



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## **Red Hat Enterprise Linux Release 9.2 Manual Pages on 'selabel\_file.5' command**

### **\$ man selabel\_file.5**

selabel\_file(5) SELinux API documentation selabel\_file(5)

#### NAME

selabel\_file - userspace SELinux labeling interface and configuration  
file format for the file contexts backend

#### SYNOPSIS

```
#include <selinux/label.h>

int selabel_lookup(struct selabel_handle *hnd,
                  char **context,
                  const char *path, int mode);

int selabel_lookup_raw(struct selabel_handle *hnd,
                      char **context,
                      const char *path, int mode);
```

#### DESCRIPTION

The file contexts backend maps from pathname/mode combinations into security contexts. It is used to find the appropriate context for each file when relabeling a file system. The returned context must be freed using freecon(3).

selabel\_lookup(3) describes the function with its return and error codes, however the following errno is clarified further for the file contexts backend:

ENOENT No context corresponding to the path and mode was found -

This will also be returned when the file contexts series of files have a context of <<none>> against the path (see

the FILE FORMAT section).

The path argument should be set to the full pathname of the file whose assigned context is being checked. The mode argument should be set to the mode bits of the file, as determined by lstat(2). mode may be zero, however full matching may not occur.

Any messages generated by selabel\_lookup(3) are sent to stderr by default, although this can be changed by selinux\_set\_callback(3).

selabel\_lookup\_raw(3) behaves identically to selabel\_lookup(3) but does not perform context translation.

The FILES section details the configuration files used to determine a file context.

## OPTIONS

In addition to the global options described in selabel\_open(3), this backend recognizes the following options:

### SELABEL\_OPT\_PATH

A non-null value for this option specifies a path to a file that will be opened in lieu of the standard file contexts file. This value is also used as the base name for determining the names of local customization files.

### SELABEL\_OPT\_BASEONLY

A non-null value for this option indicates that any local customizations to the file contexts mapping should be ignored.

### SELABEL\_OPT\_SUBSET

A non-null value for this option is interpreted as a path prefix, for example "/etc". Only file context specifications with starting with a first component that prefix matches the given prefix are loaded. This may increase lookup performance, however any attempt to look up a path not starting with the given prefix may fail. This optimization is no longer required due to the use of file\_contexts.bin files and is deprecated.

The file context files used to retrieve the default context depends on the SELABEL\_OPT\_PATH parameter passed to selabel\_open(3). If NULL, then the SELABEL\_OPT\_PATH value will default to the active policy file contexts location (as returned by selinux\_file\_context\_path(3)), otherwise the actual SELABEL\_OPT\_PATH value specified is used.

If SELABEL\_OPT\_BASEONLY is set, then the following files will be processed:

1. The mandatory file contexts file that is either the fully qualified file name from SELABEL\_OPT\_PATH.value or if NULL, then the path returned by selinux\_file\_context\_path(3).
2. The optional local and distribution substitution files that perform path aliasing on the 'in memory' version of the file contexts file.

These files have the same name as the mandatory file contexts file with the extensions .subs and .subs\_dist added.

If the SELABEL\_OPT\_BASEONLY is not set, then the following files will be processed:

1. The mandatory file contexts file that is either the fully qualified file name from SELABEL\_OPT\_PATH.value or if NULL, then the path returned by selinux\_file\_context\_path(3).
2. The optional local customizations file that has the same name as the mandatory file contexts file with the extension .local added.

selinux\_file\_context\_local\_path(3) will return the default path to this file.

3. The optional user home directory customizations file that has the same name as the mandatory file contexts file with the extension .homedirs added.  
selinux\_file\_context\_homedir\_path(3) will return the default path to this file.

4. The optional local and distribution substitution files that perform any path aliasing on the 'in memory' version of the file contexts file (and the .local and/or .homedirs if

present). These files have the same name as the mandatory file contexts file with the extensions `.subs` and `.subs_dist` added.

`selinux_file_context_subs_path(3)` and `selinux_file_con?`

`text_subs_dist_path(3)` will return the default paths to

these files.

The default file context series of files are:

`/etc/selinux/{SELINUXTYPE}/contexts/files/file_contexts`

`/etc/selinux/{SELINUXTYPE}/contexts/files/file_contexts.local`

`/etc/selinux/{SELINUXTYPE}/contexts/files/file_contexts.homedirs`

`/etc/selinux/{SELINUXTYPE}/contexts/files/file_contexts.subs`

`/etc/selinux/{SELINUXTYPE}/contexts/files/file_contexts.subs_dist`

Where `{SELINUXTYPE}` is the entry from the selinux configuration file `config` (see `selinux_config(5)`).

Only the `file_contexts` file is mandatory, the remainder are optional.

The entries within the file contexts series of files are shown in the FILE FORMAT section.

## FILE FORMAT

### File Contexts Format

Each line within the `file_contexts` and the two customization files (`.local` and `.homedirs`) is as follows:

```
pathname [file_type] context
```

Where:

`pathname`

An entry that defines the path to be labeled. May contain either a fully qualified path, or a Perl compatible regular expression (PCRE), describing fully qualified path(s). The only PCRE flag in use is `PCRE2_DOTALL`, which causes a wildcard `'.'` to match anything, including a new line. Strings representing paths are processed as bytes (as opposed to Unicode), meaning that non-ASCII characters are not matched by a single wildcard.

`file_type`

An optional file type consisting of:

- b - Block Device    -c - Character Device
- d - Directory       -p - Named Pipe
- l - Symbolic Link   -s - Socket
- - Ordinary file

context

This entry can be either:

- a. The security context that will be assigned to the file (i.e. returned as context).
- b. A value of <<none>> can be used to indicate that the matching files should not be relabeled and causes selabel\_lookup(3) to return -1 with errno set to ENOENT.

Example:

```
# ./contexts/files/file_contexts
# pathname file_type context
/*          system_u:object_r:default_t:s0
/[^/]+     --  system_u:object_r:etc_runtime_t:s0
/tmp/*     <<none>>
```

### Substitution File Format

Each line within the substitution files (.subs and .subs\_dist) has the form:

subs\_pathname pathname

Where:

pathname

A path that matches an entry in one or more of the file contexts policy configuration file.

subs\_pathname

The path that will be aliased (considered equivalent) with pathname by the look up process.

Example:

```
# ./contexts/files/file_contexts.subs
# pathname subs_pathname
```

/myweb /var/www

/myspool /var/spool/mail

Using the above example, when `selabel_lookup(3)` is passed a path of `/myweb/index.html` the function will substitute the `/myweb` component with `/var/www`, therefore the path used is:

`/var/www/index.html`

## NOTES

1. If contexts are to be validated, then the global option `SELABEL_OPT_VALIDATE` must be set before calling `selabel_open(3)`. If this is not set, then it is possible for an invalid context to be returned.
2. If the size of file contexts series of files contain many entries, then `selabel_open(3)` may have a delay as it reads in the files, and if requested validates the entries.
3. Depending on the version of SELinux it is possible that a `file_contexts.template` file may also be present, however this is now deprecated.

The template file has the same format as the `file_contexts` file and may also contain the keywords `HOME_ROOT`, `HOME_DIR`, `ROLE` and `USER`. This functionality has now been moved to the policy store and managed by `semodule(8)` and `genhomedircon(8)`.

## SEE ALSO

`selinux(8)`, `selabel_open(3)`, `selabel_lookup(3)`, `selabel_stats(3)`,  
`selabel_close(3)`, `selinux_set_callback(3)`,  
`selinux_file_context_path(3)`, `freecon(3)`, `selinux_config(5)`, `lstat(2)`,  
`selinux_file_context_subs_path(3)`,  
`selinux_file_context_subs_dist_path(3)`,  
`selinux_file_context_homedir_path(3)`,  
`selinux_file_context_local_path(3)`, `semodule(8)`, `genhomedircon(8)`