

4 LOADER

Overview

The loader (`elfloader.exe`) converts executable files into boot-loadable files for a TigerSHARC DSP. This chapter contains the following information on the loader:

- [“Loader Guide” on page 4-2](#)

This section contains procedures for using the loader within the VisualDSP++ environment.

- [“Loader Command-Line Reference” on page 4-6](#)

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

- [“Loader Glossary” on page 4-12](#)

This section contains a glossary of loader related terms.



The examples in this chapter cover the ADSP-TS001 DSP. For more information on this and other DSPs, see the appropriate DSP technical manual.

Loader Guide

The loader (`elfloader.exe`) processes your executable files, producing a boot-loadable file. Loader operations depend on a set of loader options.

The loader (`elfloader.exe`) generates boot-loadable file for TigerSHARC DSPs by processing executable files. The loader's output is a boot-loadable file with the file name extension `.LDR`. To make a loadable file, the loader processes data from a boot-kernel file (`.DXE`) and one or more other executable files (`.DXE`). Once you have fully debugged your program, use the loader to generate a set of boot-loadable files for your target system.

Loader options let you control how the loader processes your executable files, letting you select features such as loader kernel, boot type, and file format among others. These options appear on the loader command line or the Load tab of the Project Options dialog box in the VisualDSP++ environment.



The selections available via the Load options dialog box correspond to the loader's command-line switches .

All software developers using the loader should be familiar with the following operations:

- [“Working With Different Boot Types” on page 4-3](#)
- [“Boot-Kernel Features & Boot-Loading Process” on page 4-4](#)
- [“Boot-Kernel Changes and Software Issues” on page 4-5](#)
- [“Setting Loader Options” on page 4-5](#)

Working With Different Boot Types

Before selecting options for the loader's operation, you should read about how the loader's features apply to the boot-type that you are using. This section describes the boot-types for ADSP-TSxxx DSPs and describes loader features that support these boot-types.

The TigerSHARC DSPs have a special hardware feature that boot-loads a small, 256-instruction program into the DSP's internal memory at chip reset. This program can come from an external PROM, a host processor, or another ADSP-TSxxx DSP. You select the boot-type using the DSP's BMS external pin as shown in [Table 4-1 on page 4-4](#). After the boot-kernel loads, it executes and loads the rest of your program and data into the DSP. The combination of the boot-kernel and your program make up a boot-loadable file.

Any ADSP-TSxxx DSP supports three booting modes: EPROM, host, and link. Boot-loadable files for these modes pack the boot-data into 32-bit instructions and use a DMA channel of the DSP's DMA controller to boot-load the instructions:

- For booting from an EPROM through the external port, the DSP reads boot-data from an 8-bit external EPROM.
- For booting from a host processor through the external port, the DSP accepts boot-data from a 32- or 64-bit synchronous host microprocessor.
- For booting from another ADSP-TSxxx DSP through the link port, the DSP receives boot-data through its link port.

Table 4-1 shows how to select among the EPROM-, host-, and link-booting modes for a TigerSHARC DSP.

Table 4-1. Boot Mode Pins (Descriptions)

Pin	Type	Description
$\overline{\text{BMS}}$	I/O/T	Boot Memory Select — This strap option pin has a weak (100W) internal pulldown. When boot-loading from an EPROM, $\overline{\text{BMS}} = 0$. If $\overline{\text{BMS}} = 1$, link or host booting may be selected.

Boot-Kernel Features & Boot-Loading Process

For all boot-kernels, the boot-loading process begins by downloading the boot-kernel into the DSP's memory. The boot-kernel sets up the DSP as necessary and loads your memory. After the boot-kernel finishes initializing the rest of the system, the boot-kernel overwrites itself.

There are three boot-kernels that come with VisualDSP++ software. Boot kernels are loaded at reset into program segment `seg_ldr`, which is defined in the `ADSP-TS001 Loader.ldf`, linker description file. This file is stored in the `...\VisualDSP++\Ts\ldr` directory:

- `Ts001_prom.asm` is the source file for the ADSP-TSxxx PROM boot-kernel.
- `Ts001_host.asm` is the source file for the ADSP-TSxxx host boot-kernel.
- `Ts001_link.asm` is the source file for the ADSP-TSxxx link boot-kernel.



For more information on boot kernel operations, see comments in the corresponding boot kernel source file.

Boot-Kernel Changes and Software Issues

For most systems, some customization of the boot-kernel is required. Also, the operation of other tools (notably the C/C++ compiler) is influenced by whether or not you plan to use the loader.



For more information on boot kernel operations and customizations, see comments in the corresponding boot kernel source file.

Setting Loader Options

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

The loader has command-line switches that correspond to the Load dialog's options. Use the **Additional Options** field in the Load tab of the Project Options dialog box to enter the appropriate file names and command-line switches to control the loader operation.

For more information, see the *VisualDSP++ User's Guide for TigerSHARC DSPs*.

Loader Command-Line Reference

This section provides reference information on the loader command line and boot-loading. A list of all switches appears in [Table 4-3 on page 4-8](#), and a description for each switch appears in [“Loader Command-Line Switches” on page 4-9](#).



When using the loader within the VisualDSP++ environment, the settings in the Load options dialog box correspond to the loader's command-line switches. .

Loader Command-Line Syntax

Use the following syntax for the loader command line:

```
elfloader sourcefile -dADSP-TSxxx -switch [-switch ...]
```

where:

- *sourcefile* — 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 double quotation marks, "long file name".
- -dADSP-TSxxx — This switch is required for producing a DSP loadable file (where xxx specifies either 001 or 101).
- -switch — This is the name of the switch to be processed. The loader provides many optional switches that select the operations and modes for the loader.

The loader command line is case sensitive. For example, the following command line,

```
elfloader p0.dxe -dADSP-TS001 -b prom -f hex -l Ts001_prom.dxe
```

runs the loader with:

`p0.dxe` — Names an executable file to process into a boot-loadable file; because the `-o` switch does not appear on this command line, the name for the output file is `p0.ldr`.

`-dADSP-TS001` — Selects ADSP-TS001 as the DSP type for the boot-loadable file

`-b prom` — Selects EPROM booting as the boot-type for the boot-loadable file

`-f hex` — Selects Intel^R Hex-32 as the format for the boot-loadable file

`-l Ts001_prom.dxe` — Names the `Ts001_prom.dxe` file as the boot-kernel for the boot-loadable file

Many loader switches take a file name as an optional parameter. [Table 4-2 on page 4-8](#) lists the type of files, names, and extensions that the loader expects on files. File searches are important in the loader's process. The loader 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 loader only searches in that location for the file.
2. *Default directory*—If you do not include path information in the file name, the loader searches for the file in the current project directory.

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

- Enclose long file names within double quotation marks, "long file name".
- Append the appropriate file name extension to each file.

Loader Command-Line Reference

The loader follows the conventions for file name extensions that appear in [Table 4-2](#). A summary of the loader’s command line switches appears in [Table 4-3](#). Descriptions for each switch appear in “[Loader Command-Line Switches](#)” on [page 4-9](#).

Table 4-2. File Name Extension Conventions

Extension	File Description
<code>.dxe</code> ¹	Executable files and boot-kernel files
<code>.ldr</code>	Loader output file

¹ The loader recognizes shared memory (`.SM`) and overly memory (`.OVL`) files, but does not expect these files on the command line. Place `.SM` or `.OVL` files in the same directory as the `.DXE` that refers to them, and the loader finds the `.SM` and `.OVL` files when processing the `.LDR` file.

Table 4-3. Loader Switch Summary ¹

<i>filename</i>	<code>-dADSP-TS001</code> <code>-dADSP-TS001</code>	<code>[-b boot-type]</code>
<code>[-e filenames]</code>	<code>[-f format]</code>	<code>[-h]</code>
<code>[-id#exe=file]</code>	<code>[-l kernel]</code>	<code>[-o filename]</code>
<code>[-p address]</code>	<code>[-t timeout]</code>	<code>[-v]</code>

¹ The switches in [Table 4-3](#) may be used in any order on the command line. Items shown in `[ ]` are optional. Items shown in *italics* are user-defined and are described with each switch.

Loader Command-Line Switches

filename

The *filename* parameter without a switch selects an executable file for single-processor boot-loadable files. For multiprocessor boot-loadable files, use the `-id#exe=file` switch. For information on shared and overlay memory files, see [Table 4-2](#).

`-dADSP-TS001/TS101`

The `-dADSP-TS001` or `-dADSP-TS101` switch directs the loader to prepare a boot-loadable file for a corresponding ADSP-TSxxx DSP. This switch is required for producing an ADSP-TSxxx loader file—if omitted, the loader produces, by default, an ADSP-21xxx SHARC DSP loader file.

`-b boot-type`

The `-b` (boot-type) switch directs the loader to prepare a boot-loadable file for the *type* booting mode. Valid *type* selections are `PROM`, `HOST`, and `LINK`. If the `-b` switch does not appear on the command line, the default setting is `-bprom`. Note that the boot-type that you select with the `-b` switch must correspond to the boot-kernel that you select with the `-l` switch.

`-e filenames`

The `-e` (except shared memory) switch directs the loader to omit the listed shared memory (`.SM`) files from the output loader file. If you prepare a boot-loadable file for multiple processors that contains shared memory, all the processors' boot images contain the shared memory items. This switch lets you omit the shared parts of the executable from multiple processor boot files. You must run the loader again to put the shared parts of the executable in a boot file for just one of the processors. This technique lets you reduce boot-loading time by loading the shared parts of the executable only once.

Loader Command-Line Reference

-f format

The **-f** (boot-file format) switch directs the loader to prepare a boot-loadable file in the specified `format`. The available `format` selections are `hex` (Intel Hex-32), `ASCII`, and `binary`, but valid formats depend on the **-b** switch's boot-type selection. For a `prom` boot-type, use the `hex`, `binary` or `ASCII` formats; for the `host` or `link` boot-types, use the `ASCII` or `binary` formats. If the **-f** switch does not appear on the command line, the default format for boot-types are:
`hex` for `prom` and `ASCII` for `host` or `link`.

-h

The **-h** (command-line help) switch directs the loader to output the list of command line switches to standard output and exit. Combine the **-h** and **-dADSP-TS001** switches to get help for ADSP-TSxxx DSPs. The **-h** switch alone, by default, provides help for ADSP-21xxx DSPs.

-id#exe=file

The **-id#exe** (multiprocessor ID) switch directs the loader to use `#` processor ID for the corresponding executable file when producing a boot-loadable file for EPROM or host booting a multiprocessor system. This switch is used only for producing a boot-loadable file that boots multiple DSPs from a single EPROM. Valid `#` values are 0, 1, 2, 3, 4, 5, 6, and 7. You do not have to use this switch for single-processor systems; input the executable `filename` as a parameter without any switch to select an executable file for single-processor systems.

-l kernel

The **-l** (boot-kernel) switch directs the loader to use the `kernel` file for the boot-loading routine in the output boot-loadable file. Note that the boot-kernel that you select with the **-l** switch must correspond to the boot-type that you select with the **-b** switch and the processor-type from the linker description file that you select with the **-a** switch. If the **-l** switch does not appear on the command line, the loader searches for a

default boot-kernel file in the default installation directory. Based on the `-b` switch setting, the loader searches for the default boot-kernel in the current directory and the processor specific `\VisualDSP++\Ts\ldr` directory as follows:

<code>-b switch</code>	<i>Default Boot-Kernel</i>
<code>prom</code>	<code>Ts001_prom.dxe</code>
<code>host</code>	<code>Ts001_host.dxe</code>
<code>link</code>	<code>Ts001_link.dxe</code>

`-o filename`

The `-o` (output file) switch directs the loader to use the `filename` for the loader's output file. If not specified, the default name for the loader's output is "`sourcefile.ldr`."

`-p address`

The `-p` (PROM start address) switch directs the loader to start the boot-loadable file at the specified `address` in the EPROM. If the `-p` switch does not appear on the command line, the loader starts the EPROM file at address `0x0` in the EPROM; this EPROM address corresponds to address `0x4000000` in an ADSP-TSxxx DSP.

`-t timeout`

The `-t` (timeout cycles) switch directs the loader set the `timeout` number of cycles as a limit on the number of cycles that the DSP spends initializing external memory. Valid `timeout` values range from 3 to 32765 cycles; 32765 is the default `timeout` value. The `timeout` value is directly related to the number of cycles that the DSP locks the bus for boot-loading, instructing the DSP to lock the bus for no more than `2x timeout` number of cycles. If you are working with a fast host that cannot tolerate being locked out of the bus, use a relatively small value `timeout`.

`-v`

Loader Glossary

The `-v` (verbose loader messages) switch directs the loader to output status information as the loader processes your files.

Loader Glossary

Boot-kernel—The boot-kernel refers to the executable file that performs the memory initialization on the target.

Boot-loadable file—The boot file refers to the output of the loader that contains the boot loader and the formatted system configurations.

Boot-loading or booting—Booting refers to the process of loading the boot loader, initialization system memory, and starting the application on the target.

Loader—The loader refers to `elfloader` contained in the software release.