# IA-32 Assembly Language Reference Manual

**Previous**: Section Header **Next**: Symbol Tables

## Sections

A section is the smallest unit of an object file that can be relocated. Sections containing the following material usually appear in relocatable ELF files:

- Executable text
- Read-only data
- Read-write data
- Read-write uninitialized data (only **section header** appears)

Sections do not need to occur in any particular order within the object file. The sections of a relocatable ELF file contain all of the file information that is not contained in the ELF header or in the section header table. The sections in any ELF file must satisfy several conditions:

1. Every section in the file must have one section header entry in the section header table to describe the section. However, the section header table can have section header entries that correspond to no section in the file.
2. Each section occupies one contiguous sequence of bytes within a file. The section can be empty (even so, its section header entry in the section header table can have a nonzero value for the field sh_size).
3. A byte in a file can reside in at most one section. Sections in a file cannot overlap.
4. An object file can have inactive space. Inactive space is the set of all bytes in the file that are not part of the ELF header, the section header table, the program header table (for executable files), or of any section in the file. The contents of the inactive space are unspecified.

Sections can be added for multiple text or data segments, shared data, user-defined sections, or information in the object file for debugging.

**Note -**

Not all of the sections where there are entries in the file section header table need to be present.

### Predefined Sections

Sections having certain names beginning with "." (dot) are predefined, with their types and attributes already assigned. These special sections are of two kinds: predefined user sections and predefined nonuser sections.

### Predefined User Sections

Sections that an assembly language programmer can manipulate by issuing section control directives in the source file are **user sections**. The predefined user sections are those predefined sections that are also user sections.

Table 3-4 lists the names of the predefined user sections and briefly describes each.

Table 3-4 Predefined User Sections

| Section Name | Description |
|---|---|

| Section Name | Description |
|---|---|
| ".bss" | Uninitialized read-write data. |
| ".comment" | Version control information. |
| ".data" & ".data1" | Initialized read-write data. |
| ".debug" | Debugging information. |
| ".fini" | Runtime finalization instructions. |
| ".init" | Runtime initialization instructions. |
| ".rodata" & ".rodata1" | Read-only data. |
| ".text" | Executable instructions. |
| ".line" | Line # info for symbolic debugging. |
| ".note" | Special information from vendors or system builders. |

## Predefined Non-User Sections

Table 3-5 shows the predefined sections that are not user sections, because assembly language programmers cannot manipulate them by issuing section control directives in the source file.

Table 3-5 Predefined Non-user Sections

| Section Name | Description |
|---|---|
| ". dynamic " | Dynamic linking information. |
| ". dynstr " | Strings needed for dynamic linking. |
| ".dynsym" | Dynamic linking symbol table. |

| Section Name | Description |
|---|---|
| ". got " | Global offset table. |
| ". hash " | A symbol hash table. |
| ". interp " | The path name of a program interpreter. |
| ". plt " | The procedure linking table. |
| " rel **name**" & ". rela **name**" | Relocation information. name is the section to which the relocations apply. e.g., ". rel.text ", ". rela.text ". |
| ". shstrtab " | String table for the section header table names. |
| ". strtab " | The string table. |
| ". symtab " | The symbol table. |

## Relocation Tables

Locations represent **addresses in memory** if a section is allocatable; that is, its contents are to be placed in memory at program runtime. Symbolic references to these locations must be changed to addresses by the link editor.

The assembler produces a companion **relocation table** for each relocatable section. The table contains a list of relocations (that is, adjustments to locations in the section) to be performed by the link editor.

**Previous**: Section Header **Next**: Symbol Tables