

Expr Pattern Matching

Expr language provides you with a couple of functions that make it easier to check a text value against a certain pattern. Functions [MATCH](#), [CASE](#), [REPLACE](#) and [SEARCH](#) use pattern matching for your convenience.

Matching involves a value and a pattern. There are three types of patterns – the type that you use defines how matching is done.

Exact Matching

The simplest pattern type is just a text value that you expect:

- `MATCH(value, "Apples")`

This match will happen if `value` is, in fact, the text that is used as the pattern.

Although it's called "exact matching", there are some additional rules that make the matching easier. The rules are:

- All leading and trailing whitespace characters are removed from the value.
- Text comparison is case-insensitive, which means `APPLES` will match `Apples`.
- The value (without leading and trailing spaces) must match the whole pattern.

Exact matching is used by default, if the pattern is not recognized as requiring Wildcard or Regular Expression matching.

Wildcard Matching

Wildcard patterns let you use symbol "*" to specify any number of any characters.

- `MATCH(value, "App*")`

You can use multiple asterisks to build your pattern.

The same rules as for the exact matching apply:

- All leading and trailing whitespace characters are removed from the value.
- Text comparison is case-insensitive, which means `APPLES` will match `App*`.
- The value (without leading and trailing spaces) must match the whole pattern.

Exact matching is used when the pattern is not recognized to be a Regular Expression Pattern but contains at least one asterisk.

Regular Expression Matching

This type of matching lets you use powerful regular expressions to specify exactly what you need to match with.

- `MATCH(value, "^Ap+.*$$/")`

Structure uses regular expressions available with Java. For a full documentation about the regular expression language, see [Java documentation for Pattern](#).

The regular expression matching is different from other types of matching. The following rules apply:

- Leading and trailing whitespace characters are **not** removed.
- Text comparison is case-insensitive, like with the other types of matching.
- The value does not have to fully match the pattern – it is sufficient that at least one occurrence of the pattern is found in the value. To make your pattern match the whole text, use "^" and "\$" characters in the pattern.

Regular expression matching is turned on if the first and the last characters of the pattern are "/" (These characters are removed, they are not a part of the pattern.)