

**NAME**

**g\_new\_providerf**, **g\_destroy\_provider**, **g\_error\_provider** - GEOM providers management

**SYNOPSIS**

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

```
struct g_provider *
```

```
g_new_providerf(struct g_geom *gp, const char *fmt, ...);
```

```
void
```

```
g_destroy_provider(struct g_provider *pp);
```

```
void
```

```
g_error_provider(struct g_provider *pp, int error);
```

**DESCRIPTION**

A GEOM provider is the front gate at which a geom offers service. A provider is "a disk-like thing which appears in */dev*" - a logical disk in other words. All providers have three main properties: name, sectorsize and size.

The **g\_new\_providerf**() function creates a new provider on given geom *gp*. The name of the provider, which will appear as device in *devfs(5)*, is created in a *printf(3)*-like way from the rest of the arguments. After creation, the caller has to set the provider's *mediasize* and *sectorsize*, as well as other desired initializations, and then call **g\_error\_provider**() to reset the provider's error, which is initially set to *ENXIO*.

The **g\_destroy\_provider**() function destroys the given provider, cancels all related pending events and removes the corresponding *devfs* entry.

The **g\_error\_provider**() function is used to set the provider's error value. If set to a nonzero, all I/O requests will be denied, as well as increasing its access count will not be possible (error *error* will be returned).

**RESTRICTIONS/CONDITIONS**

**g\_new\_provider**():

The provider name should be unique, but this is not enforced by GEOM. If the name is not unique, one will end up with two (or more) files with the same name, which is a programmer error.

The geom *gp* has to have a *start* method defined.

The topology lock has to be held.

#### **g\_destroy\_provider():**

The provider must not have consumers attached.

The access count has to be 0.

The topology lock has to be held.

### **RETURN VALUES**

The **g\_new\_providerf()** function returns a pointer to the newly created provider.

### **EXAMPLES**

Create an example provider, set its parameters and make it usable.

```
struct g_provider *
create_example_provider(struct g_geom *gp)
{
    struct g_provider *pp;

    g_topology_lock();
    pp = g_new_providerf(gp, "example_provider");
    g_topology_unlock();
    pp->mediasize = 65536;
    pp->sectorsize = 512;
    g_error_provider(pp, 0);

    return (pp);
}
```

### **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\_geom(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)>.