

As of version 3.3.0.000 Railo now supports named arguments for Java based build in functions in the same way as it is already possible for user defined and cfml based build in functions. The only difference is that Railo throws a exception if you define an unsupported argument.

Here are some examples:

```
arr=[1,2,3,4]>
dump(arrayLen(array: arr))>
dump(arrayLen("array": arr))>
dump(arrayContains(arr,2))>
dump(arrayContains(haystack:arr,needle:2))>
dump(arrayContains(needle:2,haystack:arr))>
dump(arrayContains(object:2,array:arr))><--- using alias name --->
dump(arrayContains(o:2,arr:arr))><--- using alias name --->
dump(listAppend(value:'d',list:'a,b,c'))>
dump(isValid(type:"string", value:arr))>
dump(isValid(type:"regex", value:"string",pattern:"*"))>
dump(isValid(type:"range", value:7,min:1,max:10))>
dump(isValid(type:"range", value:17,min:1,max:10))>
```

The implications of this change are that you now don't have to obey the order of the parameters when calling a function. So instead of:

```
#dateFormat(now(), "mm-dd-yyyy")#
```

you could call:

```
#dateFormat(mask:"mm-dd-yyyy",date:now())#
```

In addition, if some arguments can be omitted and you need to pass in an argument that is at a latter position, you don't have to pass empty or dummy values for the rest of the arguments. Here's a very good example:

```
cachePut(id:myID, value:myVar, cacheName:"myCache")>
```