TXSTEP: search-tables and exchange-tables

In search tables and in string pairs used in exchange tables, two kinds of character strings may be specified for searching the text: »search strings« and »exception strings« (= character search strings which are to be skipped during a search).

When defining a search string, in place of a single character, a predefined identification for a character group or a string group may be given. In addition, a reference (pointer) can also be given, which refers to other elements (see below) of the same character string. Furthermore, frequency conditions may be added to characters, character groups, string groups and references, and ambient conditions for the character strings may also be specified.

The texts are always scanned from left to right. If a string, starting in the text at the current scan position, corresponds to more than one search string, then the longer string is given precedence in each case. For strings of equal length, priority corresponds to the order in which they were given in the parameter. Whenever a search string has been found, further search in the text will continue after the position following the string found.

If an exception string is found during the search (in accordance with the priority rules stated here), the corresponding characters in the text will be skipped over. The search in the text will then continue at the position following the exception string. However, this does not apply to exception strings specified in string groups; here the exception string is only used to specify that other character strings given in the same string group may be passed over and thus none of the search strings specified in the string group will be found beyond the current position in the text.

Caution: Based on these rules, a search of the text »...01234...« for the strings »1234« and »01« would find the string »01« but not the string »1234«. Reason: Although the string »1234« is longer than the string »01«, what is decisive here is the fact that string »01« is further to the left in the text and (since the search is carried out from left to right) is thus the first string found. Similarly, a search of the text »...01234...« by a search table containing the search string »1234« and the execption string »01« will first find the exception string »01«. The text search will then be continued at the position that follows the string »01«, thereby no longer finding the string. »1234«. In addition, attention should be given to the order in which search and exception strings of equal length are given. Specifying, for example, in an exchange table two string pairs, the first specifying the replacement string »two lower case chars« for the search string »{2}{\a}«, and the second specifying for the search string »\x\y« the replacement string »lower case xy«, will not replace »xy« by »lower case xy« because both search strings, »{2}{\a}« and »\x\y«, are two characters long and because the check in the text for any occurances of the character string »xy« will be overridden by the search string »{2}{\a}«, since it has been specified before the search string »\x\y«.

Characters, character group identifications, string group identifications and references will be refered to in the following overview as »elements« of the character search string. A frequency condition preceding a character, a character group identification, a string group identification or a reference is considered to be part of the element.

Predefined character groups

`?`	any character
`{!}`	ASCII character
`{;}`	TUSTEP character beyond ASCII characters
`{@}`	characters except letters and digits
`{%}`	character used after % for encoding accents
`{\a}`	lower case letters
`{\A}`	upper case letters
`{&a}`	lower and upper case letters
`{\0}`	normal decimal digits
`{&0}`	normal and superscript digits
`*`	zero up to any number of any characters

Character groups and string groups

– in the string defining a character group:

`{-}`	remove following characters from the group
`{+}`	include following characters into the group

– referring to a character group or a string group

`[...]`	local character group, as m[ae][iy]er
`{Z:xy}`	user defined character group xy
`{C:xy}`	same as `{Z:xy}`
`{S:xy}`	user defined string group xy

Frequency conditions in search strings

`{n}`	exactly n elements
`{n-m}`	n to m elements, as few as possible
`{n--m}`	n to m elements, as many as possible
`{0}`	0 or 1 element, same as `{0-1}`
`{00}`	1 up to infinitely many elements = `{1-0}`

Numerical values in search strings

`{#}`	number with any value
`{#n}`	number with the value n
`{!n}`	number with a value not equal n
`{#n-m}`	nuber with a value from n to m
`{!n-m}`	number with a value smaller than n or greater than m

Pointers in search strings

`{+n=}`	n-th element, counting from left a != A
`{-n=}`	n-th element, counting from right a != A
`{+n:}`	n-th element, counting from left a == A
`{-n:}`	n-th element, counting from right a == A

Fields of elements in search strings

`{|}`	Delimiter between fields of elements

Pointers in replacement strings

{+n=}	n-th element, counting from left
{-n=}	n-th element, counting from right
{+0=}	all elements of the core string
{-0=}	all elements of the core string
{+n-m=}	n-th to m-th element, conting from left
{+n-0=}	n-th to last element, counting from left
{-n-m=}	n-th to m-th element, counting from right
{-0-m=}	first to m-th element, counting from right
{=n=}	all elements of the n-th element field
{=0=}	all elements of the core string
{=n-m=}	all elements of the n-th up to the m-th element field
{...+}	... lower case --> upper case
{...-}	... upper case --> lower case
{...;}	... a, b, ..., $, ... --> ä, ^b, ..., ^$, ...
{...!}	... ä, ^b, ..., ^$, ... --> a, b, ..., $, ...

Characters to be escaped

\?	question mark
*	asterisk
\[	opening square bracket
\]	closing square bracket
\{	opening curly bracket
\}	closing curly bracket
\a	lower case letter a
\A	upper case letter A
\\	backslash
{{	< _ (opening angle bracket)
}}	> _ (closing angle bracket)

Other

{[}	left margin
{]}	right margin
{|}	in tables for sorting alphabets: switch to highest values