



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Rules Authoring Tool Help

Using Drools 5

Contents

- 1 Using Drools 5
 - 1.1 Hard Keywords
 - 1.2 Soft Keywords
 - 1.3 Escaping Hard Keywords

Using Drools 5

Drools 5 introduces the concept of hard and soft keywords.

Hard Keywords

Hard keywords are reserved—you cannot use any hard keyword when naming domain objects, properties, methods, functions and other elements that are used in the rule text. The following list of hard keywords must be avoided as identifiers when writing rules:

- true
- false
- null

Soft Keywords

Soft keywords are just recognized in their context, enabling you to use these words in any other place if you wish, although Genesys recommends avoiding them if possible to prevent confusion. The list of soft keywords is:

- | | | | | | | | | |
|------------------|--------------------|------------|--------------|------------|--------|--------------|-----------|--------|
| • lock-on-active | • activation-group | • package | • attributes | • template | • eval | • exists | • action | • init |
| | | • import | • rule | • query | • not | • forall | • reverse | |
| • date-effective | • agenda-group | • dialect | • extend | • declare | • in | • accumulate | • result | |
| | • ruleflow-group | • salience | • when | • function | • or | • collect | • end | |
| • date-expires | | • enabled | • then | • global | • and | • from | • over | |
| • no-loop | • entry-point | | | | | | | |
| • auto-focus | • duration | | | | | | | |

You can use these (hard and soft) words as part of a method name in camel case, for example `notSomething()` or `accumulateSomething()` without any issues.

Escaping Hard Keywords

Although the three hard keywords above are unlikely to be used in your existing domain models, if you absolutely need to use them as identifiers instead of keywords, the DRL language provides the ability to escape hard keywords on rule text. To escape a word, simply enclose it in grave accents, like this:

```
Holiday( `true` == "yes" ) //
```

Please note that Drools will resolve that reference to the method:

```
Holiday.isTrue()
```