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

Rocky Enterprise Linux 9.2 Manual Pages on command 'query_module.2'

$ man query_module.2

QUERY_MODULE(2)   Linux Programmer's Manual   QUERY_MODULE(2)

NAME

    query_module - query the kernel for various bits pertaining to modules

SYNOPSIS

    #include <linux/module.h>

    int query_module(const char *name, int which, void *buf,

            size_t bufsize, size_t *ret);

    Note: No declaration of this system call is provided in glibc headers;

    see NOTES.

DESCRIPTION

    Note: This system call is present only in kernels before Linux 2.6.

    query_module() requests information from the kernel about loadable mod?

    ules.   The returned information is placed in the buffer pointed to by

    buf. The caller must specify the size of buf in bufsize. The precise

    nature and format of the returned information depend on the operation

    specified by which. Some operations require name to identify a cur?

    rently loaded module, some allow name to be NULL, indicating the kernel

    proper.

The following values can be specified for which:

0   Returns success, if the kernel supports query_module(). Used to
      probe for availability of the system call.

QM_MODULES

      Returns the names of all loaded modules. The returned buffer

      consists of a sequence of null-terminated strings; ret is set to

      the number of modules.

QM_DEPS

      Returns the names of all modules used by the indicated module.

      The returned buffer consists of a sequence of null-terminated

      strings; ret is set to the number of modules.

QM_REFS

      Returns the names of all modules using the indicated module.

      This is the inverse of QM_DEPS. The returned buffer consists of

      a sequence of null-terminated strings; ret is set to the number

      of modules.

QM_SYMBOLS

      Returns the symbols and values exported by the kernel or the in‐

      dicated module.   The returned buffer is an array of structures

      of the following form

         struct module_symbol {

            unsigned long value;

            unsigned long name;

          };

      followed by null-terminated strings. The value of name is the

      character offset of the string relative to the start of buf; ret

      is set to the number of symbols.

QM_INFO

      Returns miscellaneous information about the indicated module.

      The output buffer format is:

         struct module_info {

            unsigned long address;

            unsigned long size;

```
        unsigned long flags;

    };

where address is the kernel address at which the module resides,

size is the size of the module in bytes, and flags is a mask  of

MOD_RUNNING,  MOD_AUTOCLEAN,  and so on, that indicates the cur?

rent status of the module (see the Linux kernel source file  in?

clude/linux/module.h).   ret  is  set  to  the  size of the mod?

ule_info structure.
```

RETURN VALUE

On success, zero is returned. On error, -1 is returned and errno is

set appropriately.

ERRORS

EFAULT At least one of name, buf, or ret was outside the program's ac?

cessible address space.

EINVAL Invalid which; or name is NULL (indicating "the kernel"), but

this is not permitted with the specified value of which.

ENOENT No module by that name exists.

ENOSPC The buffer size provided was too small. ret is set to the mini?

mum size needed.

ENOSYS query_module() is not supported in this version of the kernel

(e.g., the kernel is version 2.6 or later).

VERSIONS

This system call is present on Linux only up until kernel 2.4; it was

removed in Linux 2.6.

CONFORMING TO

query_module() is Linux-specific.

NOTES

Some of the information that was formerly available via query_module()

can be obtained from /proc/modules, /proc/kallsyms, and the files under

the directory /sys/module.

The query_module() system call is not supported by glibc. No declara?

tion is provided in glibc headers, but, through a quirk of history,

glibc does export an ABI for this system call. Therefore, in order to

employ this system call, it is sufficient to manually declare the in‐

terface in your code; alternatively, you can invoke the system call us‐

ing syscall(2).

SEE ALSO

create_module(2), delete_module(2), get_kernel_syms(2), init_module(2),

lsmod(8), modinfo(8)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project.   A

description of the project, information about reporting bugs, and the

latest   version   of   this   page,   can   be   found   at

https://www.kernel.org/doc/man-pages/.