Article
Alberto Fuentes · Feb 19, 2016 3m read

Object Generators: a homemade RuleEngine

The attached file contains an example of code generation using ObjectGenerators which builds a very simple homemade RuleEngine.

Code generation is an excellent way of increasing performance moving run-time calculations to compile-time.

We could generate code creating routines or implemeting methods using ObjectGenerators. In this example we are using ObjectGenerators.

Update: Rule Engine is now on GitHub https://github.com/intersystems-ib/cache-iat-ruleengine

A very simple code generated method

For starters, let's begin with a very basic code generated method:

- We want to implement *ConvertToCSV* method that returns the properties of an object separeted by a character (;)
- *ConvertToCSV* should go as fast as possible
- If we add more properties, *ConvertToCSV* should still be working!

```
Class IAT.S01.ObjectGen.Example Extends %RegisteredObject
{

Property Name As %String;

Property DOB As %Date;

Method ConvertToCSV() [ CodeMode = objectgenerator ]
{
  set myProperties = %compiledclass.Properties

  set csvLine=""
  for i=1:1:myProperties.Count() {
    #dim prop As %Dictionary.CompiledProperty
    set prop = myProperties.GetAt(i)
    if prop.Name'["%" {
      set csvLine = csvLine_".."_prop.Name_"_"";""_"
    }
  }
  // chop last underscore
  set csvLine = $extract(csvLine,1,*-1)

  do %code.WriteLine(" write "_csvLine)
}

}
```

What have we done?

- We are looping through the properties of the class using *%Dictionary* classes in compile-time.
- Then we are just writing the code that we need: in this case simply concatenating properties with a character (;)

Have a look at the generated code (Studio > View Other Code button):

```
zConvertToCSV() public {
 write ..DOB_";"_..Name_";"
}
```

So, at the end:

- We have avoided using %Dictionary classes in run-time.
- The code that is effectively executed in run-time is very simple (no loops, no opening objects, etc.)
- It could make a big difference if we need to execute this method a lot of times.

Homemade Rule Engine

Based on the simple example above we could develop a basic rule engine.

- The intention of the example is explain how something as complex as a rule engine can be modelled in Caché.
- Ensemble provides a very powerful Rule Engine that is way much better than this homemade example

Goal

- **Describe a rule using a human-readable XML, compile the rule and generate the code to evaluate that rule in run-time**

Run the example

- The example provides a patient alerts rule.
- After compiling the classes of the project, we can run the example:
  - Create a context to evaluate the rule: this context contains all needed data to evaluate the rule, in this case a patient object.
  - Evaluate the rule.
- To run the example, simply execute *##class(IAT.S01.Rules.Test.Examples).Run()* method

```
Class IAT.S01.Rules.Test.Example Extends %RegisteredObject
{

ClassMethod Run() As %Status
{
  set ret = $$$OK
  try {
    // create a patient
    set p = ##class(Patient).%New()
    set p.MRN="1234", p.Name="John", p.Surname="Snow", p.DOB=$zdh("1975-05-07",3)

    // create a rule context, set data
    set context = ##class(PatientContext).%New()
    set context.Patient = p

    // evaluate Patient Alerts Rule
    set ruleEngine = ##class(IAT.S01.Rules.Engine).%New()
    $$$TOE(sc, ruleEngine.Evaluate(
"IAT.S01.Rules.Test.PatientAlertsRule", context, .log))

    // print log
    write !,"Rule log:",!
    zwrite log
  } catch ex {
    set ret = ex.AsStatus()
    do $system.Status.DisplayError(ret)
  }
  quit ret
}

}
```

Executing it from a Terminal session:

```
USER>do ##class(IAT.S01.Rules.Test.Example).Run()

=======================
SendEmail
To:test@server.com
Body:
Patient is so old!
=======================

=======================
ShowObject
+----------------- general information ---------------
|      oref value: 1
|      class name: IAT.S01.Rules.Test.Patient
| reference count: 3
+----------------- attribute values ------------------
|                DOB = 49069
|                MRN = 1234
|               Name = "John"
|            Surname = "Snow"
=======================

Rule log:
log=4
log(1)="[2016-02-19 12:02:53] Rule: Not young anymore!"
log(2)="[2016-02-19 12:02:53] Action: 1"
log(3)="[2016-02-19 12:02:53] Action: 2"
log(4)="[2016-02-19 12:02:53] Action: 3"
```

Classes

#Code Snippet #Compiler #Object Data Model #Caché

Source URL:https://community.intersystems.com/post/object-generators-homemade-ruleengine