

**narrator**

COLLABORATORS

	TITLE : narrator		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 18, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>narrator</b>	<b>1</b>
1.1	narrator.doc . . . . .	1
1.2	narrator.device/AbortIO . . . . .	1
1.3	narrator.device/CloseDevice . . . . .	1
1.4	narrator.device/CMD_FLUSH . . . . .	2
1.5	narrator.device/CMD_Read . . . . .	2
1.6	narrator.device/CMD_RESET . . . . .	4
1.7	narrator.device/CMD_START . . . . .	4
1.8	narrator.device/CMD_STOP . . . . .	5
1.9	narrator.device/CMD_WRITE . . . . .	5
1.10	narrator.device/OpenDevice . . . . .	6

# Chapter 1

## narrator

### 1.1 narrator.doc

AbortIO()	CMD_READ	CMD_STOP
OpenDevice()	CMD_RESET	CMD_WRITE
CMD_FLUSH	CMD_START	CloseDevice()

### 1.2 narrator.device/AbortIO

#### NAME

AbortIO - Abort an IO request

#### SYNOPSIS

```
AbortIO(IORequest)
A1
```

#### FUNCTION

Exec library call to abort a specified READ or WRITE request. The IORequest may be in the queue or currently active. If currently active, the request is immediately stopped and then removed.

#### INPUTS

Pointer to the IORequest block to be aborted.

#### RESULTS

io\_Error field in the IORequest block set to #IOERR\_ABORTED.

#### SEE ALSO

### 1.3 narrator.device/CloseDevice

#### NAME

CloseDevice - terminates access to the narrator device

#### SYNOPSIS

---

```
CloseDevice(IORequest)
    A1
```

#### FUNCTION

Close invalidates the IO\_UNIT and IO\_DEVICE fields in the IORequest block, preventing subsequent IO until another OpenDevice. CloseDevice also reduces the open count. If the count goes to 0 and the expunge bit is set, the device is expunged. If the open count goes to zero and the delayed expunge bit is not set, CloseDevice sets the expunge bit.

#### INPUTS

A valid IORequest block with its io\_Message structure, and io\_Device and io\_Unit fields properly initialized. These fields are initialized by OpenDevice.

#### RESULTS

CloseDevice invalidates the unit and device pointers in the IORequest block.

#### SEE ALSO

## 1.4 narrator.device/CMD\_FLUSH

#### NAME

CMD\_FLUSH - Aborts all inprogress and queued requests

#### SYNOPSIS

Standard device command.

#### FUNCTION

Aborts all inprogress and queued speech requests.

#### INPUTS

Valid IORequest block with the io\_Command field set to CMD\_FLUSH. A valid IORequest block is one with its io\_Message structure, and io\_Device and io\_Unit fields properly initialized. The easiest way to insure proper initialization is to make a copy of the IORequest block after a successful OpenDevice call.

#### RESULTS

io\_Error in IORequest block set to 0

#### SEE ALSO

Exec input/output documentation.

## 1.5 narrator.device/CMD\_Read

#### NAME

CMD\_READ - Query the narrator device for mouth shape or other synchronization events.

---

## SYNOPSIS

Standard device command.

## FUNCTION

Currently, there are three events which the user can inquire about from the narrator device. These are: mouth shape changes, start of word, and start of syllable. Each read request returns information about any or all of these events as determined by the bits set in the sync field of the read IORequest block. In the case of mouth shape changes, each shape returned is guaranteed to be different from the previously returned shape to allow updating to be done only when necessary. Each read request is associated with a write request by information contained in the IORequest block used to open the device. Since the first field in the read IORequest block is a write IORequest structure, this association is easily made by copying the write IORequest block (after the OpenDevice call) into the voice field of the read IORequest block. If there is no write in progress or in the device input queue with the same pseudo unit number as the read request, the read will be returned to the user with an error. This is also how the user knows that the write request has finished and that s/he should not issue any more reads. Note that in this case the mouth shapes may not be different from previously returned values.

## INPUTS

mouth\_rb IORequest block with the voice field (a narrator\_rb structure) copied from the associated write request with the following fields modified:

- io\_Message - Pointer to message port for read request
- io\_Command - CMD\_READ
- io\_Error - Clear before issuing first read
- width - 0
- height - 0

## RESULTS

As long as the speech is in progress, each read returns the following information in the mouth\_rb IORequest block.

If mouth shape changes are requested the following fields are modified:

- width - Contains mouth width value in arbitrary units
- height - Contains mouth height value in arbitrary units
- shape - Compressed form of mouth shapes (internal use only)

\*\*\*\*\* NEW FOR V37 NARRATOR

If word synchronization is requested:

- sync - Bit NDB\_WORDSYNC is set

If syllable synchronization is requested:

- sync - Bit NDB\_SYLSYNC is set

Note that any or all of the above fields can be set and it is the user's responsibility to check for all possibilities.

## SEE ALSO

CMD\_WRITE  
Exec input/output documentation.

## 1.6 narrator.device/CMD\_RESET

NAME  
CMD\_RESET - Reset the device to a known state

SYNOPSIS  
Standard device command.

FUNCTION  
Resets the device as though it has just be initialized.  
Aborts all read/write requests whether active of enqueued.  
Restarts device if it has been stopped.

INPUTS  
Valid IORequest block with the io\_Command field set to CMD\_RESET.  
A valid IORequest block is one with its io\_Message structure, and  
io\_Device and io\_Unit fields properly initialized. The easiest  
way to insure proper initialization is to make a copy of the  
IORequest block after a successful OpenDevice call.

RESULTS

SEE ALSO  
Exec input/output documentation.

## 1.7 narrator.device/CMD\_START

NAME  
CMD\_START - Restarts the device after a CMD\_STOP command

SYNOPSIS  
Standard device command.

FUNCTION  
CMD\_START restarts the currently active speech (if any) and  
allows queued requests to start.

INPUTS  
Valid IORequest block with the io\_Command field set to CMD\_START  
A valid IORequest block is one with its io\_Message structure, and  
io\_Device and io\_Unit fields properly initialized. The easiest  
way to insure proper initialization is to make a copy of the  
IORequest block after a successful OpenDevice call.

RESULTS  
io\_Error set to 0.

SEE ALSO  
Exec input/output documentation.

---

## 1.8 narrator.device/CMD\_STOP

### NAME

CMD\_STOP - Stops the device.

### SYNOPSIS

Standard device command.

### FUNCTION

CMD\_STOP halts the currently active speech (if any) and prevents any queued requests from starting.

### INPUTS

Valid IORequest block with the io\_Command field set to CMD\_STOP. A valid IORequest block is one with its io\_Message structure, and io\_Device and io\_Unit fields properly initialized. The easiest way to insure proper initialization is to make a copy of the IORequest block after a successful OpenDevice call.

### RESULTS

io\_Error set to 0.

### SEE ALSO

Exec input/output documentation.

## 1.9 narrator.device/CMD\_WRITE

### NAME

CMD\_WRITE - Send speech request to the narrator device

### SYNOPSIS

Standard device command.

### FUNCTION

Sends a phonetic string to the narrator device to be spoken and, optionally, is used to direct the narrator device to return mouth shape changes, and word and syllable sync events in response to read requests from the user. The phonetic string consists of ASCII characters representing the individual phonemes. Refer to the narrator device chapter of the libraries and devices volume of the ROM Kernel Manual for detailed information.

### INPUTS

User IORequest block (struct narrator\_rb as defined in .h file). The OpenDevice call will initialize the IORequest block to a "standard male" voice. If you want to change any parms, do so after the OpenDevice call and before the DoIO (or SendIO/WaitIO). For a complete description of the narrator\_rb structure, see the narrator.h or .i include file. Note that the OpenDevice call does not initialize all the fields needed by the narrator device. The IORequest fields which must be set by the user before issuing the write request are:

io\_Command - Set to CMD\_WRITE

---



```

io_Data      - Pointer to phonetic string
io_Length    - Length of phonetic string
ch_masks     - Array of audio channel selection masks (see audio
               device documentation for description of this field)
nm_masks     - Number of audio channel selection masks

```

\*\*\*\*\* NEW FOR V37 NARRATOR

```

flags - The bit NDB_NEWIORB must be set in the flags field if
        any of the new features of the V37 narrator are used

```

In addition to producing synthetic speech, the narrator device also provides features for synchronizing the speech to animation or other user defined events. There are three types of events that the user can request. They are mouth shape changes, start of new word, and start of new syllable. Mouth shape changes are requested by setting the mouths field of the IORequest block to a non-zero value. Word and syllable sync events are requested by setting the NDB\_WORDSYNC and/or NDB\_SYLSYNC bits in the flags field of the IORequest block. Note that word and syllable sync only work in V37 and later versions of the narrator device.

#### RESULTS

The narrator device range checks and performs other validity checks for all input parms. If any input is in error, the device sets the io\_Error field of the IORequest block to an appropriate value (see include files for error codes). If everything is in order, the narrator device will produce the speech and clear the io\_Error field. The io\_Actual field is set to the length of the input string that was actually processed. If the return code indicates a phoneme error (ND\_PhonErr), io\_Actual is the NEGATIVE of the position in the input string where the error occurred.

#### SEE ALSO

```

Read command.
Audio device documentation.
Exec input/output documentation.

```

## 1.10 narrator.device/OpenDevice

#### NAME

OpenDevice - opens the narrator device.

#### SYNOPSIS

```

error = OpenDevice("narrator.device",  unit, IORequest, flags);
      D0                      A0          D0      A1          D1

```

#### FUNCTION

The OpenDevice routine grants access to the narrator device. OpenDevice checks the unit number, and if non-zero, returns an error (ND\_UnitErr). If this is the first time the driver has been opened, OpenDevice will attempt to open the audio device and allocate the driver's static buffers. If either of these operations fail, an error is returned. See the .h and .i include files for possible error return codes. Next, OpenDevice

(done for all opens, not just the first one) initializes various fields in the user's IOResult block (see below). If users wish to use non-default values for these parms, the values must be set after the open is done. OpenDevice also assigns a pseudo unit number to the IORB for use in synchronizing read and write requests. See the read command for more details. Finally, OpenDevice stores the device node pointer in the IOResult block and clears the delayed expunge bit.

\*\*\*\*\* NEW FOR V37 NARRATOR \*\*\*\*\*

Several new fields in the IOResult block have been added for V37 narrator. These fields are initialized when the device is opened if the NDB\_NEWIORB bit is set in the flags field of the user's IOResult block. Note that NDB\_NEWIORB is set in the IOResult block, NOT in the "flags" input parm to the OpenDevice call.

#### INPUTS

```
device      - "narrator.device"
unit        - 0
IOResult    - Pointer to the user's IOResult block
flags       - 0
```

#### RESULTS

The narrator device will initialize the IOResult block as follows (assume IORB points to the IOResult block):

```
IORB->rate = 150;          /* Speaking rate in words/minute */
IORB->pitch = 110;         /* Baseline pitch in Hertz      */
IORB->mode = NATURALF0;    /* Pitch (F0) mode              */
IORB->sex = MALE;          /* Sex of voice                  */
IORB->volume = 64          /* Volume, full on              */
IORB->sampfreq = 22200     /* Audio sampling freq          */
IORB->mouths = 0           /* Don't generate sync events   */
```

and if the NDB\_NEWIORB bit is set:

```
IORB->F0enthusiasm = 0     /* F0 excursion factor          */
IORB->F0perturb = 32       /* F0 perturbation (in 32nds)   */
IORB->F1adj = 0            /* F1 adjustment in \ensuremath{\pm}5% steps ←
*/
IORB->F2adj = 0            /* F2 adjustment in \ensuremath{\pm}5% steps ←
*/
IORB->F3adj = 0            /* F3 adjustment in \ensuremath{\pm}5% steps ←
*/
IORB->A1adj = 0            /* A1 adjustment in decibels    */
IORB->A2adj = 0            /* A2 adjustment in decibels    */
IORB->A3adj = 0            /* A3 adjustment in decibels    */
IORB->articulate = 100     /* Transition time multiplier   */
IORB->centralize = 0       /* Degree of vowel centralization */
IORB->centphon = ""        /* Pointer to central ASCII phon */
IORB->AVbias = 0           /* AV bias                      */
IORB->AFbias = 0           /* AF bias                      */
IORB->priority = 100       /* Priority while speaking      */
```

#### SEE ALSO

The include files contain the complete IOResult block definition,

---

default settings, and error return codes.  
Exec input/output documentation.

---