

VORDE PROGRAMMERS MANUAL

Neda Document Number: 103-102-01

Last Updated: 2000/03/23 18:31:29

Doc. Revision: 1.1.1.1

Neda Communications, Inc.
17005 SE 31st Place
Bellevue, WA 98008

June 15, 1999

Intel is a trademark of the Intel Corporation.
UNIX is a trademark of AT&T.
XENIX is a registered trademark of Microsoft Corporation.
MS-DOS is a registered trademark of Microsoft Corporation.
VRTX is a trademark of Hunter & Ready Inc.
VMS and VAX are trademarks of Digital Equipment Corporation.
IBM and PC are trademarks of International Business Machines Corporation.

This document describes the VoRDE Library, a utility which aids in the consistency of portable implementation of Voice Response Development Environment.

Copyright ©1988, 1989, 1994 Mohsen Banan. Copyright ©1994, 1995 Neda Communications, Inc.

Published by Neda Communications, Inc.
17005 SE 31st Place
Bellevue, WA 98008 USA

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Copyright holders.

List of Tables

7.1	International Phonetic Alphabet	29
7.2	Text-To-Speech Conversion Facilities	30

List of Figures

2.1 Functional Components of VoRDE	12
5.1 Overview of VoRDE Components	20
7.1 Functional components of TTSCF	28

Contents

1	Introduction	9
1.1	About This Product	9
1.2	About This Book	9
1.3	Related Documents	9
2	Overview	11
2.1	Voice Processing Hardware	11
2.1.1	Dialogic Board	11
2.2	Host	11
2.2.1	Board Specific Host Software	11
2.2.2	Neda Open Source Voice Development Environment	11
2.3	Overview Of Text-To-Speech Conversion Facilities (TTSCF)	13
3	Development & Target Environments	15
4	Voice Response Application Development	17
5	C-Module	19
5.1	Introduction	19
5.2	System Specific Environment	21
5.3	Voice Response Virtualization Layer (VRVL)	21
5.3.1	Phone Port Module (PP_)	21
5.3.2	Voice Message Module (VM_)	21
5.4	Hosted Voice Response Layer	22
5.4.1	Synchronous	22
5.4.2	Asynchronous	22
5.5	Generic Voice Response Applications	23
5.5.1	V Compiler	23
5.6	Specific Voice Response Applications	23
5.7	User Interface (UI_)	23
5.7.1	Nested Menu Handler	24

6	Voice Response System Management	25
6.1	Console Interface	25
6.2	Root Menu	25
6.3	Nested Menu Handler	26
7	Text to Speech Conversion Facilities	27
7.1	Phonetic Speech Synthesizer	27
7.1.1	Phonetic Back Ground	27
7.2	Text-To-Speech Conversion	27
7.2.1	Conversion Layers	30
7.2.2	Exception Dictionaries	30
7.2.3	English to IPA Conversion Rules	30
7.3	IPA to SC01 Conversion Rules	31
A	Some Basic Telephone Principles	33
A.1	TONES:	33
A.2	DIAL PULSES:	33
B	Old Unsupported Environment	37
B.1	Delivery Diskettes	37
B.2	Generation Procedure	38
B.2.1	MS-DOS Environment	38
B.2.2	XENIX Environment	38
B.3	Installation Procedure	39
B.4	Delivery Diskettes	39
B.5	Generation Procedure	39
B.5.1	MS-DOS Environment	39
B.5.2	XENIX Environment	40
B.6	Installation Procedure	40
C	Public License	41
C.1	GNU LIBRARY GENERAL PUBLIC LICENSE	41
D	Abbreviations	49

Chapter 1

Introduction

1.1 About This Product

VoRDE is Neda Communications, Inc.'s Voice Response Development Environment. VoRDE provides a convenient environment for developing and managing Voice Response applications. A simple yet powerful language called "V" enables ordinary computer users to create Voice Response applications. More complex applications can be developed in "C" through the services provided by VoRDE. Finally, user-developed "V" application programs, powerful "C" generated applications and a great number of Voice Response Management capabilities are all integrated into a system that makes Voice Response a reality of today.

1.2 About This Book

This publication applies to VoRDE – Neda Communications, Inc.'s Voice Response Development Environment as implemented for a computer equipped with Dialogic hardware and MSDOS operating system.

VoRDE is a collection of generalized commands, voice programming tools, software libraries, and related publications. Typically, a voice application programmer, a voice response system manager, and a casual end user will require access to different subsets of this collection. Anyone coming to VoRDE for the first time will find it useful to read through an overview of concepts and facilities.

The following section provides such an overview. Where appropriate, reference to other sections of this publication or other related publications will be made.

This manual is written in \LaTeX info. \LaTeX info is a documentation system that uses a single source file for both on-line documentation and a printed manual. See the \LaTeX info Manual for more details [8].

The on-line documentation is in the form of an Info file. An on-line Info file is a file formatted so that the Info documentation reading program can operate on it. Info files are divided into pieces called *nodes*, each of which contains the discussion of one topic. M-x info in emacs and xinfo under X11 are two Info documentation reading programs that can be used for on-line manipulation of this manual.

1.3 Related Documents

Other Voice Processing documents that are available through VoRDE.ORG:

- VoRDE Programmers Manual[6] (This document)
- IVR Resources[2]
- Phone Port Control Protocol (PPCP)[4]
- Voice Response Virtualization Layer (VRVL)[5]
- VoRDE V-Module[7]
- Neda's IVR Applications[3]

Chapter 2

Overview

This section describes VoRDE in terms familiar to a computer user. Figure 2.1 represents the functional component of VoRDE.

2.1 Voice Processing Hardware

2.1.1 Dialogic Board

The Dialogic board contains a processor with its own ROM and RAM. Contained in the read-only portion of the memory on the Dialogic board is firmware (DX-FIRMWARE) that provides the basic voice response capabilities for controlling telephone lines.

2.2 Host

2.2.1 Board Specific Host Software

On the host computer, VoRDE components provide an interface to the Dialogic board (i.e. the Board Specific Host Software) and a voice response device independent interface (i.e. Neda Open Source Voice Development Environment).

1. "DX-DRIVER" The Dialogic Board (i.e. Dialog-40 board, Dialog-41 board, etc.) is accessed through the DX-DRIVER (i.e. for Dialog-40 board, the driver would be D40-DRIVER). This driver must be installed before any program can access the Dialogic board.
2. "DX-LIB" DX-LIB is a "C" library that makes use of the driver-provided services conveniently available to "C" application programs.

2.2.2 Neda Open Source Voice Development Environment

1. "VR-LIB" VR-LIB is a device independent Voice Response "C" library. It provides basic Voice Response Services to application programs. It is strongly recommended that application programs always use this interface module.

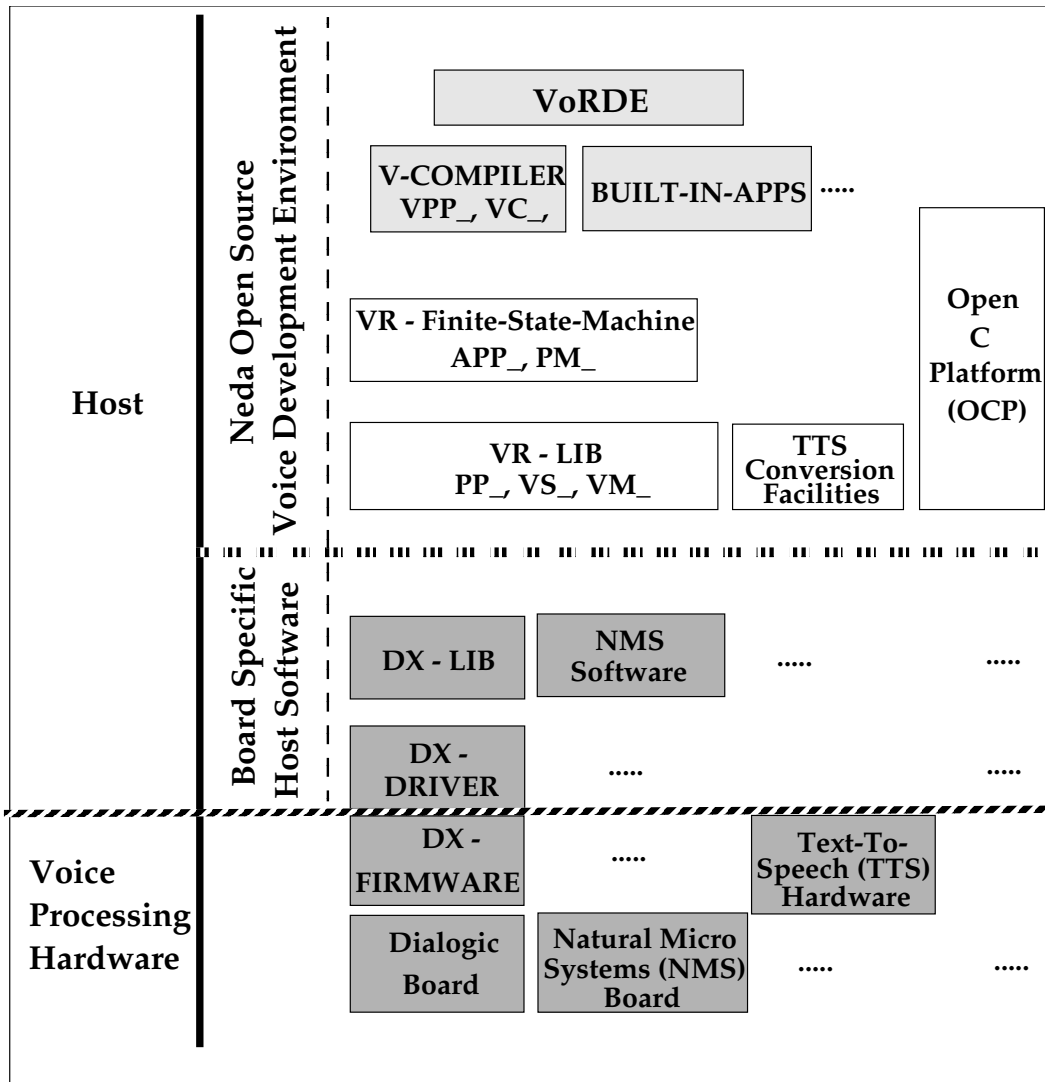


Figure 2.1: Functional Components of VoRDE

2. "VR-Finite-State-Machine:" is a set of facilities that allow for generation and execution of state tables. A voice response application can be created and executed through the services provided by this module.
3. "V-COMPILER" "V" is a simple yet powerful Voice Response development language. The V-COMPILER generates applications based on programs written in "V". See V-GUIDE for more information.
4. "BUILT-IN-APPS" Powerful Voice Response applications developed in "C" can co-exist with applications generated in "V". One or more of these built-in applications can be integrated with the V-COMPILER. See BUILT-GUIDE for more information.
5. "VoRDE" VoRDE integrates the built-in applications, the "V-COMPILER", and the voice response system management capabilities into a single Voice Response Program.

VoRDE services can logically be divided into two basic categories:

1. Voice Response Application Development
2. Voice Response System Management

Details of each will be described later.

2.3 Overview Of Text-To-Speech Conversion Facilities (TTSCF)

1. "Hardware Interface:" This module provides the interface to the TTSCF board.
2. "SC01 Interface:" This module provides the low level routines necessary for the conversion of SCA to SCB conversion. Setting the parameters of SC01 are also done through this module.
3. "IPA2SC conversion:" a [left] [match] [right] [out] set of rules are used to convert IPA strings to SCA strings.
4. "ENG2IPA conversion:" ENG2IPA conversion is done through NRL rules.
5. "Binary Table Editor:" Is a very simple data base manager. It handles simple tables (relations). All data base items are of the form ;"key", "value"; where both "key" and "value" are strings.
6. "SCA Dictionary:" Tuples of this relation are of the form "English-word" "sca-phonemes". English-word is used as the key for locating sca-phonemes.
7. "IPA Dictionary:" Tuples of this relation are of the form "English-word" "ipa". English-word is used as the key for locating ipa.
8. "Phrase Analyzer:" Decides on how an English phrase should be treated. Depending on the option settings, does the translation in the proper format.

TTSCF services can logically be categorized as Phonetic Speech Synthesizer.

Chapter 3

Development & Target Environments

For more detail information on Development and Target Environments, please refer to **Open C Platform** document.
[\[1\]](#)

Chapter 4

Voice Response Application Development

Voice Response applications can be developed through a simple Voice Response application generator language called "V". The V-Module section of this document describes this language.

Voice Response applications can also be developed in "C". A complete library of Voice Response Development services (VR-LIB) is provided. Applications developed through (VR-LIB) can co-exist with other applications written in "V". The C-Module section of this document describes the proper use of VR-LIB.

A Voice Response application is a sequence of desired Voice Response actions. A Voice Response application can be assigned to one or more telephone lines (Phone-Port). Once assigned to a Phone-Port, that telephone line will perform the actions defined in the application.

Chapter 5

C-Module

5.1 Introduction

VoRDE consists of a number of modules that are clustered into five principle groups.

1. V Compiler.
 - (a) The pre-processor.
 - (b) VC. The V Compiler.
2. Voice Response Finite State Machine.
 - (a) Application Management Module.
 - (b) PM. Phone Port Finite State Machine Module.
3. Voice Response Facilities.
 - (a) Phone Port Module.
 - (b) VS. Voice Stream Module.
 - (c) VM. Voice Message Module.
 - (d) D4. Dialog 40 interface Module.
4. Global Facilities.
 - (a) OCP. Open C Platform.
5. User Interface
 - (a) UI. User Interface and Menu Module.

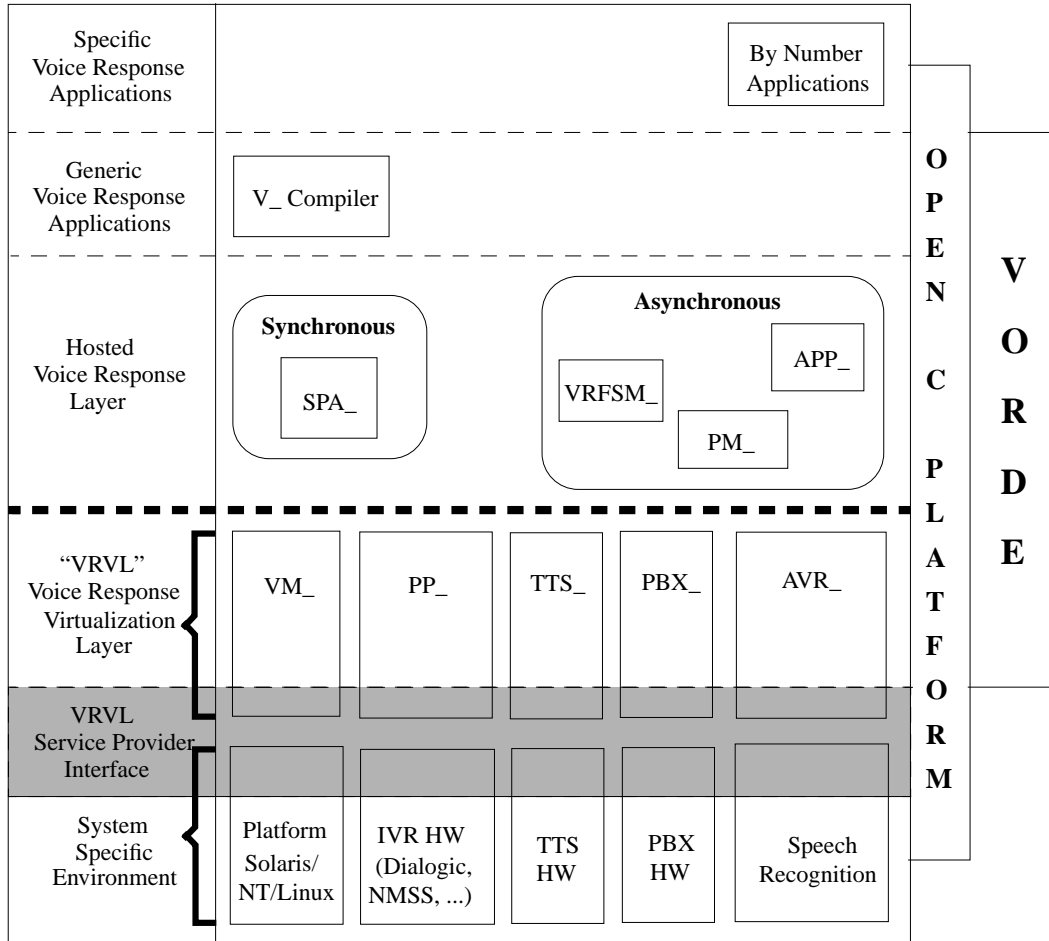


Figure 5.1: Overview of VoRDE Components

5.2 System Specific Environment

5.3 Voice Response Virtualization Layer (VRVL)

5.3.1 Phone Port Module (PP_)

The PP_ module is the interface to an abstract telephone handler module. This abstract interface is asynchronous. An action is requested and some time later an event is observed.

several entities that may be using it. This is supported through the concept of a Service Access Point (SAP). A Service Access Point is defined as the interface between a service user and a service provider.

Each SAP is identified by the PP_ module through a SAP-Address (PP_SapAddr).

delete a SAP (PP_sapDelete).

user entity.

One or more Phone-Ports can be associated with a SAP. Association of a Phone-Port to a SAP is accomplished through the PP_attach facility. Once a Phone-Port has been attached to a SAP, all events associated with that Phone-Port will be delivered to the SAP to which it is attached. A Phone-Port can be not be attached to more than one SAP.

Action Primitives

An action primitive is a request to the

PP_Events

```
typedef struct PP_Event {
struct PP_Event *next;
    struct PP_Event *prev;
    PP_PortDesc portDesc;
    PP_EventId evtId;
    union EventData {
        PP_PlayCnf playCnf;
        PP_GetDtmfCnf getDtmfCnf;
        PP_RecordCnf recordCnf;
        PP_DialCnf dialCnf;
    } evtData;
} PP_Event;
```

5.3.2 Voice Message Module (VM_)

VM_ module operates on voice messages. A voice message in the D40/MS-DOS implementation is a DOS file.

5.4 Hosted Voice Response Layer

5.4.1 Synchronous

Synchronous Phone Port Application Layer (SPA_)

```
#include "spa.h"

void
SPA_vMsgDomainSet(String domain)

String
SPA_vMsgDomainGet(void)

SPA_PlayResult
SPA_vMsgPlayN(PP_PortDesc portDesc, String vMsg, Int nuDtmfs)

SPA_PlayResult
SPA_wordSeqPlayN(PP_PortDesc portDesc, String wordSeq, Int nuDtmfs)

SuccFail
SPA_dtmfWaitFor(PP_PortDesc portDesc, Int nuDtmfs, String dtmfTerms,
Int wait, Char *buf, Char *bufEnd)

SuccFail
SPA_mfWaitFor(PP_PortDesc portDesc, Int nuDtmfs, String dtmfTerms,
Int wait, Char *buf, Char *bufEnd)

SuccFail
SPA_dtmfGenerate(PP_PortDesc portDesc, String dtmfStr)
```

5.4.2 Asynchronous

Application Management (APP_)

A voice response application in VoRDE environment is a state table that is executed by PM_Machine. PP_ module can be used to assign a name to an application and to assign the application to one or more Phone-Ports.

Phone Port Machine (PM_)

PM_ module provides the service of dynamic creation of state tables and event driven execution of these state tables. Several state tables (Applications) may exist within a program (process). Each of these state tables may be associated with one or more Phone-Ports.

Sections 5.4.2 address the categories of services provided by the PM_ module.

Voice Response Finite State Machine

The PM_ state machine, takes as input the
the Phone-Port.

There is a PM_Machine associated with each Phone-Port. Each PM_Machine consists of:

- A Stack.
- Memory for static variables.
- State Information.

Generation of a State

PM_ states may dynamically be generated and linked together to generate state tables (Voice Response Applications).

A state is normally generated through a facility which has the form PM_genStateName. % 'StateName'% is the name of the desired state. For example to generate a state that would play a file, the user would:

```
{
    PM_State *state;

    state = PM_genPlayN(0, "hello.vox");
}
```

Linkage of States

A state table is a linked set of states. PM_States are linked together through PM_Events.

PM_addState facility is used to link two states together with an event.

5.5 Generic Voice Response Applications**5.5.1 V Compiler**

The V compiler consists of two elements. A macro pre-processor identical in capabilities to the C pre-processor, and an application state generator.

Pre-Processor

The pre-processor is a portation of "Decus CPP" to MS-DOS environment.

State Generator

The state generator, parses the syntax specifies in the V-Module section and produces states according to its input.

5.6 Specific Voice Response Applications**5.7 User Interface (UI_)**

```
#include "ui.h"
```

```
#include "ui_menu.h"
```

The user interface to VoRDE is a Nested Menu Handler (NMH).

5.7.1 Nested Menu Handler

NMH interacts with the user through the console (Key Board and Screen). Application programs interface to NMH through two data structures.

1. "MenuItem" A data abstraction describing a menu item. A MenuItem has associated with it either a function or a MenuDef.
2. "MenuDef" A data abstraction describing a menu. A menu is a set of MenuItems.

Appendix 'A' has a description of the commands and features supported by NMH.

User of NMH defines a menu by creating a MenuDef and a set of MenuItems that go with it. Each MenuItem has associated with it either a pointer to a function or a pointer to a MenuDef. When the console user issues the command corresponding to a MenuItem which has a function associated with it. NMH indirectly invokes the user supplied function. Syntax of the invocation is:

```
(*menuFunc)(argc, argv, env);  
Int argc;  
String *argv;  
UI_Environment *env;
```

argc and argv are the translated information that was obtained from the console user. env describes the user interface environment and must only be used in conjunction with UI facilities. It is the responsibility of the application program to interpret argc and argv.

Chapter 6

Voice Response System Management

Once one or more applications have been declared, defined and assigned to Phone-Ports, each Phone-Port executes the application assigned to it. Voice response applications can be developed in V or in C, as described in "Voice Response Application Development" of this publication.

VC is the name of the program that integrates C developed applications, the V compiler, and Voice Response management capabilities. While voice response applications are being executed, the status of the Voice Response system can be monitored and managed through an interactive menu oriented interface running on the console. The following sections describe the features and the services provided by the console menu interface.

6.1 Console Interface

A simple yet powerful user interface is provided through the Nested Menu Handler (NMH). Appendix NMH-APDX outlines the basic features provided by this menu handler. The following sections describe specifics of each menu level.

6.2 Root Menu

The prompt for the root menu is "VoRM". A list of available menu options follows:

1. "A- Available Applications"

A list of available applications is reported.

2. "S- Phone-Port Status"

The current status of each Phone-Port is reported.

3. "P- Phone-Port Assignments"

The current association of Phone-Ports to applications is reported.

6.3 Nested Menu Handler

Nested Menu Handler (NMH) is a simple menu based user interface.

A menu command is the first character of the first word on the command line. Only the first character is significant. A '?' provides a list of available commands. Illegal commands result in a reminder of the availability of '?'.

If the command corresponds to a menu function, that function is invoked. If the menu function expects arguments, the arguments can be issued on the command line.

If the command corresponds to another menu, that menu becomes active. Every menu level has a "menu prompt" associated with it. When the user enters a menu, previous menu prompts are concatenated to the current prompt and displayed. This reminds the user of which submenu he is currently at.

Characters listed in the following table are treated specially by the menu handler:

'?'	Display all menu items.
' '	Argument Separator.
';'	Command Separator.
'\bs '	Command terminator.
'x'	Pop the menu Stack.
'\bs '	Go to Root level menu.

All Neda Communications, Inc. programs conform to the Proposed Syntax Standards for Unix System Commands put forth by Kathy Hemenway and Helene Armitage of AT&T Bell Laboratories in Unix/World, Vol.1, No.3, 1984. Some of these rules are:

1. Command names must be between 2 and 9 characters.
2. Command names must include lower case letters and digits only.
3. An option name is a single character.
4. Options are delimited by '-'.
5. Options with no arguments may be grouped (bundled) behind one delimiter.
6. Option-arguments may not be optional.
7. Arguments immediately follow options, separated by whitespace.
8. The order of options does not matter.
9. A '-' preceded and followed by whitespace means standard input.
10. A group of bundled options may end with an option that has an argument.
 1. "SCA" SC01 Ascii codes.
 2. "SCB" SC01 binary codes.
 3. "IPA" International Phonetic Alphabets.
 4. "NRL" Naval Research Laboratories.

CONTENTS"

Chapter 7

Text to Speech Conversion Facilities

This section describes TTSCF in terms familiar to a computer user. Figure 7.1 represents the functional components of TTSCF.

7.1 Phonetic Speech Synthesizer

7.1.1 Phonetic Back Ground

There are a limited number of sounds that make up words in human languages. In linguistics these sounds are referred to as phonemes. Any word may be created by a sequence of phonemes. Phoneme Synthesis Technique is best suited for applications that require unlimited vocabularies.

Table 7.1 presents a list of English phonemes. International Phonetic Alphabet (IPA) assigns a set of symbols to phonemes. Most of these symbols are not available on an ordinary typewriter setting. A set of one and two character Latin alphabet codes are assigned to phonemes. These codes will be used for IPA expressions.

Electronic Phoneme synthesizers are devices capable of producing the phoneme sounds through corresponding digital code. Votrax SC01 is one popular phoneme synthesizers currently available.

7.2 Text-To-Speech Conversion

Ideally the input to a phonetic speech synthesizer should be a string of ASCII characters that is readable by a human. In order to provide complete text-to-speech capabilities, TTSCF uses a multi step process that converts ASCII text into natural sounding speech waveforms. Individual steps of this process are:

1. "Normalization:" The text normalizer takes the ASCII input and expands abbreviations, numbers and monetary amounts to their full word form. It also processes punctuation, nonalphabetic input characters and special pronunciation.
2. "Syntactic Analysis:" A crude syntactic analyses of the sentence is performed based on the syntactic role of function words and verbs that were detected.
3. "Phonemic Translation:" Incoming text is compared against an exception dictionary. If a match for a word is found, the associated phoneme string and stress information are retrieved and passed directly to the next process.

IPA CODE	Example	Phonetic Description
EY	gAte	
EH	gEt	
AE	fAt	Front Vowel
AA	fAther	
AO	lAWn	
OW	lOne	
UH	fUll	
UW	fOOl	
ER	mURdER	
AX	About	
AH	bUt	
AY	hIde	
AW	hOW	
OY	tOY	
p	Pack	Unvoiced
b	Back	Voiced
t	Time	Unvoiced
d	Dime	Voiced
k	Coat	Unvoiced
g	Goat	Voiced
f	Fault	Unvoiced Fricative Consonant
v	Vault	Voiced Fricative Consonant
TH	eTHer	Unvoiced Fricative Consonant
DH	eiTHer	Voiced Fricative Consonant
s	Sue	Unvoiced Fricative Consonant
z	Zoo	Voiced Fricative Consonant
SH	leaSH	Unvoiced Fricative Consonant
ZH	leiSure	Voiced Fricative Consonant
HH	How	Whisper Consonant
m	suM	Nasal Consonant
n	suN	Nasal Consonant
NG	suNG	Nasal Consonant
l	Laugh	Liquid Semivowels
w	Wear	Liquid Semivowels
y	Young	Glide Semivowel
r	Rate	Glide Semivowel
CH	CHar	Affricate Consonant
j	Jar	Affricate Consonant
WH	WHere	

Table 7.1: International Phonetic Alphabet

Input	Function	Output
eng_string	x_eng2ipa	ipa_string
ipa_string	x_ipa2sca	sca_string
sca	sca2b	scb
scb	scb2a	sca
sca_string	sca_say	VOICE
scb_string	scb_say	VOICE
sca	sca_utter	VOICE
scb	scb_utter	VOICE

Table 7.2: Text-To-Speech Conversion Facilities

If the dictionary search fails to find a match, the word is assigned a phoneme string and stress pattern based on an extensive set of rules that are similar to the rules used for reading English aloud. The mechanism of assigning phonemic and stress information to a word is called synthesis-by-rule.

4. "Parameter Generation:" Given the phoneme string, lexical stress and syntactic information, appropriate parameters should be generated and supplied to the acoustic synthesizer.

7.2.1 Conversion Layers

The following table provides a list of the functions used for conversion required for transformation of text to speech.

7.2.2 Exception Dictionaries

Two user maintained phonetic dictionaries are used by TTSCF software. These phonetic dictionaries may be used to conversion rules when desired.

English to IPA Dictionary

English to IPA dictionary is a relation that contains tuples of the form: ;"english-word": "ipa-string";. This dictionary can be maintained by the "Binary Table Editor".

English to SCA Dictionary

English to SCA dictionary is a relation that contains tuples of the form: ;"english-word": "sca-string";. This dictionary can be maintained by the "Binary Table Editor".

7.2.3 English to IPA Conversion Rules

English to IPA conversion is done through a set of phonetic rules published by the Naval Research Laboratories. Each rule has the form:

```
[left] [match] [right] [out]
```

7.3 IPA to SC01 Conversion Rules

A set of rules are used to determine the conversion. Each rule has the form:

[left] [match] [right] [out]

Appendix A

Some Basic Telephone Principles

In the United States you can dial a phone number using two completely independent methods: Tones and Dial Pulses. The information presented below has been gathered from Refs. 1 and 3. Since in this project the existence of DTMF facilities is being assumed and is the method used, more emphasis will be placed on the Tone method.

A.1 TONES:

Each time you hold down a key on your push button telephone set a pair of audio frequency signals is transmitted over the telephone voice channel. Central-office switching facilities decode these tones and connect the desired circuits based on the sequence of tone pairs received. Each tone must last long enough and there must be adequate separation between them. A tone pair duration of about 150 ms and a separation of about 75 ms works.

Each of these tones is composed of two pure sine waves of different frequencies superimposed on each other. These two frequencies explicitly represent one of the digits on the telephone key-pad.

The telephone key-pad can be thought of as a 4 row by 3 column matrix. Associated with each row is a specific frequency belonging to the low group (697 to 941 HZ) and corresponding to each column is a unique frequency of the high group (1209 to 1633 HZ). All the keys in a given row or column have one tone in common (see Table 1.1). For example, pressing the digit "9" (row 3 and column 3) produces 852 Hz and 1477 Hz tones simultaneously, while pressing a "5" produces 770 and 1336 HZ tones.

The full DTMF-encoding standard defines four rows and four columns for a total of 16 two-tone combination. Standard telephones use only 12 of these combinations. Depending on the application, these extra codes may be useful. Most tone decoding devices allow a 2 per cent tolerance on DTMF frequencies. This creates a range of acceptable frequencies, which is demonstrated in Table (1.2).

A.2 DIAL PULSES:

When you pick up the receiver on a telephone, an electrical connection is made to the lines leading to the central office. When you replace the receiver on the cradle the connection is broken or interrupted. This applies to both push-button and rotary dial telephones.

By periodically breaking the connections leading to the central office a number can be dialed. The number of interruptions is equal to the digit dialed, with the exception that ten interruptions corresponds to zero. These pulses may be generated at the rate of ten times per second and there should be a 1/2 second delay between each two digits.

DTMF DIALING MATRIX

		HIGH GROUP			
		column 0	column 1	column 2	column 3
		1209hz	1336 hz	1477 hz	1633 hz
LOW GROUP	Row 0	1	2	3	A
	697 hz				
	Row 1	4	5	6	B
	770 hz				
	Row 2	7	8	9	C
	852 hz				
	Row 3	*	0	#	D
	941 hz				

The dialing matrix of DTMF (Dual Tone Multiple Frequency) signaling system. The low group frequencies correspond to the matrix row; the high group frequencies correspond to the column. Column 3 is for special applications and is not normally used.

ACCEPTABLE DTMF FREQUENCIES

DTMF FREQUENCY (HZ)	LOWER DETECTION FREQUENCY LIMIT (HZ)	HIGHER DETECTION FREQUENCY LIMIT (HZ)	ACCEPTABLE FREQUENCY RANGE (HZ)
697	683	711	28
770	755	786	31
852	834	869	35
941	922	960	38
1209	1184	1233	49
1336	1309	1363	54
1477	1447	1507	60
1633	1600	1666	66

The standard DTMF frequencies with the minimum and maximum values accepted within the 2 percent tolerance of most digital tone decoding devices (see Ref. 1).

The rotary dial on the telephone is a mechanical device which periodically breaks the connection leading to the central office. When the rotary dial is released, as it travels back to its resting position, it breaks the connection at a rate of ten times per second thus dialing the digit.

Numbers can also be dialed by pushing the cradle switch button at a rate of ten times per second. A solenoid plunger that is mounted to depress and release the cradle switch on the telephone set may be used to dial numbers using the dial pulses technique.

Appendix B

Old Unsupported Environment

B.1 Delivery Diskettes

VoRDE is distributed on three diskettes.

1. A port of Decus CPP to MS-DOS environment. The source is in "Shell Archive" format. The archive file is "vpp.shr". A mini-shell called "sharsh.exe" is provided for expansion of the archive in MS-DOS environment. A brief description of the files on the source diskette follows:
 - (a) "readme" Electronic copy of this section of the manual.
 - (b) "vpp.shr" C Source to VPP.
 - (c) "sharsh.exe" A mini-shell capable of expanding VC.SHR.
2. VC Source diskette contains the C source to VoRDE and the V compiler. The source is in "Shell Archive" format. The archive file is "vc.shr". A mini-shell called "sharsh.exe" is provided for expansion of the archive in MS-DOS environment. A brief description of the files on the source diskette follows:
 - (a) "readme" Electronic copy of this section of the manual.
 - (b) "vc.shr" C Source to VoRDE and the V Compiler.
 - (c) "sharsh.exe" A mini-shell capable of expanding VC.SHR.
3. Binary diskette contains executable for the V-Compiler, a sample V-source file and the voice files necessary for the example program. A brief description of the files on the binary diskette follows:
 - (a) "readme" Electronic copy of this section of the manual.
 - (b) "vpp.exe" The V pre-processor. Needed by the V compiler.
 - (c) "vc.exe" The V Compiler.
 - (d) "config.v" Configuration part of the example V program.
 - (e) "snow.v" Definition of the example V program.
 - (f) "choose.vox,steven.vox,white.vox,crystal.vox,thankyou.vox" Voice files used by the example V program.

B.2 Generation Procedure

This section describes the procedure for the generation of VoRDE from the C source. This Generation procedure supports two environments.

1. MSC 4.0 in MS-DOS environments.
2. DOS cross development environment under SCO XENIX.

VoRDE has been developed with portability in mind. VoRDE may easily be ported to any environment that has a C compiler that conforms to the proposed ANSI X3J11 specification.

B.2.1 MS-DOS Environment

Make sure that your MSC compiler is properly installed. Make sure that the compiler may be invoked through "msc" command. Follow these steps:

1. Create an empty directory on your hard disk.
2. Go to that directory.
3. Copy vpp.shr and % 'sharsh.exe'% to the empty directory.
4. Invoke "sharsh vpp.shr".
5. Copy vc.shr and % 'sharsh.exe'% to the empty directory.
6. Invoke "sharsh vc.shr".
7. Invoke "build".

When build terminates, vc.exe and vpp.exe are produced.

B.2.2 XENIX Environment

Follow these steps:

1. Create an empty directory.
2. Go to that directory.
3. Copy vpp.shr to the empty directory.
4. Invoke "sh vpp.shr".
5. Copy vc.shr to the empty directory.
6. Invoke "sh vc.shr".
7. Invoke "make".

When make terminates, vc/vc.exe and vpp/vpp.exe are produced.

B.3 Installation Procedure

Copy vc.exe and % 'vpp.exe'% from the binary diskette to your executables directory. Install the Dialog 40 driver. Verify the installation of the V Compiler by invoking "vc config.v" from the binary diskette directory. Call the system and see it perform.

B.4 Delivery Diskettes

PCSTB-1 is distributed on two diskettes.

1. PCSTB-1 Source diskette contains the C source to PCSTB-1. The source is in "tar" format. The archive file is "pcstb.tar". A tar program called "tar.exe" is provided for expansion of the archive in MS-DOS environment. A brief description of the files on the source diskette follows:
 - (a) "readme" Electronic copy of this section of the manual.
 - (b) "pcstb.tar" C Source to PCSTB-1 software.
 - (c) "tar.exe" An archive program capable of expanding pcstb.tar.
2. Binary diskette contains executable for the PCSTB-1 nested menu handler user interface and a number of other related programs along with the data files associated with them. A brief description of the files on the binary diskette follows:
 - (a) "readme" Electronic copy of this section of the manual.
 - (b) "pcstb.exe" Nested menu handler interface to PCSTB-1 board.
 - (c) "eng2ipa.exe" English to IPA conversion program.
 - (d) "ipa2sca.exe"

B.5 Generation Procedure

This section describes the procedure for the generation of PCSTB-1 from the C source. This Generation procedure supports two environments.

1. MSC 4.0 in MS-DOS environments.
2. DOS cross development environment under SCO XENIX.

PCSTB-1 has been developed with portability in mind. PCSTB-1 may easily be ported to any environment that has a C compiler that conforms to the proposed ANSI X3J11 specification.

B.5.1 MS-DOS Environment

Make sure that your MSC compiler is properly installed. Make sure that the compiler may be invoked through "msc" command. Follow these steps:

1. Create an empty directory on your hard disk.

2. Go to that directory.
3. Copy pcstb1.tar and % 'tar.exe'% to the empty directory.
4. Invoke "tar xvf pcstb1.tar".
5. Invoke "build".

When build terminates, pcstb1/bs pcstb1.exe and eng2ipa/bs eng2ipa.exe are produced.

B.5.2 XENIX Environment

Follow these steps:

1. Create an empty directory.
2. Go to that directory.
3. Copy pcstb1.tar to the empty directory.
4. Invoke "tar xvf pcstb1.tar".
5. Invoke "make".

When make terminates, pcstb1/pcstb1 and eng2ipa/eng2ipa are produced.

B.6 Installation Procedure

Copy pcstb1.exe and % 'eng2ipa.exe'% from the binary diskette to your executables directory. Execute demo.bat and see the system perform.

Appendix C

Public License

C.1 GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright ©1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what

they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

2. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

3. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- (a) The modified work must itself be a software library.
- (b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- (c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- (d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

4. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

5. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

6. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

7. As an exception to the Sections above, you may also compile or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- (a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- (b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- (c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- (d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the “work that uses the Library” must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

8. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:
 - (a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
 - (b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
9. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
10. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

11. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
12. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

13. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
14. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

15. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

16. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
17. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

one line to give the library’s name and an idea of what it does. Copyright (C) year name of author

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library ‘Frob’ (a library for tweaking knobs) written by James Random Hacker.

signature of Ty Coon, 1 April 1990 Ty Coon, President of Vice

That’s all there is to it!

Appendix D

Abbreviations

1. "SCA" SC01 Ascii codes.
2. "SCB" SC01 binary codes.
3. "IPA" International Phonetic Alphabets.
4. "NRL" Naval Research Laboratories.

Bibliography

- [1] Neda Public Document. *Open C Platform*. Neda Published Document 103-103-01, Neda Communications Inc, Bellevue, WA, October 1996. Online document is available at <http://www.public.neda.com/pubs/biblio/103-103-01/index.html>.
- [2] Neda's Voice Processing Public Document. *IVR Resources*. Neda Published Document 103-102-02, Neda Communications Inc, Bellevue, WA, October 1996. Online document is available at <http://www.vorde.org/pubs/biblio/103-102-02/index.html>.
- [3] Neda's Voice Processing Public Document. *Neda's IVR Applications*. Neda Published Document 103-102-06, Neda Communications Inc, Bellevue, WA, November 1999. Online document is available at <http://www.vorde.org/pubs/biblio/103-102-06/index.html>.
- [4] Neda's Voice Processing Public Document. *Phone Port Control Protocol (PPCP)*. Neda Published Document 103-102-03, Neda Communications Inc, Bellevue, WA, November 1999. Online document is available at <http://www.vorde.org/pubs/biblio/103-102-03/index.html>.
- [5] Neda's Voice Processing Public Document. *PP Layer – PP Dialogic Device Driver*. Neda Published Document 103-102-04, Neda Communications Inc, Bellevue, WA, November 1999. Online document is available at <http://www.vorde.org/pubs/biblio/103-102-04/index.html>.
- [6] Neda's Voice Processing Public Document. *VoRDE Programmers Manual*. Neda Published Document 103-102-01, Neda Communications Inc, Bellevue, WA, November 1999. Online document is available at <http://www.vorde.org/pubs/biblio/103-102-01/index.html>.
- [7] Neda's Voice Processing Public Document. *VoRDE V-Module*. Neda Published Document 103-102-05, Neda Communications Inc, Bellevue, WA, November 1999. Online document is available at <http://www.vorde.org/pubs/biblio/103-102-05/index.html>.
- [8] Robert J. Chassell Richard Stallman and Michael Clarkson. *The LaTeXinfo Documentation Format*. The Free Software Foundation, 675 Massachusetts Ave., Cambridge MA, 02139, 1990.