

**misc**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> misc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 28, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>misc</b>	<b>1</b>
1.1	misc.doc . . . . .	1
1.2	misc.resource/AllocMiscResource . . . . .	1
1.3	misc.resource/FreeMiscResource . . . . .	2

# Chapter 1

## misc

### 1.1 misc.doc

```
AllocMiscResource()  
FreeMiscResource()
```

### 1.2 misc.resource/AllocMiscResource

#### NAME

AllocMiscResource - allocate one of the miscellaneous resources

#### SYNOPSIS

```
CurrentUser = AllocMiscResource( unitNum, name )  
D0                                D0          A1
```

```
char * AllocMiscResource(ULONG, char *);
```

#### FUNCTION

This routine attempts to allocate one of the miscellaneous resources. If the resource had already been allocated, an error is returned. If you do get it, your name is associated with the resource (so a user can see who has it allocated).

This function may not be called from interrupt code

#### DESCRIPTION

There are certain parts of the hardware that a multitasking- friendly program may need to take over. The serial port is a good example. By grabbing the misc.resource for the serial port, the caller would "own" the hardware registers associated with that function. Nobody else, including the system serial driver, is allowed to interfere.

Resources are called in exactly the same manner as libraries. From assembly language, A6 must equal the resource base. The offsets for the function are listed in the resources/misc.i include file (MR\_ALLOCMISCRESOURCE for this function).

#### INPUTS

---

unitNum - the number of the resource you want to allocate  
(eg. MR\_SERIALBITS).  
name - a mnemonic name that will help the user figure out  
what piece of software is hogging a resource.  
(havoc breaks out if a name of null is passed in...)

#### RESULTS

CurrentUser - if the resource is busy, then the name of  
the current user is returned. If the resource is  
free, then null is returned.

#### BUGS

#### SEE ALSO

resources/misc.i, FreeMiscResource()

### 1.3 misc.resource/FreeMiscResource

#### NAME

FreeMiscResource - make a resource available for reallocation

#### SYNOPSIS

```
FreeMiscResource( unitNum )  
                DO
```

```
void FreeMiscResource(ULONG);
```

#### FUNCTION

This routine frees one of the resources allocated  
by AllocMiscResource. The resource is made available  
for reuse.

FreeMiscResource must be called from the same task that  
called AllocMiscResource. This function may not be called from  
interrupt code.

#### INPUTS

unitNum - the number of the miscellaneous resource to be freed.

#### RESULTS

Frees the appropriate resource.

#### BUGS

#### SEE ALSO

resources/misc.i, AllocMiscResource()

---