# flexible search™

# Translate Table Reference Manual

*( Click on a link to access the desired section )* Version 2.0

SLICCWARE™

Publication Date -- January 24, 2003

# Table of Contents

# Section 1 – Translate Table Overview

Process Control Values

The Stemming Process

**The Translate Table** is a file that provides the information necessary to properly match similar words based upon synonym pairings and root extraction or stemming rules. The SLICCWARE Application Tool Set (ATS) is used to generate the file. Any number of Translate Tables may be generated. Although no Keyword Index is required to have a Translate Table associated with it, having a Translate Table associated with a Keyword Index is often desired. Each Keyword Index may have its own Translate Table or multiple Keyword Indexes may share a Translate Table.

The Translate Table contains six sections plus a header. The first section contains six values seperated by tabs. The six values are:

**Resource ID**, a value assigned to the Translate Table to identify it from other resources used by the installation. The resource ID must be different from that of any other resource defined for the installation.

**Processing ID**, a value identifying a special, language-specific routine used to finalize the stemming process. Currently only two choices are available: zero (0), indicating no special processing; or one (1), indicating English language special processing.

**Character Set ID**, the resource ID for a loaded character set. Currently this is unsupported, and should be set to zero (0) for an internal case insensitive character set, and to one (1) for an internal case sensitive character set.

**Trim Plurals Flag**, a flag indicating whether recognized plurals are to be trimmed using English language standards. The choices are: zero (0), indicating trimming of plurals should not be done; and one (1), indicating trimming of plurals should be done. Besides simple trimming of **“s”** and **“es”**, **“ies”** is replaced by **“y”** and **“lves”** is replaced by **“lf”**. However, Latin plurals ending in **“ae”** or **“i”** are not handled, nor are special spellings such as **“mice”** from **“mouse”** or **“oxen”** from **“ox”**.

**Compress Doubles Flag**, a flag indicating that all double consonants are to be compressed down to a single character. This can be very useful in combatting simple spelling errors, both within documents and within search requests.

**Process Threshold**, the number of characters a word must contain before trimming of plurals or stemming will take place.

## Section 1.2 – The Stemming Process

**The stemming process** is defined within the second section. The stemming process allows a user to match words based upon their roots. For instance, the user may wish to consider “banker” and “banking” a match because they share the same root, “bank”.

To accomplish this, the trailing chacters of a word are compared against different patterns representing suffixes that are to be removed. The comparisons are made on a character by character basis with no consideration for upper or lower case. In addition to alpha-numeric characters, four wildcard and one special character are supported by the matching algorithm.

**Wildcard Character “?”**, signified by a question mark, is used to match any single character.

**Wildcard Consonant “%”**, signified by a percent sign, is used to match any single consonant. ( b - d, f - h, j - n, p - t, v - z )

**Wildcard Vowel “@”**, signified by an at sign, is used to match any single vowel ( a, e, i, o, u ).

**Wildcard Digit “#”**, signified by a pound sign, is used to match any single digit ( 0 - 9 ).

**Double Character Flag “!”**, signified by an exclamation point, is used to signify the next character must appear twice. This flag is most commonly used with a wildcard since a specific double character can be identified directly.

This stemming process, as it is called, is performed in a series of one or more passes. If a match is found, the suffix is removed from the word. In addition to identifying a pattern for suffix removal, the designer may also identify a replacement string to be attached to the root after the suffix is removed. Occasionally, the designer may not want to strip, from the end of the word, all the characters that were matched by the pattern. To allow for this, the replacement string may contain the following special character one or more times.

> **Retention Character “.”**, signified by a period or decimal point, is used to cause the character previously stripped from a location to be reattached. This is often used to reattach a single consonant after matching and removing a consonant pair.

Once a match has been made during any pass, the remaining patterns defined for the pass are ignored and the process moves on to the next pass.

**Examples of pattern matching replacement**

| | | | |
|---|---|---|---|
| **6** | **!%ing** | **.** | **compress double consonant** |
| **6** | **%cing** | **.ce** | **replace “ing” with “e”** |
| **6** | **%%ing** | **..** | **no substitution for “ing”** |
| **6** | **!%@%ing** | **....** | **no substitution for “ing”** |
| **6** | **%@%ing** | **...e** | **replace “ing” with “e”** |
| **6** | **@@%ing** | **...** | **no substitution for “ing”** |

**clapping ==>> clap**
**rule 1**
**fencing ==>> fence**
**rule 2**
**punting ==>> punt**
**rule 3**
**flattening ==>> flatten rule 4**
**stoning ==>> stone**
**rule 5**
**waiting ==>> wait**
**rule 6**

If a pattern match has been made, and in the prior section, the Processing ID was set to one, and the last pattern match did not have a replacement string associated with it, the software will attempt to terminate the root properly. This allows the designer to use a much less complex stemming algorithm since the double consonant and long vowel logic is left to the English-specific special processing function.

**Example of pattern matching removal with Processing ID set to 0 and 1**

**6 ing no substitution for “ing”**

| Processing ID : | | 0 | 1 |
|---|---|---|---|
| clapping | ==>> | clapp | clap |
| fencing | ==>> | fenc | fence |
| punting | ==>> | punt | punt |
| flattening | ==>> | flatten | flatten |
| stoning | ==>> | ston | stone |
| waiting | ==>> | wait | wait |

To help prevent a portion of the root being mistaken for a suffix, a threshold size is defined for *each* match. In the examples 6 is being used. If the number of characters in the word is less that or equal to the threshold, it will not be tested against that pattern. This is similar to the process threshold identified earlier, except that it is defined seperately for each suffix pattern, allowing for more exact tuning of the stemming process.

# Section 2 – Special Processing Lists

Defining a Stop List

Defining an Exception List

Defining a Start List

Defining a Synonym List

## Section 2.1 – Defining a Stop List

**The Stop List** is defined within the third section. The Stop List is simply an alphabetized list of stop words. Stop words are a common method used to reduce the size of an index. Certain words which are deemed to be of questionable importance within a search are added to the Stop List. When a word contained within the Stop List is encountered, it is discarded rather than indexed, thus reducing the size of the index.

**Example of a simple Stop List**

**a**
**about**
**an**
**and**
**are**
**at**
**be**
**but**
**by**
**her**
**his**
**i**
**in**
**is**
**it**
**me**
**mine**
**my**
**nor**
**not**
**on**
**or**
**our**
**over**
**the**
**their**
**they**
**under**
**was**
**we**
**were**
**with**
**within**
**without**
**you**
**your**

# Section 2.2 – Defining an Exception List

**The Exception List** is defined within the fourth section. The Exception List is a list of words that are not to be stemmed. This could be proper names or words that have little to do with the root that would be extracted. Usually they will be words of specific importance within the context of the index.

**Example of a simple Exception List of proper names**

**Alfred**
**Carter**
**Fisher**
**Peking**
**Trantor**
**Wilfred**

# Section 2.3 – Defining a Start List

**The Start List** is defined within the fifth section. The Start List provides a simple means of categorizing data. If a Start List exists, only words found within the Start List are indexed. All other words are discarded. However, it is not the word itself that is indexed. Instead, there is a replacement word associated with the start word. It is that word which is indexed without any further replacement or stemming.

The major use of a Start List is in categorizing data. A category is created by associating a number of individual start words with the same replacement word. Once the desired number of categories have been created within the Start List, the list can be used with an index created over the data to categorize it for searching or histogramming.

**Example of a Start List being used to categorize records by make of automobile**

| | |
|---|---|
| **626** | **mazda** |
| **camaro** | **chevrolet** |
| **cavalier** | **chevrolet** |
| **chevrolet** | **chevrolet** |
| **chevy** | **chevrolet** |
| **cobra** | **ford** |
| **corvette** | **chevrolet** |
| **cougar** | **mercury** |
| **focus** | **ford** |
| **ford** | **ford** |
| **galaxy** | **ford** |
| **impala** | **chevrolet** |
| **malibu** | **chevrolet** |
| **maverick** | **ford** |
| **mazda** | **mazda** |
| **mercury** | **mercury** |
| **miata** | **mazda** |
| **millenia** | **mazda** |
| **mpv** | **mazda** |
| **mustang** | **ford** |
| **prizm** | **chevrolet** |
| **protege** | **mazda** |
| **sable** | **mercury** |
| **thunderbird** | **ford** |
| **taurus** | **ford** |
| **venture** | **chevrolet** |
| **villager** | **mercury** |
| **windstar** | **ford** |
| **zx2** | **ford** |

# Section 2.4 – Defining a Synonym List

**The Synonym List** is defined within the sixth section. The Synonym List can be used as a thesaurus or to handle special plurals such as “mice” which is the plural of “mouse”. Although the pattern being matched must be a single word, the replacement string can be a combination of words.

Once a match has been found and the replacement made, normal stemming proceeds as it would with any other word. If the word has been replaced by a combination of words while searching, all the replacement words must exist within the searchable object for a match to take place. This provides for some great searching opportunities as described below.

**Example of a simple Synonym List**

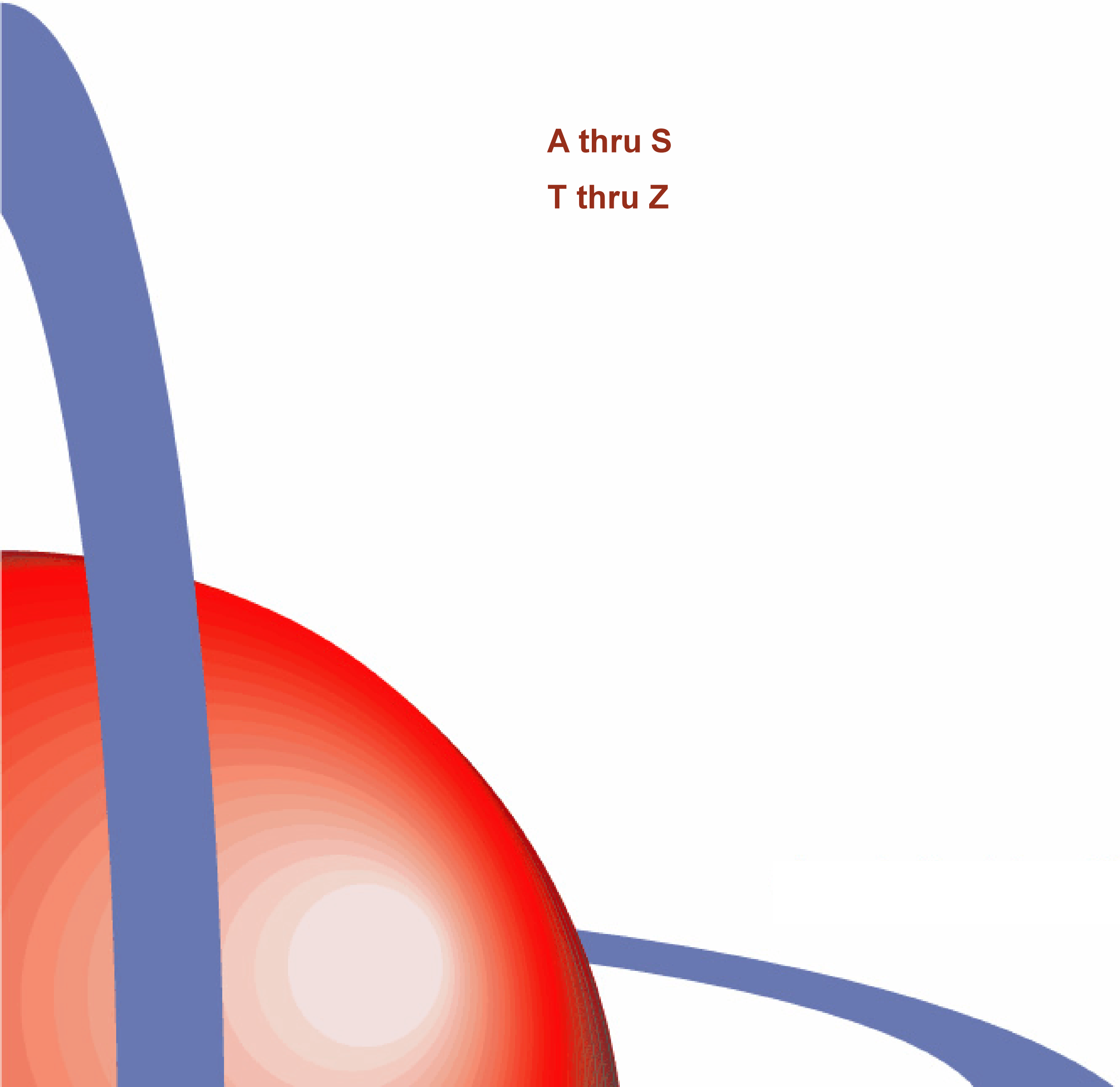
| | |
|---|---|
| **car** | **automobile** |
| **fettuccini** | **fettuccini italian pasta** |
| **goose** | **geese** |
| **man** | **men** |
| **mouse** | **mice** |
| **ox** | **oxen** |
| **po** | **post office** |
| **ravioli** | **ravioli italian pasta** |
| **spaghetti** | **spaghetti italian pasta** |
| **usmc** | **united states marine corps** |
| **woman** | **women** |

In the example above, **”ravioli”** is being replaced by **”ravioli italian pasta”**. As a result any search looking for **“italian”** or **“pasta”** or **“italian and pasta”** would find searchable objects containing **“ravioli”**. Such a search would also find searchable objects containing **“fettuccini”** or **“spaghetti”** for the same reason. However, a search for **“ravioli”** would only find searchable objects containing **“ravioli”** since the search request for **“ravioli”** would translate to a search request for **“ravioli and italian and pasta”**.

# INDEX

A thru S

T thru Z



# INDEX – A thru S