Introduction to Regular Expressions (Regexp)
for the GATE Family of Text Engineering Tools

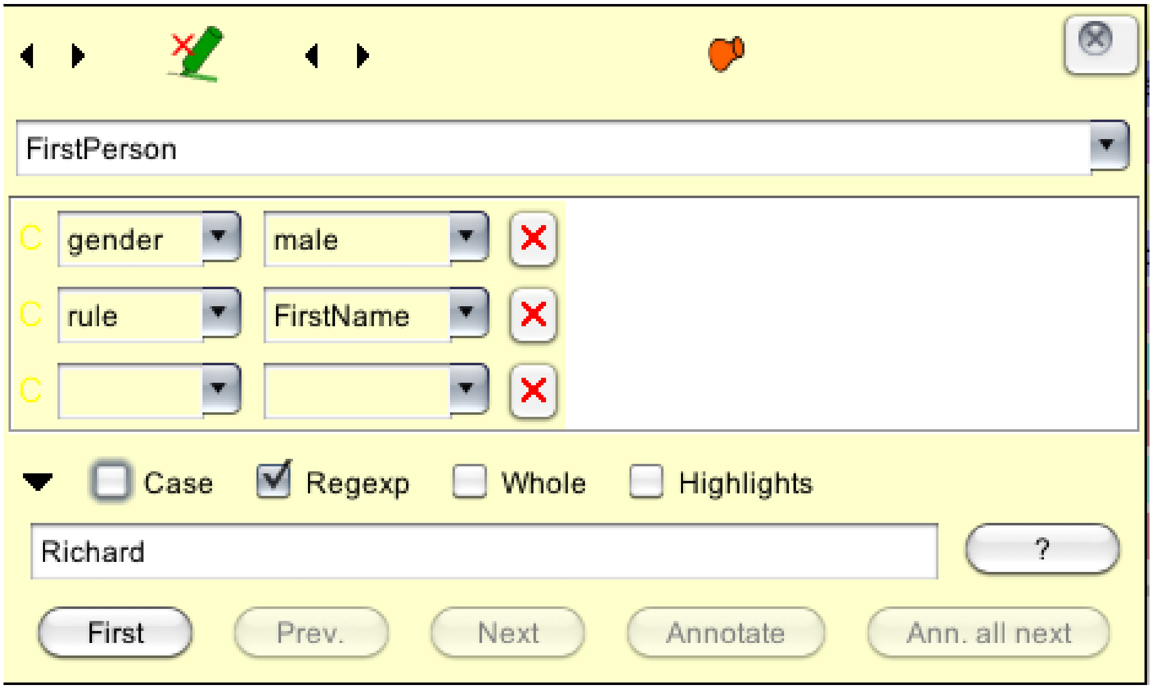
GATE allows you to search inside documents using a “regular expression” (regexp). Regular expressions provide a succinct and adaptable way of searching for particular words or patterns of characters. The expressions are written in a formal language that is processed and interpreted by a “regular expression engine.” In the software world, different regular expression engines exist in different scripting languages, and not all are compatible with each other. GATE makes use of the Java regular expression library.

NOTE: Italicized text in this guide indicates where GATE/Java regular expression syntax differs from other regular expression engines.

GATE Search Expression Builder

The GATE search expression builder is a function available from within the annotation editing pop-up box in the GATE Developer and GATE Teamware user interfaces. This functionality is also supported in the GATE Embedded Java class library.

In order to search documents using the GATE search expression builder, open and view a document, and highlight a word. Hover the mouse over the highlighted word, and a window will pop up. Click the push-pin at the top of the new window to prevent it from closing. Then, click on “Open Search & Annotate Tool” at the bottom of the pop-up menu.



FirstPerson

gender	male
rule	FirstName

☐ Case ☑ Regexp ☐ Whole ☐ Highlights

Richard

Check off the Regexp box and enter one or a combination of the pre-defined Java regular expressions for the string for which you want to search.

Most of the standard Java regular expressions are listed in the table below.

Search expression	GATE Regular Expression
word boundary	\b
any character	.
at the beginning of a line	^
at the end of a line	$
all letters	\p{L}
uppercase letters	\p{Lu}
lowercase letters	\p{Ll}
titlecase letters	\p{Lt}
letter modifier	\p{Lm}
letter other	\p{Lo}
all numbers	\p{N}
number decimal digit	\p{Nd}
number letter	\p{Nl}
number other	\p{No}
all punctuation	\p{P}
punctuation connector	\p{Pc}
punctuation dash	\p{Pd}
punctuation open	\p{Ps}
punctuation close	\p{Pe}
*punctuation initial quote	\p{Pi}
*punctuation final quote	\p{Pf}
punctuation other	\p{Po}
all symbols	\p{S}
symbol math	\p{Sm}
symbol currency	\p{Sc}
symbol modifier	\p{Sk}
symbol other	\p{So}
all separators	\p{Z}
separator space	\p{Zs}
separator line	\p{Zl}
separator paragraph	\p{Zp}
all marks	\p{M}
mark nonspacing	\p{Mn}
mark spacing combining	\p{Mc}
mark enclosing	\p{Me}
all others	\p{C}
other control	\p{Cc}
other format	\p{Cf}
other surrogate	\p{Cs}

other private use	\p{Co}
other not assigned	\p{Cn}
**any character except Category	\P{Category}
**Category1 and/or Category2	[\p{Category1}\p{Category2}]
**Category1 and Category2	[\p{Category1}&&\p{Category2}]
either the selection or X	(?:\p{N})|(?:X)
once or not at all	(?:\p{N})?
zero or more times	(?:\p{N})*
one or more times	(?:\p{N})+
capturing group	(?:\p{N})
non-capturing group	(?:\p{N})

*Note that not all of these expressions work in Java versions below 1.6. If you are unable to upgrade your Java version, please make use of the work-around in Sun's bug tracker.
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4829857

**Note that Java, and therefore GATE, does not support the long-hand version of character Categories as do some other regular expression engines. So, in order to search for matches in the Category "letter", you must use \p{L} rather than \p{Letter}.

In addition, the pre-defined character classes (below) are supported, as are the POSIX character classes (such as \p{Lower}, \p{Digit}), Unicode escape sequences (such as \u2014), etc.

Search Expression	GATE Regular Expression
a digit [0-9]	\d
a non-digit [^0-9]	\D
a whitespace character [\t\n\x0B\f\r]	\s
a non-whitespace character [^\s]	\S
a word character [a-zA-Z_0-9]	\w
a non-word character [^\w]	\W

GATE Regexp Special Characters

Several characters have special meaning (metacharacters) and will be discussed first. In order to search any of these thirteen characters literally, it must be escaped, i.e., preceded by a back-slash, which is further discussed below. They include:

[opening square bracket
This symbol defines a character class. [A-Z] will retrieve all uppercase letters. The closing square bracket] need not be escaped.

\ back-slash
If you want to use any of these 13 special characters as a literal in regexp, you need to escape it with a back-slash. Following suit, if you want to search for the back-slash, you must use \\.

^ caret
The caret is an anchor expression. It does not match any particular character, but indicates that the character appear at the beginning of a line.

$ dollar sign
The dollar sign is an anchor expression. It indicates that the character appear at the end of a line.

. period
Matches any character. For example, ... will match any three-letter string.

| vertical bar/pipe symbol
Acts as an "OR" between two terms.

? question mark
Matches a single character. Used in a string, it makes the preceding token in the regexp optional. *A GATE regexp may not begin with a question mark, although other regexp engines allow it.*

asterisk
Matches zero or an infinite number of characters. *As with the question mark, the preceding token in the regexp is optional, and a GATE regexp cannot start with this metacharacter.*

+ plus sign
Matches at least one or more characters. *As with the question mark and the asterisk, a GATE regexp may also not begin with a plus sign.*

( ) rounded brackets
By placing part of a regexp within rounded brackets, you are able to group that part of the expression together.

{ } curly brackets

Most regular expression search engines do not require that curly brackets be escaped when they are being searched for literally. However, Java/GATE does.

GATE Regexp Use Examples

As noted earlier, different regular expression engines are not always fully compatible with each other. As a result, we will analyze this Java flavor of regexp with respect to its use in GATE.

- The metacharacter \b, like the caret or dollar sign, marks the position of a word boundary and is of zero-length. For example:

  ```
  \b\p{Lu}\p{N}\p{N}\p{Lu}\b
  ```

  will find you G17F, but not 124BG17F42

- If you are interested in searching for a person’s name, enter:

  ```
  \p{Lu}\p{L}+, \p{Lu}\.(?\p{Lu}\.)*
  ```

  (i.e., uppercase word followed by comma followed by uppercase followed by optional middle initial)

- If you would like to search for a number, enter the following:

  ```
  \b[\p{N}][\p{N},.]*\b
  ```

  (i.e., a number or numbers optionally followed by a comma and/or a decimal point)

- To search for “any” symbol, such as an equals sign “=”, enter:

  ```
  \p{S}
  ```

- If you want to search for a particular symbol, such as “%”, then escape the symbol with a back-slash:

  ```
  \%
  ```

- Right-hand truncation is permitted. For example, searching for

DU* returns every “D” or “d” in the document, as does

DU? since the token immediately before the metacharacter is considered optional.

- However, left-hand truncation is not permitted. Searching

  *DU
  ?DU

  returns an error.

- Note that if you want to search for 1+2=3, you must construct it as follows

  1\+2\=3

  because 1+2=3 is a legitimate regular expression.

- You can search for plain text, such as the word or phrases below. Do not use quotation marks around phrases (such as the latter example) as such symbols are viewed literally.

  compound
  an effective amount