

NAME

`auplugin_fgets`, `auplugin_fgets_more`, `auplugin_fgets_eof`, `auplugin_fgets_clear`, `auplugin_setvbuf`, `auplugin_setvbuf_r` – buffered line reader helpers

SYNOPSIS

```
#include <auplugin.h>
```

```
int auplugin_fgets(char *buf, size_t blen, int fd);
int auplugin_fgets_more(size_t blen);
int auplugin_fgets_eof(void);
void auplugin_fgets_clear(void);
int auplugin_setvbuf(void *buf, size_t buff_size, enum auplugin_mem how);
int auplugin_setvbuf_r(auplugin_fgets_state_t *st, void *buf, size_t buff_size, enum auplugin_mem how);
```

DESCRIPTION

auplugin_fgets reads from *fd* into *buf* up to *blen* bytes or through the next newline. Text is accumulated across calls in an internal buffer so that complete lines can be returned. The string is NUL terminated.

auplugin_fgets_more checks whether the buffer holds a newline or at least *blen* - 1 bytes.

auplugin_fgets_eof indicates whether end of file was reached on *fd*

auplugin_fgets_clear resets the internal buffer and EOF state, discarding any stored text. When the memory type is **MEM_MMAP_FILE**, the buffer is rewound to the beginning making the entire file available again.

auplugin_setvbuf points the internal buffer at *buf* and sets how it will be released when **auplugin_fgets_destroy** is called. The *how* parameter should be one of

MEM_SELF_MANAGED - Internal buffer owned by **fgets**

MEM_MALLOC - User-supplied malloc buffer; **fgets** frees it

MEM_MMAP - User-supplied mmap buffer; **fgets** munmaps it

MEM_MMAP_FILE - Read-only mmap file data

The default is **MEM_SELF_MANAGED** which means its lifecycle is automatically handled. When **MEM_MMAP_FILE** is used, the buffer is treated as a preloaded buffer (the entire file) and no reads will be performed on the descriptor provided to **auplugin_fgets_r**. The reentrant form **auplugin_setvbuf_r** operates on an explicit state handle.

These functions maintain static state and are therefore not thread safe. For thread-safety, use the re-entrant variants.

RETURN VALUE

auplugin_fgets returns -1 on error, 0 when no data is available, or the number of characters copied otherwise.

auplugin_fgets_more and **auplugin_fgets_eof** return 1 for true and 0 for false.

auplugin_fgets_clear returns no value.

auplugin_setvbuf returns 0 on success and 1 on failure.

BACKGROUND

The reason that this family of functions was created is because in `auditd` plugins, the event stream is `stdin`, which is descriptor 0. A typical pattern is to call `select`, `poll`, or `epoll` to wait for a record to arrive. As soon as it does, you need to read it. If you use `fgets`, you will wind up with big problems because you cannot mix low level descriptors with high level constructs like `struct FILE`. This family of functions allows you to correctly work only using descriptors but with the convenience of `fgets`.

SEE ALSO
fgets(3)