## NAME

**vm_page_aflag_clear**, **vm_page_aflag_set**, **vm_page_reference** - change page atomic flags

## SYNOPSIS

**#include <sys/param.h>**
**#include <vm/vm.h>**
**#include <vm/vm_page.h>**

*void*
**vm_page_aflag_clear**(*vm_page_t m, uint8_t bits*);

*void*
**vm_page_aflag_set**(*vm_page_t m, uint8_t bits*);

*void*
**vm_page_reference**(*vm_page_t m*);

## DESCRIPTION

The **vm_page_aflag_clear**() atomically clears the specified bits on the page's *aflags*.

The **vm_page_aflag_set**() atomically sets the specified bits on the page's *aflags*.

The **vm_page_reference**(*m*) call is the same as

```
vm_page_aflag_set(m, PGA_REFERENCED);
```

and is the recommended way to mark the page as referenced from third-party kernel modules.

These functions neither block nor require any locks to be held around the calls for correctness.

The functions arguments are:

*m* The page whose *aflags* are updated.

*bits* The bits that are set or cleared on the page's flags.

The following *aflags* can be set or cleared:

*PGA_REFERENCED* The bit may be set to indicate that the page has been recently accessed. For instance, pmap(9) sets this bit to reflect the accessed attribute of the page

mapping typically updated by processor's memory management unit on the page access.

*PGA_WRITEABLE* A writeable mapping for the page may exist.

Both PGA_REFERENCED and PGA_WRITEABLE bits are only valid for the managed pages.

**AUTHORS**

This manual page was written by Chad David <*davidc@acns.ab.ca*>.