



MPFS2 Utility

MPLAB Harmony Integrated Software Framework v1.07.01

MPFS2 Utility

This section provides a description of the Microchip Proprietary File System (MPFS2) utility.

Introduction

This topic provides an overview of the MPFS2 Utility in MPLAB Harmony.


Description

The MPFS2 Utility packages web pages into a format for efficient storage in an embedded system. It is a graphical application for personal computers that can generate MPFS2 images for storage in external storage or internal Flash program memory.

When used to build MPFS2 images, the MPFS2 Utility also indexes the dynamic variables found. It uses this information to generate `http_print.h`, which ensures that the proper callback functions are invoked as necessary. It also stores this index information along with the file in the MPFS2 image, which alleviates the task of searching from the embedded device.

Finally, when developing an application that uses external storage, the MPFS2 Utility can upload images to the external storage device using the upload functionality built into the HTTP web server or FTP server. The `http_print.idx` file is also generated by the utility, which keeps information of all the dynamic variable details.

The MPFS2 utility generates the `MPFS2SettingDetails.xml` file, which contains all of the run-time configured parameters.

 **Note:** If there is any change in `http_print.idx`, the updated MPFS image will be generated. It is recommended to remove the `http_print.idx` file before generating a new MPFS image.

Building MPFS2 Images

This topic provides information for building MPFS2 images.

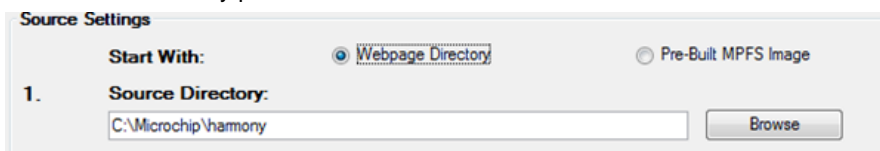
Description

The MPFS2 Utility has four steps, which are denoted on the left hand side of the dialog.

To build an MPFS image, select **Start With: Webpage Directory** in Step 1 and choose the directory in which the web pages are stored.

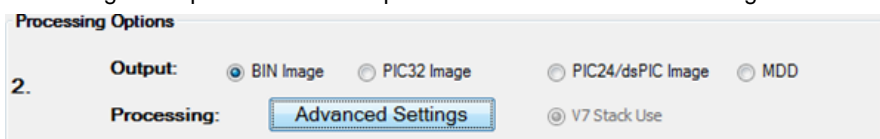
By default, the source directory path is `C:\Microchip\harmony`. Users need to provide the web page directory path from the `apps\tcpip\ demonstration` project.

The Source Directory path is stored in `MPFS2SettingDetails.xml` for future access.



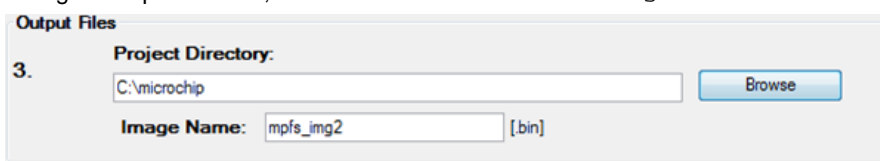
Step 2 selects the output format. If storing the web pages in external EEPROM or serial Flash, choose the **BIN Image** output format. If internal program memory will be used, select **PIC32 Image** for use with 32-bit devices, or **ASM30 Array** for 16-bit targets. To store the web pages on a device formatted with the FAT file system without compressing them into an MPFS image, select **MDD**. Refer to [Advanced MPFS2 Settings](#) (see page 3) for more information.

The configurable parameters of Output file format and Advance Settings are stored in `MPFS2SettingDetails.xml`.



Step 3 asks for the TCP/IP MPLAB X IDE project directory. The MPFS utility will write the image file to the project directory, and will also update the `http_print.h` file there if needed. Select the correct directory so that the right files are modified.

By default, the project directory path is one path behind the source directory. Both the Project Directory and Image Name are configurable parameters, which are stored in `MPFS2SettingDetails.xml`.



Step 4 controls the upload settings. When external EEPROM or serial Flash is used for storage, the option to upload the newly created image to the board is available. Check the box next to **Upload Image To** to enable this feature. The target host name (or IP address), upload protocol, and upload path may need to be changed to the one chosen when the board was first configured. You may also need to modify the user name and password used to access the secured functionality in your application, like web page upload. Use the **Settings** button to edit these values (see [MPFS2 Upload Settings](#) (see page 6) for more information). If internal program memory is being used, the image will be compiled in with the project and so direct uploads are not available. Make sure to include the output source file indicated in Step 3 as part of the project.

Once all the correct settings have been chosen, click the **Generate** button to create the image. If uploads are enabled, this will also attempt to upload the file to the device.

Uploading Prebuilt MPFS2 Images

This topic provides information on uploading prebuilt MPFS2 images.

Description

There are two ways to upload a prebuilt image to external storage. The first involves uploading from the browser directly. The second is to use the MPFS2 Utility to upload the image. You can select HTTP or FTP uploading to match the protocol that your application uses.

To use the MPFS2 Utility to upload an image, begin by selecting Start With: **Pre-Built MPFS Image** in Step 1. The Source: Directory path and Start With: radio button are configurable parameters, which are stored in `MPFS2SettingDetails.xml`. Choose the image file to upload.

Steps 2 and 3 are not required for prebuilt images. Proceed directly to Step 4 and verify that the upload settings are correct. The target host name (or IP address), upload protocol, and upload path may need to be changed to the one chosen when the board was first configured. You may also need to modify the user name and password used to access the secured functionality in your application, like web page upload. Use the **Settings** button to edit these values (see [MPFS2 Upload Settings](#) (see page 6) for more information).

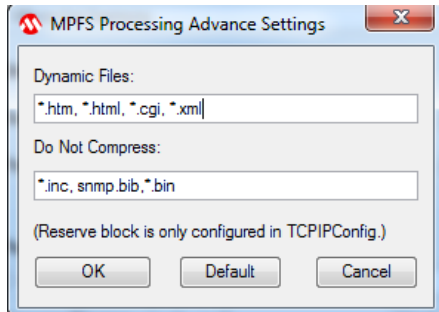
Once all the settings are correct, click **Upload**. The image will be uploaded to the board.

Advanced MPFS2 Settings

This topic provides information on advanced MPFS2 settings.

Description

The Advanced Settings dialog found in Step 2 provides greater control over how files are processed.



The **Dynamic Files** list indicates which file types to parse for dynamic variables. By default, all files with the extensions .htm, .html, .cgi, or .xml are parsed. If an application has dynamic variables in other file types, these types must be added to the list. This field must be a comma-separated list of extensions and file names.

Default Dynamic Files are *.htm, *.html, *.cgi, and *.xml.

The **Do Not Compress** field indicates which file types should never be compressed. Compressing files with GZIP saves both storage space and transmission time. However, this is only suitable for static content such as CSS or JavaScript. Any files with dynamic variables will automatically be excluded. In addition, any file that the PIC may need to process internally should be excluded. Files included via ~inc:filename~ should not be compressed, nor should any BIB file used for the SNMP module (if present). Additional file types can be added to this list if a custom application will be accessing the MPFS.

Default Do Not Compress Files are *.inc, snmp.bib, and *.bin.

The GZIP compressor will attempt to shrink all files. In some cases, especially with images, little or no compression is achieved. When this occurs the file is stored "as is" in the MPFS image.

MPFS2 Command Line Options

This topic provides a description of the MPFS2 command line options.

Description

To facilitate batch files and automation, the MPFS2 Utility also supports execution from the command line. The syntax is as follows:

```
MPFS2.jar [options] <SourceDir> <ProjectDir> <OutputFile>
```

The SourceDir, ProjectDir, and OutputFile options are required and should be enclosed in quotation marks. The OutputFile option will be relative to ProjectDir, and cannot be a full path name. The various option switches are described in the following table.

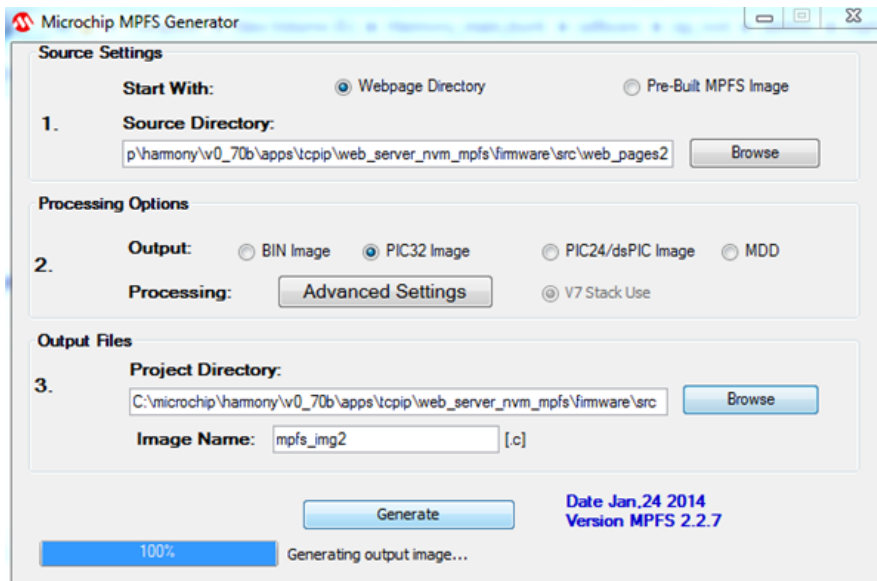
Switch	Short	Description
/BIN	/b	Output a BIN image (Default)
/PIC32	/c	Output a XC32 image
/ASM16	/s	Output an ASM16 image
/mpfs2	/2	Use the MPFS2 format (Default)
/html "..."	/h "..."	File types to be parsed for dynamic variables (Default: "*.htm, *.html, *.cgi, *.xml")
/xgzip "..."	/z "..."	File types to be excluded from GZIP compression (Default: "*.bib, *.inc, *.bin")

SourceDir, ProjectDir, and OutputFile are required and should be enclosed in quotes.

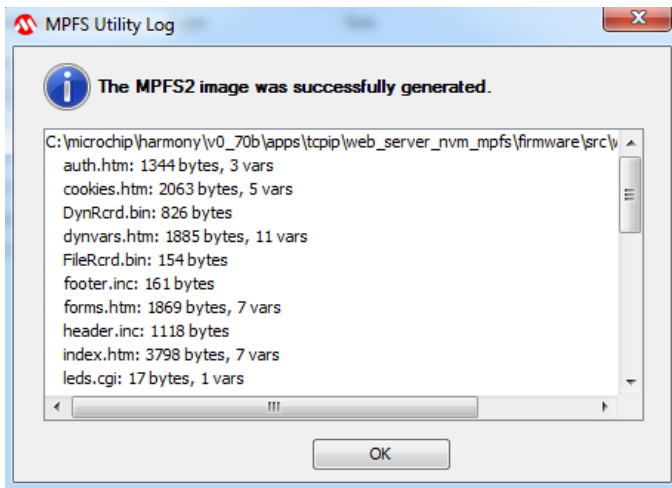
OutputFile is placed relative to ProjectDir and cannot be a full path name.

For unified stack (/v) should be used with PIC32 output image format.

PIC32 Output Option Example:



Successful MPFS2 Image Generation Example:



MPFS2 Utility Output Log Details

```
C:\microchip\harmony\v0_70b\apps\tcpip\web_server_nvm_mpfs\firmware\src\web_pages2 :
auth.htm: 1344 bytes, 3 vars
cookies.htm: 2063 bytes, 5 vars
DynRcd.bin: 826 bytes
dynvars.htm: 1885 bytes, 11 vars
FileRcd.bin: 154 bytes
footer.inc: 161 bytes
forms.htm: 1869 bytes, 7 vars
header.inc: 1118 bytes
index.htm: 3798 bytes, 7 vars
leds.cgi: 17 bytes, 1 vars
mchp.css: 915 bytes (gzipped by 72%)
mchp.gif: 1250 bytes (gzipped by 2%)
mchp.js: 1474 bytes (gzipped by 61%)
snmp.bib: 493 bytes
status.xml: 183 bytes, 7 vars
temp.cpf: 236 bytes (gzipped by 31%)
upload.htm: 943 bytes, 4 vars
C:\microchip\harmony\v0_70b\apps\tcpip\web_server_nvm_mpfs\firmware\src\web_pages2\snmp :
snmp/snmpconfig.htm: 1341 bytes, 9 vars
C:\microchip\harmony\v0_70b\apps\tcpip\web_server_nvm_mpfs\firmware\src\web_pages2\protect :
protect/config.htm: 1888 bytes, 12 vars
protect/index.htm: 963 bytes, 3 vars
protect/reboot.cgi: 8 bytes, 1 vars
protect/reboot.htm: 1779 bytes, 4 vars
```

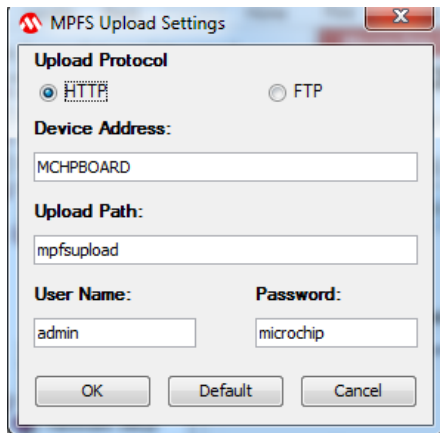
```
C:\microchip\harmony\v0_70b\apps\tcpip\web_server_nvm_mpfs\firmware\src\web_pages2\email :  
email/index.htm: 1876 bytes, 6 vars  
C:\microchip\harmony\v0_70b\apps\tcpip\web_server_nvm_mpfs\firmware\src\web_pages2\dyndns :  
dyndns/index.htm: 1461 bytes, 12 vars  
GENERATED MPFS2 IMAGE: 28932 bytes
```

MPFS2 Upload Settings

This topic provides a description of the MPFS2 upload settings.

Description

The Upload Settings dialog found in Step 4 of [Building MPFS2 Images](#) ([see page 2](#)) provides configuration to upload the binary file to the external EEPROM or serial Flash using either the HTTP or FTP protocol.



The Device Address is either a board name or an IP address. Please note that only IPv4 is supported.

By default, the Upload Path is "mpfsupload", the MPFS upload User Name is "admin", and the Password is "Microchip".

Index

A

Advanced MPFS2 Settings 3

B

Building MPFS2 Images 2

I

Introduction 2

MPFS2 Utility 2

M

MPFS2 Command Line Options 4

MPFS2 Upload Settings 6

MPFS2 Utility 2

U

Uploading Prebuilt MPFS2 Images 3