

NewAntFeaturesInDetail NewIntrospectionRules

Tasks and data-types can now accept arbitrary classes that implement a certain interface or extend a certain class as nested elements by using `add` (`Type`) as the adder method. Details can be found [here](#).

Some of the built-in tasks/types have been enhanced with interface adders of that kind, you can now add a new condition by `<typedef>`ing the class (to give it a name) and nesting it into the `<condition>` task. The same is possible for selectors and filterreaders, it is described in more detail [here](#).

The short version looks like this

Define a condition that checks whether a given value ends with FOO.

```
package org.example;
import org.apache.tools.ant.taskdefs.condition.Condition;

public class EndsWithFoo implements Condition {
    private String value;

    public void setValue(String value) {
        this.value = value;
    }

    public boolean eval() {
        return value.endsWith("FOO");
    }
}
```

name it

```
<typedef name="endswithfoo" classname="org.example.EndsWithFoo"/>
```

and simply use it

```
<condition property="fooEndsWithFoo">
  <endswithfoo value="foo"/>
</condition>
```