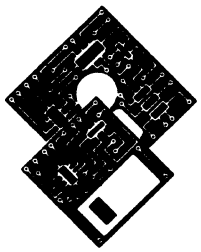

APPLE
PROGRAMMER'S
AND DEVELOPER'S
ASSOCIATION

290 SW 43rd. Street
Renton, WA 98055
206-251-6548

MacinTalk Development Package

Version 1.0

APDA#: KMSTD1

About the MacinTalk Development Package

Contents: Section:

Also included with this package is one disk. It contains the MacinTalk resource, interface code, an example program, and useful tools to aid in speech phoneme generation.

Contents of the MacinTalk disk:

```
ExceptionEdit          18K   6/3/85  9:59 AM   1/1/83  9:22 AM
MacinTalk              28K  7/28/86  1:09 PM   1/1/83  9:13 AM
sample speech           2K   5/6/86  6:08 PM  12/2/85  3:42 PM
SpeakFile               3K   6/3/85  8:44 AM   6/3/85  8:44 AM
Speech Tutor           16K   1/9/86  3:46 PM   1/9/86  3:54 PM
SpeechAsm.Rel           2K   6/3/85  7:01 AM   6/3/85  6:54 AM
SpeechLab              12K   6/3/85 10:00 AM   1/1/83  5:48 AM
TextToSpeak             3K  8/21/85 10:04 AM   1/1/83  1:28 AM

:Code Folder:Example Code:
EXAMPLE/SPEAKFILE.T…    3K   6/3/85  7:06 AM   1/1/83  5:05 AM
EXAMPLE/SPEAKFILER.…    1K   6/3/85  7:04 AM   1/1/83  5:06 AM

:Code Folder:Lisa Code:
intrfc/speechintf.t…    3K   6/2/85  2:31 AM   1/1/83  5:06 AM
OBJ/SPEECHASM.OBJ       1K   6/2/85  2:45 AM   6/2/85  2:44 AM
OBJ/SPEECHINTF.OBJ      3K   6/2/85  2:46 AM   6/2/85  2:46 AM
source/SpeechAsm.te…   13K   6/2/85  2:20 AM   1/1/83  5:06 AM

:Code Folder:MPW Code:
SpeechIntf.p            3K  7/23/86 11:15 AM  3/31/86  8:41 PM

:Code Folder:MPW Code:Better Speech:
SpeechImplementatio…   14K   4/5/86  1:01 PM   4/5/86 12:55 PM
SpeechIntf.o            1K   4/5/86  1:04 PM   4/5/86  1:04 PM

:Code Folder:MPW Code:Standard Speech:
Standard SpeechImpl…   14K  3/31/86  6:22 PM  3/31/86  6:22 PM
Standard SpeechIntf…    1K  3/31/86  8:34 PM  3/31/86  8:34 PM

:Code Folder:Original stuff:
MacinTalk              28K    1/2/4  3:58 AM   1/1/83  9:13 AM
MySpeech.a             14K  3/31/86  6:22 PM  3/31/86  6:22 PM
SpeechAsm.Rel           2K   6/3/85  7:01 AM   6/3/85  6:54 AM
SpeechAsm.text         14K  8/19/85  3:48 PM   1/1/83  5:06 AM
TLACvt.Directives       1K  7/29/85 11:46 PM  6/16/85  3:00 AM
```

ExceptionEdit is an application which allows the user to substitue the pronunciation that the READER package generates for words. This is useful when READER generates a pronunciation which is not correct for a word. For example, READER creates the words "Mitch Ale" for the input word "Michael." With ExceptionEdit, the correct pronunciation can be substituted.

MacinTalk is the driver for speech generation. It must be on the same disk as the application which uses MacinTalk, or may reside in the System Folder of the Startup Disk.

SpeakFile is an application which will use READER to speak a file of English text. The file that it searches for is called "`TextToSpeak`."

Speech Tutor is useful for learning to write in phonemes. It will translate English text to phonemes, and allows editing the resulting phoneme text for adjustments and improvements. The file `SpeakFile` can be opened with this application.

SpeechAsm.rel is the MDS .rel file for linking the MacinTalk routines to an assembly language application.

SpeechLab is another tool for learning to write in phonemes. It will translate English text to phonemes and also allows editing of either English or phoneme text. Not as powerful as `Speech Tutor`.

EXAMPLE/SPEAKFILE.TEXT is a Lisa Pascal file which uses MacinTalk.
EXAMPLE/SPEAKFILER.TEXT is the resource file for the application.

The files in the **Lisa Code** folder (which is in the **Code Folder**) are the sources and compiled objects of the interface and glue files for using MacinTalk under Lisa Pascal or Assembly. These files must be tranferred to the Lisa with MacCom.

The **SpeechIntf.o** file in the **MPW Code** folder (which is in the **Code Folder**) is the Pascal interface file for use under the Macintosh Programmer's Workshop. It is to be used with either of the implementations given in the two folders here.

The **Standard Speech** folder contains the interface source and object files (for use with the Macintosh Programmer's Workshop) for accessing MacinTalk in the standard manner (according to the documentation here).

The **Better Speech** folder contains the interface source and object files (for use with the Macintosh Programmer's Workshop) for accessing MacinTalk in a better way. The code here has been changed such that the application must open MacinTalk itself, that is, the application must look for the MacinTalk resource, and open it with Resource Manager calls.

The **Original Stuff** folder has the original source to Macintalk. It can be used with MDS, the Macintosh Development System.

THE MACINTALK 1.1 TOOLKIT

MACINTALK, the Macintosh Speech synthesizer, is a software driver that runs under the Macintosh operating system. This driver will convert an ASCII string of phonetic codes to high quality synthetic speech utilizing a male, non-regional, standard American dialect. The speaking rate and the basic pitch of the voice can be programmatically set and modified from an application program using **MACINTALK**. Because **MACINTALK** works best with phonetic text rather than standard English text as input, a special program, called **SPEECH TUTOR**, is provided to help prepare the special phonetic version of text that is required.

One component of the driver, called **Reader**, will convert standard English text into a phonetic transcription of the text that is suitable for passing to the **MACINTALK** driver. This translation process is not 100% accurate and can result in some rather interesting interpretations of English pronunciation. From experimentation, it has been found to be about 75% - 80% accurate. When it fails, however, the mistakes in pronunciation can be quite unintelligible for a person unfamiliar with the actual content that is being spoken. Therefore, it is not recommended that straight translated English text be used as input for speech. The phonetic version can be adjusted into very intelligible spoken English that is far superior to any raw translations from the **Reader** procedure.

MACINTALK VERSION 1.1

This document refers to version 1.1 of the MacinTalk driver, which was distibuted with the May 1985 Macintosh Software Supplement (the identical driver is also included with the December 1985 Software Supplement Additions). The Speech Tutor application mentioned in this document is included with the December Supplement Additions; ExceptionsEdit and interface files required for writing applications which use MacinTalk are included in the May Supplement.

USING THE MACINTALK DRIVER

Any application program that intends to use the MACINTALK driver makes three basic calls. The first, **SpeechOn**, initializes the driver and returns a handle that is used for subsequent calls to the driver. The call made most often, **MacinTalk**, is used to make the actual speech by passing it a handle to the phonetic string. When a program is through with the driver, a call to **SpeechOff** will close the driver and clean up.

There are two calls that can be made to adjust the sound of the speech in terms of the speaking rate and the base pitch of the speaking voice. **SpeechRate** sets the speaking rate in terms of "words per minute". This rate can be between 85 and 425 words per minute. Depending on the base pitch of the voice, any rate over 200 is only suitable for imitating cartoon characters.

The procedure **SpeechPitch** adjusts the base pitch of the voice. **MACINTALK** has two basic speech "styles" - natural or robotic. Natural speech utilizes a complex pattern of rising and falling pitch to imitate the normal inflections of speech. In this mode, the base pitch defines the center around which the rise and fall occurs. Robotic mode maintains a constant pitch - the base pitch - and no inflections are used. This mode has a machine-like quality which may or may not be suitable for human consumption.

The base pitch can be set within the range of 65 to 500 Hertz. A pitch around 100 Hertz is considered normal. Pitches above 150 (again dependent on the speech rate) can take on a humorous quality that may be good for a laugh, but not very intelligible.

Experimenting with the synergistic effects between speech rate and base pitch can give you a

variety of different sounding speech that may be more suited for the type of effect you wish to acheive. However, the ability of a person *understanding* the speech can be impaired by these effects.
The driver defaults to a "normal" speaking voice which has a base pitch of 95 Hertz and a speaking rate of 150 words per minute using a "natural" mode of speaking. Both the pitch and the rate can be changed prior to any call to the **MacinTalk** procedure. By breaking the phonetic strings into parts, the Macintosh can be made to sing!

PREPARING PHONETIC TEXT

The **MACINTALK TOOLKIT** allows for two different ways for preparing phonetic text for output through the driver.

As a programmer, you can patiently sit, translating your intended English text into its phonetic form and type it into your program as strings that can be passed to the driver.

Or, using **SPEECH TUTOR**, you can generate phonetic text from the intended English text and then edit the phonentic text for intelligibility. All the required phonetic strings and their accompanying English form are then stored as resources in a resource file, which your program can access for output to the driver.

USING TRANSLATED SPEECH

Though it is not recommended that you simply translate English text using the **Reader** procedure, you can govern to a large extent how it translates by using what is called an "Exceptions File". This is a file that is used by the **Reader** procedure to control how it translates specific patterns, words, punctuation marks, and abreviations. The **Reader** procedure uses a standard set of rules ("exceptions") that can be added to or modified by using the **EXCEPTIONS EDITOR**. This program allows you to specify how the **Reader** procedure will translate its input. Your program can then tell the driver which Exceptions File to use when translating.

This ability allows you to customize the **Reader** procedure for specific types of text you wish to translate.

USING SPEECH TUTOR

Speech Tutor is an application that is used to prepare English text and its phonetic equivalent for use by another application program using **MACINTALK**. You enter English text and then it is translated to its phonetic equivalent. The phonetic form can be listened to, and then edited to improve its intelligibility. You can listen to the complete phonetic string or portions of the string. **Speech Tutor** allows you to prepare a number of text pairs - English text paired with its phonetic form - and then it will save the strings in a resource file that can then be used in conjunction with your application program. Your program can access both the English text, for display purposes, and the phonetic text to pass on to the **MACINTALK** driver.

Setting up a Speech Resource File

When **Speech Tutor** is initially run, from the **File menu** you can select either **NEW** to create a new resource file or **OPEN** to open an existing speech resource file (these are files whose creator is SPKR).

Opening a New file allows you to save the contents as a new file using the **SAVE AS...** selection, to throw away your work using the **THROW AWAY WORK...** selection, or to close

and quit (equivalent to **SAVE AS...** and **QUIT**) using the **CLOSE FILE AND QUIT** selection.

An existing file can be saved with either **CLOSE CURRENT FILE** or **CLOSE FILE AND QUIT**.

Any selection other than quiting will return to the initial state where you can either create a new file or open an existing file.

Editing the Text

Speech Tutor displays two windows, one for English text and one for phonetic text. The windows display the name of the current file as well as the type of text and the text record number.

This program uses the standard TextEdit interface and supports the usual **CUT**, **COPY**, **PASTE**, and **CLEAR** edit commands. **UNDO** is not supported.

Speech Tutor has a special edit function called **TRANSLATE AND PASTE**. First select a portion of the English text and then **COPY** it. Then select an insertion point in the phonetic text window. Selecting the **TRANSLATE AND PASTE** function from the **Edit menu** will cause the English selection to be translated into its phonetic form and then pasted into the phonetic text.

Each portion, either English or phonetic, of the text record is restricted in size to 2048 characters. Phonetic text, translated from English, is about 1.5 times bigger than the accompanying English text, so the practical size for English text is about 1300 characters. If your text exceeds this size, an error is reported. You can save the extra text by cutting it, moving to a new window and then pasting it there.

Hearing the Speech

The **Say It menu** has two options. The first, **TRANSLATE**, will translate any selected English text or the entire English text, if no portion of the text is selected. If the phonetic text window is empty, the phonetic results of the translation will be placed in it. If it is not empty, no phonetic text will be pasted there. In either case, the translated text will be spoken by the Macintosh.

The **SPEAK** selection will speak any selected portion of the phonetic text or the entire contents of the phonetic text window if no portion is selected. If an error is found in the phonetic text string, it will be reported and the insertion point (the flashing bar) will be placed immediately after the offending character in the string.

You can adjust the volume of the speech by selecting the **CONTROL PANEL** desk accessory and adjust the volume there.

Setting Voice Options

Under the **Voice Options menu**, the **SET SPEECH PARAMETERS** selection will display a dialog box that will allow you to set the basic pitch and speaking rate. There are two "preset selections" which override the pitch and rate settings. The first, **Normal**, is the default setting of the driver. The second, **Drunk**, is a reasonable imitation of "Crazy" Guggenheim. You can set the rate and speech values by changing the numbers in the appropriate boxes.

The two "check boxes", **Natural** and **Robotic**, set the voice mode as described above.

Editing Phonetic Text and Using the Phonetic Alphabet

The phonetic alphabet uses special sequences of standard English letters to stand for specific phonemes (sounds) found in English. Most of the consonants map to themselves, e.g. r is **R**, f is **F**, and so on. These letters are easy to remember and can be simply typed in to the phonetic text.

There are some special consonant sounds that require special representation. These are usually consonant pairs such as 'sh' and 'ch'. These sounds can be found in the **SPECIAL CHARACTERS menu**. Select an insertion point in the phonetic text and then select the desired sound. The correct phonetic sequence will be inserted into the phonetic text.

All of the vowel sounds are mapped into two-character sequences. The letters used in these sequences may or may not have any relation to the sound expressed. For example, **IY** represents the long-e sound found in the word **beet**. All of the vowel sounds can be found in the **"Vowel" menus**. For example, if you need the short-i sound (as in '**hit**'), select an insertion point in the phonetic text, pull down the **I menu** and select **bIt.** This will insert the correct phonetic sequence. The vowel menus also have all the diphthongs.

Inflection (either rise or fall in pitch) is represented by a number (1 - 9) placed after a vowel phoneme. This number has the effect of causing a slight lengthening of the vowel sound coupled with a rise in pitch. The actual effect is dependent on the position of the vowel in the word and the position of the word in the sentence. For example, the word "carbon" could be written as KAARBIHN with no inflection or as KAA5RBIHN with emphasis on the first syllable. The judicious of these inflection values can tremendously increase both the intellegibility of the text as well as add the 'human' dimension to the style of speech.

There are a limited set of punctuation marks recognized by the driver. A period ('.') marks the end of a sentence and causes a fall in pitch at the end of the sentence. A question mark ('?') at the end of the sentence will cause a rise in pitch denoting a question. A comma (',') will cause a pause with a slight rise in pitch. A comma should be used to divide clauses in the sentence. Another way of making a pause is to use a dash ('-'). This sign is used to demarcate phrases in a sentence. It does not cause a rise in pitch preceeding it. In order to emphasis noun phrases in a sentence, parenthesis can be used to mark the noun phrase at the beginning and end. You should only use these for noun phrases with two or more "content" words. They are most effective around large noun phrases, such as "the big ugly cat who ate the mouse".

If you translate English text, only commas and periods are translated into their appropriate form. Question marks are translated as periods and should be changed back to question marks in the phonetic text. All other punctuation marks such as quotes, '/', etc. are translated into their word equivalents. For example "carbon" is translated as "-KWOWT- KAARBIHN- AHNKWOWT", that is "quote, carbon, unquote".

HOW TO WRITE PHONETICALLY FOR MACINTALK

This section describes in more detail the method for writing phonentic input for the **MACINTALK** driver. The intelligibility of the speech produced is directly proportional to your ability to express the sounds of speech using the phonetic alphabet. The translation function produces only a "rough draft" of good phonentic input and it requires finer tuning in order to get the best results.

Phonetic Spelling

A phoneme is the basic sound unit of speech, a "speech atom" in a sense. The phonetic alphabet

supplies a symbol for almost all English phonemes. For example, the word "cat" has three phonemes (which should not be confused with the three letters in the word). The first is the 'k' sound represented by the phoneme "K". The second is the vowel sound represented by the phoneme "AE", which is followed by the third, "T". Thus the word "cat" would be spelled phonetically as "KAET".

Some phonemes are expressed with two letters in English. The "sh" sound as in "should" and the "ch" sound in "chucksteak" are some examples. Note that the "sh" sound is also in the word "sugar" where the "sh" letters are not found. An important rule is to think about how the word sounds and not how it is spelled.

Choosing the Right Vowel

The phonetic alphabet supplies 18 basic vowel sounds. When choosing a vowel sound, say the word aloud and listen to the vowel. Compare this sound with the sample word vowel sounds. For example, the "a" in "apple" has the same sound as the "a" in "cat", not like the "a" in "made", "about", or "talk".

Some vowel sounds are called dipthongs. These are combinations of two vowel sounds that are usually spelled with two vowels. The "oi" in "boil" is an example. Each of these sounds have have a single phoneme to represent them. In this case "OY" stands for the "oi" sound in "boil".

In many English words, the vowel sounds are shortened almost out of existence. In the word "Macintosh", the "i" is hardly spoken. Normally, this sound would be written as "IX" but there are some special phonemes that can be used to represent these types of sounds. Instead of "IX" the phoneme "IN" can be used. This effect is called "relaxation of the vowel" and it usually occurs before the letters "l", "m", and "n". A little experimentation is appropriate when trying to choose the correct phoneme.

Choosing the Right Consonant

In most cases, picking the right consonant is very easy. But there is an important distinction between two types of consonants, making the right choice very important. Basically, consonants come in two forms: voiced and unvoiced. A voiced consonant is spoken with the vocal cords vibrating. In the word "pleasure" the "s" is voiced. When an unvoiced consonant is spoken, the vocal cords do not vibrate. The "s" in "soup" is unvoiced. Each form of consonant has its own phonemic representation. Again, pay attention to the sound and not to the spelling.

The correct meaning of your sentences is conveyed by their sound. If you make a mistake you may convey either the wrong meaning or one that is meaningless. The word "close" is a good example. When you say "Close the door" the "s" in "close" is voiced. When you say "Put that bread close to the oven" the "s" in "close" is unvoiced.

The sounds that are associated with the letter "r" can cause some confusion. There are three phonemes for "r" in the phonetic alphabet. The first "R" is used as a consonant as in "rat" - "RAET". Many times "r" is used in conjuction with a vowel as in "flirt". Here the other phoneme "ER" is used as in "FLERT". The third is called the "post-vocalic R" and is found after "a" in such words as "car". The rule is - use "R" if it proceeds or follows another vowel sound in the word, as in "car", "write", or "rabbit". Use "ER" if the "r" sound is part of the vowel sound, as in "absurd", "bird", and "herd". Use "RX" after "a" when appropriate.

Special Symbols

There is also a phoneme for something that is less of a sound and more of a pause. This is called a "glottal stop". In the word "kitten" one occurs between the two "t's". For many cases, **MACINTALK** will correctly place a "glottal stop" in the word, but there are some cases when it fails to recognize the need for one. Carefully listen to your text and occasionally you will find that a "Q" (glottal stop phoneme) can improve the sound.

Stress and Intonation

By stressing the parts of words through accenting syllables and generally describing the intonation patterns (the rise and fall in pitch), you can improve both the sound of your text and its expression of meaning.

Stress and intonation are represented by the addition of an integer (in the range 1 - 9) after a vowel phoneme. Stress is accomplished by lengthening the vowel sound. The presence of a number after a vowel phoneme will cause its sound to be lengthened. The value of the number represents the intonation. The higher the value, the greater the potential for either a rise or fall in pitch. The actual result is relative to the word's position in the sentence, which is governed by its proximity to punctuation marks.

There are a few simple rules for placing these numbers (also called "stress marks"):

1. Always place a stress mark in a "content word", i.e. a word that expresses meaning in the sentence. Nouns, verbs, and adjectives are content words.

2. Always place a stress mark on the accented syllable(s) of a polysyllabic word whether it is a content word or not. A dictionary is a good source for information about accents. For words with more than one accented syllable, use a value of 3 - 9 for the primary accent and a value of 1 - 2 for the secondary accent. Compound words like "baseball" and "lunchbag" should be treated as two words when placing stress marks.

3. A general scheme for what value to use is shown below:

Nouns	5
Pronouns	2
Verbs	4
Adjectives	5
Adverbs	7
Quantifiers	7
Exclamations	9
Articles	no stress mark
Prepositions	no stress mark
Conjunctions	no stress mark

These rules are not to be slavishly followed. The best rule is "does the speech correctly express the meaning I wish to convey". There are times when you may need to stress a preposition and not stress a vowel. Listen to the speech and always ask yourself, "Does it say what I mean?"

Further Hints for Good Speech

1. Often a polysyllabic word is easier to understand than a monosyllabic word. Try to vary the text so that its meaning is carried in more than one way. This makes for better understanding.

2. Keep your sentences to a reasonable length. The Macintosh can talk with out taking a breath which sounds unnatural. Also, your listeners can only keep so much information in their heads for a given amount of time. Keep the sentences fairly short and pause once in a while so that your listener can keep abreast of your meaning.

3. If your text is introducing new terms or using words in an unusual sense, stress these words the first time they are introduced. Then treat them in the normal fashion.

4. Listen to your text with your eyes closed. Don't read along while you are trying to improve its sound. Try it on an unsuspecting victim, see if they can understand your text.

USING TEXT RECORDS CREATED BY SPEECH TUTOR

SPEECH TUTOR creates a resource file that contains both the English and phonetic strings you have created. This resource file can then be used in conjuction with your application to "vocalize" its output.

Each string is a "named" resource and it is best to access them by using the **GetNamedResource** call. English strings are named "Eng Text - n" where 'n' is an integer denoting the record number. Phonetic strings are named "Phn Text - n" where, again, 'n' is the record number. These records are paired together just as they were when you used **SPEECH TUTOR** to make them. Thus there is "Eng Text - 1" and "Phn Text - 1", etc., which correspond to the window contents displayed by **SPEECH TUTOR.**

Please refer to the Resource Manager documentation for information about using resource files and the resources they contain.

BUILDING EXCEPTION FILES WITH THE EXCEPTION EDITOR

The translation English text by the **Reader** procedure is governed by about 400 context sensitive rules. But even with these rules the translations are not necessarily correct. For example, the word "doughnut" comes out as "duffnut". These rules can be added to or modified by creating what is called an Exceptions File, which is simply a dictionary of rules for translating the text. The **Reader** procedure will match any patterns found in the text with patterns defined in the Exceptions File and choose the defined definition for translating them. For example, you can define that "doughnut" be translated as "DOW3NAHT" which is much better than "DAHFNAHT".

To make an Exceptions File, run the Exception Editor program. To create a new file, select **CREATE NEW EXCEPTIONS FILE** in the **File menu**. To open an existing Exceptions File, select **OPEN** in the **File menu.**

The Exception Editor program will periodically save your work to the file. If you are working on an existing file, duplicate the old file first, before working with it with the editor. This will save the old version in case you need it later.

The program displays two windows: the "Say" window and the "As" window. You enter a word or pattern in the "Say" window. Typing <return> will display the default translation in the "As" window. You can edit the phonetic text in the "As" window to the desired pronunciation. Once the word and its translation are to your liking you click the **ADD** button to add it to the file. You can delete a word from the file by typing its English form in the "Say" window and typing <return>. Then click the **DELETE** button to delete it.

When you are done, select **SAVE** in the **File menu**. You can then begin work on another new or old file. Selecting **QUIT** will exit the program. **Do not select QUIT before saving because you will lose all the work done after the last automatic update to the file.**

FUNCTION SpeechOn(ExceptionsFile: Str255;
VAR theSpeech: SpeechHandle): SpeechErr;

SpeechOn initializes the **MACINTALK** driver and performs the following actions depending on the value of "ExceptionsFile":

ExceptionsFile = " {null string} - Bring in Reader with default rules

ExceptionsFile = 'noReader' - Do not load the Reader package

ExceptionsFile = 'filename' - Bring in Reader and use the rules in the Exceptions File 'filename'

SpeechOn will return a handle in theSpeech that is used for subsequent calls to the driver. This function should be called before any other **MACINTALK** procedures are called. It sets the driver to the default speech mode of natural, the speaking rate to 150 words per minute, and the base pitch to 95 Hertz.

SpeechOn will return any errors. The possible error values are:

noErr	- driver was opened successfully
memFullErr	- the required buffers cannot be allocated
resFNotFound	- the specified Exceptions File cannot be found
resNotFound	- a required driver resource cannot be found
fullUnitT	- the driver unit table is full

PROCEDURE SpeechOff(theSpeech: SpeechHandle);

SpeechOff closes the driver and cleans up. The driver is removed from the unit table, any buffers are released, and any resource files used by the driver are closed.

PROCEDURE SpeechPitch(theSpeech: SpeechHandle; thePitch: INTEGER;
theMode: F0Mode);

SpeechPitch will set the base pitch and/or it will set the speech mode. The pitch can be any value between 65 and 500. Any value outside this range is ignored. By passing a value of 0, only the mode can be changed. theMode can take the values NoChange (to only change the pitch), Natural for natural mode, and Robotic for robotic mode.

PROCEDURE SpeechRate(theSpeech: SpeechHandle; theRate: INTEGER);

SpeechRate will set the speaking rate in words per minute. The rate must be between the range 85 to 425. Any other value is ignored.

FUNCTION MacinTalk(theSpeech: Speechhandle;
Phonemes: Handle): SpeechErr;

Phonemes is a handle to a packed array of characters that holds the phonetic string to be spoken. This string must be in upper case and consist only of legal phonetic characters, stress marks, and punctuation marks. The input length is determined by either a '#' terminating the string or after processing a GetHandleSize(Phonemes) by the driver.

Possible return (error) values are:

noError	- No error
nilHandleErr	- the input handle or master pointer is NIL
positive integer	- an error in the input string has been found and the integer reflects the position in the string of the offending character

This function will break the input into sentences, treating each as a separate utterance. It will attempt to get a buffer large enough to hold the whole utterance. If a buffer cannot be found, the function will recursively break the utterance into smaller and smaller pieces until a buffer can be found. This splitting may cause a two-character phoneme to be broken in half which will create an illegal character sequence that will cause an error. This is unlikely to occur, except for long sentences with no spaces.

The driver requires about 20K of space. It allocates buffers, which are about 800 bytes in size, as needed.

FUNCTION Reader(theSpeech: SpeechHandle; EnglishPtr: Ptr;
InputLength: LONGINT;
PhoneticOutput: Handle): SpeechErr;

The Reader function translates English text into its phonetic form. EnglishPtr is a pointer to the packed array of characters that holds the English text. InputLength is the length of the array in characters. The string can be terminated by the sequence '##' or by reaching the InputLength character in the string. The handle PhoneticOutput points to the packed array of phonetic characters. This handle may possible be empty (but not equal to NIL).

Possible Errors returned are:

noError	- no error
nilHandleErr	- PhoneticOutput handle is invalid
memFullErr	- no room to allocate output handle

The Reader function is loaded indendently of the other **MACINTALK** procedures based on the values passed in the SpeechOn call. Reader takes up about 10K + the size of the output buffer.

MACINTALK PHONEME TABLE

VOWELS

made	- EY	beet	- IY	hide	- AY	low	- OW	use	- UW
bat	- AE	bet	- EH	bit	- IH	hot	- AA	under	- AH
about	- AX	better	- ER	bird	- ER	power	- AW	urban	- ER
talk	- AO					look	- UH		
						soon	- UW		
						border	- OH		
						toy	- OY		

VOWEL CONTRACTIONS

AXL can be written as UL	IXL can be written as IL
AXM can be written as UM	IXM can be written as IM
AXN can be written as UN	IXN can be written as IN

QX can be used to represent a silent vowel.

CONSONANTS

but	- B	soup	- S
dog	- D	table	- T
fed	- F	very	- V
guest	- G	wax	- W
hole	- /H	axe	- KS
judge	- J	yak	- Y
kitchen	- K	zipper	- Z
lot	- L	check	- CH
must	- M	loch	- /C
new	- N	rush	- SH
push	- P	pleasure	- ZH
quit	- KW	thin	- TH
rat	- R	then	- DH

SPECIAL SYMBOLS

pity (tongue flap)	- DX
kitt_en (glottal stop)	- Q
call	- LX
car	- RX

1 - 9	are stress marks
.	sentence terminator
?	sentence terminator
,	clause delimiter
-	phrase delimiter
()	noun phrase

USING MACINTALK

To use **MACINTALK** on a 400K diskette, simply place the Macintalk driver file anywhre on your application disk. The System Folder is a good place to put it. To use **MACINTALK** when your application is on an 800K disk or a Hard Disk 20 (which use the Hierarchical File System), either place the Macintalk driver file on in System Folder or place both the application and driver together in the same directory.

Pascal based programs need to include the file SPEECHINTF in a USES statement:

```
$USES {obj/SpeechIntf} SpeechIntf;
```

The resulting object code must be linked to the files OBJ/SPEECHINTF and OBJ/SPEECHASM.

All the **MACINTALK** routines are stack-based and follow the general parameter passing conventions, so assembly-language based programs can also access them.

LICENSING MACINTALK

You may not distribute a product which uses **MACINTALK** without specific written permission of Apple Computer, Inc. Licenses are available for a $250 annual fee through the Apple Software Licensing Department; they can be reached at (408) 973-4667.

TECHNICAL SUPPORT

Apple's Technical Support group does not offer support for MacinTalk. Apple *does not* guarantee that Macintalk will be supported on any future Macintosh hardware.

MACINTALK 1.1

THE MACINTOSH SPEECH SYNTHESIZER

MACINTALK 1.1

INTRODUCTION:

The Macintosh speech synthesizer (MACINTALK) is a software driver which runs under the MACINTOSH operating system. Running in real time, MACINTALK converts an ASCII string of phonetic codes to high quality synthetic speech utilizing a male, non-regional, standard American dialect. The MACINTALK driver also provides for user control of the speaking rate and pitch. Another program, READER, converts unrestricted English text into phonetic codes directly usable as input to MACINTALK.

Also provided are an application, SpeechLab, for experimenting with the system, and ExceptionEdit, an exceptions editor for creating custom rule sets for READER.

The MACINTALK driver consists of the following procedures and functions:

FUNCTION SpeechOn(ExceptionsFile: Str255; VAR theSpeech: SpeechHandle): SpeechErr;

SpeechOn initializes the Macintalk speech driver and takes the following actions depending on the setting of "ExceptionsFile".

ExceptionsFile null ==> Bring in READER using only the default rules.
ExceptionsFile := 'noReader' ==> Do not use READER package (phonetic input only).
ExceptionsFile := 'Filename' ==> Bring in READER with 'Filename' exceptions file.

SpeechOn allocates a handle to the required static buffers and sets default speaking rate, baseline pitch, pitch mode, and speaker sex. This function should be called exactly once, before any other speech routine. The default parameter values set by SpeechOn are:

Speaking Rate:	150 words/minute
Baseline Pitch:	110 Hz
Pitch Mode:	Natural
Speaker Sex:	Male

SpeechOn returns a handle to the driver's internal parm block in the variable "theSpeech". This parm block should not be modified by the user, but must be passed to each MacinTalk routine.

SpeechOn return code:

noErr	- driver open was successful
memFullErr	- the required buffers cannot be allocated
resFNotFound	- the specified exceptions file can't be found
resNotFound	- one of the required resources DRVR, TALK, or RULZ can't be found (these resources are found in the MacinTalk file, which must be on the same disk as the application)
fullUnitT	- the driver unit table is full

PROCEDURE SpeechOff(theSpeech: SpeechHandle);

SpeechOff closes the speech driver and cleans up. The driver is removed from the unit table, any still existing buffers are released, and all resource files are closed.

PROCEDURE SpeechRate(theSpeech: SpeechHandle; theRate: INTEGER);

SpeechRate sets the speaking rate, in words per minute. The rate is constrained to lie between 85 and 425 words per minute.

PROCEDURE SpeechPitch(theSpeech: SpeechHandle;
thePitch: INTEGER; theMode: F0Mode);

SpeechPitch sets the MACINTALK'S baseline pitch in Hertz, and it's pitch mode (either Natural or Robotic).

In the "Natural" mode, a sentence's pitch follows a complex, generally declining, contour of rises and falls, approximating the natural intonations of human speech. In this mode, thePitch is the frequency around which the sentence's pitch gestures are made. In the "Robotic" mode, the pitch is held constant throughout the sentence, giving the speech a mechanical or robot-like quality and is primarily for use in games where such perversions are commonplace.

If the specified mode is "NoChange", the mode will not be changed. This allows the user to change the baseline pitch without effecting the pitch mode. In either mode, the pitch is constrained to lie between 65 and 500 Hertz. If the input parm "thePitch" lies outside this range, the synthesizer's pitch will not be changed from its last value. This allows the user to change the pitch mode without effecting the baseline pitch.

PROCEDURE SpeechSex(theSpeech: SpeechHandle; theSex: Sex);

Reserved for future implementation.

FUNCTION MacinTalk(theSpeech: SpeechHandle; Phonemes: Handle): SpeechErr;

The input handle is a doubly indirect pointer to a packed array of ASCII characters containing the phonemes to be spoken. The input string must be in upper case, and consist only of valid phonetic codes, stress numbers, and prosodic indicators (see "How to Write Phonetically..."). The input is terminated upon encountering a "#" or after processing "GetHandleSize(Phonemes)" number of characters. This is done so that the handle size need not agree with the actual size of the phoneme

array.

Macintalk return codes are:

noError	- No error.
nilHandleErr	- Input handle or master pointer NIL.
+ive integer	- Phoneme code error. Value of return code is the location in the input array of the error. Possible errors are: invalid phoneme code, attempt to stress a non-vowel, or use of lower case letter or ASCII control code.

Overflow Processing:

MACINTALK will break the input into sentences, treating each sentence as a separate utterance, attempting to get a buffer large enough for the entire utterance (approx. 800 bytes per second of speech). MACINTALK also has some potentially overflowable buffers within it. They can hold 512 phonemes or 128 syllables, whichever fills up first. If any buffer overflows, MACINTALK will attempt to intelligently and recursively break the utterance into smaller fragments at punctuation marks. If this still overflows, MACINTALK will attempt to break the utterance into groups of words, single words, or if necessary, into parts of words. This last-gasp attempt would only be used in the highly unlikely event of a sentence having no spaces. This may split a phoneme code in half causing an error; eg. '...DHAXMAE5KINTAOKSPI Y5CHSIH5NTHIXSAYZER...'

phoneme error----------^

IY split

RAM requirements:

Code + Static buffers: 20K

Dynamic buffers: allocated as needed. Typically, 800 bytes per second of uninterrupted speech, and nearly all sentences of reasonable size are less than 10 seconds long.

DON'T PANIC

Some Notes on MacinTalk

- The MacinTalk speech driver arbitrates the unit (aka driver) number at run time. If the driver table is full, the SpeechOn will return a "fullUnitT" (-4000 decimal) error.
- To install MacinTalk on a user diskette, simply move the MacinTalk icon and any user defined exceptions dictionaries to that diskette.
- The MacinTalk driver must be on the same diskette as the application using it.
- The volume of the speaking voice is controlled via the control panel. A separate volume call may be added at a later time.
- The ExceptionEdit program creates and edits a user defined exceptions dictionary. This is a first release, so care should be taken in its use. In particular, always copy a pre-existing exceptions file before attempting to modify it. This will protect you in case of a system crash.
- The sex and language options are not included in this release. They may be implemented in the future.
- The SpeechLab and ExceptionEdit applications supplied with the MacinTalk driver may have some bugs.
  These bugs are in the applications and **not** in the driver.

READER

ENGLISH TEXT TO PHONETICS CONVERTER

READER is a Macintosh package which converts English text to phonetic codes acceptable in format to the MACINTALK speech driver. The input string can consist of unrestricted English text, including letters, symbols (such as "$" and "%"), and digits. The output of READER consists of a handle to a packed array of characters which contains the phonetic codes. This handle can be edited, saved on disk, and/or directly input to the MACINTALK speech driver.

The READER package should only be used when the text to be spoken is not known in advance by the programmer. READER should be used on text that comes from the outside world, such as modem or user supplied input. When the text to be spoken is "canned", contained within the application, it should be in human encoded phonetic form. READER may be used to translate small words, names or phrases which you can then insert into previously written phonetic strings (MAD-LIBS style).

```
FUNCTION Reader(theSpeech: SpeechHandle; EnglishInput: Ptr;
                InputLength: LongInt;
                PhoneticOutput: Handle): SpeechErr;
```

"EnglishInput" is a pointer to a packed array of characters containing the English string to be translated.

"InputLength" is a Longint which specifies the length of the input text. Translation of the input will terminate upon encountering either a "##" in the input, or by processing "InputLength" number of characters, whichever comes first.

"PhoneticOutput" is a possibly empty (NOT NIL!!!) handle whose master pointer will, upon return, point to a packed array of characters which will contain the phonetic codes to be spoken. READER will dynamically grow this handle as needed to accomodate the size of the output.

READER returns:

noError	- Good return.
nilHandleErr	- Invalid output handle.
memFullError	- Cannot get enough memory to hold the output.

RAM Requirements: 10K + output buffer.

In general, the output buffer can be expected to be approximately 1.5 times the input buffer in size.

USING THE EXCEPTIONS EDITOR

The READER package uses in excess of 400 context sensitive rules to derive the phonetic spelling of English text, but even with that many rules it is still subject to errors. This is because it has no knowledge of parts of speech, grammar, entymology, etc. Fortunately, we can eliminate some of the more annoying errors by using an "exceptions" dictionary for when the word does not "fit the rule". For example, if we are listening to a paper on somebody named "Michael", we might get a little tired of hearing about "Mitch ale". So we enter that name into an exceptions dictionary along with the proper pronunciation, MAY5KUL. We can tailor these dictionaries to various fields such as medicine, computing, or sports. The ability to create separate dictionaries allows us to get higher performance in a given application without using up vast amounts of memory.

To enter words into an exceptions dictionary, launch the "ExceptionEdit" application. You must now do one of the following: if the dictionary you wish to work with does not yet exist, you must create one by pulling down the "File" menu and selecting the "Create New Exceptions File" item. If you are adding to an existing file, select "Open". Both of these items use the standard file package, and their use should be self-explanatory. Note, however, that exceptions files are special. Do not try to edit one by any means other than ExceptionEdit.

You are now ready to enter words. Click in the "Say" window, type the English word in question and **hit <return> to have it translated.** The phonetic equivalents will appear below in the "As" window and be spoken. Now click in the "As" window and edit the phonetic code until it is to your liking. Use the "Say It" button to hear the text currently in the "As" window. When you are satisfied with the pronunciation, click on the "Add" button to enter the exception into the dictionary. The "As" window may contain any legal string of phonetic characters, including blanks and punctuation, but remember to enter letters in upper case only. The word will now be literally translated into the string below. A typical example may be:

Say: NYSE
As: NUW5 YOH2RK STAA5K EHKSCHEY5NJ

Typing the word in the "Say" window again and hitting <return> should result in the new pronunciation appearing in the "As" window. Repeat the above procedure for all the words you wish to add. Each word that is to be entered into an exceptions dictionary must be done one at a time.

Should you decide to remove something from the exceptions dictionary, enter that word into the "Say" window and hit return (or enter). When the translation appears in the "As" window, click on the "Delete" button. Re-entering the word in the "Say" window should cause the "default" pronounciation to appear in the "As" window.

"ExceptionEdit" will automatically checkpoint your file from time to time as you add words, so if you want to keep an old version, copy it before you begin.

When you are finished with your session, pull down the "File" menu and select "Save". You may now work on a new file or quit by selecting the "Quit" item from the "File" menu.

Selecting the "Quit" item before saving your file will result in the loss of entries made after the last checkpoint was performed.

USING SPEECHLAB

SpeechLab is an application program designed to get you familiar with the MACINTALK system, and in particular with phonetic spelling. When SpeechLab is launched, two windows appear and you may select and edit text in either one. Remember to type a '.' or '?' at the end of your sentence(s).

The window for English text will accept any sequence of characters and attempt to translate them into phonetic code using the default rule set. If you want to bring in one of your own exception files, pull down the "File" menu and select "Use Exceptions File". The result appears in the phonetic input window and is spoken. You may then select the phonetic code window, edit the phonetics and hear the text spoken again.

You may also enter phonetic code directly in the phonetic window adhering to the spelling rules given in the "How to Write Phonetically for MacinTalk" document. You may type more than one sentence in the window and terminate the last sentence with a '#' ('##' in the English window). If you make a phonetic spelling error, MACINTALK will say all the sentences up to the one containing the error and then say nothing. To stop the use of exceptions, select "Use Basic Rules Only".

The various menus allow you to change the operating parameters of the synthesizer and are self explanatory. Remember that the ranges are constrained.

A word of advice:
It is recommended that the phonetic spelling system not be learned from the READER package. In other words, don't use READER to give you a "first shot" at the phonetic spelling. It's much easier to type phonetics from scratch. Believe us, we know. The authors of MACINTALK want good quality speech flowing from the applications that use it. Therefore we strongly urge you to take the one day necessary to learn and become proficient in the use of the phonetic spelling system.

HOW TO WRITE PHONETICALLY FOR MACINTALK

INTRODUCTION

This section will describe in detail the procedure used to specify phonetic strings to the Macintalk speech synthesis driver. No previous experience with phonetics is required. The only thing you may need is a good pronouncing dictionary for those times when you doubt your own ears. The beauty of writing phonetically is that you do not have to know how a word is spelled, just how it is said.

The Macintalk speech synthesizer works on utterances at the sentence level. Even if you want to say only one word, Macintalk will treat it as a complete sentence. Therefore, Macintalk wants one of two punctuation marks to appear at the end of every sentence. These are the period (.) and the question mark (?). If no punctuation appears at the end of a string, Macintalk will append a period to it. The period is used for almost all utterances and will cause a final fall in pitch to occur at the end of a sentence. The question mark is used at the end of yes/no questions only, and results in a final rise in pitch. For example, the question, "Do you enjoy using your Macintosh?", would take a question mark at the end because the answer to the question is either yes or no. The question, "What is your favorite color?" would not take a question mark, and should be followed with a period. Macintalk recognizes other punctuation marks as well, but these will be left for later discussion.

PHONETIC SPELLING

Utterances are usually written phonetically using an alphabet of symbols known as I.P.A., for International Phonetic Alphabet. This alphabet is found at the front of most good dictionaries. The symbols can be hard to learn and are not available on computer keyboards, so the Advanced Research Projects Agency (ARPA) came up with Arpabet, a way of representing each symbol using one or two upper case letters. Macintalk uses an expanded version of Arpabet to specify phonetic sounds.

What is a phonetic sound or "phoneme"? It is a basic speech sound, almost a speech atom. Working backwards, we break sentences into words, words into syllables, and syllables into phonemes. The word "cat" has three letters and (coincidentally) three phonemes. Looking at the table of phonemes we find the three sounds that make up the word "cat". They are: K, AE and T, written as KAET. The word "cent" translates as: S, EH, N and T, or SEHNT. Notice that both words begin with a "c" but because the "c" says "k" in "cat" we use the phoneme K. In "cent" the "c" says "s" so we use the phoneme S. You might also have noticed that there is no C phoneme. The above example clearly illustrates that a word rarely sounds like it looks in English spelling. These examples introduce you to a very important concept: **spell it like it sounds, not like it looks.**

CHOOSING THE RIGHT VOWEL

Phonemes, like letters, are divided into the two catagories of vowels and consonants. A loose definition of a vowel is that it is a continuous sound made with the vocal cords vibrating and air exiting the mouth (as opposed to the nose). All vowels use a two letter code. A consonant is any other sound, such as those made by rushing air (like S or TH), or by interruptions in air flow by the lips or tongue (like B or T). Consonants use a one or two letter code.

In English we write with only five vowels: a, e, i, o and u. Things would be easy if we only said five vowels. Unfortunately we say in excess of 15 vowels. Macintalk provides for most of them. The way to choose the proper vowel is to listen to it. Say the word out loud, perhaps extending the vowel sound you want to hear. Compare the sound you are making to the sounds made by the vowels in the example words to the right of the phoneme list. For example, the "a" in "apple"

sounds the same as the "a" in "cat", not like the "a" in "about", "talk" or "made". Notice also that some of the example words in the list do not even use any of the same letters contained in the phoneme code, like AA as in "hot".

Vowels are divided into two groups, those which maintain the same sound throughout their durations and those which change their sound. The ones which change are called "diphthongs". They are the last six vowels listed in the table. Say the word "made" out loud very slowly. Notice how the "a" starts out like the "e" in "bet" but ends up like the "e" in "beet". The "a" therefore is a diphthong in this word and we would use EY to represent it. Some speech synthesis systems require you to specify the changing sounds in diphthongs as separate elements, but Macintalk takes care of the assembly of diphthongal sounds for you.

CHOOSING THE RIGHT CONSONANT

Consonants are divided into many catagories by phoneticians, but we need not concern ourselves with most of them. Picking the correct consonant is very easy if you pay attention to just two catagories: voiced and unvoiced. A voiced consonant is one made with the vocal cords vibrating, and an unvoiced one is made when they are silent. Sometimes English uses the same letter combinations to represent both. Compare the "th" in "thin" and in "then". Notice that the first is made with air rushing between the tongue and upper teeth. In the second, the vocal cords are vibrating also. The voiced "th" phoneme is DH, the unvoiced is TH. Therefore "thin" is spelled TH, IH, N or THIHN and "then" is spelled DH, EH, N or DHEHN. A sound that is particularly subject to mistakes is voiced and unvoiced "s" spelled Z or S. To put it clearly, "bats" ends in S, "suds" ends in Z. What kind of "s" does "closet" have? How about "close"? Say all of these words out loud to find out. Actually "close" changes its meaning when the "s" is voiced or unvoiced: "I love to be close to you." versus "What time do you close?"

Another sound that causes some confusion is the "r" sound. There are two different r-like phonemes in the Macintalk alphabet, R under the consonants and ER under the vowels. When do you use which? Use ER if the "r" sound <u>is</u> the vowel sound in the syllable. Words that take ER are "absurd", "computer" and "flirt". Use R if the "r" sound precedes or follows <u>another</u> vowel sound in that syllable, such as in: "car", "write" or "craft". "Rooster" uses both kinds of "r". Can you tell which is which?

CONTRACTIONS AND SPECIAL SYMBOLS

There are several phoneme combinations that appear very often in English words. Some of these are caused by our laziness in pronunciation. Take the word "Macintosh" for example. The "i" in the second syllable is almost swallowed out of existence. We would not use the IH phoneme, but would use the IX instead. It is because of this "relaxation" of vowels that we find ourselves using AX and IX very often. Since this relaxation frequently occurs before l, m and n, Macintalk has a shortcut for typing these combinations. So instead of "Macintosh" being spelled MAEKIXNTAASH, we can spell it MAEKINTAASH, making it a little more readable. "Anomaly" goes from AXNAAMAXLIY to UNAAMULIY, and KAAMBIXNEYSHIXN becomes KAAMBINEYSHIN for "combination". Sometimes it may be hard to decide whether to use the AX or IX brand of relaxed vowel. The only way to find out is to try both and see which sounds best.

The other special symbols are used internally by Macintalk. Sometimes they are inserted into, or substituted for part of your input sentence. You can type them in directly if you wish. The most useful is probably the Q or glottal stop; an interruption of air flow in the glottis. The word "Atlantic" has one between the "t" and the "l". Macintalk knows there should be a glottal stop there and saves you the trouble of typing it. But Macintalk is only close to perfect, so sometimes a word or word pair might slip by that would have sounded better with a Q stuck in someplace.

STRESS AND INTONATION

It isn't enough to tell Macintalk what you want said. For the best results you must also tell it how you want it said. This way you can alter a sentence's meaning, stress important words and specify the proper accents in polysyllabic words. These things improve the naturalness and thus the intelligibility of Macintalk's spoken output.

Stress and intonation are specified by numbers. These numbers are single digits 1-9 following a vowel phoneme code. Stress and intonation are two different things but are specified by a single number. Stress is, among other things, the elongation of a syllable. It is a logical term, that is, a syllable is either stressed or not. The presence of a number after the vowel in a syllable indicates stress on that syllable. The value of the number indicates the intonation. From this point onward, these numbers will be referred to as "stress marks". Intonation here means the pitch pattern or contour of an utterance. The higher the stress mark, the higher the potential for an accent in pitch (a rise and fall). A sentence's basic contour is comprised of a quickly rising pitch gesture up to the first stressed syllable in the sentence, followed by a slowly declining tone throughout the sentence, and finally a quick fall to a low pitch on the last syllable. The presence of additional stressed syllables causes the pitch to break its slow, declining pattern with rises and falls around each stressed syllable. Macintalk uses a very sophisticated procedure to generate natural pitch contours based on your marking the stressed syllables.

HOW AND WHERE TO PUT THE STRESS MARKS

The stress marks go immediately to the right of vowel phoneme codes. The word "cat" has its stress marked after the AE so we get KAE5T or KAE9T. You generally have no choice about the location of a number, that is, there is definitely a right and wrong. Either a number should go after a vowel or it shouldn't. Macintalk won't flag an error if you forget to put a stress mark in or if you place one on the wrong vowel. It will only tell you if a stress mark is in the wrong place, such as after a consonant.

The simple rules are:

1) Always place a stress mark in a "content" word. A content word is one that contains some meaning. Nouns, verbs, and adjectives are all content words. "Boat", "huge", "tonsils" and "hypertensive" are all content words; they tell the listener what you're talking about. Words like "but", "the", "if" and "is" are not content words. They don't convey any real world meaning at all, but are required to make the sentence function. So they are given the name "function" words.

2) Always place a stress mark on the accented syllable(s) of polysyllabic words, whether content or function. A polysyllabic word is any word of more than one syllable. "Macintosh" has its stress (or accent as it is often called) on the first syllable and would be spelled MAE5KINTAASH. "Computer" is stressed on the second syllable, giving KUMPYUW5TER. If you are in doubt about which syllable gets the stress, look the word up in a dictionary and you will find an accent mark over the stressed syllable. If more than one syllable in a word receives stress, they usually are not of equal value. These are referred to as primary and secondary stresses. The word "understand" has its first and last syllables stressed, with "stand" getting primary stress and "un" secondary, giving AH1NDERSTAE4ND. Syllables with secondary stress should be marked with a value of only 1 or 2. Compound words (words with more than one root) such as "base/ball", "soft/ware", "lunch/wagon" and "house/boat" can be written as one word but should be thought of as separate words when marking stress. So "lunchwagon" would be spelled LAH5NCHWAE2GIN. Notice that "lunch" got a higher stress mark than "wagon". This is common in compound words; the first word usually receives the primary stress.

WHAT STRESS VALUE DO I USE?

If you get the spelling and stress mark positions correct, you are 95% of the way to a good sounding sentence. The next thing to do is decide on the stress mark values. They can be roughly related to parts of speech, and as a guide, you can use the following table to assign values:

Nouns	5	
Pronouns	2	
Verbs	4	
Adjectives	5	
Adverbs	7	
Quantifiers	7	
Exclamations	9	
Articles	0	(no stress)
Prepositions	0	
Conjunctions	0	
Secondary stress	1, 2	

The above values merely suggest a range. If you want attention directed to a certain word, raise its value. If you want to downplay one, lower it. Sometimes even a function word can be the focus of a sentence. It is quite conceivable that the word "to" in the sentence "Please deliver this to Mr. Wozniak." could receive a stress mark of 9. This would add focus to the word "to" indicating that the item should be delivered to Mr. Wozniak no less than in person.

PUNCTUATION

In addition to the period or question mark that is required at the end of a sentence, Macintalk recognizes several other punctuation marks. These are the dash, comma, and parenthesis. The comma goes where you would normally put a comma in an English sentence. It causes Macintalk to pause with a slightly rising pitch, indicating that there is more to come. The use of additional commas, that is, more than would be required for written English is often helpful. They serve to set clauses off from one another. There is a tendency for a listener to lose track of the meaning of a sentence if the words run together. Read your sentence aloud pretending to be a newscaster. The locations for additional commas should leap out at you.

The dash serves almost the same purpose as the comma, except that the dash does not cause the pitch to rise so severely. A rule of thumb is: Use dashes to divide phrases, commas to divide clauses. For a definition of these terms, consult a high school English book.

The parentheses provide additional information to Macintalk's intonation routine. They should be put around noun phrases of two or more content words. This means that the noun phrase, "a giant yacht" should be surrounded with parentheses because it contains two content words, "giant" and "yacht". The phrase "my friend" should not have paren's around it because it contains only one content word. Noun phrases can get pretty big like, "the silliest guy I ever saw" or "a big basket of fruit and nuts". The paren's really are most effective around these large phrases; the smaller ones can sometimes go without. The effect of the paren's is a subtle one and in some sentences you might not even notice their presence, but in sentences of great length they help provide for a very natural contour.

CONCLUDING REMARKS

This guide should get you off to a good start in phonetic writing for Macintalk. Of course, the only

way to get really proficient is to practice. Many people become good at it in as little as one day. Others make continual mistakes because they find it hard to let go of the rules of English spelling (if there are any).

HINTS FOR INTELLIGIBILITY

There are a few tricks you can use to improve the intelligibility of a sentence. Often, a polysyllabic word is more recognizable than a monosyllabic word. For instance, instead of saying "Mac", say "Macintosh". The longer version contains information in every syllable, thus giving the listener three times the chance to hear it correctly. This can be taken to extremes, so try not to do things like "This program has several insects in it".

Another good practice is to keep sentences to an optimal length. Writing for reading and writing for talking are two different things. Try not to write a sentence that cannot be spoken in one breath. This tends to give the impression that the speaker has an infinite lung capacity. Try to keep sentences confined to one main idea. Run-on sentences tend to loose their meaning after a while.

New terms should be highly stressed the first time they are heard. If you are doing a tutorial or something similar, stress a new term at its first occurrence. All subsequent occurences of that term need not be stressed as highly because it is now "old news".

The above techniques are but a few ways to enhance the performance of Macintalk. You will probably find some of your own. Have fun.

MACINTALK PHONEME TABLE

VOWELS

IY	b**ee**t	IH	b**i**t
EH	b**e**t	AE	b**a**t
AA	h**o**t	AH	**u**nder
AO	t**a**lk	UH	l**oo**k
ER	b**ir**d	OH	b**or**der
AX	**a**bout	IX	sol**i**d

AX and IX should never be used in stressed syllables

DIPHTHONGS

EY	m**a**de	AY	h**i**de
OY	b**oi**l	AW	p**ow**er
OW	l**ow**	UW	cr**ew**

CONSONANTS

R	**r**ed	L	ye**ll**ow
W	**w**ay	Y	**y**ellow
M	**m**en	N	me**n**
NX	si**ng**		
S	**s**ail	SH	ru**sh**
F	**f**ed	TH	**th**in
Z	ha**s**	ZH	plea**s**ure
V	**v**ery	DH	**th**en
CH	**ch**eck	J	**j**u**dg**e
/H	**h**ole	/C	lo**ch**
B	**b**ut	P	**p**ut
D	**d**og	T	**t**oy
G	**g**uest	K	**c**amp

SPECIAL SYMBOLS

DX	pity	Q	kitt_en
(tongue flap)		(glottal stop)	
RX	car	LX	call
	(postvocalic R and L)		
QX =	silent vowel		
UL =	AXL	IL =	IXL
UM =	AXM	IM =	IXM
UN =	AXN	IN =	IXN
	(contractions, see text)		

digits 1-9 stress marks

.	sentence terminator
?	sentence terminator
-	phrase delimeter
,	clause delimeter
( )	noun phrase delimeters

Example of English and Phonetic Texts

Cardiomyopathy. I had never heard of it before, but there it was listed as the form of heart disease that felled not one or two but all three of the artificial heart recipients. A little research produced some interesting results. According to an article in the Nov. 8, 1984, New England Journal of Medicine, cigarette smoking causes this lethal disease that weakens the heart's pumping power. While the exact mechanism is not clear, Dr. Arthur J. Hartz speculated that nicotine or carbon monoxide in the smoke somehow poisons the heart and leads to heart failure.

KAA1RDIYOWMAYAA5PAXTHIY. AY /HAED NEH1VER /HER4D AXV IHT BIXFOH5R, BAHT DHEH5R IHT WAHZ - LIH4STIXD AEZ (DHAX FOH5RM AXV /HAA5RT DIHZIY5Z) DHAET FEH4LD (NAAT WAH5N OHR TUW5), BAHT (AO7L THRIY5 AXV DHIY AA5RTAXFIHSHUL /HAA5RT RIXSIH5PIYINTS). (AH LIH5TUL RIXSER5CH) PROHDUW5ST (SAHM IH5NTRIHSTIHNX RIXZAH5LTS). AHKOH5RDIHNX TUW (AEN AA5RTIHKUL IHN DHAX NOWVEH5MBER EY2TH NAY5NTIYNEYTIYFOH1R NUW IY5NXGLIND JER5NUL AXV MEH5DIXSIN), (SIH5GEREHT SMOW5KIHNX) KAO4ZIHZ (DHIHS LIY5THUL DIHZIY5Z) DHAET WIY4KINZ (DHAX /HAA5RTS PAH4MPIHNX PAW2ER). WAYL (DHIY IHGZAE5KT MEH5KINIXZUM) IHZ NAAT KLIY5R, DAA5KTER AA5RTHER JEY2 /HAA5RTS SPEH5KYULEYTIHD DHAET NIH5KAXTIY1N OHR KAA5RBIN MUNAA5KSAYD IHN DHAX SMOW5K - SAH5M/HAW1 POY4ZINZ DHAX /HAA5RT - AEND LIY4DZ TUW (/HAA5RT FEY5LYER).

Building Applications Using MacinTalk

Program using MacinTalk written in Lisa Pascal must include the statement

```
$USES {obj/SpeechIntf} SpeechIntf;
```

Such programs (and Lisa Assembler programs) must be linked with the file `obj/SpeechAsm.obj`. The file `intrfc/SpeechIntf.text` contains a human readable interface. All of these files are on the **MacinTalk** disk and must be transferred to the Lisa with MacCom.

Macintalk programs can also be written with the Macintosh 68000 Development System (MDS) or with languages which use the MDS linker. Link these programs with the file `SpeechAsm.Rel`.

All the MacinTalk routines described in this document are stack-based; assembly language programs should simply push the routine parameters on the stack before doing a JSR to the routine. As usual, space for any function result is pushed on the stack first, followed by parameters are in left to right order; just pass the address of any parameters larger than a longword (e.g. a parameter of type Str255). For more information please refer to Programming Macintosh Applications in Assembly Language in Inside Macintosh.

Regardless of your development system, make sure that a copy of the Macintalk driver file is on the same disk as the application. The `MacinTalk` file can be found on the **MacinTalk** disk. The disk also contains the tools `SpeechLab` and `ExeceptionEdit` and the example program `SpeakFile` with its data file `TextToSpeak`. The Lisa Pascal source to SpeechLab can be found in the file `example/SpeakFile.text`.

Licensing MacinTalk

You may not distribute a product which uses MacinTalk without the specific written permission of Apple Computer, Inc. Licenses which permit distributing the latest MacinTalk software are available for a moderate annual fee. Contact Apple's Software Licensing Department at (408) 973-4667 for more information.

There may be updates to MacinTalk, but we don't expect the interface to change; i.e. software which works with MacinTalk 1.1 will compile without changes with any subsequent version. If there are updates to the MacinTalk software they will be sent to people who have licensed it. Contact the Licensing Department to determine the latest version before shipping a product which uses MacinTalk.