

7 SPLITTER

Overview

The ADSP218x splitter/loader tool can prepare non-bootable PROM image files, which execute from DSP external memory. In most all instances, developers working with an ADSP-21xx DSP should use the loader instead of the splitter. [For more information, see “ADSP-218x Loader”](#). When working within the VisualDSP++ environment, using the splitter to produce a PROM file is an output option for projects.

In ADSP-218x DSPs, the splitter (`elfspl21.exe`) would be used to generate images for the external PMOVLAY (=1 and 2) and DMOVLAY (=1 and 2) memory pages. If you use the splitter to produce ROM images (for example, ADSP-2181 program memory pages 1 and 2), the generated code must target ROM. Define appropriate ROM segments in your architecture (.LDF) file.

This chapter contains the following information on the splitter:

- [“Using the Splitter” on page 7-2](#)

This section describes how you can use the splitter via the VisualDSP++ environment.

- [“Splitter Command-Line Reference” on page 7-3](#)

This section contains reference information on the splitter commands and operations.

- [“Splitter Glossary” on page 7-9](#)

This section contains a glossary of splitter related terms.

Using the Splitter

The splitter/loader (`elfspl21.exe`) processes your executable files, producing a non-bootable-PROM-image file. Splitter operations depend on a set of splitter options.

Splitter options let you control how the splitter processes your executable files, letting you select features such as memory type and file format among others. These options come from splitter command-line switches or (when used within the VisualDSP++ environment) come from settings in the **Load** dialog box (via the **Project Options** selection of the VisualDSP++ IDDE). These **Loader/Splitter** dialog box settings correspond to command-line switches.

When developing a DSP project, you may find it useful to modify the splitter's default option settings in the VisualDSP++ environment.

Use the **Additional Options** field from the **Load** tab (in **Project Options** dialog box) to enter splitter settings and command switches. For more information, see the *VisualDSP++ 2.0 User's Guide for ADSP-21xx DSPs*.

The splitter has command-line switches that correspond to the **Split** dialog's options. For more information on these two controls on the split operation, see [“Splitter Command-Line Reference” on page 7-3](#).

Splitter Command-Line Reference

The splitter `elfspl21.exe`) generates non-bootable PROM image files for ADSP-218x DSPs by processing executable files. The splitter's output is a PROM file with the file name extension `.bnl`, `.bnu`, or `.bnm`. To make an PROM file, the splitter processes data from a linker's executable file (`.DXE`).

This section provides reference information on the splitter command line and splitting. A list of all switches and their descriptions appear in [Table 7-2 on page 7-6](#).

Use the following syntax for the splitter command line:

```
elfspl21 [-switch ...] -pm &| -dm &| executable
```

where:

- `[-switch ...]` — This is the name of one or more switches to be processed. The splitter has many optional switches. These select the operations and modes for the splitter.
- `-pm &| -dm &|` — The `&|` symbol between these three switch indicates AND-OR.
- *executable* — This is the name of the executable file (`.DXE`) to be processed for a single-processor boot-loadable file. A file name can include the drive, directory, file name and file extension; enclose long file names within straight-quotes, "long file name".

The splitter command line is case sensitive. The following two command lines, for example:

```
elfspl21 -pm -o pm_stuff my_proj.dxe
elfspl21 -dm -o dm_stuff my_proj.dxe
```

Splitter Command-Line Reference

run the splitter twice, once producing PROM files for program memory and again producing PROM files for data memory. The switches on these command lines are as follows:

```
-pm -dm
```

Selects program and data memory as sources in the executable for extraction and placement in the image. Because these are the only switches used for identifying the memory source, the source segments are any PM or DM ROM segments. Because no other contents-switches appear on these command lines, the format for the output defaults to Motorola S3 format, and the output PROM width defaults to 8-bits wide for all PROMs.

```
-o pm_stuff -o dm_stuff
```

Provides names for the output files, using differing names so that the output of the second run does not overwrite the output of the first run. The output names are `pm_stuff.s_#` and `dm_stuff.s_#`.

```
my_proj.dxe
```

Provides an executable file to process into a non-bootable-PROM-image file.

Many splitter switches take a file name as an optional parameter. [Table 7-1 on page 7-5](#) lists the type of files, names, and extensions that the splitter expects on files. File searches are important in the splitter's process. The splitter supports relative and absolute directory names, default directories, and user-selected directories for file search paths.

File searches occur as follows:

1. *Specified path* — If you include relative or absolute path information in a file name, the splitter only searches in that location for the file.
2. *Default directory* — If you do not include path information in the file name, the splitter searches for the file in the current working directory.

The splitter follows the conventions for file names in [Table 7-1](#).

Table 7-1. File Name Extension Conventions

Extension	File Description
.dxe	Executable files and boot-kernel files
.bnm	Splitter binary output file — middle
.bnu	Splitter binary output file — upper
.bnl	Splitter binary output file — lower

When you provide an input or output file name as a command-line parameter, use the following guidelines:

- Enclose long file names within straight-quotes, "long file name".
- Append the appropriate file name extension to each file.

Splitter Command-Line Reference

Descriptions for each switch appear in [Table 7-2](#).

Table 7-2. Splitter Command-Line Switches

Switch	Description
<i>executable</i>	The <i>executable</i> parameter selects an executable file for processing into PROM files.
-dm	The -dm (include Data Memory) switch directs the splitter to extract segments from the executable that you have declared as Data Memory ROM. The -dm switch influences the operation of the -ram and -norom switches, adding Data Memory as their target.
-o <i>image</i>	The -o (output file) switch (optional) directs the splitter to use the <i>image</i> name for the splitter's output file(s). If not specified, the default name for the splitter's output is " <i>executable.ext</i> ," where <i>ext</i> depends on the output format.
-norom	The -norom (no ROM in PROM) switch directs the splitter to ignore any ROM segments in the <i>executable</i> when extracting information for the output image. The -dm and -pm switches control selection of Data or Program memory for the -norom switch.
-pm	The -pm (include Program Memory) switch directs the splitter to extract segments from the executable that you have declared as Program Memory ROM. The -pm switch influences the operation of the -ram and -norom switches, adding Program Memory as their target.
-ram	The -ram (include RAM in PROM) switch directs the splitter to extract any RAM segments in the <i>executable</i> when extracting information for the output image. The -dm and -pm switches control selection of Data or Program memory for the -ram switch. The operation of the -s switch is not influenced by the -ram switch.
-readall	The -readall (include RAM and ROM in PROM) switch directs the splitter to extract both RAM and ROM segments from the input file(s)

Table 7-2. Splitter Command-Line Switches (Cont'd)

Switch	Description
-u	The -u (user flags) switch directs the splitter to use an appropriate byte-stacked format file. -u s1 format (default) -us s1 format with bytes stacked for 8 bit memory -us2 s2 format with bytes stacked for 8 bit memory -ui Intel format with bytes stacked for 8 bit memory
-i	This switch directs the splitter to use the Intel hex format
-byte	This switch directs the splitter to use the byte stream output format
-2181	When used with -2181 -loader, this switch generates a byte memory image without a boot loader.
-218x	This switch supports potential internal PMOVLAY/DMOVLAY pages. Use in place of -2181. For example, -2187, -2188, or -2189.
-noloader	When used with -2181 -loader, this switch generates a byte memory image without a boot loader.
-bdma <i>inputfile</i> <i>start_address</i>	For use with -2181 -loader. This option allows the user to specify an additional .dxe file (inputfile) to be placed in byte memory, starting at the specified address. The splitter will return an error if the specified address is in use by the boot loader, or another additional -bdma specified file.
-bdmaload <i>inputfile</i> <i>start_address</i>	This switch allows the user to specify an additional .dxe file to be placed in byte memory, starting at the specified address. (For use with -2181 -loader). The difference for the -bdma switch is that the splitter generates a load frame (BDMA boot code) as it does for the primary file. It also includes the 32-word preloader twice. The starting address has to be a byte page border (a multiple of 0x4000).
-unload <i>file</i>	This switch reads the 32-word BDMA boot loader (preloader) from an object file (.DOJ). The file length must not exceed 32 instructions.

Splitter Command-Line Reference

Table 7-2. Splitter Command-Line Switches (Cont'd)

Switch	Description
offsetaddr #	This switch provides the byte memory address offset (this switch only valid with -2181 -noloader)
offsetpage #	This switch provides the byte memory page offset (switch only valid with -2181 -noloader)

Splitter Glossary

Splitter — The splitter refers to `ElfSpl21k` contained in the software release.

Non-bootable PROM image files — The output from the splitter consists of PROM files that cannot be used to boot load a system. Use the loader to produce boot-loadable files. For more information, see the overview of the loader on [.](#)

Splitter Glossary