

**NAME**

**properties\_read**, **property\_find**, **properties\_free** - functions to allow creating simple property lists from ASCII file data

**LIBRARY**

System Utilities Library (libutil, -lutil)

**SYNOPSIS**

```
#include <libutil.h>
```

```
properties
```

```
properties_read(int fd);
```

```
char *
```

```
property_find(properties list, const char *name);
```

```
void
```

```
properties_free(properties list);
```

**DESCRIPTION**

```
typedef struct _properties {  
    struct _properties *next;  
    char *name;  
    char *value;  
} *properties;
```

The function **properties\_read**() reads *name = value* pairs from the file descriptor passed in *fd* and returns the head of a new property list, assuming that the file's contents have been parsed properly, or NULL in case of error.

The **property\_find**() function returns the associated value string for the property named *name* if found, otherwise NULL. The value returned may be up to PROPERTY\_MAX\_VALUE bytes in length.

The **properties\_free**() function is used to free the structure returned by **properties\_read**() when it is no longer needed.

**FILE FORMAT**

Each property in the file is assumed to have the format of *name = value* where *name* is an alphanumeric string (and any punctuation not including the '=' character) and *value* is an arbitrary string of text terminated by a newline character. If newlines are desired, the entire value should be enclosed in { }

(curly-bracket) characters. Any line beginning with a # or ; character is assumed to be a comment and will be ignored.

**AUTHORS**

Jordan Hubbard

**BUGS**

Simplistic.