Speech Synthesis Control

Escape Sequences

FlexTalk accepts a number of special input strings, or Escape Sequences , which change the resulting output speech. They are commonly used to fine-tune particular sentences for specific applications such as perception tests. Vowels can be shortened, pauses lengthened, stresses changed, etc.

Basic Translation Rules

There are two types of text translation rules: the Basic Translation Rules and the Text Class Rules . The Basic Translation Rules handle things such as end of sentence detection, punctuation expansion and abbreviation expansion. These rules are usually simple and do not use context beyond the immediately neighboring words to disambiguate the translation. Also, they are internal, that is, they are hard coded within FlexTalk and cannot be easily changed. There is a limited way to control some Basic Rules. For examples the expansion of abbreviations can be turned off or on, or an end of sentence can be forced using an escape sequence. Also frequently the abbreviation expansion rules use the settings of the Text Class Detectors (off, conservative or risky) to determine how (and if) to expand abbreviations.

Text Class Rules

The Text Class Rules attempt to detect certain classes of text (addresses, phone numbers, etc.) and perform translations appropriate for the class. For example when a telephone number is detected, each digit should be read individually (in serial mode) with pauses inserted at appropriate places. The Text Class Rules are generally more complex than the basic rules and look at a number of words when trying to detect syntactic patterns.

Special Translation Modes

FlexTalk provides four Special Translation Modes . They are turned on and off using escape sequences as shown below. If more than one mode is on at the same time, the first one on the list below is used. If none of the special translation modes is on, FlexTalk is in a standard translation mode.

Phonemic Input

It is also possible to explicitly specify pronunciations by inserting phoneme strings in the input text. Phonemes should be those defined in the CECILbet Phonetic Alphabet. Phoneme strings are placed between \(and \) delimiters. Double quote places primary stress on the following vowel. Hyphen forces syllabification.

Examples:	This is an \(ig"zam-p&l \) of embedded phonetics.
	My name is \("m @ r k "byUt.nA-g&l \) .

Original orthography can be supplied within curly brackets {}, and is highly recommended because some TTS modules depend on it.

The \({test} t"est \) was \({silly} "sil-E \) .

Explicit durations can be supplied for any phoneme...

\(.O[75]_kla"hO-ma \)

where the number in [] gives the duration of the previous phone in 100ths of a second, i.e. "O[100]" is one second.

Go Back To Introduction and Welcome