

# Struts 2 Plugin OGNL Support

## OGNL (Object Graph Navigation Library) Support

Support for OGNL-expressions with many language specific features is available since IDEA 11. In 12.1+, Parser and PSI have been rewritten completely using [GrammarKit](#).

 Using OGNL in non-Struts 2 context  
Using "IntelliLang" plugin OGNL-language support can be enabled anywhere: method calls, XML attribute values, ... (see Settings->Project Settings->Language Injections)

 Editing tip  
Please use intention "Edit OGNL fragment" (Alt+Enter while cursor on OGNL expression) for better editing capabilities. To enable it: Settings->Intentions->Language Injection "Edit Injected Fragment"

 PLEASE NOTE: Support is under development, some expressions might not be parseable (yet) and will highlight bogus errors. See [list of known issues](#).

Symbol	Status
	fully supported
	partly supported
	not done yet
	future?

## Integration with Struts 2

The goal is to support OGNL-expressions everywhere in Struts 2 applications in the same way as done for EL-expressions in JSPs.

OGNL language is available automatically at the following places:

 JSP(X)

Struts UI, JQuery

- all attributes containing "%{" (multiple occurrences possible)
- all attributes with list expression "{ e1, e2, ..}"
- `<s:iterator>` "value" (without any prefix)

 Freemarker

Struts UI, JQuery

- all attributes containing "%{" (multiple occurrences possible)
- all attributes with list expression "{ e1, e2, ..}"

 Velocity

 validation.xml

- Validator "fieldexpression" -> "expression"

## Editor highlighting

 Syntax highlighting

Basic syntax highlighting (identifiers, numbers, string literals, keywords, ..)

```
%{myArray[42] * 3}  
%{'String literal'}  
%{ myInt gt 5 || myBool neq true}
```

✓ Escape sequences in strings

Highlight valid/invalid escape sequences

```
%{"valid escapes: My App\nVersion 1.0 \u00a9 2011 My Company"}  
%{"invalid escape: \uXXX "}
```

+ Semantic highlighting

Highlight missing closing brackets, parentheses, expressions, ...

```
%{myArray[42]}  
] expected
```

```
<s:textfield disabled="%{ id == 5 ? true : }"/>  
'else' expression expected
```

```
<s:textfield disabled="%{ id in {1, } }"/>  
sequence expected
```

✓ Highlight matching parentheses, brackets and braces

```
%{ (myInt gt 5) || myBool neq true}  
doubleList="%{top == 'North' ? {'Oregon', 'Washington'} : {'Texas', 'Florida'}}"
```

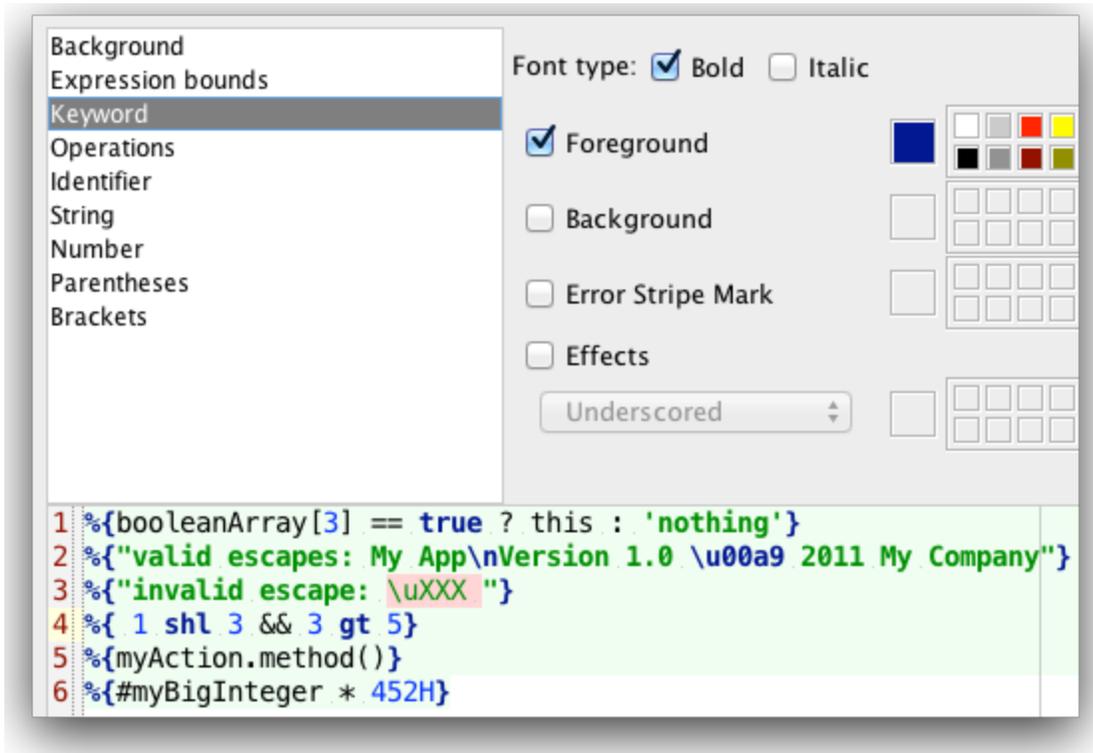
✓ Spellchecker

Strings and identifiers are checked for spelling errors using IDEA's integrated functionality.

✓ Color settings page

Settings->Editor->Colors & Fonts->OGNL

Customize the appearance (color, style) to your needs:



## Completion

- ✓ Auto-complete closing quote/tick
- ✓ Auto-complete closing '}'

Will be inserted automatically after typing "%{" and "{".

- + Keywords

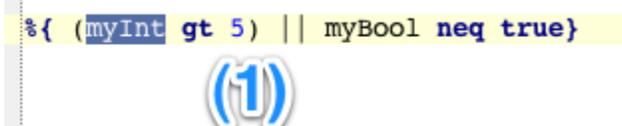
Context-sensitive variants.

- Identifiers

## Editing

- ✓ CTRL+W support

Expand/shrink selected part of expression semantically by repeatedly pressing CTRL-(Shift)-W:



```
%{ (myInt gt 5) || myBool neq true}
```

(2)

```
%{ (myInt gt 5) || myBool neq true}
```

(3)

```
%{ (myInt gt 5) | myBool neq true}
```

(4)

#### + Reformat

Basic whitespace-formatting (before/after operators, parentheses, brackets, ...) using Java-settings. To reformat, use intention "Edit OGNL fragment..." and run "Code->Reformat Code" from there.

#### + Live templates

Create your own OGNL-specific live templates (Context "OGNL") or use one of the predefined templates (see Settings->Live Templates).

NOTE Expanding live templates in injected context works with CTRL+J

#### - Complete Current Statement

Complete expression syntactically, e.g. adding missing closing ")", "}"

#### - Surround With

Provide specific "Surround with.." functionality.

## Navigation

#### - Go To Declaration

for identifiers

#### - Go To Code references

Navigate to Class/Method references.

## Documentation

#### - QuickDoc

Provide quick documentation on hovering over elements (type, name).