Basic Translation Rules

Changing Text Expansion Modes

The escape sequences are:

\!ab	begin acronym spell mode (default),
\!ae	end acronym spell mode
\!eb	begin abbreviation expansion mode (default),
\!ee	end abbreviation expansion mode,
\!hnv	set numerical hyphen verbose mode (default)
\!hnt	set numerical hyphen terse mode
\!hav	set alphabetic hyphen verbose mode,
\!hat	set alphabetic hyphen terse mode (default),
\!xb	begin detection of externally defined text classes (default),
\!xe	end detection of externally defined text classes.

Abbreviation Expansion

The first set of basic rules and expansions applied to the incoming text is abbreviation expansion. Each word (except numbers and punctuation) is compared against a list of all known abbreviations. If a match is found, the table provides information about the abbreviation class of this abbreviation. Given the class, FlexTalk performs a particular set of tests to determine if and how to expand the abbreviation.

If an abbreviation has at most one possible expansion and does not require any special handling (for example hr to hour), it belongs to one of the common abbreviation classes. If the abbreviation requires application of special rules to determine if and how to expand it, it belongs to its own special abbreviation class with its own special rules. There are several reasons why an abbreviation may require its own special abbreviation class: it has more than one expansion (for example fl to Florida, floor, fluid or falls), or it is also a common word (for example apt can be apartment or apt), or it requires some type of special handling that is not provided by any of the common classes.

All abbreviations can be followed by an optional period which is either deleted or kept as an end of sentence period. If an abbreviation is within a matched pattern that includes word(s) that follow the abbreviation, the period is obviously not an end of sentence and is deleted. For example, personal titles requires that the abbreviation be followed by at least one word (proper name). In other cases, an end of sentence algorithm is applied to determine what to do about the period. See the **End of Sentence Detection** below. Note, also, that expansion of some abbreviation classes depends on class detector settings, such as address risky mode.

Expansion of an abbreviation can be inhibited by enclosing the text in double quotes. In addition, expansion of abbreviations can be enabled or disabled through the FlexTalk Property Sheet (Reading Style page). Expansion of all abbreviations can also be turned off via the escape sequence \!ee and turned on (default) via \!eb.

End of Sentence Detection

End of sentence detection can be driven either by normal sentence punctuation or by the end of a text line (CR character). End of sentence punctuation includes: .!? (period, exclamation point and question mark). Selection of these modes can be accomplished through the FlexTalk Property Sheet (Reading Style page). Modes can also be selected through the use of control tags. End of line mode is selected by the tag \Ctx=“line”\. Punctuation mode is selected by the tag \Ctx=“sentence”\.

Acronyms and Alphanumerics

An acronym contains only all upper case letters that is not an abbreviation. An alphanumeric contains both letters and numbers. If the acronym spell mode is on, all acronyms and the alphabetic portion of the alphanumeric are spelled out (separated by spaces). If the acronym spell is off, these alphabetic fields are not changed and the text synthesizer may pronounce them as a word. Refer to the Numbers below for expansion of numbers within alphanumerics. The acronym spell mode can be controlled through the Reading Style page of the FlexTalk Property Sheet . It can also be turned on (default) via the \!ab escape sequence and turned off via the \!ae escape sequence.

In the case insignificant mode, acronyms are not recognized. To be pronounced correctly, they must be either in the abbreviation dictionary, or the letters must be separated by spaces or periods. Some examples are i.b.m. or i b m.

Numbers

Number expansion is performed after the text class detection and expansion and before handling punctuation. There are four ways (modes) in which FlexTalk can expand a number.

Serial: each digit is translated as a single word. For example 1234 expands into one two three four.

Comma: the number is read with all place quantifiers made explicit. For example 1234 expands into one thousand two hundred thirty four.

Pair: each pair of digits is grouped and translated as a unit. For odd length strings, the first digit is translated alone. For example 1234 expands into twelve thirty four and 12345 expands to one twenty three forty five.

Hundreds: the first two digits are grouped together and translated together, followed by their place quantifier. This mode only applies to numbers in range 1100-9999 (e.g. dates). For example 1234 expands into twelve hundred thirty four.

FlexTalk contains a complex set of rules that select between these modes based on the context in which the number appears. These rules enable FlexTalk to reliably speak numbers in many settings, such as phone numbers, dates, and addresses. While there is no direct user control over these interpretations, some elements may be controlled through the Class Detectors page of the FlexTalk Property Sheet .

Roman Numerals

The recognized roman numeral must contain more than one letter, must conform to the roman numeral syntax and must only contain letters I, V, X, i, v or x. The highest recognized numeral is XXXVIII, the lowest is II. The lower case numerals are only recognized when in the case insignificant mode. There is no way to inhibit expansion of roman numerals.

Examples:

Valid numerals: II, XIV, XXXVIII

Invalid numerals: X (single letter), IXV (syntax), XXXX (syntax), LX (unrecognized letter).

Unrecognized numerals are either spelled out as acronyms (if all capital letters) or left unchanged.

Once detected, roman numerals are expanded in two ways.

1) If the numeral is detected as a part of a name (names text class) or the name risky mode is set, the numeral is expanded as an ordinal number, for example: Henry VIII as Henry the eighth.

2) In all other cases the numeral is expanded as cardinal number, for example: Chapter IV as Chapter four.

Punctuation

Punctuation characters are: {}!@#$%^&*()-_=+[]\|`~;:'",<>./?. Punctuation expansion is one of the last steps that FlexTalk performs. The rules described in this section apply only to the punctuation that was not handled in one of the previous steps (for example by a text class detector or during number translation).

There are three ways in which FlexTalk handles punctuation:

1)Punctuation delimiting the end of sentence is passed to the text synthesizer unchanged.

2)The punctuation characters: -'"().%#& (hyphens, quotes, parentheses, periods, percents, pound signs, and ampersands) that occur within specific syntax are translated using special rules which are described below. This special handling takes effect only if none of the special translation modes is set (math mode, spell mode, etc.).

3) Punctuation that does not fit into one of the above two categories is translated to its word equivalent (for example & to

ampersand) or left unchanged, depending on the context. The translations depend on the current translation mode (generic mode, math mode, spell mode, etc.).

Hyphens

The hyphens are handled by classifying them into three categories as listed below:

1) Word hyphenation hyphen. Such hyphens must be immediately preceded by a word (no space), and must be immediately followed by a newline and a word. The hyphen is deleted from the word.

2) Numeric Hyphen. A hyphen is classified as numeric if it is not immediately followed by another hyphen (with no intervening space) and matches one of the two rules below:

- preceded by word and space(s) and immediately followed by a number, punctuation, alphanumeric or a plain word, for example value of -1, or

- preceded by a number or punctuation or alphanumeric, optional spaces(s) and hyphen(s) and followed by optional space(s), hyphen(s) and number or punctuation or alphanumeric, for example 3B2-20.

3) Alphabetic hyphen. A hyphen is classified as alphabetic if it does not fall in any of the above categories.

Although the expansion rules for the numeric and alphabetic hyphens are essentially identical, the default behavior is to expand numerical hyphens as dash and ignore the alphabetic hyphens.

Quotes

Use of quotes (single or double) will have cause few unexpected results. Contractions will be handled correctly and enclosing words in quotes will cause either a trailing pause (single quotes) or emphasis (double quotes). Mismatched quotes and leading quotes may sometimes cause FlexTalk to pronounce the word “apostrophe” or “quote” when confusing punctuation is encountered.

There is no direct user control over FlexTalk’s handling of quotes. However, some class detectors, especially measurement risky mode will affect interpretation.

Parentheses

A left parenthesis is replaced with a comma (pause) if it is the first (outside) one in a parenthesized expression and if it is preceded by a space or quotes. In other cases it is expanded to the words “open paren” or “left parenthesis” (Math Mode).

A right parenthesis is replaced with a comma (pause) only if it has a previous matching left parenthesis which was replaced by a comma. In other cases it is expanded to the words “close paren” or “right parenthesis” (Math Mode).

For example:

printf() function	printf open paren close paren function
see printf (3c) in the	see printf , three c , in the
chapter 3 (section 6(c))	chapter three , section open paren c close paren ,
nested ((twice)) are	nested , open paren twice close paren , are

Other Punctuation

A period that is immediately followed by a number is translated to point. A percent sign that is preceded by a number and an optional space is translated to percent. A pound sign that is immediately followed by a number or alphanumeric is translated to number. Ampersand is always expanded to and. In all other cases, the expansion is to the word “ampersand”.

Go Back To Speech Synthesis Control
Go Back To Introduction and Welcome