title: Scala description: Sematext infrasctructure and application monitoring service provides Scala transaction tracing, giving insight into end-to-end request code execution that can span across multiple running applications and multiple servers

Tracing Scala Apps

Sematext application and infrastructure monitoring platform lets you trace transactions for Scala apps. In order to define custom pointcuts for a Scala app you need to follow the convention Scala uses to generate JVM classes. Below you can find a simple example that covers all basic cases:

```
trait UserService {
  def getUsers(): List[String]
}
class UserServiceImpl extends UserService {
  def getUsers(): List[String] =
    "Max_Rockatansky"::"Nux"::"Joe"::"Furiosa"::Nil
}
sealed trait Type {
  def typeName: String
}
case object Original extends Type {
  val typeName: String = "original"
}
case object Retweet extends Type  {
  val typeName: String = "retweet"
}
case class Tweet(text: String, t: Type = Original) {
  def vowelsCount: Int = {
    text.toLowerCase.filter("aeiou".toCharArray.contains).size
  }
}
class TweetService {
  def getTweets(user: String): List[Tweet] =
    Tweet("What a lovely day!")::Nil
}
object StatisticsService extends App {
  val userService = new UserServiceImpl
  val tweetService = new TweetService
  def vowelsCount(): Int = {
    val count = for {
      user <- userService.getUsers()
      tweet <- tweetService.getTweets(user) if tweet.t.typeName == "original"
    } yield tweet.vowelsCount
    count.sum
```

```
  }
  def serve(): Unit = {
    println(vowelsCount())
    Thread.sleep(1000)
    serve()
  }
  serve()
}
```

Custom pointcuts definition:

```
<instrumentation-descriptor name="scala">
  <pointcuts>
    <pointcut name="foo" entry-point="true">
      <method signature="int StatisticsService$#vowelsCount()"/>
    </pointcut>
    <pointcut name="userService">
      <method signature="scala.collection.immutable.List UserService#getUsers()"/>
    </pointcut>
    <pointcut name="tweetService">
      <method signature="scala.collection.immutable.List TweetService#getTweets(java.lang.St
    </pointcut>
    <pointcut name="type">
      <method signature="java.lang.String Original$#typeName()"/>
      <method signature="java.lang.String Retweet$#typeName()"/>
    </pointcut>
  </pointcuts>
</instrumentation-descriptor>
```