

**translator**

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

	<i>TITLE :</i> translator		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 29, 2025	

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

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>translator</b>	<b>1</b>
1.1	translator.doc . . . . .	1
1.2	translator.library/Translate . . . . .	1

# Chapter 1

## translator

### 1.1 translator.doc

Translate()

### 1.2 translator.library/Translate

#### NAME

Translate -- Convert an English string into narrator device phonemes.

#### SYNOPSIS

```
rtnCode = Translate(inString, inLength, outBuffer, outLength)
D0                      A0          D0          A1          D1
```

```
LONG Translate( STRPTR inString, LONG inLength, STRPTR outBuffer,
               LONG outlen );
```

#### FUNCTION

The translate function converts an English string into a string of phonetic codes suitable as input to the narrator device.

#### INPUTS

inString - pointer to English string  
inLength - length of English string  
outBuffer - a char array which will hold the phonetic codes  
outLength - the length of the output array

#### RESULTS

rtnCode - zero if no error has occurred.

The only error that can occur is overflowing the outBuffer. If Translate() determines that an overflow will occur, it will stop the translation at a word boundary before the overflow happens. If this occurs, rtnCode will be a negative number whose absolute value indicates where in inString Translate() stopped. The user can then use the offset -rtnCode from the beginning of inString in a subsequent Translate() call to continue the translation.

---

SEE ALSO  
narrator.device/CMD\_WRITE