NAME

g_new_geomf, **g_destroy_geom** - geom management

SYNOPSIS

#include <geom/geom.h>

*struct g_geom **
g_new_geomf(*struct g_class *mp, const char *fmt, ...*);

void
g_destroy_geom(*struct g_geom *gp*);

DESCRIPTION

The geom (do not confuse "geom" with "GEOM") is an instance of a GEOM class. For example: in a typical i386 FreeBSD system, there will be one geom of class MBR for each disk. The geom's name is not really important, it is only used in the XML dump and for debugging purposes. There can be many geoms with the same name.

The **g_new_geomf**() function creates a new geom, which will be an instance of the class *mp*. The geom's name is created in a printf(3)-like way from the rest of the arguments.

The **g_destroy_geom**() function destroys the given geom immediately and cancels all related pending events.

The *g_geom* structure contains fields that should be set by the caller after geom creation, but before creating any providers or consumers related to this geom (not all are required):

*g_start_t * start*
Pointer to a function used for I/O processing.

*g_spoiled_t * spoiled*
Pointer to a function used for consumers spoiling.

*g_dumpconf_t * dumpconf*
Pointer to a function used for configuration in XML format dumping.

*g_access_t * access*
Pointer to a function used for access control.

*g_orphan_t * orphan*

Pointer to a function used to inform about orphaned consumer.

*g_ioctl_t * ioctl*
Pointer to a function used for handling ioctl requests.

*void * softc*
Field for private use.

RESTRICTIONS/CONDITIONS

If you intend to use providers in this geom you must set field *start* of your geom.

If you are planning to use consumers in your geom you must set fields *orphan* and *access* for it.

g_new_geomf():

Class *mp* must be valid (registered in GEOM).

The topology lock has to be held.

g_destroy_geom():

The geom cannot possess any providers.

The geom cannot possess any consumers.

The topology lock has to be held.

RETURN VALUES

The **g_new_geomf**() function returns a pointer to the newly created geom.

EXAMPLES

Create an example geom.

```
static void
g_example_start(struct bio *bp)
{

        [...]
}
```

```
static void
g_example_orphan(struct g_consumer *cp)
{

        g_topology_assert();

        [...]
}

static void
g_example_spoiled(struct g_consumer *cp)
{

        g_topology_assert();

        [...]
}

static int
g_example_access(struct g_provider *pp, int dr, int dw, int de)
{

        [...]
}

static struct g_geom *
create_example_geom(struct g_class *myclass)
{
        struct g_geom *gp;

        g_topology_lock();
        gp = g_new_geomf(myclass, "example_geom");
        g_topology_unlock();
        gp->start = g_example_start;
        gp->orphan = g_example_orphan;
        gp->spoiled = g_example_spoiled;
        gp->access = g_example_access;
        gp->softc = NULL;

        return (gp);
```

```
}

static int
destroy_example_geom(struct g_geom *gp)
{

        g_topology_lock();
        if (!LIST_EMPTY(&gp->provider) ||
            !LIST_EMPTY(&gp->consumer)) {
                g_topology_unlock();
                return (EBUSY);
        }
        g_destroy_geom(gp);
        g_topology_unlock();

        return (0);
}
```

SEE ALSO

geom(4), DECLARE_GEOM_CLASS(9), g_access(9), g_attach(9), g_bio(9), g_consumer(9), g_data(9), g_event(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*>.