## Matching, wildcards and ranges

In a list-lookup, we will test if there exist any records in the list whereincoming Key (e.g. caller's phone number, entered DTMF or Subject in an email) matches the Key in a row in the list. If a match is found, the (first) matching record's value is returned.

Please do not use these special characters in the key field since they will not return a match even if there is a matchú, û, ü, $\div$, õ, ý, ù

The incoming Key can only be a single (constant) element. Wildcards or ranges are not allowed here.
The Key field of a record in a list may be asingle element (constant), a range, or a single element containing one or more wildcards Combinations of range and wildcard(s) within a list row's Key is not allowed.

## Ranges in the list's Key field

- The from-value and the to-value is separated by : (colon)
- The to-value is included in the range. Example Key range: 21000000 : 21999999


## Wildcards in the Key field

The wildcard matching, as well as the substitution (next chapter), is based on syntax known from Perl-based regular expressions. Though, there exists a set of restrictions compared to the full Perl syntax. This is what's supported as part of a Key:

- Represent any number of characters: .* (i.e. a single dot followed by an asterisk)
- Represent any single character: . (i.e. a single dot)


## Examples:

|  | Description |
| :--- | :--- |
| Match for any number starting with 47 | Key |
| Match for numbers starting with 47, and which are 10 digits in total length (i.e. eight digits after 47) | $47 .{ }^{*}$ |
| Match any number starting with 47 and ending with 99. Between the starting 47 and the ending 99, there <br> should be at least two characters. | $47 . . . . . .$. |
| Match for text ending with abc | $47 . . .{ }^{*} 99$ |
| Match for text beginning with abc | .*abc |
| Match for text containing abc | abc.* |

## Advanced:

## Quoting needed for the wildcard characters

If a single dot is going to be part of the key, and not treated as a wildcard, you need to prefix (escape) the dot with a backslash. The same is true for an asterisk if the asterisk follows a single dot. Though, asterisk alone (i.e. not right after a dot) is understood to be part of the key without the backslash. Examples, quoting:

| Description | Key |
| :---: | :---: |
| Make a Key that should match an incoming Key wiPeeseriptiarvalue 47*1 | 4\%ey |
| Make a Key that should match an incoming Key starting with 47, then it should have one character of any kind, immediately followed by *9 (sample of such an incoming Key is $473^{*} 9$ ) | 47.1*9 |
| Make a Key that should match an incoming Key with the fixed value 47.1. | 47\. 1 |

## Substitution:

A return value (from the match in the look-up) can be manipulated so it contains parts of (or the whole) incoming Key. This is how:

- A wildcard expression might be surrounded by parentheses, thereby making a wildcard group
- There might be more than one wildcard group within the same Key. The characters in an incoming Key that matches a particular wilcard group could be part of the returned value.
- Placeholders (in the specified return Value) for these wildcard groups are $\$ 1$ for the first group, $\$ 2$ for the second, $\$ 3$ for the third, and so on.


## Examples:

| Description | Set <br> Key to | Set <br> Value to |
| :--- | :--- | :--- |
| Any incoming Key starting with 22, and which are exactly 8 digits, should be prefixed with the <br> value 0047 | $22(\ldots . .)$. | $004722 \$ 1$ |
| Incoming Key starting with 0047, and containing an unknown number of characters thereafter, <br> should have 0047 stripped off. | $0047\left(.^{*}\right)$ | $\$ 1$ |

