OpenVMS RTL DECtalk (DTK$) Manual

Order Number: AA–PS6CA–TE

May 1993

This manual documents the DECtalk support routines contained in the DTK$ facility of the VMS Run-Time Library.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

Index

Preface

This manual provides users of the VMS operating system with detailed usage and reference information on DECtalk support routines supplied in the DTK$ facility of the Run-Time Library.

Run-Time Library routines can only be used in programs written in languages that produce native code for the VAX hardware. At present, these languages include VAX MACRO and the following compiled high-level languages:

VAX Ada
VAX BASIC
VAX BLISS-32
VAX C
VAX COBOL
VAX COBOL-74
VAX CORAL
VAX DIBOL
VAX FORTRAN
VAX Pascal
VAX PL/I
VAX RPG
VAX SCAN

Interpreted languages which can also access Run-Time Library routines include VAX DSM and DATATRIEVE.

Intended Audience

This manual is intended for system and applications programmers who want to call Run-Time Library routines.

Document Structure

This manual is organized into two parts as follows:

- Chapter 1 provides guidelines on using the DTK$ DECtalk routines.

- Part II provides detailed reference information on each DECtalk support routine contained in the DTK$ facility of the Run-Time Library. This information is presented using the documentation format described in the *OpenVMS Programming Interfaces: Calling a System Routine*. Routine descriptions appear in alphabetical order by routine name.

Associated Documents

The Run-Time Library routines are documented in a series of reference manuals. This manual provides an overview of the VMS Run-Time Library DECtalk (DTK$) facility, along with detailed information on each DTK$ routine. A general overview of the VMS Run-Time Library is presented in the *OpenVMS Programming Interfaces: Calling a System Routine*. Descriptions of the other RTL facilities and their corresponding routines and usages are discussed in the following books:

- The *OpenVMS RTL Library (LIB$) Manual*

- The *OpenVMS VAX RTL Mathematics (MTH$) Manual*

- The *OpenVMS RTL General Purpose (OTS$) Manual*

- The *OpenVMS RTL Parallel Processing (PPL$) Manual*

- The *OpenVMS RTL Screen Management (SMG$) Manual*

- The *OpenVMS RTL String Manipulation (STR$) Manual*

The VAX Procedure Calling and Condition Handling Standard, which is documented in the *OpenVMS Programming Interfaces: Calling a System Routine*, contains useful information for anyone who wants to call Run-Time Library routines.

Applications programmers of any language may refer to the *Guide to Creating OpenVMS Modular Procedures* for the Modular Programming Standard and other guidelines.

High-level language programmers will find additional information about calling Run-Time Library routines in their language reference manual. Additional information may also be found in the language user's guide provided with your VAX language.

The *OpenVMS User's Manual* may also be useful.

For a complete list and description of the manuals in the VMS documentation set, see the *Overview of OpenVMS Documentation*.

Conventions

In this manual, every use of VMS means both the OpenVMS AXP and the OpenVMS VAX operating system.

The following conventions are used in this manual:

Ctrl/*x*	A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 *x*	A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1, then press and release another key or a pointing device button.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)

. . .	A horizontal ellipsis in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.
[ ]	In format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in a VMS file specification, or in the syntax of a substring specification in an assignment statement.)
{ }	In format descriptions, braces surround a required choice of options; you must choose one of the options listed.
boldface text	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason. Boldface text is also used to show user input in online versions of the manual.
italic text	Italic text emphasizes important information, indicates variables, and indicates complete titles of manuals. Italic text also represents information that can vary in system messages (for example, Internal error *number*), command lines (for example, /PRODUCER=*name*), and command parameters in text.
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
-	A hyphen in code examples indicates that additional arguments to the request are provided on the line that follows.
numbers	All numbers in text are assumed to be decimal, unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

Other conventions used in the documentation of Run-Time Library routines are described in the *OpenVMS Programming Interfaces: Calling a System Routine*.

1
DECtalk Guidelines

This section discusses the Run-Time Library routines that control and perform various functions on DIGITAL's DECtalk device. The DECtalk routines are meant solely for use with DIGITAL's DECtalk device.

1.1 Overview of the DTK$ Facility

The DECtalk device accepts alphanumeric text from a computer system and converts it to human-quality speech. DECtalk speaks this data through its internal speaker, an external audio system, or over the telephone.

The DTK$ Facility consists of routines that control the functions of the DECtalk device. These routines not only control DECtalk's characteristics, but they also provide functional controls such as reading keys entered on a phone keypad or hanging up the phone. In general, the DTK$ routines supply a simple interface between the user and the DECtalk device.

1.2 Controlling the DECtalk Environment

The DTK$ facility supplies several routines that control the environment in which the DECtalk device functions. These routines are grouped according to function and described in the following subsections.

1.2.1 Initializing DECtalk

Before issuing any commands to the DECtalk device, you must first initialize it. The routine DTK$INITIALIZE initializes a DECtalk device and returns its associated voice identifier. This routine also specifies a file specification or logical name to which the output associated with the DECtalk device is written.

1.2.2 The DECtalk Device

Once a DECtalk device has been initialized, you can select various operating modes for the device. One of these modes involves the specification of phonemic text; a phoneme is one of the smallest possible units of speech that can be used to distinguish one word from another.

DTK$SET_MODE enables you to select any operating mode for DECtalk. The DTK$SET_MODE routine controls how DECtalk processes text, according to the modes in the following list. Note that it is possible to perform a logical OR operation to set more than one mode at a time, and any mode that is not specified is reset.

- DTK$M_SQUARE treats square brackets ([]) as phonemic text delimiters. The DTK$M_SQUARE mode also lets the close parenthesis character ")", when it appears before a word, indicate an alternative pronunciation from the built-in dictionary.

- DTK$M_ASCII uses single-character phonemic translation instead of a multiletter translation. This is valid only for the DTC01 device.

- DTK$M_MINUS pronounces a hyphen (–) as "minus."

- DTK$M_EUROPE uses European number format (where a comma separates the number from its decimal portion and periods mark off digits into groups of 3). This is valid only for the DTC03 device.

- DTK$M_SPELL spells out all text instead of interpreting the text as words. This is valid only for the DTC03 device.

DTK$SET_MODE can be used to set new modes, to examine the current mode settings, or to change the modes and later return them to their previous state.

The DTK$SET_SPEECH_MODE routine toggles DECtalk's speech on and off. In the case of turning the speech off, you can either cause DECtalk to stop speaking immediately or to stop speaking when the current text is completed.

DTK$SET_VOICE controls the actual voice characteristics of the DECtalk device. The following is a list of the available voice mode settings for DECtalk, along with their characteristics.

Voice Mode	Characteristics
DTK$K_VOICE_MALE	Standard male voice
DTK$K_VOICE_FEMALE	Standard female voice
DTK$K_VOICE_CHILD	Standard child voice
DTK$K_VOICE_DEEP_MALE	Deep male voice
DTK$K_VOICE_DEEP_FEMALE	Deep female voice
DTK$K_VOICE_OLDER_MALE	Older male voice
DTK$K_VOICE_LIGHT_FEMALE	Light female voice

The DTK$SET_VOICE routine also enables you to control DECtalk's speech rate, as well as the length of the pauses after commas and periods. The speech rate of the DECtalk device is measured in words per minute, and the valid range is 120 to 350 words per minute. The values that control the comma and period pauses are measured in milliseconds.

1.2.3 The Terminal

The DECtalk device enables you to connect a local terminal to the DECtalk device. The DTK$ facility supplies two routines that you can use to control the local terminal.

The DTK$SET_LOGGING_MODE routine determines the information that is displayed on the local terminal connected to the DECtalk device. The following is a list of modes that can be set or reset using DTK$SET_LOGGING_MODE. Note that you can perform a logical OR operation to set more than one mode at a time, and any mode not specified is reset.

- DTK$M_TEXT causes all spoken text to be printed on the terminal.

- DTK$M_PHONEME causes all spoken text to be converted to its phonemic equivalent and then printed on the terminal.

- DTK$M_RAWHOST causes all data received from the host to be transmitted, including escape sequences, to the terminal in exactly the form in which it is received.

- DTK$M_INHOST causes all text received from the host to be printed, including escape sequences, on the terminal. However, all control characters are first translated to a readable form.

- DTK$M_OUTHOST causes all characters sent to the host to be printed at the terminal. Before printing, control characters are converted to a readable form. This readable form is sometimes a mnemonic (for example, ESC for escape) and sometimes a letter that has meaning when used with the CONTROL key (for example, C for CTRL/C).

- DTK$M_ERROR causes all DECtalk error messages to be printed on the terminal.

- DTK$M_TRACE causes all text received from the host to be printed on the terminal, including escape sequences. However, the escape sequences are first converted into their symbolic meaning before being printed.

The routine DTK$SET_TERMINAL_MODE controls the attributes of the local terminal. This routine can be used to set new modes, to examine the current mode settings, or to change the modes and later return them to their previous state. All of the following modes are valid.

DTK$M_HOST	When set, any characters typed on the local terminal are transmitted to the host computer.
DTK$M_SPEAK	When set, characters typed on the local terminal are spoken by DECtalk. (Note that they will be interspersed with characters sent from the host.)
DTK$M_EDITED	When set, text entered from the local terminal is processed one line at a time.
DTK$M_HARD	When set, text is echoed on the local terminal in a manner appropriate for hardcopy terminals. When this mode is not set, echoing is performed in a manner appropriate for video display terminals.
DTK$M_SETUP	When set, DECtalk speaks the set-up dialogue.
DTK$M_FILTER	When set, DECtalk does not transmit DECtalk-specific escape sequences to the local terminal. DECtalk also ignores most non-DECtalk escape sequences transmitted from the host computer.

Once again, it is possible to perform a logical OR operation to set more than one mode at a time. Any mode that is not specified is automatically reset.

1.2.4 The Index

When DECtalk speaks text, the speech rate for the device is much slower than the rate of data transmission. This may create a problem if a certain task in your program is contingent upon certain speech being complete. For example, it is quite likely that the DECtalk device will still be speaking some text while your program is already executing the next statement. The DTK$ facility therefore provides you with indexes. An index is a number that you insert at an appropriate location in your text. Once DECtalk finishes speaking all text up to that index, the index value is returned to your program. Using this index, your program can then keep track of text that has already been spoken.

An index could be inserted, for example, following some instructional text regarding user input. Once DECtalk speaks the instructional text, it returns the index to the program, and this triggers the program to wait for the user's input. You can set an index using the DTK$SET_INDEX routine. The index value must be in the range of 1 to 32767. The DTK$RETURN_LAST_INDEX routine returns the last index spoken.

1.2.5 The Dictionary

DECtalk comes equipped with two dictionaries. The first of these is stored in ROM and therefore cannot be changed in any way. A certain amount of dynamic storage is allocated for a supplemental dictionary to which you can add your own words and pronunciations.

To load a word into this supplemental dictionary, you use the DTK$LOAD_ DICTIONARY routine. By specifying both the actual spelling of the word and its phonemic definition, you create a new entry in the supplemental dictionary.

1.2.6 Terminating DECtalk

To deassign the voice identifier that was assigned to the DECtalk device when it was initialized, you must call DTK$TERMINATE. This routine terminates all use of the specified DECtalk device by deallocating the voice control block and all of its substructures.

1.3 Controlling DECtalk's Speech

The DTK$ facility provides three different methods for specifying the text that DECtalk is to speak.

DTK$SPEAK_TEXT sends the specified text to the DECtalk device to be spoken. You can optionally set one of the following modes:

- DTK$K_IMMED returns control to the user immediately (this is the default).

- DTK$K_WAIT waits until the text is completely spoken before returning control to the user.

- DTK$K_STATUS waits until the text is completely spoken, and then returns a phone status.

These modes can be set for DTK$SPEAK_FILE, which speaks the text contained in a specified file, and also for DTK$SPEAK_PHONEMIC_TEXT. This routine sends the specified phonemic text to the DECtalk device to be spoken. Phonemic text contains the phonemic representations of the words to be spoken; that is, the words are spelled as they are pronounced.

1.4 Controlling DECtalk for Telephone Use

One of the most common applications for DECtalk is to serve as an access to a remote database via the telephone. The DTK$ facility therefore supplies several routines that improve the interface between DECtalk and the user.

1.4.1 The Telephone Keypad

The routine DTK$SET_KEYPAD_MODE enables and disables DECtalk's recognition of the telephone keypad. If keypad recognition is enabled with autostop, DECtalk stops speaking when a terminator is entered. Otherwise, keypad recognition is disabled immediately.

1.4.2 Input Using the Telephone Keypad

The routine DTK$READ_KEYSTROKE reads a key entered on the telephone keypad and returns that key's equivalent key code value. These key codes are in the form DTK$K_TRM_*xxxx*, and they are defined in the DIGITAL supplied library $DTKDEF. The optional prompt argument enables you to specify some text that DECtalk speaks before waiting for input. The timeout argument controls the number of seconds that DECtalk waits for input.

DTK$READ_STRING reads a series of keys entered on a telephone keypad and returns the key series as a single string. Again, you can specify an optional prompt and a timeout value for this routine. The optional termination code argument returns the key code value of the terminating key entered.

1.4.3 Controlling the Telephone Functions

In addition to accepting user input, the DTK$ facility also supplies routines that control the actual operation of the telephone.

DTK$ANSWER_PHONE waits for the phone connected to the DECtalk device to ring and then answers it. You can specify optional text that DECtalk speaks after answering the phone. The DTK$ facility also supplies a routine for hanging up the phone: DTK$HANGUP_PHONE. This routine speaks an optional message (if specified) and then hangs up the phone. DTK$DIAL_PHONE dials the specified number on the telephone. You can select either pulse dialing or tone dialing with the mode argument for this routine, and you can specify how many seconds DECtalk should wait for the phone to be answered. The text argument enables you to specify optional text that DECtalk speaks after the phone is answered.

1.5 Exit Handlers

The DTK$ facility supplies its own exit handler to terminate access to the DECtalk device. This exit handler hangs up the phone and resets any terminal characteristics changed by the DTK$ facility. Users must not call any of the DTK$ routines from within their own exit handlers.

DTK$ Reference Section

This section provides detailed discussions of the routines provided by the VMS RTL DECtalk (DTK$) Facility.

DTK$ANSWER_PHONE—Wait for Phone to Ring and Answer

The Wait for Phone to Ring and Answer routine waits for the phone connected to the DECtalk device to ring and then answers it.

Format

DTK$ANSWER_PHONE   voice-id [,number-of-rings] [,text] [,timeout]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

number-of-rings
VMS Usage: longword_signed
type: longword (signed)
access: read only
mechanism: by reference

Number of rings DECtalk waits for before answering the phone. The optional **number-of-rings** argument is the address of a signed longword containing this number. The default is 1 ring.

text
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Text that DECtalk speaks after answering the phone. The optional **text** argument is the address of a descriptor pointing to the text.

timeout
VMS Usage: longword_signed
type: longword (signed)
access: read only
mechanism: by reference

Number of seconds that DECtalk allows the phone to ring before answering. The optional **timeout** argument is the address of a signed longword containing this timeout value. There is no default value.

DTK$ANSWER_PHONE

Description

DTK$ANSWER_PHONE waits for the phone connected to the DECtalk device to ring and then answers it. If the **number-of-rings** argument is not specified, DECtalk answers the phone after 1 ring. If both the **number-of-rings** argument and the **timeout** argument are specified, DECtalk answers the phone after the first argument occurs. For example, if you specify 2 rings and 30 seconds, DECtalk will answer after 2 rings. After the phone is answered, DECtalk speaks any optional text that you specified.

Condition Values Returned

RMS$_NORMAL	Normal successful completion.
SS$_xxxx	Any condition value returned by $QIOW.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.

DTK$CHECK_HDWR_STATUS—Check Hardware Status

The Check Hardware Status routine checks the DECtalk hardware for hardware malfunctions.

Format

DTK$CHECK_HDWR_STATUS   voice-id ,hdwr-status

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

hdwr-status
VMS Usage: mask_longword
type: longword (unsigned)
access: write only
mechanism: by reference

Receives the hardware status of the DECtalk machine. The **hdwr-status** argument is the address of an unsigned longword bit mask that receives the status.

Valid values for **hdwr-status** are:

DTK$_NOMALFUN1	No malfunctions on first test
DTK$_NOMALFUN2	No malfunctions on second test
DTK$_COMFAIL	Communication failure
DTK$_INPBUFOVR	Input buffer overflow
DTK$_NVROPRFAI	NVR operation failed
DTK$_ERRPHOTRA	Phonemic transmission error
DTK$_CONSEQERR	Control sequence error
DTK$_DECTSFAI	Self-test failed

DTK$CHECK_HDWR_STATUS

Description

The DTK$CHECK_HDWR_STATUS routine checks the DECtalk hardware (described by the **voice**-**id** argument) for hardware malfunctions and returns the status to the caller.

If more than one hardware malfunction occurs, you can invoke the DTK$CHECK_HDWR_STATUS routine as many times as necessary to retrieve all error status codes. A status of "no malfunctions" indicates that there are no further error status codes to be retrieved.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIOW.
DTK$_INVVOIID	Invalid **voice**-**id**.
DTK$_WRONUMARG	Wrong number of arguments.
LIB$_xxxx	Any error from LIB$GET_VM or LIB$FREE_VM.

DTK$DIAL_PHONE—Dial the Telephone

The Dial the Telephone routine dials the specified number on the telephone.

Format

DTK$DIAL_PHONE   voice-id ,phone-number [,dial-mode] [,text] [,timeout]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk machine. The **voice-id** argument is the address of
an unsigned longword containing this identifier. The voice identifier is returned
by the DTK$INITIALIZE routine.

phone-number
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Phone number to dial. The **phone-number** argument is the address of a
descriptor pointing to the specified phone number.

dial-mode
VMS Usage: longword_unsigned
type: longword (unsigned)
access: read only
mechanism: by reference

Mode to use when dialing the phone. The optional **dial-mode** argument is the
address of an unsigned longword containing this mode.

The valid modes are:

DTK$K_DIAL_PULSE   Use pulse dialing.
DTK$K_DIAL_TONE   Use tone dialing.

Pulse dialing is the default. Tone dialing requires a touch tone telephone; there
are, however, touch tone phones that have a pulse dialing mode.

DTK$DIAL_PHONE

text

VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Text to be spoken after the phone is answered. The **text** argument is the address of a descriptor pointing to the specified text.

timeout

VMS Usage: longword_signed
type: longword (signed)
access: read only
mechanism: by reference

Number of seconds to wait for the phone to be answered. The optional **timeout** argument is the address of a signed longword containing this timeout value. If omitted, DECtalk dials the phone and immediately returns control to the calling program.

The valid range for this argument is 10 to 120 seconds, inclusive.

Description

DTK$DIAL_PHONE dials the specified number on the telephone. If a call is currently active, DECtalk does not hang up the phone.

Note that this routine does not ensure that the phone is answered; it simply dials the specified telephone number. If the user specifies the optional **text** argument, DECtalk speaks this text before returning control to the calling program.

If the **timeout** argument is specified, the DTC01 device always waits for the specified number of seconds before returning control, even if the phone is answered before the specified number of seconds has elapsed. On the other hand, the DTC03 device interprets the **timeout** argument as the maximum number of seconds to wait before returning control to the calling program. That is, the DTC03 device returns control either when the phone is answered or when the **timeout** argument has expired, whichever occurs first.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIOW.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_TOOLONG	Phone number is too long.
DTK$_INVMODE	Invalid **mode** specified.
DTK$_OFFHOOK	Phone is off hook (phone is already active).

DTK$HANGUP_PHONE—Hang Up the Phone

The Hang Up the Phone routine speaks an optional message and then hangs up the phone.

Format

DTK$HANGUP_PHONE   voice-id [,text]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

text
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Text to be spoken before hanging up the phone. The optional **text** argument is the address of a descriptor pointing to the specified text.

Description

DTK$HANGUP_PHONE hangs up the phone after speaking an optional message.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIOW.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.

DTK$INITIALIZE—Initialize DECtalk

The Initialize DECtalk routine initializes a DECtalk device and returns the device's assigned voice identifier.

Format

DTK$INITIALIZE   voice-id ,output-device [,device-type]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: write only
mechanism: by reference

Voice identifier of the newly created DECtalk device. The **voice-id** argument is the address of an unsigned longword that receives this identifier.

output-device
VMS Usage: device_name
type: character string
access: read only
mechanism: by descriptor

File specification or logical name to which the output associated with the DECtalk device is written. The **output-device** argument is the address of a descriptor pointing to this logical name.

device-type
VMS Usage: longword_signed
type: longword (signed)
access: write only
mechanism: by reference

Device type of the newly created DECtalk device. The optional **device-type** argument is the address of a signed longword that receives the device-type information. The two valid device types are:

DTK$K_DTC_01   For DECtalk I.

DTK$K_DTC_03   For DECtalk III.

If the device-type information is not received in time, the assigned **device-type** is DTK$K_DTC_UNKNOWN.

Description

DTK$INITIALIZE creates a DECtalk device and returns its assigned voice identifier, **voice-id**. **Output-device** is the device to which the output associated with this newly created DECtalk is written.

If DTK$INITIALIZE is called to create a second DECtalk on a device that already has a voice identifier associated with it, DTK$INITIALIZE simply returns the identifier of the already existing DECtalk, along with the condition code DTK$_VOIALREXI, which signifies that DECtalk already exists for this device.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $GETDVI.
RMS$_xxxx	Any error from $PARSE.
LIB$_xxxx	Any error from LIB$GET_VM or LIB$GET_EF.
LIB$_INSVIRMEM	Insufficient virtual memory to allocate needed buffer.
DTK$_VOIALREXI	DECtalk already exists for this device.
DTK$_WRONUMARG	Wrong number of arguments.

DTK$LOAD_DICTIONARY—Load a Word into the DECtalk Dictionary

The Load a Word into the DECtalk Dictionary routine loads a phonemic definition of a word into the DECtalk dictionary.

Format

DTK$LOAD_DICTIONARY   voice-id ,text ,substitution

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

text
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Word to be loaded into the DECtalk dictionary. The **text** argument is the address of a descriptor pointing to the specified word.

substitution
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Phonemic definition of the word specified by the **text** argument. The **substitution** argument is the address of a descriptor pointing to the phonemic representation of the specified word.

Description

DTK$LOAD_DICTIONARY loads the phonemic definition of a specified word into the DECtalk dictionary.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVVOIID	Invalid **voice**-**id**.
DTK$_NOROOM	No room in the dictionary to add the specified word.
DTK$_DEFTOOLONG	Word definition is too long.

DTK$READ_KEYSTROKE—Read a Key Entered on the Keypad

The Read a Key Entered on the Keypad routine reads a key entered on the phone keypad.

Format

DTK$READ_KEYSTROKE   voice-id ,key-code [,prompt-string] [,timeout]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

key-code
VMS Usage: longword_signed
type: longword (signed)
access: write only
mechanism: by reference

The DTK$K_TRM_xxxx code for the key entered on the keypad. The **key-code** argument is the address of a signed longword that receives this code. The valid codes are listed in DTKDEF.

prompt-string
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Text to be spoken before waiting for input. The optional **prompt-string** argument is the address of a descriptor pointing to this text.

timeout
VMS Usage: longword_signed
type: longword (signed)
access: read only
mechanism: by reference

Number of seconds to wait for input. The optional **timeout** argument is the address of a signed longword containing the specified number of seconds

the DECtalk device waits for input. If the **timeout** argument is omitted, DTK$READ_KEYSTROKE waits for input indefinitely.

Description

DTK$READ_KEYSTROKE reads a key entered on the phone keypad. If the optional **text** argument is specified, DECtalk speaks this text before waiting for input. If the keypad mode has not yet been set, this routine will set the phone keypad to auto-stop mode.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIOW.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_ONHOOK	Phone is on the hook (inactive).
DTK$_WINK	A wink has occurred.

DTK$READ_STRING—Read a Series of Keys Entered on the Keypad

The Read a Series of Keys Entered on the Keypad routine reads a series of keys entered on the phone keypad.

Format

DTK$READ_STRING   voice-id ,resultant-string [,prompt-string] [,timeout]
[,longword-integer-termin-code]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

resultant-string
VMS Usage: char_string
type: character string
access: write only
mechanism: by descriptor

String into which the keys being read are written. The **resultant-string** argument is the address of a descriptor pointing to this string.

prompt-string
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Text to be spoken before waiting for input. The optional **prompt-string** argument is the address of a descriptor pointing to this text.

timeout
VMS Usage: longword_signed
type: longword (signed)
access: read only
mechanism: by reference

Number of seconds DECtalk waits for input. The optional **timeout** argument is the address of a signed longword containing the number of seconds DECtalk

waits for input. If the **timeout** argument is omitted, DTK$READ_STRING waits for input indefinitely.

longword-integer-termin-code

VMS Usage:	longword_signed
type:	longword (signed)
access:	write only
mechanism:	by reference

The DTK$K_TRM_xxxx code for terminating key entered. The optional **longword-integer-termin-code** argument is the address of a signed longword that receives this code. The valid codes are located in DTKDEF.

Description

DTK$READ_STRING reads a series of keys entered on the phone keypad and stores them in **resultant-string**. If the optional **prompt-string** argument is specified, DECtalk speaks the specified text before waiting for input. The valid terminators are the number sign (#) and the asterisk (*). If the keypad mode has not yet been set, this routine will set the phone keypad to auto-stop mode.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIOW.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_ONHOOK	Phone is on the hook (inactive).
DTK$_WINK	A wink has occurred.

DTK$RETURN_LAST_INDEX—Return Last Index Spoken

The Return Last Index Spoken routine returns the last index spoken.

Format

DTK$RETURN_LAST_INDEX   voice-id ,p-index

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of
an unsigned longword containing this identifier. The voice identifier is returned
by the DTK$INITIALIZE routine.

p-index
VMS Usage: longword_signed
type: longword (signed)
access: write only
mechanism: by reference

Index to be returned. The **p-index** argument is the address of a signed longword
that receives the index identifier.

Description

DTK$RETURN_LAST_INDEX returns the last index spoken. An index is
inserted into the output stream with the DTK$SET_INDEX routine.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIOW.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.

DTK$SET_INDEX—Insert an Index at the Current Position

The Insert an Index at the Current Position routine inserts an index into the current output stream.

Format

DTK$SET_INDEX   voice-id ,p-index

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

p-index
VMS Usage: longword_signed
type: longword (signed)
access: read only
mechanism: by reference

Index to be inserted. The **p-index** argument is the address of a signed longword containing the index value. Valid values are in the range of 1 to 32767. An index of zero is reserved for use by DIGITAL.

Description

DTK$SET_INDEX inserts an index into the current position in the output stream. Allowable values for **p-index** are in the range of 1 to 32767. An index of zero is reserved for use by DIGITAL.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIOW.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVARG	Invalid argument.

DTK$SET_KEYPAD_MODE—Turn the Phone Keypad On and Off

The Turn the Phone Keypad On and Off routine turns recognition of the telephone keypad on or off.

Format

DTK$SET_KEYPAD_MODE   voice-id ,keypad-mode

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

keypad-mode
VMS Usage: longword_unsigned
type: longword (unsigned)
access: read only
mechanism: by reference

Mode that determines the status of the telephone keypad. The **keypad-mode** argument is the address of an unsigned longword containing this mode. The valid mode specifiers are:

DTK$K_KEYPAD_ON   Turns the keypad on.

DTK$K_KEYPAD_OFF   Turns the keypad off.

DTK$K_KEYPAD_AUTO   Turns the keypad on with autostop.

Description

DTK$SET_KEYPAD_MODE turns the recognition of the telephone keypad on or off. Depending upon the **keypad-mode** specified, the keypad can be turned on, off, or on with autostop. Autostop means that DECtalk stops speaking when a terminator is entered.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_ONHOOK	Phone is on the hook (inactive).
DTK$_INVMODE	Invalid **mode** specified.
DTK$_WINK	A wink has occurred.

DTK$SET_LOGGING_MODE—Set the Logging Mode for the Video Terminal Connected to the DECtalk Device

The Set the Logging Mode for the Video Terminal Connected to the DECtalk Device routine controls the information that is displayed on the video terminal while the DECtalk device is functioning.

Format

DTK$SET_LOGGING_MODE   voice-id [,new-mode] [,old-mode]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

new-mode
VMS Usage: mask_longword
type: longword (unsigned)
access: read only
mechanism: by reference

DECtalk mode to be set. The optional **new-mode** argument is the address of a longword bit mask containing the specified mode. Valid values for **new-mode** are:

- DTK$M_TEXT
- DTK$M_PHONEME
- DTK$M_RAWHOST
- DTK$M_INHOST
- DTK$M_OUTHOST
- DTK$M_ERROR
- DTK$M_TRACE
- DTK$M_DEBUG

It is possible to perform a logical OR operation on the bits in the bit mask together to set more than one mode at a time. Any mode not specified is reset.

old-mode

VMS Usage: mask_longword
type: longword (unsigned)
access: write only
mechanism: by reference

Current mode settings of the DECtalk device. The optional **old-mode** argument is the address of a longword bit mask that receives the current DECtalk settings.

Description

DTK$SET_LOGGING_MODE sets or resets the specified modes on the DECtalk device. It controls the information that is displayed on a video terminal connected to the DECtalk device. Note that any modes not explicitly set are reset by DTK$SET_LOGGING_MODE.

DTK$SET_LOGGING_MODE has two optional parameters, **new-mode** and **old-mode**. By specifying different combinations of these arguments, you can use DTK$SET_LOGGING_MODE in various ways.

- To use DTK$SET_LOGGING_MODE to determine the current mode settings, use the following format:

  DTK$SET_LOGGING_MODE (voice_id ,,old_mode)

- To use DTK$SET_LOGGING_MODE to set the bits without regard to their current setting, use the following format:

  DTK$SET_LOGGING_MODE (voice_id ,new_mode)

- To use DTK$SET_LOGGING_MODE to save the current settings, set new modes, and later restore the original settings, use the following format:

  DTK$SET_LOGGING_MODE (voice_id ,new_mode ,save_old_settings)

  This retrieves the current bit settings and then sets the mode according to the **new-mode** argument.

  Later, to restore the mode to its former state, specify the following format:

  DTK$SET_LOGGING_MODE (voice-id ,save_old_settings)

  This sets the new mode settings according to those previously retrieved.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVVOID	Invalid **voice-id**.
DTK$_INVMODE	Invalid mode specified.

DTK$SET_MODE—Set the Mode for the DECtalk Terminal

The Set the Mode for the DECtalk Terminal routine sets or resets the mode settings of the DECtalk terminal.

Format

DTK$SET_MODE   voice-id [,new-mode] [,old-mode]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

new-mode
VMS Usage: mask_longword
type: longword (unsigned)
access: read only
mechanism: by reference

DECtalk mode to be set. The optional **new-mode** argument is the address of a longword bit mask containing the specified mode. Valid values for **new-mode** are:

- DTK$M_SQUARE

- DTK$M_ASCII (valid for the DTC01 device only)

- DTK$M_MINUS

- DTK$M_EUROPE (valid for the DTC03 device only)

- DTK$M_SPELL (valid for the DTC03 device only)

It is possible to perform a logical OR operation on the bits in the bit mask together to set more than one mode at a time. Any mode not specified is reset. If the **new-mode** argument is omitted, the current mode is unchanged.

old-mode
VMS Usage: mask_longword
type: longword (unsigned)
access: write only
mechanism: by reference

Current mode settings of the DECtalk device. The optional **old-mode** argument is the address of a longword bit mask that receives the current DECtalk settings.

Description

DTK$SET_MODE controls the mode settings for the DECtalk device. Note that any modes not explicitly set are reset by DTK$SET_MODE.

DTK$SET_MODE has two optional parameters, **new-mode** and **old-mode**. By specifying different combinations of these arguments, DTK$SET_MODE can be used in various ways.

- To use DTK$SET_MODE to determine the current mode settings, use the following format:

  DTK$SET_MODE (voice_id ,,old_mode)

- To use DTK$SET_MODE to set the bits without regard to their current setting, use the following format:

  DTK$SET_MODE (voice_id ,new_mode)

- To use DTK$SET_MODE to save the current settings, set new modes, and later restore the original settings, use the following format:

  DTK$SET_MODE (voice_id ,new_mode ,save_old_settings)

  This retrieves the current bit settings and then sets the mode according to the **new-mode** argument.

  Later, to restore the mode to its former state, specify the following format:

  DTK$SET_MODE (voice-id ,save_old_settings)

  This sets the new mode setting according to those previously retrieved.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVVOID	Invalid **voice-id**.
DTK$_INVMODE	Invalid mode specified.

DTK$SET_SPEECH_MODE—Turn Speech Mode On and Off

The Turn Speech Mode On and Off routine either starts or stops the DECtalk device's speech.

Format

DTK$SET_SPEECH_MODE   voice-id ,new-mode [,old-mode]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id

VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

new-mode

VMS Usage: longword_unsigned
type: longword (unsigned)
access: read only
mechanism: by reference

Mode to be set. The **new-mode** argument is the address of an unsigned longword containing the specified mode. Valid values are:

DTK$K_SPEAK   Start speaking.

DTK$K_STOP   Stop speaking when current text is completed.

DTK$K_HALT   Stop speaking immediately.

If the **new-mode** argument is omitted, the current mode is unchanged.

old-mode

VMS Usage: longword_unsigned
type: longword (unsigned)
access: write only
mechanism: by reference

Current speech mode of the DECtalk device. The optional **old-mode** argument is the address of an unsigned longword that receives the current mode setting before enabling the new mode. The values returned in **old-mode** are the same as those valid for the **new-mode** argument.

Description

DTK$SET_SPEECH_MODE starts or stops the speech of the DECtalk device. When stopping DECtalk's speech, the user can either stop the DECtalk device immediately or stop it after it has finished speaking. Because DTK$SET_ SPEECH_MODE is used to set a new speech mode, the **new-mode** argument is required.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVMODE	Invalid mode specified.

DTK$SET_TERMINAL_MODE—Set the Mode for the Video Terminal Connected to the DECtalk Device

The Set the Mode for the Video Terminal Connected to the DECtalk Device routine controls the attributes of the video terminal connected to the DECtalk device.

Format

DTK$SET_TERMINAL_MODE   voice-id [,new-mode] [,old-mode]

Returns

VMS Usage:	cond_value
type:	longword (unsigned)
access:	write only
mechanism:	by value

Arguments

voice-id

VMS Usage:	identifier
type:	longword (unsigned)
access:	read only
mechanism:	by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

new-mode

VMS Usage:	mask_longword
type:	longword (unsigned)
access:	read only
mechanism:	by reference

DECtalk mode to be set. The optional **new-mode** argument is the address of a longword bit mask containing the specified mode. Valid values for **new-mode** are:

- DTK$M_HOST
- DTK$M_SPEAK
- DTK$M_EDITED
- DTK$M_HARD
- DTK$M_SETUP
- DTK$M_FILTER

It is possible to perform a logical OR operation on these values to set more than one mode at a time. Any mode not specified is reset. If the **new-mode** argument is omitted, the current mode is unchanged.

old-mode

VMS Usage: mask_longword
type: longword (unsigned)
access: write only
mechanism: by reference

Current mode settings of the DECtalk device. The optional **old-mode** argument is the address of a longword bit mask that receives the current DECtalk settings.

Description

DTK$SET_TERMINAL_MODE controls the mode settings for the video terminal connected to the DECtalk device. Note that any modes not explicitly set are reset by DTK$SET_TERMINAL_MODE.

DTK$SET_TERMINAL_MODE has two optional parameters, **new-mode** and **old-mode**. By specifying different combinations of these arguments, DTK$SET_TERMINAL_MODE can be used in various ways.

- To use DTK$SET_TERMINAL_MODE to determine the current mode settings, use the following format:

  DTK$SET_TERMINAL_MODE (voice_id ,,old_mode)

- To use DTK$SET_TERMINAL_MODE to set the bits without regard to their current setting, use the following format:

  DTK$SET_TERMINAL_MODE (voice_id ,new_mode)

- To use DTK$SET_TERMINAL_MODE to save the current settings, set new modes, and later restore the current settings, use the following format:

  DTK$SET_TERMINAL_MODE (voice_id ,new_mode ,save_old_settings)

  This retrieves the current bit settings and then sets the mode according to the **new-mode** argument.

  Later, to restore the mode to its former state, specify the following format:

  DTK$SET_TERMINAL_MODE (voice-id ,save_old_settings)

  This sets the new mode settings according to those previously retrieved.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVVOID	Invalid **voice-id**.
DTK$_INVMODE	Invalid mode specified.

DTK$SET_VOICE—Set Voice Characteristics

The Set Voice Characteristics routine changes the DECtalk voice characteristics to match those specified.

Format

DTK$SET_VOICE   voice-id [,new-voice] [,speech-rate] [,comma-pause]
[,period-pause]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

new-voice
VMS Usage: longword_signed
type: longword (signed)
access: read only
mechanism: by reference

Type of voice. The optional **new-voice** argument is the address of a signed longword containing any valid **new-voice** value. Valid values for **new-voice** are:

DTK$K_VOICE_MALE	Standard male voice
DTK$K_VOICE_FEMALE	Standard female voice
DTK$K_VOICE_CHILD	Standard child voice
DTK$K_VOICE_DEEP_MALE	Deep male voice
DTK$K_VOICE_DEEP_FEMALE	Deep female voice
DTK$K_VOICE_OLDER_MALE	Older male voice
DTK$K_VOICE_LIGHT_FEMALE	Light female voice

speech-rate
VMS Usage: longword_signed
type: longword (signed)
access: read only
mechanism: by reference

Rate at which DECtalk speaks, measured in words per minute. The optional **speech-rate** argument is the address of a signed longword containing this rate. The valid range of values for **speech-rate** is 120 to 350 words per minute.

comma-pause

VMS Usage: longword_unsigned
type: longword (unsigned)
access: read only
mechanism: by reference

Number of milliseconds by which to increase the time DECtalk pauses after a comma. The optional **comma**-**pause** argument is the address of a signed longword containing this number. A value of zero resets the pause time to the hardware default value.

period-pause

VMS Usage: longword_unsigned
type: longword (unsigned)
access: read only
mechanism: by reference

Number of milliseconds by which to increase the time DECtalk pauses after a period. The optional **period**-**pause** argument is the address of a signed longword containing this number. A value of zero resets the pause time to the hardware default value.

Description

DTK$SET_VOICE changes the DECtalk voice characteristics to match those specified. DTK$SET_VOICE can change the voice type, the rate of speech, and the number of milliseconds DECtalk pauses after commas and periods.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIO.
DTK$_INVVOIID	Invalid **voice**-**id**.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVARG	Invalid argument.
OTS$_xxxx	Any error from OTS$CVT_L_TU.

DTK$SPEAK_FILE—Speak the Text in a Specified File

The Speak the Text in a Specified File routine speaks the text contained in the specified file.

Format

DTK$SPEAK_FILE   voice-id ,filespec [,completion-mode]

Returns

VMS Usage:	cond_value
type:	longword (unsigned)
access:	write only
mechanism:	by value

Arguments

voice-id

VMS Usage:	identifier
type:	longword (unsigned)
access:	read only
mechanism:	by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

filespec

VMS Usage:	char_string
type:	character string
access:	read only
mechanism:	by descriptor

File specification of the file containing the text to be spoken. The **filespec** argument is the address of a descriptor pointing to this file.

completion-mode

VMS Usage:	longword_unsigned
type:	longword (unsigned)
access:	read only
mechanism:	by reference

Mode characteristic. The optional **completion-mode** argument is the address of an unsigned longword containing the specified mode. Valid values for the **completion-mode** argument are:

DTK$K_IMMED	Return to the user immediately (default).
DTK$K_WAIT	Wait until the text is completely spoken.
DTK$K_STATUS	Wait until the text is completely spoken, and then return a phone status.

Description

DTK$SPEAK_FILE speaks the text contained in the specified file.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIO.
RMS$_xxxx	Any error generated by RMS.
DTK$_INVVOIID	Invalid **voice**-**id**.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVMODE	Invalid **mode** specified.

DTK$SPEAK_PHONEMIC_TEXT—Speak the Specified Phonemic Text

The Speak the Specified Phonemic Text routine sends the specified phonemic text to the DECtalk device to be spoken.

Format

DTK$SPEAK_PHONEMIC_TEXT   voice-id ,text [,completion-mode]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

text
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Phonemic text to be spoken. The **text** argument is the address of a descriptor pointing to the specified phonemic representation of the text.

completion-mode
VMS Usage: longword_unsigned
type: longword (unsigned)
access: read only
mechanism: by reference

Mode characteristic. The optional **completion-mode** argument is the address of an unsigned longword containing the specified mode. Valid values for the **completion-mode** argument are:

DTK$K_IMMED	Return to the user immediately (default).
DTK$K_WAIT	Wait until the text is completely spoken.
DTK$K_STATUS	Wait until the text is completely spoken, and then return a phone status.

Description

DTK$SPEAK_PHONEMIC_TEXT sends the specified phonemic representation of some text to the DECtalk device. This text contains the phonetic representations of the words to be spoken; that is, the words are spelled as they are pronounced. The DECtalk device then speaks this text.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIO.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVMODE	Invalid **mode** specified.
DTK$_ONHOOK	Phone is on the hook (inactive).

DTK$SPEAK_TEXT—Speak the Specified Text

The Speak the Specified Text routine sends the specified text to the DECtalk
device to be spoken.

Format

DTK$SPEAK_TEXT   voice-id ,text [,completion-mode]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of
an unsigned longword containing this identifier. The voice identifier is returned
by the DTK$INITIALIZE routine.

text
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Text to be spoken. The **text** argument is the address of a descriptor pointing to
the specified text.

completion-mode
VMS Usage: longword_unsigned
type: longword (unsigned)
access: read only
mechanism: by reference

Mode characteristic. The optional **completion-mode** argument is the address
of an unsigned longword containing the specified mode. Valid values for the
completion-mode argument are:

DTK$K_IMMED	Return to the user immediately (default).
DTK$K_WAIT	Wait until the text is completely spoken.
DTK$K_STATUS	Wait until the text is completely spoken, and then return a phone status.

Description

DTK$SPEAK_TEXT sends the specified text to the DECtalk device. The DECtalk device then speaks this text.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIO.
DTK$_INVVOIID	Invalid **voice-id**.
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVMODE	Invalid **mode** specified.
DTK$_ONHOOK	Phone is on the hook (inactive).

DTK$SPELL_TEXT—Spell Text

The Spell Text routine causes DECtalk to pronounce each letter of the specified text.

Format

DTK$SPELL_TEXT   voice-id ,text [,completion-mode]

Returns

VMS Usage: cond_value
type: longword (unsigned)
access: write only
mechanism: by value

Arguments

voice-id
VMS Usage: identifier
type: longword (unsigned)
access: read only
mechanism: by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

text
VMS Usage: char_string
type: character string
access: read only
mechanism: by descriptor

Text to be spelled out by DECtalk. The **text** argument is the address of a descriptor pointing to the specified string.

completion-mode
VMS Usage: mask_longword
type: longword (unsigned)
access: read only
mechanism: by reference

The optional completion mode characteristic. The **completion-mode** argument is the address of a longword bit mask containing the specified mode. Valid values for the **completion-mode** argument are:

DTK$K_IMMED	Return to the user immediately (default).
DTK$K_WAIT	Wait until the text is completely spoken.
DTK$K_STATUS	Wait until the text is completely spoken, then return a phone status.

Description

DTK$SPELL_TEXT causes DECtalk to pronounce each letter of the specified text individually, rather than treat the text as a word. For example, DECtalk would normally pronounce USA as "oosa." Calling DTK$SPELL_TEXT causes DECtalk to pronounce USA as "U, S, A."

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $QIOW.
DTK$_INVVOIID	Invalid **voice-id.**
DTK$_WRONUMARG	Wrong number of arguments.
DTK$_INVMODE	Invalid **completion-mode** specified.
DTK$_ONHOOK	Phone is on the hook (inactive).

DTK$TERMINATE—Terminate DECtalk

The Terminate DECtalk routine terminates the use of an initialized DECtalk device.

Format

DTK$TERMINATE   voice-id

Returns

VMS Usage:	cond_value
type:	longword (unsigned)
access:	write only
mechanism:	by value

Arguments

voice-id

VMS Usage:	identifier
type:	longword (unsigned)
access:	read only
mechanism:	by reference

Voice identifier of the DECtalk device. The **voice-id** argument is the address of an unsigned longword containing this identifier. The voice identifier is returned by the DTK$INITIALIZE routine.

Description

DTK$TERMINATE terminates the use of the specified DECtalk device.

Condition Values Returned

SS$_NORMAL	Normal successful completion.
SS$_xxxx	Any error from $DASSGN.
LIB$_xxxx	Any error from LIB$FREE_VM or LIB$FREE_EF.
DTK$_WRONUMARG	Wrong number of arguments.

Index