NAME

pvs – display the internal version information of dynamic objects

SYNOPSIS

pvs [**–dnorsv**] [**–N** *name*] *filename*. . .

AVAILABILITY

SUNWtoo

DESCRIPTION

pvs displays any internal version information contained within an ELF file. Commonly these files are dynamic executables and shared objects, and possibly relocatable objects. This version information can fall into one of two categories:

- version definitions
- version dependencies

Version *definitions* describe the interfaces made available by an ELF file. Each version definition is associated to a set of global symbols provided by the file. Version definitions may be assigned to a file during its creation by the link-editor using the **–M** option and the associated *mapfile* directives (see the *Linker and Libraries Guide* for more details).

Version *dependencies* describe the binding requirements of dynamic objects on the version definitions of any shared object dependencies. When a dynamic object is built with a shared object, the link-editor records information within the dynamic object indicating that the shared object is a dependency. This dependency must be satisfied at runtime. If the shared object also contains version *definitions,* then those version definitions that satisfy the global symbol requirements of the dynamic object will also be recorded in the dynamic object being created. At process initialization, the runtime linker will use any version *dependencies* as a means of validating the interface requirements of the dynamic objects used to construct the process.

OPTIONS

–d Print version definition information.

–n Normalize version definition information. By default, all version definitions within the object are displayed. However, version definitions may inherit other version definitions, and under normalization only the head of each inheritance list is displayed.

–o Create one-line version definition output. By default, file, version definitions, and any symbol output is indented to ease human inspection. This option preseeds each output line with the file and version definition name and may be more useful for analysis with automated tools.

–r Print version dependency (requirements) information.

–s Print the symbols associated with each version definition. Any data symbols are accompanied with the size, in bytes, of the data item.

–v Verbose output. Indicates any weak version definitions, and any version definition inheritance. When used with the **–N** and **–d** options, the inheritance of the base version definition is also shown. When used with the **–s** option, the version symbol definition is also shown.

–N *name* Print only the information for the given version definition *name* and any of its inherited version definitions (when used with the **–d** option), or for the given dependency file *name* (when used with the **–r** option).

If neither the **–b**, or **–r** options are specified, both will be enabled.

EXAMPLES

The following example displays the version definitions of **libelf.so.1**:

```
example% pvs -d /usr/lib/libelf.so.1
        libelf.so.1;
        SUNW.1.1
```

A normalized, one-liner display, suitable for creating a *mapfile* version control directive can be created using the **–n** and **–o** options:

```
example% pvs -don /usr/lib/libelf.so.1
/usr/lib/libelf.so.1 -     SUNW.1.1;
```

The following example displays the version requirements of **ldd**, and **pvs**:

```
example% pvs -r /usr/bin/ldd /usr/bin/pvs
/usr/bin/ldd:
        libelf.so.1 (SUNW.1.1);
        libc.so.1 (SUNW.1.1);
/usr/bin/pvs:
        libelf.so.1 (SUNW.1.1);
        libc.so.1 (SUNW.1.1);
```

EXIT CODES

If the requested version information is not found a non-zero value is returned, otherwise a zero value is returned. Version information is determined not found when; the **–d** option is specified and no version definitions are found; the **–r** option is specified and no version requirements are found; neither the **–d** or **–r** option is specified and no version definitions or version requirements are found.

SEE ALSO

ld(1), **ldd**(1), **elf**(3E)

Linker and Libraries Guide