

NAME

posix_spawn_file_actions_addopen, **posix_spawn_file_actions_adddup2**,
posix_spawn_file_actions_addclose, **posix_spawn_file_actions_addclosefrom_np**,
posix_spawn_file_actions_addchdir_np, **posix_spawn_file_actions_addfchdir_np** - add open, dup2,
close, closefrom, or chdir/fchdir actions to spawn file actions object

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <spawn.h>
```

int

```
posix_spawn_file_actions_addopen(posix_spawn_file_actions_t *file_actions, int fildes,  
    const char *restrict path, int oflag, mode_t mode);
```

int

```
posix_spawn_file_actions_adddup2(posix_spawn_file_actions_t *file_actions, int fildes, int newfildes);
```

int

```
posix_spawn_file_actions_addclose(posix_spawn_file_actions_t *file_actions, int fildes);
```

int

```
posix_spawn_file_actions_addclosefrom_np(posix_spawn_file_actions_t *file_actions, int from);
```

int

```
posix_spawn_file_actions_addchdir_np(posix_spawn_file_actions_t *restrict file_actions,  
    const char *restrict path);
```

int

```
posix_spawn_file_actions_addfchdir_np(posix_spawn_file_actions_t *file_actions, int fildes);
```

DESCRIPTION

These functions add an open, dup2 or close action to a spawn file actions object.

A spawn file actions object is of type *posix_spawn_file_actions_t* (defined in *<spawn.h>*) and is used to specify a series of actions to be performed by a **posix_spawn()** or **posix_spawnnp()** operation in order to arrive at the set of open file descriptors for the child process given the set of open file descriptors of the parent.

A spawn file actions object, when passed to **posix_spawn()** or **posix_spawnnp()**, specify how the set of open file descriptors in the calling process is transformed into a set of potentially open file descriptors for the spawned process. This transformation is as if the specified sequence of actions was performed exactly once, in the context of the spawned process (prior to execution of the new process image), in the order in which the actions were added to the object; additionally, when the new process image is executed, any file descriptor (from this new set) which has its `FD_CLOEXEC` flag set is closed (see **posix_spawn()**).

The **posix_spawn_file_actions_addopen()** function adds an open action to the object referenced by *file_actions* that causes the file named by *path* to be opened (as if

```
open(path, oflag, mode)
```

had been called, and the returned file descriptor, if not *fildes*, had been changed to *fildes*) when a new process is spawned using this file actions object. If *fildes* was already an open file descriptor, it is closed before the new file is opened.

The string described by *path* is copied by the **posix_spawn_file_actions_addopen()** function.

The **posix_spawn_file_actions_adddup2()** function adds a `dup2` action to the object referenced by *file_actions* that causes the file descriptor *fildes* to be duplicated as *newfildes* (as if

```
dup2(fildes, newfildes)
```

had been called) when a new process is spawned using this file actions object, except that the `FD_CLOEXEC` flag for *newfildes* is cleared even if *fildes* is equal to *newfildes*. The difference from **dup2()** is useful for passing a particular file descriptor to a particular child process.

The **posix_spawn_file_actions_addclose()** function adds a close action to the object referenced by *file_actions* that causes the file descriptor *fildes* to be closed (as if

```
close(fildes)
```

had been called) when a new process is spawned using this file actions object.

The **posix_spawn_file_actions_addclosefrom_np()** function adds a close action to close all file descriptors numerically equal or greater than the argument *from*. For each open file descriptor, logically the close action is performed, and any possible errors encountered are ignored.

The **posix_spawn_file_actions_addchdir_np()** and **posix_spawn_file_actions_addfchdir_np()** functions

add a change current directory action to the object referenced by *file_actions* that affects actions (opens with relative path) performed after the operation, in the order of insertion into the *file_actions* object. It also sets the working directory for the spawned program. The **posix_spawn_file_actions_addchdir_np()** function takes the *path* to set as the working directory, while **posix_spawn_file_actions_addfchdir_np()** takes the directory file descriptor.

RETURN VALUES

Upon successful completion, these functions return zero; otherwise, an error number is returned to indicate the error.

ERRORS

These functions fail if:

- | | |
|----------|---|
| [EBADF] | The value specified by <i>fildes</i> or <i>newfildes</i> is negative. |
| [ENOMEM] | Insufficient memory exists to add to the spawn file actions object. |

SEE ALSO

close(2), dup2(2), open(2), posix_spawn(3), posix_spawn_file_actions_destroy(3), posix_spawn_file_actions_init(3), posix_spawnnp(3)

STANDARDS

The **posix_spawn_file_actions_addopen()**, **posix_spawn_file_actions_adddup2()** and **posix_spawn_file_actions_addclose()** functions conform to IEEE Std 1003.1-2001 ("POSIX.1"), with the exception of the behavior of **posix_spawn_file_actions_adddup2()** if *fildes* is equal to *newfildes* (clearing FD_CLOEXEC). A future update of the Standard is expected to require this behavior.

The **posix_spawn_file_actions_addchdir_np()**, **posix_spawn_file_actions_addfchdir_np()**, and **posix_spawn_file_actions_addclosefrom_np()** functions are non-standard functions implemented after the similar functionality provided by glibc.

HISTORY

The **posix_spawn_file_actions_addopen()**, **posix_spawn_file_actions_adddup2()** and **posix_spawn_file_actions_addclose()** functions first appeared in FreeBSD 8.0. The **posix_spawn_file_actions_addchdir_np()**, **posix_spawn_file_actions_addfchdir_np()**, and **posix_spawn_file_actions_addclosefrom_np()** functions first appeared in FreeBSD 14.0.

AUTHORS

Ed Schouten <ed@FreeBSD.org>