

WOOgnl Framework

Overview

OGNL stands for "Object Graph Navigation Language", and it defines an entire family of key-value coding-like abilities. As Jonathan Rentzsch put it in his CAWUG presentation on Project WONder, "Think: Key-Value Coding on Steroids". You can get more information on the specifics on OGNL at [the official OGNL website](#).

WOOgnl provides a framework that integrates the OGNL syntax into WO's standard binding resolution. By simply including the WOOgnl framework on your build path and preceding your binding value with a "~", it will be interpreted by WOOgnl.

Here are some examples that demonstrate just a tiny bit of the really cool things you can do:

- `value=~"Hello Mr.\" + session.user.firstName";`
- `value=~name.length().(#this>100?2*#this:20+#this);`
- `value=~#A=new NSMutableArray(),#A.addObject(name),#A";`

Here are some examples provided by Max Muller, WOOgnl's original author:

```
// Calling methods with arguments
Repetition1: WOREpetition {
    item = arrayItem;
    list = "~sort(anArray, \"name\")";
}
```

```
// Calling static methods
Repetition2: WOREpetition {
    item = arrayItem;
    list =
"@er.extensions.ERXArrayUtilities@sortedArraySortedWithKey(anArray,
\"name\")";
}
```

```
// Accessing static ivars
String1: WOSTring {
    value = "~@ognl.webobjects.WOOgnl@OgnlSpecialCharacters";
}
```

```
// Accessing static ivars within inner class
String1: WOSTring {
    value =
"@ognl.webobjects.WOOgnl$MyInnerClass@OgnlSpecialCharacters";
}
```

```
// Use of conditionals, note that every previous value of the . is
// pushed into the ivar #this
String2: WOString {
    value = "~name.length().(#this > 100? 2*#this : 20+#this)";
}
```

```
// String concat
String3: WOString {
    value = "~\"Hello Max \" + name";
}
```

```
// Use of set operator in. can also use in against NSArray and
NSSet objects
String4: WOString {
    value = "~name in {\"Main\", \"Something\"} ? \"Yes\" : \"No\"";
}
```

```
// Variable declaration. Note that commas allow multiple actions
// per expression.
String5: WOString {
    value = "~#A=new
com.webobjects.foundation.NSMutableArray(),#A.addObject(name),
#A.addObjectsFromArray(session.languages), #A";
}
```

Helper system

For more detail, check [WOOGNL Helper Functions](#). With these two lines in your properties file and Project WONDER,

```
ognl.helperFunctions=true
ognl.inlineBindings=true
```

You can declare your own helper class like this:

```
public class StringHelper {  
    public String capitalize(String str) {  
        // ...  
    }  
}
```

And use that in your binding by "piping your data"

```
<wo:str value="$person.name|capitalize" />
```