## Preprocessing Question Adjustments

When Transcelerator generates guessed translation, the guesses are based on its ability to statistically match parts of previously translated questions with parts of the English question. Theses “parts” typically consist of words or short phrases, sometimes even partial words. Since many of the English questions are fairly short, a relatively high percentage of the text of the question is sometimes grammatical fluff (e.g., helping words such as “did” or “does”) that might not be reflected in the vernacular grammar. Additionally, the basic word order used in English questions and the vernacular questions might differ. These kinds of differences might slow Transcelerator down in its ability to figure out what vernacular parts correspond to what English parts. Writing effective Preprocessing Question Adjustments is not particularly intuitive, but if you can figure out a few phrase substitution rules that help align the English to the vernacular, it might speed things up, especially in the early stages of using Transcelerator. If you aren’t able to analyze the differences or can’t figure out how to convey the necessary adjustment via a rule, don’t worry. This is an advanced part of Transcelerator and is in no way required to use the program successfully.

The easiest way to know when a question adjustment is needed is when you are looking at a question and see that the guessed translation is close to correct but has messed up word order, repeated words, or extra words that don’t make sense. When that happens, on the Advanced menu, select Preprocessing Question Adjustments, and then do the following:

1. Not that the question you had selected in the main window is displayed in the Preview Sample Question. You will probably want to refer to that when writing your rule.
2. To begin to create a new rule, clicking in the blank line at the bottom of the list of rules.
3. In the column **Word or Phrase to Replace**, type the portion of the English question that you want to “fix” by reordering, deleting, adding, etc.
4. In the **Replacement** column, type the words or phrase as you want them to be adjusted. Note that these should still be English words, not the target language. To delete the entire word or phrase you typed in step 3 (e.g., to remove an auxiliary verb), just leave this cell blank.
5. If the rule is case-sensitive, select that check box. (Most rules probably don’t need to be.)
6. In the **Preview Result** column, look to see whether your new rule produced a change for the currently selected sample question. If not, then you probably made a mistake in step 3. Or you typed the same thing in the Replacement box.

If you have other rules, an earlier rule may have also changed the text. If the two changes are completely independent and are likely to be independent for other questions to which they both apply, then the order does not matter. If it is important for the rules to be considered and applied in a particular order, you can move a rule up or down in the list by selecting it and clicking the green up or down arrow to the right of the grid. The **Preview Result** column will adjust dynamically as the order of the rules changes.

*Hint:*

* If you want to force Transcelerator to treat two adjacent words as a single “part” when looking for a matching piece in the vernacular, write a rule to join the tow words using a hyphen.

**Regular Expression** rules are a much more powerful way of doing phrase substitutions because they can match and replace based on a pattern rather than requiring an exact text match. If you are unfamiliar with regular expressions, please understand that even a simple regular expression can be fairly difficult to grasp. It’s impossible to fully document how regular expressions work, but we will illustrate two examples here. For more help, find someone with experience in regular expressions to assist you or search on-line to learn about the syntax. (Note: There are many dialects of regular expressions. Transcelerator uses the dialect that is used in .net/C#.)

**Regular expression example 1**

There are many English questions in Transcelerator that have the form, “What did \_\_\_ say…” The blank might be a single word, such as a proper name, or a longer phrase. A regular expression to match this phrase is:

*What did (.\*) say*

The period means any character. The asterisk means any number of the preceding thing. The parentheses indicate that the matched characters should be treated as a group. If you wanted to Adjust this phrase for a language (e.g. Spanish) that does not use auxiliary verbs in questions but uses the simple past, your replacement would look like this:

*What said $1*

The dollar sign indicates that a matched group should be inserted at this location in the replacement. The number 1 indicates the group number to insert. In this example, there is only one group. If you select *What did Elizabeth say about God's kindness towards Mary?* As the Preview Sample Question, you will see that this rule results in the following:  
*What said Elizabeth about God's kindness towards Mary?*

**Regular expression example 2**

There are many English questions in Transcelerator that have the form, “How long did *noun phrase* *verb*…” or “How often did *noun phrase* *verb*…”The following regular expression will match some of the common verbs who simple past is formed by adding an *-ed* suffix:

*How (long|often) did (.\*) (last|stay|fish|help|enjoy|remain|mourn|expect|wait|rain)*

The second group is defined just as the group in the previous example. The first and last groups – also indicated by parentheses – have lists of specific words. If a particular question has some other verb (e.g., *continue*, which gets only a *–d* suffix to form the simple past), it won’t match this rule because it will fail to match the third group. Now you can write a replacement like this:

*How $1 $3ed $2*

If you select *How long did the stay there?* As the Preview Sample Question, you will see that this rule results in the following:  
*How long stayed they there?*

Using what you learned in this example, can you figure out how to write the rule for a list of verbs like *continue* that end in *e* and therefore only need a *–d* suffix?