



**Text to Speech
Guide
ODTVision**

Table of Contents

Text-to-Speech (Optional)	5
Variables.....	5
VoiceRate = -10 to +10.....	5
VoiceVolume = 0 to 100.....	5
VoiceEnableXML = True/False.....	5
VoiceName = “Microsoft Mary”	5
VoicePunctuation = True/False.....	5
Commands.....	6
VoiceText “Text to speak”.....	6
VoiceXML type, [parms] [,text]	7
Txt = VoiceXML type, [parms] [,text].....	7
VoiceFile “file.ext”	8
rtn = VoiceSpeakToFile file.ext, [txt]	9
rtn = VoiceSpeak [txt] [, <i>stopchar</i> , <i>maxtime</i> , <i>prompt</i>]	9
VoiceSpeak [txt]	9
VoiceXML types	10
VoiceXML types – State control	10
VoiceXML types - Insertion	14
VoiceXML types – Context	15
VoiceXML types – Context (dates and times).....	16
VoiceXML types – Context (numbers).....	21
VoiceXML types – Context (misc.).....	24
Using Text to Speech.....	26
American English Phoneme Representation.....	29
Symbolic and Numerical Representation.....	29
American English Phoneme Table.....	30

Text-to-Speech (Optional)

Variables

VoiceRate = -10 to +10

Sets the speaking rate (speed) of the voice for text-to-speech.

Values are in the range from -10 to 10, which represent the slowest and the fastest speaking rates, respectively.

VoiceVolume = 0 to 100

Sets the base volume level (loudness) of the voice for text-to-speech.

Values are in the range from 0 to 100, representing the minimum and maximum volume levels, respectively.

VoiceEnableXML = True/False

If this is true, the input text will be parsed for XML markup.
The default is False

```
voiceenablexml = true
```

VoiceName = "Microsoft Mary"

Changes the default voice used for text-to-speech.

Examples of Voice Names are "Microsoft Mary" and "Microsoft Mike."

Use the Text-to-Speech test program, or the Speech control panel option to determine what voices are available.

VoicePunctuation = True/False

If this is set to True, Punctuation characters should be expanded into words (e.g. "This is it." would become "This is it period").

The default is False

Commands

VoiceText “Text to speak”

Puts text in the Text-to-speech Queue. Multiple VoiceText, VoiceXML, and VoiceFile commands can be queued. The Text-to-speech Queue is emptied when either the VoiceSpeakToFile or the VoiceSpeak commands are performed.

Example:

OrderQty = 22

Item = “Widgets”

VoiceText “You ordered ”

VoiceXML number, OrderQty

VoiceText Item

Rtn = VoiceSpeak

Would speak:

“You ordered twenty-two Widgets”

VoiceXML type, [parms] [,text]

Wraps the text supplied in XML tags as specified by the type parameter and puts the XML text in the Text-to-speech Queue. Multiple VoiceText, VoiceXML, and VoiceFile commands can be queued. The Text-to-speech Queue is emptied when either the VoiceSpeakToFile or the VoiceSpeak commands are performed.

See “VoiceXML tags for more information.

Example:

```
OrderQty = 22  
Item = “Widgets”
```

```
VoiceText “You ordered ”  
VoiceXML number, OrderQty  
VoiceText Item  
Rtn = VoiceSpeak
```

Would speak:
“You ordered twenty-two Widgets”

Txt = VoiceXML type, [parms] [,text]

Wraps the text supplied in XML tags as specified by the type parameter and returns the XML text in the returned variable. This allows you to perform multiple XML tags on text before adding them to the Queue with a VoiceText command.

See “VoiceXML tags for more information.

Example:

```
vSilence = VoiceXML SILENCE 1000  
vText = “Say this first ” CAT vSilence  
vText = vText CAT “ Say this second”  
vText = VoiceXML EMPHASIS vText  
VoiceText vText
```

Would speak:
“Say this first” (one second of silence) “Say this second”
and the whole sentence would be spoken with emphasis.

VoiceFile “file.ext”

Puts all of the text contained in the supplied file in the Text-to-speech Queue. Multiple VoiceText, VoiceXML, and VoiceFile commands can be queued. The Text-to-speech Queue is emptied when either the VoiceSpeakToFile or the VoiceSpeak commands are performed.

There is no “default” directory for the file name specified. Enter the full path and file name of the file to speak as in the example below.

Example:

```
VoiceFile “C:\Announcements\Sale.txt”  
Rtn = VoiceSpeak
```


rtn = VoiceSpeakToFile file.ext, [txt]

Puts the (optional) text supplied as a parameter in the Text-to-Speech Queue and creates a voice file from all of the text that has been placed in the Text-to-Speech Queue. The Text-to-Speech Queue will be empty after this operation.

If the extension of the file is .WAV, only a wav file will be created. If the extension is .VOX or no extension is given, both a WAV and a VOX file is created.

This command is allowed in a control window script.

rtn = VoiceSpeak [txt] [, stopchar, maxtime, prompt]

VoiceSpeak [txt]

Puts the (optional) text supplied as a parameter in the Text-to-Speech Queue and creates a voice file from all of the text that has been placed in the Text-to-Speech Queue. The Text-to-Speech Queue will be empty after this operation.

If a return value is not specified, the voice file is placed on the Voice speak Queue and will not be played until the next Speak or Play command that returns a value.

VoiceXML types

VoiceXML types – State control

Volume, nn

Where nn = 0 to 100

One hundred represents the default volume of a voice. Lower values represent percentages of this default. That is, 50 corresponds to 50% of full volume.

Values specified using the Volume tag will be combined with the values set using the VoiceVolume variable. For example, if you combine a VoiceVolume = 50 with a <volume level="50"> tag, the volume of the voice should be 25% of its full volume.

Example:

VoiceXML Volume, 80, TextIn

RateAbsolute, nn

Where nn = -10 to +10.

A value of 0 sets a voice to speak at its default rate (speed). A value of -10 sets a voice to speak at one-third (or 1/3) of its default rate. A value of +10 sets a voice to speak at 3 times its default rate. Each increment between -10 and +10 is logarithmically distributed such that incrementing/decrementing by 1 is multiplying/dividing the rate by the 10th root of 3 (about 1.12). Values more extreme than -10 and +10 will be passed to an engine, but SAPI 5 compliant engines may not support such extremes and instead may clip the rate to the maximum or minimum rate it supports.

Values specified using the RateAbsolute tag will override the values set using the VoiceRate variable.

Example:

VoiceXML RateAbsolute, 2, TextIn

RateAdjust, nn

Where nn = -10 to +10.

A value of 0 sets a voice to speak at its default rate (speed). A value of -10 sets a voice to speak at one-third (or 1/3) of its default rate. A value of +10 sets a voice to speak at 3 times its default rate. Each increment between -10 and +10 is logarithmically distributed such that incrementing/decrementing by 1 is multiplying/dividing the rate by the 10th root of 3 (about 1.12). Values more extreme than -10 and +10 will be passed to an engine, but SAPI 5 compliant engines may not support such extremes and instead may clip the rate to the maximum or minimum rate it supports.

Values specified using the RateAdjust tag will be combined with the values set using the VoiceRate variable.

Example:

VoiceXML RateAdjust, 2, TextIn

PitchAbsolute, nn

Where nn = -10 to +10.

A value of 0 sets a voice to speak at its default pitch. A value of -10 sets a voice to speak at three-fourths (or $\frac{3}{4}$) of its default pitch. A value of +10 sets a voice to speak at four-thirds (or $\frac{4}{3}$) of its default pitch. Each increment between -10 and +10 is logarithmically distributed such that incrementing/decrementing by 1 is multiplying/dividing the pitch by the 24th root of 2 (about 1.03). Values more extreme than -10 and 10 will be passed to an engine but SAPI 5 compliant engines may not support such extremes and instead may clip the pitch to the maximum or minimum pitch it supports.

Values specified using the PitchAbsolute tag will set the new default pitch for all subsequent text-to-speech commands.

Example:

VoiceXML PitchAbsolute, 2, TextIn

PitchAdjust, nn

Where nn = -10 to +10.

A value of 0 sets a voice to speak at its default pitch. A value of -10 sets a voice to speak at three-fourths (or $\frac{3}{4}$) of its default pitch. A value of +10 sets a voice to speak at four-thirds (or $\frac{4}{3}$) of its default pitch. Each increment between -10 and +10 is logarithmically distributed such that incrementing/decrementing by 1 is multiplying/dividing the pitch by the 24th root of 2 (about 1.03). Values more extreme than -10 and 10 will be passed to an engine but SAPI 5 compliant engines may not support such extremes and instead may clip the pitch to the maximum or minimum pitch it supports.

Values specified using the PitchAdjust tag will change the pitch relative to the last pitch change for all subsequent text-to-speech commands.

Example:

VoiceXML PitchAdjust, 2, TextIn

Emphasis

Emphasis instructs the voice to emphasize the word or section of text that was passed to the VoiceXML command.

The method of emphasis may vary from voice to voice.

Example:

VoiceXML Emphasis, TextIn

Spell

Spell forces the voice to spell out all text supplied, letter by letter, rather than using its default word and sentence breaking rules, normalization rules, and so forth. All characters will be expanded to corresponding words (including punctuation, numbers, and so forth).

Words that contain punctuation, such as "U.S.A." will spell out the letters as well as the punctuation.

Example:

VoiceXML Spell, TextIn

VoiceXML types - Insertion

Silence nnn

Where nnn = number of milliseconds, from zero to 65535, of silence. Silence adds a specified number of milliseconds of silence to the end of the text supplied. If no text was supplied, only silence will be returned or queued.

Example:

VoiceXML Silence, 1000

Pronounce sym, [word]

Pronounce specifies a specific pronunciation. The voice will process the sequence of phonemes exactly as they are specified.

If a word (or words) are specified with the pronunciation, it will be interpreted the text that is being pronounced.

Sym is a string of white space separated phonemes.

See the section on Text-to-speech phonemes for more information.

Example:

VoiceXML Pronounce, "h eh l l ow", "Hello"

VoiceXML types – Context

PartOfSpeech part, word

PartOfSpeech tells the voice what “part of speech” will be used to speak the supplied text. Use this tag to enable the voice to pronounce a word with multiple pronunciations correctly depending on its part of speech. The PartOfSp tag cannot be empty.

Part is a string corresponding to a SAPI part of speech. Only SAPI defined parts of speech are supported - "Unknown", "Noun", "Verb", "Modifier", "Function", "Interjection".

Example:

VoiceText “Please ”

VoiceXML PartOfSpeech, "verb", “Record”

VoiceText this

VoiceXML PartOfSpeech, "noun", “Record”

; would say “Please record this record.”

VoiceXML types – Context (dates and times)

Date and time context specifies that the text passed to the engine is a date or a time.

Dates will generally have the format of:

number [delimiter] number [delimiter] number

or

number [delimiter] number

where the delimiter can be a '.', '/' or '-', and numbers are typically between 01 and 12 for months, 01 and 31 for days. A year is generally a two or four-digit number.

Times will generally have the format of:

number [delimiter] number [delimiter] number

or

number [delimiter] number

where the delimiter is ':', '' or ' "' and numbers are typically between 01 and 24 for hours, 01 and 59 for minutes and seconds.

When a zero is present in numbers between 01 and 09, the voice may ignore this, or normalize it as "oh". The voice may also place an "and" in the normalized time.

MDY value
date_mdy value

Example:

VoiceXML MDY, "12/21/1999"

12/21/1999 will be normalized to
"December twenty first nineteen ninety nine"

DMY value
date_dmy value

Example:

VoiceXML DMY, "21/12/1999"

21/12/1999 will be normalized to
"December twenty first nineteen ninety nine"

YMD value
date_ymd value

Example:

VoiceXML YMD, "1999/12/21"

1999/12/21 will be normalized to
"December twenty first nineteen ninety nine"

YM value
date_ym value

Example:

VoiceXML YM, "1999/12"

1999/12 will be normalized to
"December nineteen ninety nine"

MY value
date_my value

Example:

VoiceXML MY, "12/1999"

12/1999 will be normalized to
"December nineteen ninety nine"

DM value
date_dm value

21.12 will be normalized to
"December twenty first"

MD value
date_md value

Example:

VoiceXML MD, "12/21"

12/21 will be normalized to
"December twenty first"

Y value
YEAR value
date_year value

Example:

VoiceXML YEAR, "2001"

2001 will be normalized to
"Two thousand one"

time value

Example:

VoiceXML time, "12:30"

12:30 will be normalized to

"twelve thirty"

Example 2:

VoiceXML time, "01:21"

01:21 is normalized as

"one twenty one"

or

"oh one twenty one"

Example 3:

; The varTime has the value of 1'21"

VoiceXML MDY, varTime

1'21" is normalized as

"one minute twenty one seconds"

or

"one minute and twenty one seconds"

VoiceXML types – Context (numbers)

Number value
number_cardinal value

Example:

VoiceXML number, 3432

3432 will be normalized to
"three thousand four hundred thirty two"
or
"three thousand four hundred and thirty two"

Digits value
number_digit value

Example:

VoiceXML Digits, 3432

3432 will be normalized to
"three four three two"

fraction, value
number_fraction value

Example:

VoiceXML fraction, "3/15"

3/15 will be normalized to
"three fifteenths"
or
"three over fifteen"

Decimal value
number_decimal value

Example:

VoiceXML decimal, 423.1243

423.1243 will be normalized to
"four hundred and twenty three point one two four three"

phone value
Phone_Number value

Example:

VoiceXML Phone, "614-555-1212"

614-555-1212 may be normalized to
"six one four five five five one two one two"
The exact implementation of Phone is left to the voice supplier.

\$ value
Currency value

Example:

VoiceXML \$, "\$34.90"

\$34.90 may be normalized to
"thirty four dollars and ninety cents"
The exact implementation of Currency is left to the voice supplier.

VoiceXML types – Context (misc.)

URL value
web_url value

Example:

VoiceXML URL, "www.ebay.com"

www.ebay.com will be normalized to

"w w w dot ebay dot com"

The exact implementation of URL is left to the voice supplier.

EMAIL value
E-mail_address value

Example:

VoiceXML EMail, "joe@ODT1000.com"

joe@ODT1000.com</context>

is normalized to "Joe at D T T 1 0 0 0 dot com"

The exact implementation of EMAIL is left to the voice supplier.

address value

Example:

VoiceXML address, CustAddr

35 West Broad Street, Redmond, WA, 98052

will be normalized to

"Thirty five west broad street Redmond Washington nine eight zero five two"

The exact implementation of Address is left to the voice supplier

ZIP value

address_postal value

Example:

VoiceXML zip, CanadaZip

A2C 4X5 will be normalized to

"A 2 C 4 X 5"

The exact implementation of Zip is left to the voice supplier

Using Text to Speech

ItemNumber = "893728"

ItemDesc = "Widgets"

OrderQty = 2145

OrderTotal = 127,322.56

OrderWeight = 235.5

Address= "54 West Belt Drive, Chicago, IL"

ZipCode = "43215"

PhoneNo = "(614) 555-1212"

- ' You ordered 2145 of item 893728, Widgets,
- ' for a total of \$127,322.56.
- ' The total weight of your order is 235.5 pounds.
- ' Your order will be delivered on December 27th 2004 at 12:30.
- ' Your delivery window is 20 minutes.

VoiceText "You ordered "

VoiceXML number, OrderQty

VoiceText " of item "

VoiceXML digits, ItemNumber

VoiceText ", "

VoiceText ItemDesc

VoiceText ", "

VoiceText " for a total of "

VoiceXML "currency", orderTotal

VoiceText ". The total weight of your order is "

VoiceXML decimal, OrderWeight

VoiceText " pounds."

VoiceText “Your order will be delivered on “

VoiceXML mdy “12/27/2004”

VoiceText “ at “

VoiceXML time, “12:30”

VoiceText “. Your delivery window is “

VoiceXML time, “20”

VoiceText = “. ”

' Your order will be delivered to
' 54 West Belt Drive, Chicago, IL 43215.
' An order confirmation will be faxed to (614) 555-1212'

VoiceText "Your order will be delivered to "

VoiceXML address, Address

VoiceXML zip, zipCode

VoiceText ". An order confirmation will be faxed to "

VoiceXML phone, PhoneNo

'Please visit our website at www.ODT1000.com

'or send an E mail to sales@ODT1000.com.

VoiceText "Please visit our web site at "

VoiceXML URL, "www.ODT1000.com"

VoiceText ", or send an E mail to "

VoiceXML EMAIL, "sales@ODT1000.com"

' Speak all of the text to a wave file

rtm = VoiceSpeakToFile "OrderSpeech.WAV"

' or speak it to the caller

rtm = VoiceSpeak

American English Phoneme Representation

This is a brief introduction to the use and implementation of the Text-to-speech phoneme representations.

Symbolic and Numerical Representation

ODT1000 developers can create pronunciations for words that are not currently in the lexicon by using the English phonemes represented in the following table.

The developer will be able to enter the SYM representation to create the pronunciation using the VoiceXML Pronounce command. Each phoneme entry should be space delimited.

Example: pronunciation for "hello":

```
VoiceXML Pronounce "h eh l ow"
```

For improved accuracy, the primary (1), secondary (2) stress markers, and the syllabic markers (-) can be added to the pronunciation.

Example: pronunciation for "hello" using the primary stress (1) and syllabic (-) markers:

```
VoiceXML Pronounce "h eh - l ow 1"
```

American English Phoneme Table

SYM	Example	PhoneID
-	syllable boundary (hyphen)	1
!	Sentence terminator (exclamation mark)	2
&	word boundary	3
,	Sentence terminator (comma)	4
.	Sentence terminator (period)	5
?	Sentence terminator (question mark)	6
_	Silence (underscore)	7
1	Primary stress	8
2	Secondary stress	9
aa	F <u>a</u> ther	10
ae	C <u>a</u> t	11
ah	C <u>u</u> t	12
ao	D <u>o</u> g	13
aw	F <u>o</u> ul	14
ax	A <u>g</u> o	15
ay	B <u>i</u> te	16
b	<u>B</u> ig	17
ch	Ch <u>i</u> n	18
d	<u>D</u> ig	19
dh	Th <u>e</u> n	20
eh	P <u>e</u> t	21
er	F <u>u</u> r	22
ey	<u>A</u> te	23
f	<u>F</u> ork	24
g	<u>G</u> ut	25
h	<u>H</u> elp	26

ih	<u>F</u> ill	27
iy	Fee <u>l</u>	28
jh	<u>J</u> oy	29
k	<u>C</u> ut	30
l	<u>L</u> id	31
m	<u>M</u> at	32
n	No	33
ng	<u>S</u> ing	34
ow	<u>G</u> o	35
oy	Toy	36
p	<u>P</u> ut	37
r	<u>R</u> ed	38
s	<u>S</u> it	39
sh	She	40
t	<u>T</u> alk	41
th	Thin	42
uh	<u>B</u> ook	43
uw	Too	44
v	<u>V</u> at	45
w	<u>W</u> ith	46
y	<u>Y</u> ard	47
z	<u>Z</u> ap	48
zh	Plea <u>s</u> ure	49