title: Custom Pointcuts description: Instrument custom method calls with the build-in extension mechanism not handled by the Sematext infrastructure and monitoing tracing agent

The built-in extension mechanism can be used to instrument custom method calls that are not handled by the Sematext tracing agent out of the box. An extension is set of pointcuts described in XML files placed in /opt/spm/spm-monitor/ext/tracing/ directory. The tracing agent scans all .xml files in that directory when it starts.

Below is the description of the XML format:

```
<instrumentation-descriptor name="descriptor-name">
  <pointcuts>
     <pointcut name="pointcut-1" [entry-point="true"] [transaction-name="custom-transaction-
         <class name="com.company.example.ClassName"/>
         <class name="com.company.example.ClassName$InnerClassName"/>
         <constructor signature="com.company.example.Service(int intParam, java.lang.String
         <method signature="java.lang.String com.company.example.Service#getUserName(com.co
     </pointcut>
  </pointcuts>
</instrumentation-descriptor>
```

An extension represents a set of pointcuts. Each pointcut defines a set of rules to match joinpoints which will be instrumented. A joinpoint is a particular method/constructor that will be instrumented if it satisfies given rules.

A pointcut has two optional parameters:

- **entry-point** - if set to true a new transaction will be created for each joinpoint.
- **transaction-name** - transaction name will inherit the joinpoint name unless this parameter is specified

The following rules are supported:

- **class** - all non-static methods of this class. The 'name' parameter should be a fully qualified java class name (see http://docs.oracle.com/javase/7/docs/api/java/lang/Class.html)
- **constructor** - constructor for a given class which matches given parameter types. The 'signature' should specify a fully qualified java class name followed by a list of comma-separated parameters inside parenthesis (e.g. (int foo, java.lang.String bar))
- **method** - method for a given class and all its subclasses whose methods match the specified method name, return type, and parameters.

Examples

An extension that enables instrumentation of *all* methods in given classes.

```
<instrumentation-descriptor name="spring-petclinic-extension">
  <pointcuts>
    <pointcut name="jpa-repository">
      <class name="org.springframework.samples.petclinic.repository.jpa.JpaOwnerRepositoryIm
      <class name="org.springframework.samples.petclinic.repository.jpa.JpaPetRepositoryImpl
      <class name="org.springframework.samples.petclinic.repository.jpa.JpaVetRepositoryImpl
      <class name="org.springframework.samples.petclinic.repository.jpa.JpaVisitRepositoryIm
    </pointcut>
    <pointcut name="jdbc-repository">
      <class name="org.springframework.samples.petclinic.repository.jdbc.JdbcOwnerRepository
      <class name="org.springframework.samples.petclinic.repository.jdbc.JdbcPetRepositoryIm
      <class name="org.springframework.samples.petclinic.repository.jdbc.JdbcVetRepositoryIm
      <class name="org.springframework.samples.petclinic.repository.jdbc.JdbcVisitRepository
    </pointcut>
  </pointcuts>
</instrumentation-descriptor>
```

In the extension below a new transaction is created each time methods CustomDescriptionTracerTest$Job\#doJob and CustomDescriptionTracerTest$CustomJob\#doJob are invoked. For the former the transaction name will be 'com.sematext.spm.tracing.agent.tracer.CustomDescrip while for the latter we set it to 'CustomTransactionName' using the transaction-name attribute.

```
<instrumentation-descriptor name="custom-description-tracer-test">
  <pointcuts>
    <pointcut name="method-pointcut-test">
      <method signature="void com.sematext.spm.tracing.agent.tracer.CustomDescriptionTracerT
      <method signature="void com.sematext.spm.tracing.agent.tracer.CustomDescriptionTracerT
    </pointcut>
    <pointcut name="whole-class-test">
      <class name="com.sematext.spm.tracing.agent.tracer.CustomDescriptionTracerTest$Service
    </pointcut>
    <pointcut name="constructor-test">
      <constructor signature="com.sematext.spm.tracing.agent.tracer.CustomDescriptionTracerT
      <constructor signature="com.sematext.spm.tracing.agent.tracer.CustomDescriptionTracerT
      <constructor signature="com.sematext.spm.tracing.agent.tracer.CustomDescriptionTracerT
    </pointcut>
    <pointcut name="job" entry-point="true">
      <method signature="void com.sematext.spm.tracing.agent.tracer.CustomDescriptionTracerT
    </pointcut>
    <pointcut name="custom-named-job" entry-point="true" transaction-name="CustomTransactior
      <method signature="void com.sematext.spm.tracing.agent.tracer.CustomDescriptionTracerT
    </pointcut>
  </pointcuts>
```

```
</instrumentation-descriptor>
```

To disable a pointcut definition from the /opt/spm/spm-monitor/ext/tracing/ directory simply rename the file so it doesn't have the .xml extension. For example, you might rename foo.xml to foo.xml.disabled. After that, restart your application.

IMPORTANT: instrumenting methods that are invoked frequently and execute quickly can and will increase the agent overhead. Use with caution.