

**NAME**

**g\_post\_event**, **g\_waitfor\_event**, **g\_cancel\_event** - GEOM events management

**SYNOPSIS**

```
#include <geom/geom.h>
```

*int*

```
g_post_event(g_event_t *func, void *arg, int flag, ...);
```

*int*

```
g_waitfor_event(g_event_t *func, void *arg, int flag, ...);
```

*void*

```
g_cancel_event(void *ref);
```

*struct g\_event \**

```
g_alloc_event(int flag);
```

*void*

```
g_post_event_ep(g_event_t *func, void *arg, struct g_event *ep, ...);
```

**DESCRIPTION**

The GEOM framework has its own event queue to inform classes about important events. The event queue can be also used by GEOM classes themselves, for example to work around some restrictions in the I/O path, where sleeping, heavy weight tasks, etc. are not permitted.

The **g\_post\_event()** function tells the GEOM framework to call function *func* with argument *arg* from the event queue. The *flag* argument is passed to **malloc(9)** for memory allocations inside of **g\_post\_event()**. The only allowed flags are **M\_WAITOK** and **M\_NOWAIT**. The rest of the arguments are used as references to identify the event. An event can be canceled by using any of the given references as an argument to **g\_cancel\_event()**. The list of references has to end with a **NULL** value.

The **g\_waitfor\_event()** function is a blocking version of the **g\_post\_event()** function. It waits until the event is finished or canceled and then returns.

The **g\_post\_event\_ep()** function posts the event with a pre-allocated *struct g\_event*. An event may be pre-allocated with **g\_alloc\_event()**.

The **g\_cancel\_event()** function cancels all event(s) identified by *ref*. Cancellation is equivalent to calling the requested function with requested arguments and argument *flag* set to **EV\_CANCEL**.

**RESTRICTIONS/CONDITIONS****g\_post\_event():**

The argument *flag* has to be M\_WAITOK or M\_NOWAIT.

The list of references has to end with a NULL value.

**g\_waitfor\_event():**

The argument *flag* has to be M\_WAITOK or M\_NOWAIT.

The list of references has to end with a NULL value.

The **g\_waitfor\_event()** function cannot be called from an event, since doing so would result in a deadlock.

**g\_alloc\_event():**

The argument *flag* has to be M\_WAITOK or M\_NOWAIT.

The returned *struct g\_event \** must be freed with **g\_free()** if **g\_post\_event\_ep()** is not called.

**RETURN VALUES**

The **g\_post\_event()** and **g\_waitfor\_event()** functions return 0 if successful; otherwise an error code is returned.

**EXAMPLES**

Example of a function called from the event queue.

```
void
example_event(void *arg, int flag)
{
    if (flag == EV_CANCEL) {
        printf("Event with argument %p canceled.\n", arg);
        return;
    }

    printf("Event with argument %p called.\n", arg);
}
```

**ERRORS**

Possible errors for the **g\_post\_event()** function:

[ENOMEM]           The *flag* argument was set to M\_NOWAIT and there was insufficient memory.

Possible errors for the **g\_waitfor\_event()** function:

[EAGAIN]           The event was canceled.

[ENOMEM]           The *flag* argument was set to M\_NOWAIT and there was insufficient memory.

**SEE ALSO**

geom(4), DECLARE\_GEOM\_CLASS(9), g\_access(9), g\_attach(9), g\_bio(9), g\_consumer(9),  
g\_data(9), g\_geom(9), g\_provider(9), g\_provider\_by\_name(9), g\_wither\_geom(9)

**AUTHORS**

This manual page was written by Pawel Jakub Dawidek <[pjd@FreeBSD.org](mailto:pjd@FreeBSD.org)>.