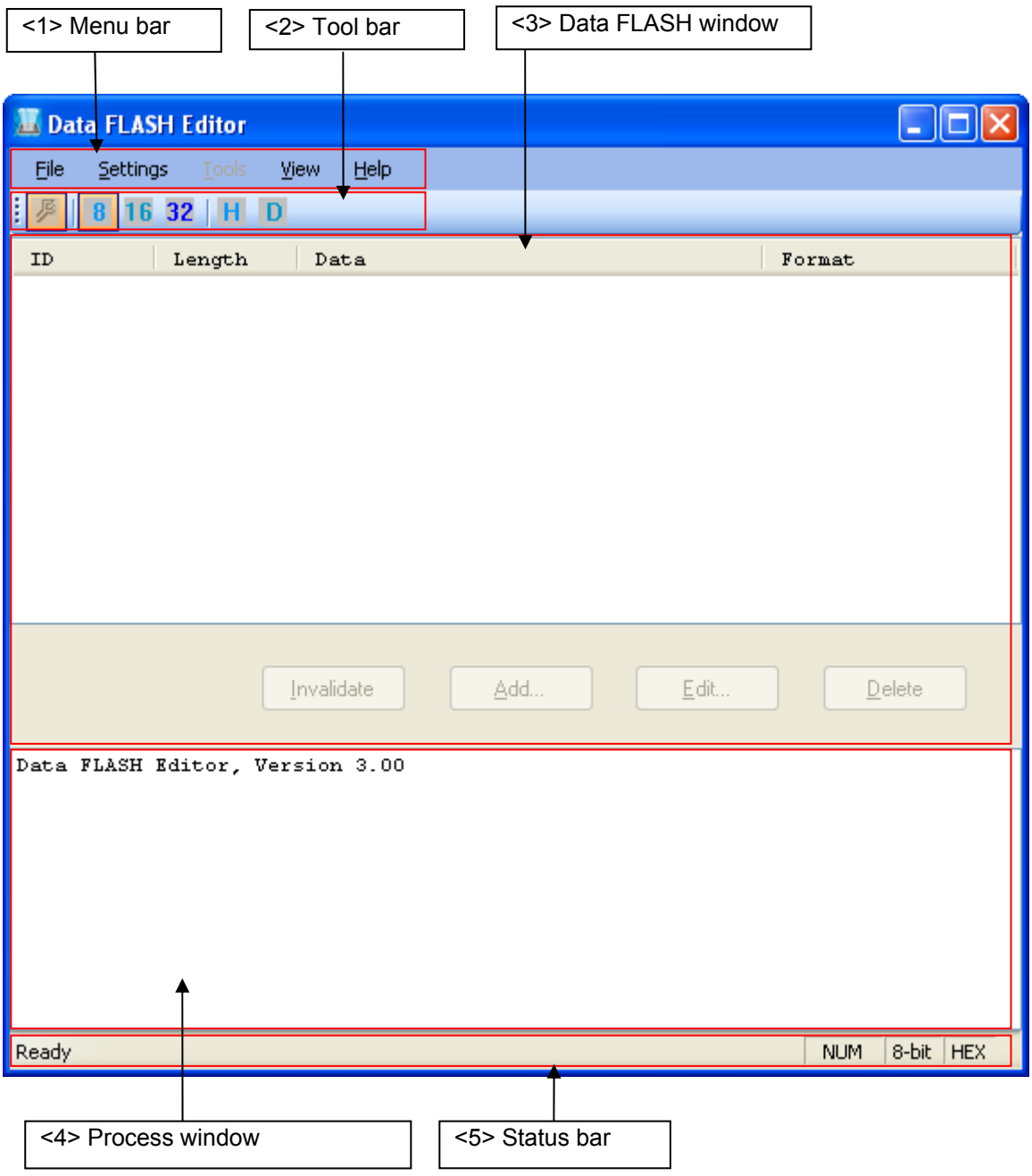


Figure 22 Data FLASH Editor main window

The main window consists of the following areas.

	Name	Displayed items
<1>	Menu bar	Menu items executable by the Data FLASH Editor
<2>	Toolbar	Frequently used commands
<3>	Data FLASH window	Shows the EEPROM emulation data of an loaded Data FLASH memory image, allows editing of EEPROM emulation data
<4>	Process window	Shows tool command and process information
<5>	Status bar	Show current status and configuration of the Data FLASH Editor

Table 4 Main window composition

9.1 Toolbar

The toolbar contains buttons to configure the Data FLASH Editor. The following buttons and corresponding functions are offered by the Data FLASH Editor.

	Setup Data FLASH
	Display all EEPROM emulation data entries in 8-bit data format (byte format)
	Display all EEPROM emulation data entries in 16-bit data format (half-word format)
	Display all EEPROM emulation data entries in 32-bit data format (word format)
	Display all EEPROM emulation data entries in Decimal format
	Display all EEPROM emulation data entries in Hexadecimal format

Table 5 Toolbar Buttons

9.2 The Menu

Analogue to the Toolbar the main menu of the Data FLASH Editor allows the configuration and execution of all Data FLASH Editor functions.

9.2.1 File menu

By using the ***File*** menu a Data FLASH memory image can be loaded, stored or if necessary closed. Additionally, the memory mapping of the Data FLASH area can be controlled via the ***File*** menu.

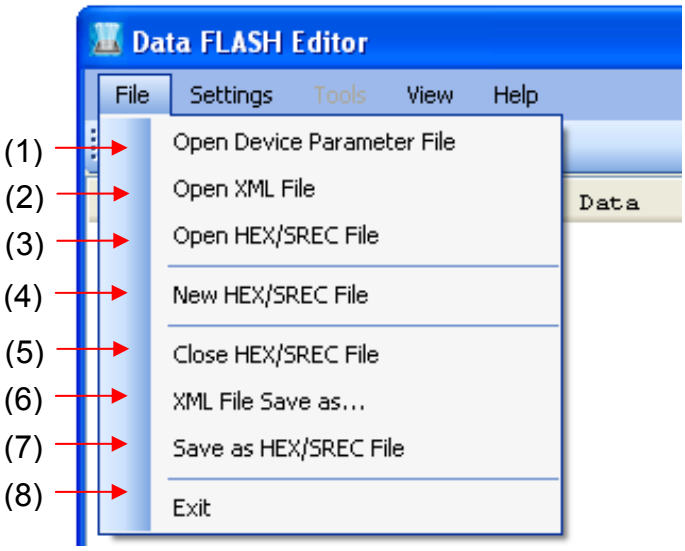


Figure 23 File menu

(1) Open Device Parameter File

The ***Open Device Parameter File*** menu allows selecting and loading an existing Device Parameter File, which file name always includes the extension *.pr5. The Data FLASH Editor gets the device FLASH type e.g. V850E/ES(MF2/UX4), V850E2(UX6LF) or 78K0R(MF2) information from the device parameter file. Additionally for a V850E2(UX6LF) or 78K0R(MF2) based device more Data FLASH information like the block number, FLASH range, start address will be read from the device parameter file.

By loading a new Device Parameter File the current configuration of the Data FLASH Editor is set to default.

(2) Open XML File

The ***Open XML File*** menu is only enabled for V850E2(UX6LF) and 78K0R(MF2) FLASH Technology based devices. It allows selecting and loading an existing XML based raw data description by the Data FLASH Editor. The XML file must fit to the selected Device Parameter File. Otherwise the Data FLASH Editor generates a corresponding error message.

(3) Open HEX/SREC File

The ***Open…*** menu allows selecting and loading an existing Data FLASH memory image by the Data FLASH Editor. After loading the Data FLASH memory image the corresponding Data FLASH entries are shown in the Data FLASH window.

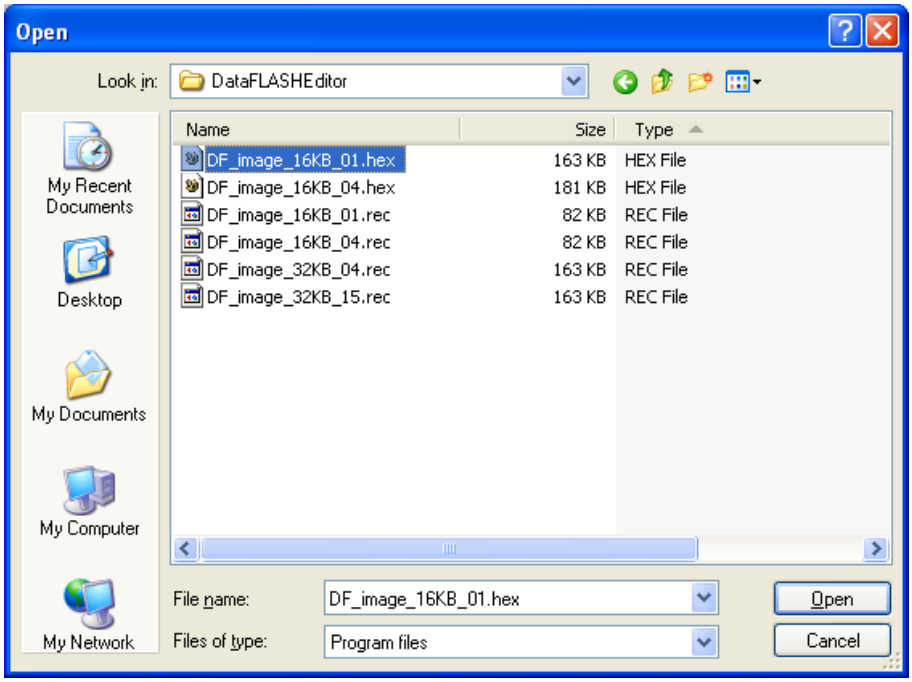


Figure 24 Open dialog

Note: Before a Data FLASH memory image can be loaded the Data FLASH area must be specified within the Settings menu.

In case of V850E/ES(MF2/UX4) the Data FLASH start address and size must be specified first.

In case of V850E2(UX6LF) or 78K0R(MF2) an existing XML based raw data description file including Data FLASH management information must be loaded first.

Note: When loading a Data FLASH memory image it must fit to the chosen Data FLASH start address and the Data FLASH size. Otherwise the file loading will be aborted and a corresponding error message is generated.

The most recently used directory a file has been loaded from will be offered in this file open menu. The Data FLASH memory image can be loaded in two different formats, the Intel-hex or Motorola-S record file format.

After a Data FLASH memory image is loaded, the corresponding EEPROM emulation data entries, including ID-tag, length and data record information were displayed in the Data FLASH window of the editor.

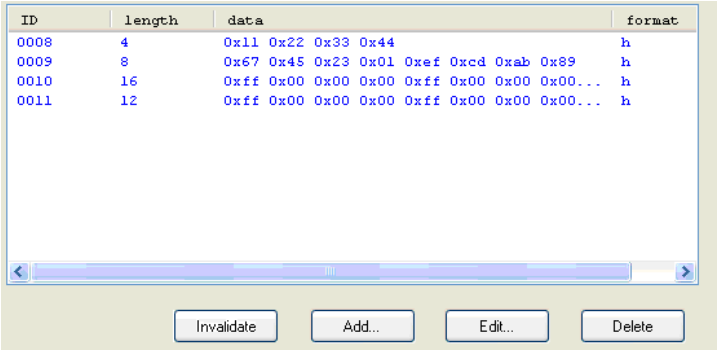


Figure 25 Data FLASH window

Additional to the Data FLASH memory the file reading process and the corresponding file and address map information are displayed in the process window.

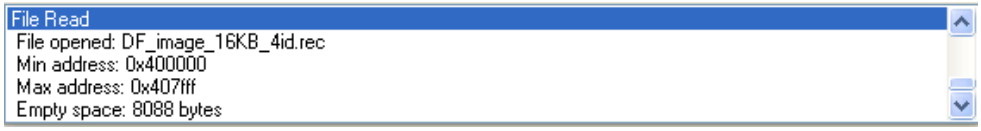


Figure 26 Process window

(4) New HEX/SREC File

The ***New HEX/SREC File*** menu is enabled only for V850E2(UX6LF) and 78K0R(MF2) FLASH Technology based devices. It allows preparing a new HEX/SREC file by the Data FLASH Editor.

(5) Close HEX/SREC File

Via the ***Close…*** menu the working on a Data FLASH memory image can be finished and the corresponding Data FLASH memory image will be closed. In case the Data FLASH memory image was changed a file save dialog offers the possibility to save the last changes.

(6) XML File Save as…

The ***Save as…*** menu is enabled only for V850E2(UX6LF) and 78K0R(MF2) FLASH Technology based devices. It allows saving of a modified or newly created XML file by the Data FLASH Editor. Especially, the invalidated data sets will be saved in the XML file and the memory space they occupied will be included in the data FLASH memory check function.

(7) Save as HEX/SREC File

The ***Save as HEX/SREC File*** menu allows saving of a modified Data FLASH memory image by the Data FLASH Editor. Especially for V850E2(UX6LF) and 78K0R(MF2) FLASH Technology based devices, the invalidated data sets will be ignored in the saved file.

(8) Exit

Via the ***Exit*** menu the Data FLASH Editor can be closed. In case the Data FLASH memory image was changed a file save dialog offers the possibility to save the last changes.

In addition, the current Data FLASH Editor configuration is stored in the “CurrentSettings.xml” configuration file. The configuration file is located in the installation folder of the Data FLASH Editor. By restarting the Data FLASH Editor the latest settings are restored.

Note: Write permissions to the installation folder and to the “CurrentSettings.xml”file are required to allow the storage of the current configuration settings by the Data FLASH Editor.

9.2.2 Settings menu

Within the settings menu the corresponding Data FLASH settings can be displayed and modified.

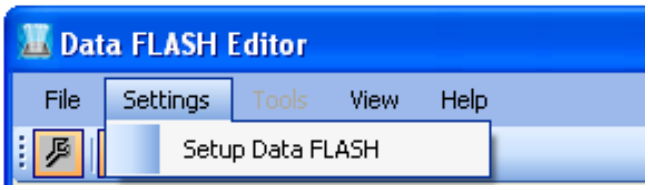


Figure 27 Settings menu

The *Setup Data FLASH* menu allows to setup the corresponding Data FLASH of the chosen device. According to the FLASH Technology of the specified device V850E/ES(MF2/UX4) or V850E2(UX6LF) / 78K0R(MF2) two different setup menus will be displayed.

Data FLASH setup for V850E/ES (MF2/UX4):

For V850E/ES (MF2/UX4) based devices the *Setup Data FLASH* menu allows to setup the start address and the size of the Data FLASH area. By selecting this menu item the following dialog window appears and offers the possibility to specify the Data FLASH mapping.

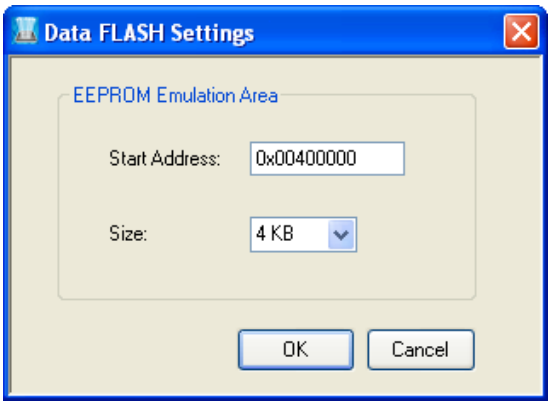


Figure 28 Setup Data FLASH start address

Within the Data FLASH Settings dialog the start address of the Data FLASH area must be specified. Only hexadecimal input values are permissible. The entered start address should be equal to the physical start address of the Data FLASH area of the corresponding V850E/ES(MF2/UX4) device. The size of the Data FLASH can be specified by the pull-down menu. Supported sizes are 4 KByte, 8 KByte, 16 KByte and 32 KByte.

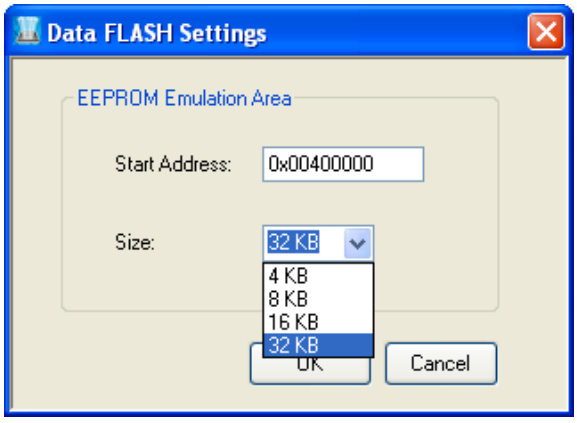


Figure 29 Setup Data FLASH size

***Tip*:** After a Data FLASH memory image was loaded or edited, the Data FLASH Start Address can be modified by the user. This offers the possibility to generate and store the Data FLASH memory image for different memory locations. In case of V850E/ES(MF2/UX4) the debugging address space or otherwise the FLASH programmer address space can be specified.

Data FLASH setup for V850E2(UX6LF) and 78K0R(MF2):

For V850E2(UX6LF) and 78K0R(MF2) based devices the *Data FLASH Settings* menu allows to setup and modify the Administrative Section. This Administrative Section represents the Data FLASH mapping and partitioning of a V850E2(UX6LF) or 78K0R(MF2) based device.

Additionally, by using the *Data FLASH Settings* menu the Administrative Section of an XML raw data description file including Data FLASH management information can be easily created. The tags of this section are predefined. The input values must match to the selected device parameter file.

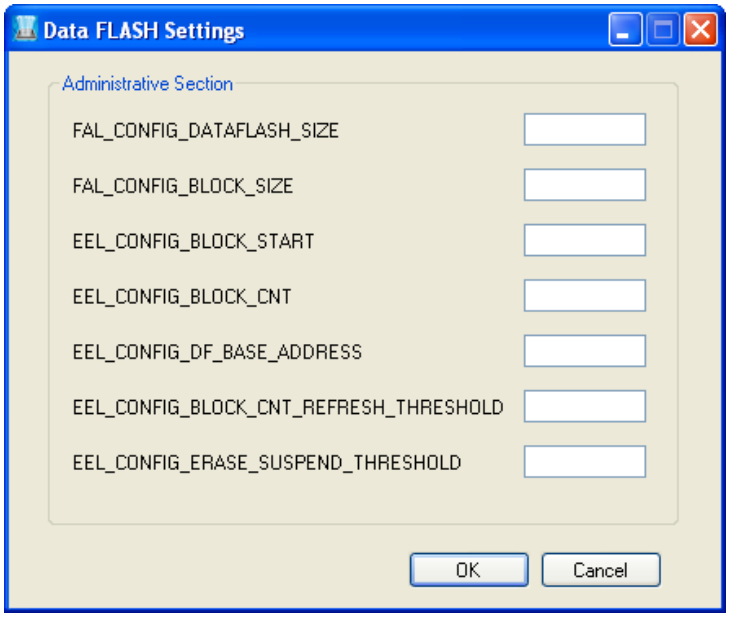


Figure 30 Administrative Settings

***Tip*:** After a Data FLASH memory image was loaded or edited, the Data FLASH Start Address can be changed by the user, by modifying the “EEL_CONFIG_DF_BASE_ADDRESS” entry. This offers the possibility to generate and store the Data FLASH memory image for different memory locations. In case of 78K0R(MF2) the debugging address space or otherwise the FLASH programmer address space can be specified.

9.2.3 Tools menu

The ***Tools*** menu is enabled only for V850E/ES(MF2/UX4) FLASH Technology based devices. It controls the refreshing of the active Data FLASH section. In case EEPROM emulation data entries have been modified, remove or added, the “Refresh” function updates the active Data FLASH section and makes the changes valid.

Figure 31 Tools menu

9.2.4 View menu

By using the ***View*** menu the configuration of the Data FLASH Editor can be changed. The following functions are offered by the View menu.

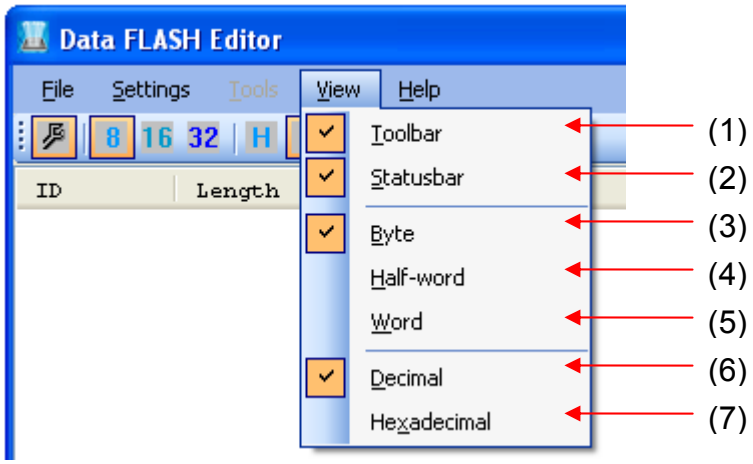


Figure 32 View menu

The functionality of each ***View*** menu items is described in the following table.

(1) Toolbar	Enables / Disables Toolbar
(2) Status bar	Enables / Disables Status bar
(3) Byte	Display all EEPROM emulation data entries in 8-bit data format (byte format)
(4) Half-word	Display all EEPROM emulation data entries in 16-bit data format (half-word format)
(5) Word	Display all EEPROM emulation data entries in 32-bit data format (word format)
(6) Decimal	Display all EEPROM emulation data entries in Decimal format
(7) Hexadecimal	Display all EEPROM emulation data entries in Hexadecimal format

Table 6 View menu

9.2.5 Help menu

Clicking the ***Help*** menu displays the following pull-down menu.

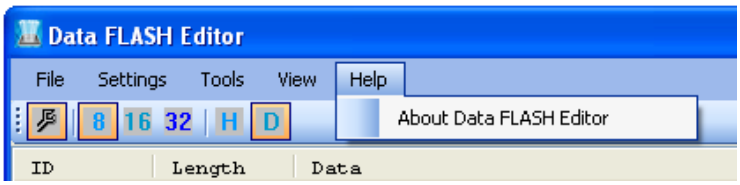


Figure 33 Help menu

The ***About Data FLASH Editor…*** command opens the “About Data FLASH Editor” window as shown below and indicates the program version.

Figure 34 About Data FLASH Editor window

9.3 Data FLASH window

The Data FLASH window visualizes the EEPROM emulation data records of a loaded Data FLASH memory image. The Data FLASH window supports the following functions:

- Display of EEPROM emulation data records, including:
  - ID tag
  - Length information
  - Invalidated state
  - Data
  - Format
- Modification of EEPROM emulation data:
  - Adding new EEPROM emulation data record
  - Editing of existing EEPROM emulation data record
  - Deleting existing EEPROM emulation data record
  - Invalidate/Validate existing EEPROM emulation data record

The Data FLASH window is shown in the figure below.

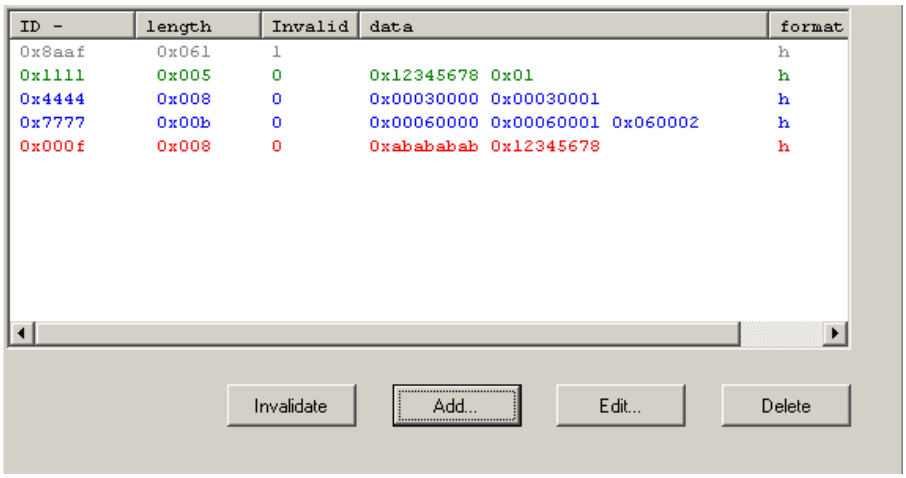


Figure 35 Data FLASH window

The meaning of the different colours of the displayed EEPROM data records can be found in the following table.

BLUE	Original / unchanged EEPROM data record
GREEN	Modified EEPROM data record
RED	Added / new EEPROM data record
GRAY	Invalidated data record

Table 7 EEPROM data record colouring

By using the “Refresh” function in the Tools menu, all changes that have been done will be written to the active Data FLASH section. After the active Data FLASH section has been updated successfully, all EEPROM data records are displayed in **blue** colour.

The functionality of each corresponding Data FLASH window button is described in the following table.

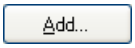
	Add new EEPROM emulation data entry
	Edit existing EEPROM emulation data entry
	Delete existing EEPROM emulation data entry
	Invalidate existing data record
	Recover or validate existing data record

Table 8 Data FLASH window buttons

9.3.1 Add new EEPROM emulation data entry

To add a new EEPROM data record, please press the “Add” button. The following screen does appear.

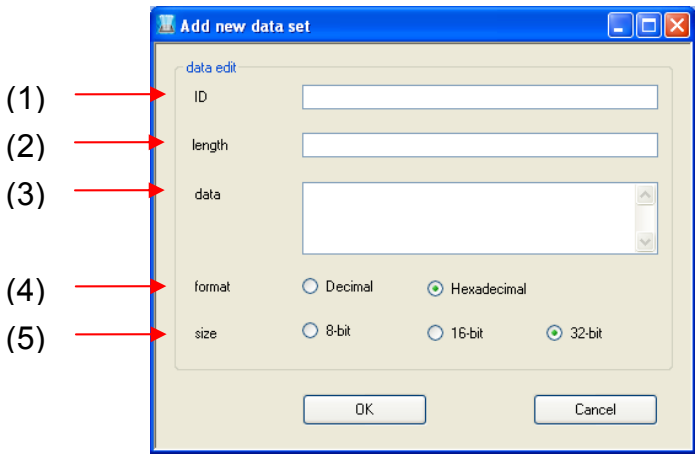


Figure 36 Add new data set menu

The functionality of each ***Add*** menu item is described in the following table.

(1) ID	Identifier of the data record (ID tag). Only decimal values are supported
(2) Length	Length definition of the data record. The minimum data length is equal to four bytes. The length can be specified in 4 Byte *n steps. Only decimal values are supported.
(3) Data	Specifies the data record, any hexadecimal or decimal value. The data record must fit to the specified Length and format.
(4) Format	Specifies the input and display format of the data record.
(5) Size	Specifies the display size of the data record, the following sizes are supported: → 8-bit data format (byte format) → 16-bit data format (half-word format) → 32-bit data format (word format)

Table 9 Add new data set items

After a new EEPROM emulation data record has been entered, please press the <OK> button.

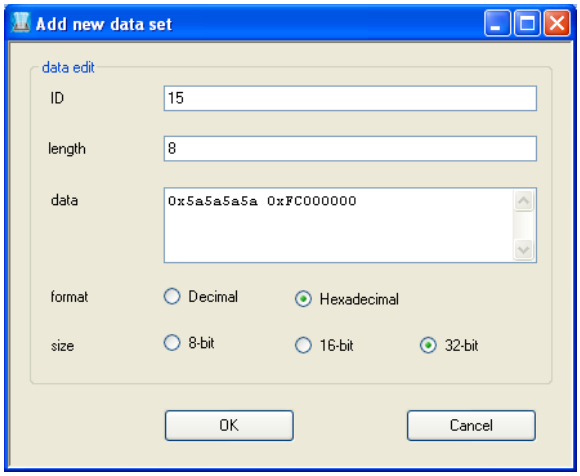


Figure 37 Example 1, add new data set

The new EEPROM data record is now visible, highlighted in **red** colour, in the Data FLASH window.

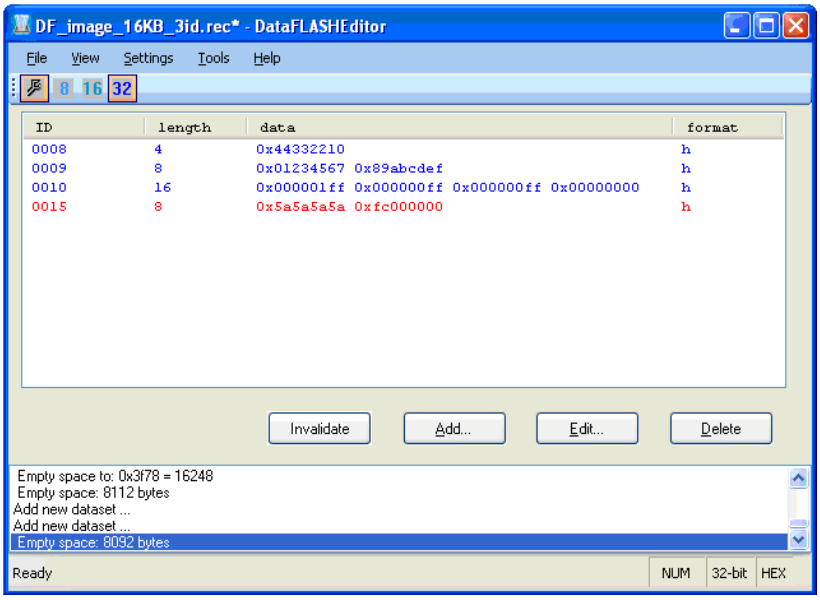


Figure 38 Example 1, Data FLASH window (Add function)

To make the changing valid and finally add the new EEPROM data record to the active Data FLASH section, please execute the “Refresh” function within the Tools menu.

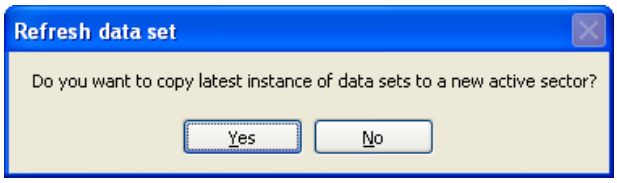


Figure 39 Example 1, refresh data set

After the active Data FLASH section has been updated successfully, the new EEPROM data record is displayed in **blue** colour.

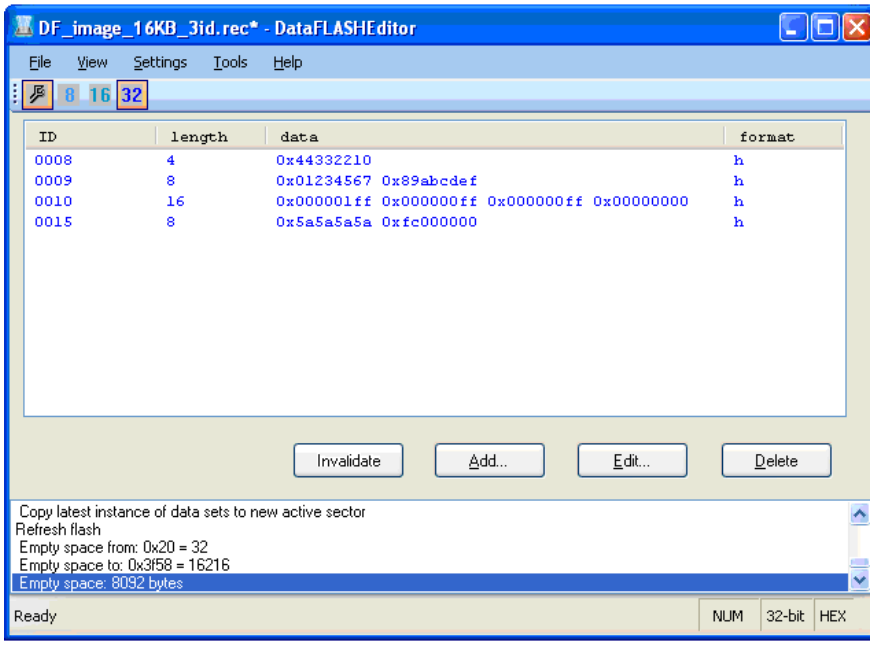


Figure 40 Example 1, added new EEPROM data record

9.3.2 Edit EEPROM emulation data entry

To edit an EEPROM data record, please mark the corresponding entry in the Data FLASH window and press the “Edit” button. Analogue to this, you can double-click on the corresponding EEPROM data record. The following screen does appear.

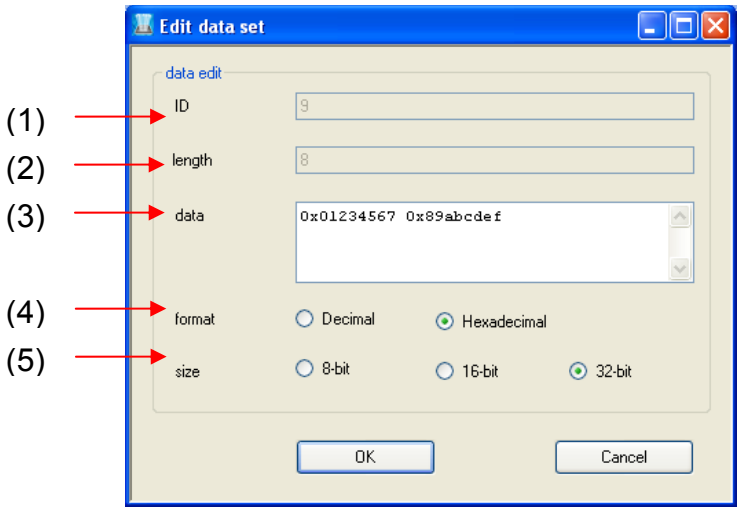


Figure 41 Edit data set menu

The ***Edit*** menu does only support the modification of the data record, the format and the size. The ID and Length of the corresponding data record can not be changed. The functionality of each ***Edit*** menu item is described in the following table.

(1) ID	Current identifier of the data record (ID tag). Can not be modified.
(2) Length	Current length of the data record. Can not be modified.
(3) Data	Specifies the data record, any hexadecimal or decimal value. The data record must fit to the specified Length and format.
(4) Format	Specifies the input and display format of the data record.
(5) Size	Specifies the display size of the data record, the following sizes are supported: → 8-bit data format (byte format) → 16-bit data format (half-word format) → 32-bit data format (word format)

Table 10 Edit data set items

After an EEPROM emulation data entry has been modified, please press the <OK> button.

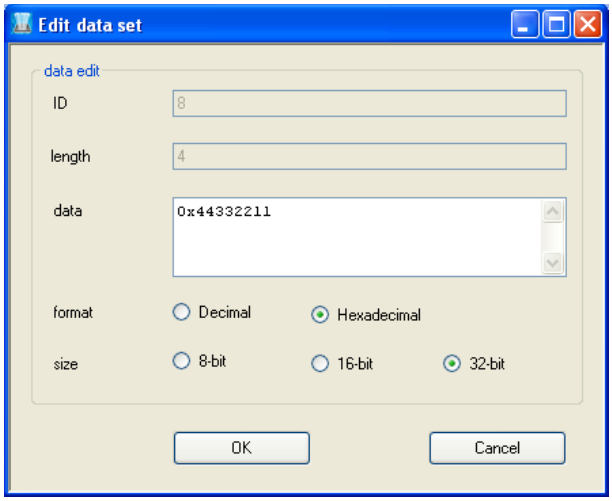


Figure 42 Example 2, edit data set

The modified EEPROM data record is now visible, highlighted in **green** colour, in the Data FLASH window.

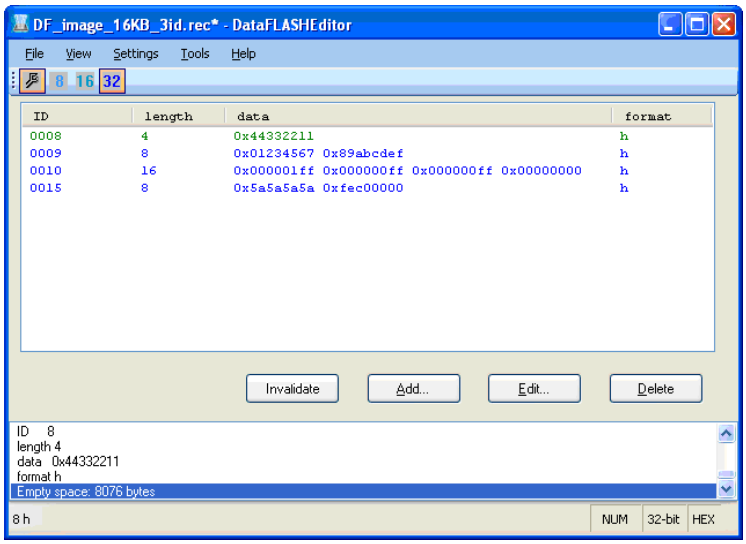


Figure 43 Example 2, Data FLASH window (Edit function)

To make the changing valid and finally add the new EEPROM data record to the active Data FLASH section, please execute the “Refresh” function in the Tools menu.

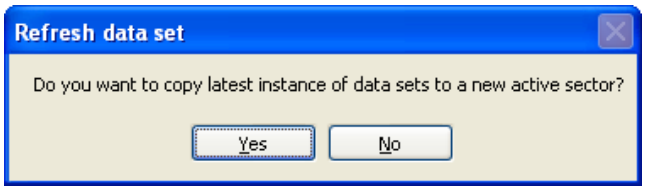


Figure 44 Example 2, refresh data set

After the active Data FLASH section has been updated successfully, the new EEPROM data record is displayed in **blue** colour.

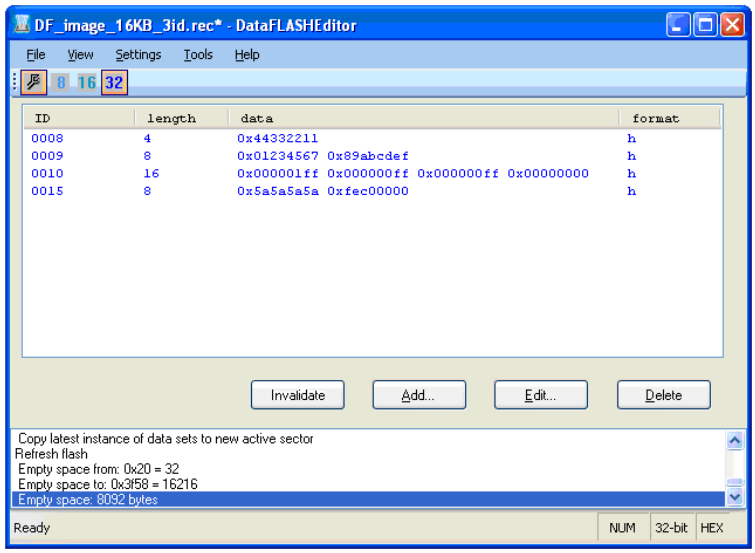


Figure 45 Example 2, modified EEPROM data record

9.3.3 Delete EEPROM emulation data entry

To delete an EEPROM data record, please mark the corresponding entry in the Data FLASH window and press the “Delete” button.

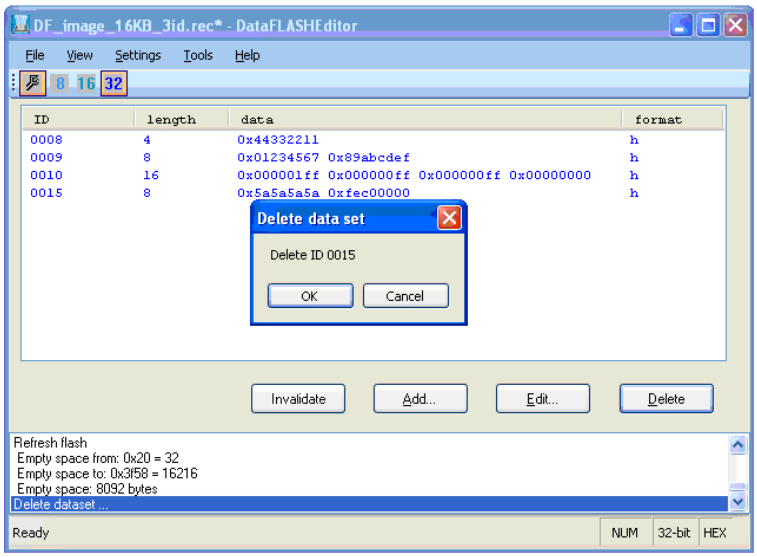


Figure 46 Delete data set

To delete the selected EEPROM data record, please press the <OK> button. To remove the selected EEPROM data record finally from the active Data FLASH selection, please execute the “Refresh” function in the Tools menu.

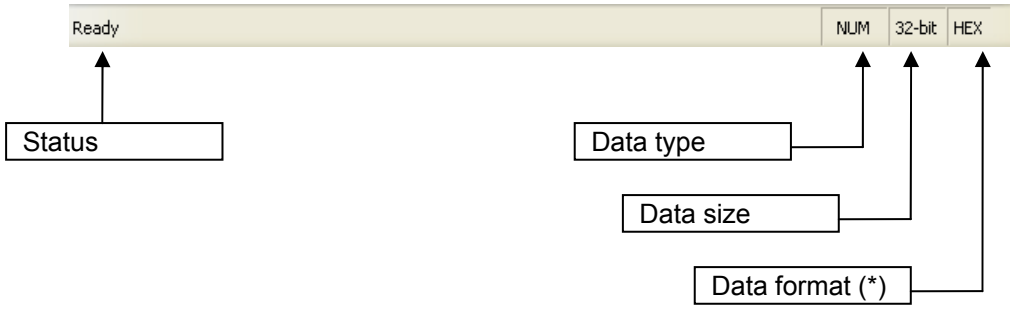
9.4 Process window

The process window shows the executed tool commands and process information. Additionally it stores the command history. You can use the scroll bar to navigate through the command history.

Figure 47 Process window

9.5 Status bar

The status bar shows the actual status and configuration of the Data FLASH Editor.



(*) = Only by loading a new Data FLASH memory image the selected data format will be taken under consideration. Changing the display format within the Settings menu after a Data FLASH memory was loaded has no influence and does not change the global format of the represented EEPROM emulation data entries. You can change the display format of an individual EEPROM emulation data entry in the ***Edit*** menu by double-clicking on it or using the “Edit” button within the Data FLASH window.

Figure 48 Status bar

Chapter 10 Error Messages

Error messages will be output in the error dialog boxes of the Data FLASH Editor.

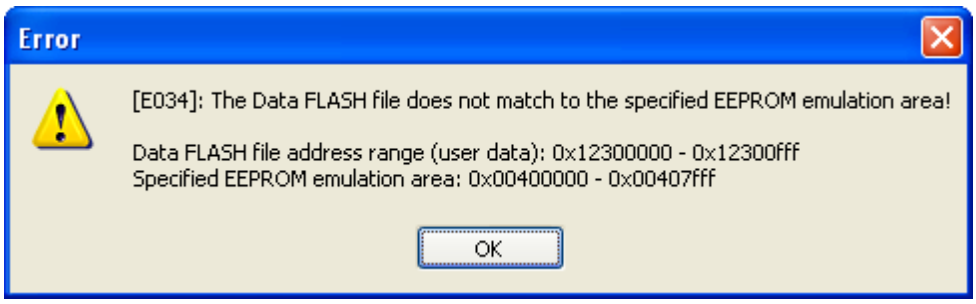


Figure 49 Error message, dialog box

The following table shows the error messages generated by the Data FLASH Editor.

Error Number	Message / Description
[E002]	Could not open FLASH file!
[E003]	Could not read FLASH file!
[E004]	Could not save FLASH file!
[E005]	Could not load program configuration file <CurrentSettings.xml>!
[E006]	Could not write program settings XML file!
[E007]	Could not save XML file!
[E008]	The FLASH file contains checksum error!
[E009]	The FLASH file contains unknown lines!
[E010]	The FLASH file contains wrong Intel HEX lines!
[E011]	The FLASH file contains wrong Motorola SREC lines!
[E012]	The FLASH file contains wrong information!
[E013]	The data record could not be found!
[E014]	Active FLASH sector error when searching sector 0!
[E015]	Active FLASH sector error when searching sector 1!
[E016]	The active FLASH sector could not be found!
[E017]	Error during memory copy!
[E018]	FLASH file does not include data for RecordID <id>!
[E019]	The RecordID <id> has been deleted. A refresh is needed before it can be used again!
[E020]	The RecordID <id> is already used!
[E021]	Illegal character, must be an integer value, use digits 0-9!
[E022]	Error during decimal conversion, element=<id>!
[E023]	Error during hexadecimal conversion, element=<id>!
[E024]	The new data set does not fit in current FLASH!
[E025]	Problem to update empty FLASH space size!
[E026]	The length must be 4-65532
[E027]	Error when ID should be inserted!
[E028]	Error when data should be inserted!

[E029]	No inactive sector found!
[E030]	The sector is consumed and needs to be erased!
[E031]	The sector is already active!
[E032]	The length must be a multiple of 4!
[E033]	Error the data is not valid, element=<id>!
[E034]	The Data FLASH file does not match to the specified EEPROM emulation area! Data FLASH file address range (user data): <id1> - <id2> Specified EEPROM emulation area: <id3> - <id4>
[E035]	The Data FLASH file does not fill out the specified EEPROM emulation area. The file must have the same size as the EEPROM emulation area! Data FLASH file address range (user data): <id1> - <id2> Specified EEPROM emulation area: <id3> - <id4>"
[E036]	The ID must be a value from 1 to 65534!
[E037]	The file data exceeds the configured EEPROM emulation area! Data FLASH file address range: <id1> - <id2> Specified EEPROM emulation area: <id3> - <id4>
[E038]	The Data FLASH file is to large. Please import the Data FLASH file according to the specified Data FLASH settings.
[E040]	Wrong data FLASH base address specified. Please insert data FLASH base address as hexadecimal value! (for instance, Base address: 0x001F8000)
[E041]	Wrong data FLASH base address specified. The value has been updated according to alignment rules!
[E042]	A data set must be selected!
[E043]	Illegal start address!
[E044]	Wrong reference write pointer in block <blk> header section! Block number=<blk>
[E045]	The settings within XML file do not match to the device parameter file!
[E046]	The RecordID <ID> includes invalid characters in XML file!
[E047]	The RecordID <ID> value exceeds range (1-0xFFFE) in XML file!
[E048]	Same RecordID <ID> found in XML file!
[E049]	Inputs contains invalid characters!
[E051]	The Record length <length> exceeds range!
[E052]	The Record length <length> include invalid character in XML file!
[E053]	<name> value is out of range in XML file!
[E054]	Can not find valid data record in the FLASH file!
[E055]	length must be a value from (1-<max>)!
[E057]	Block <blk> RWP is wrong!
[E059]	Empty flash size is not enough to validate this data set!
[E060]	Data size might exceed active pool
[E061]	Could not open Device Parameter File!
[E062]	Could not read Device Parameter File!

Table 11 Error Messages

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	March 31, 2011	-	First Edition Issued. Replaces document U19829EE2V0UM00.

Renesas Electronics Corporation

SALES OFFICES http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Data FLASH Editor

Renesas Electronics Corporation

R01UT0176ED0000