Schema documentation for config.xsd

june 19, 2011

Table of Contents

Namespace: "" 1
Schema(s) 1
Main schema `config.xsd` 1
Element(s) 1
Element `jmxpoller` 1
Element `formatter` 2
Element `cluster` 2
Element `mbean` 3
Element `attribute` 4
Element `jmxserver` 5
Attribute(s) 7
Attribute `formatter / @className` 7
Attribute `attribute / @name` 7
Attribute `attribute / @outputname` 7
Attribute `mbean / @domain` 7
Attribute `mbean / @properties` 8
Attribute `jmxserver / @host` 8
Attribute `jmxserver / @jmxpass` 8
Attribute `jmxserver / @jmxport` 8
Attribute `jmxserver / @jmxuser` 9
Attribute `jmxserver / @jvmDescription` 9
Attribute `jmxserver / @pid` 9
Attribute `jmxserver / @pidFile` 9
Attribute `jmxserver / @pidCommand` 10
Attribute `cluster / @name` 10
Attribute `cluster / @description` 10

Namespace: ""

Schema(s)

Main schema `config.xsd`

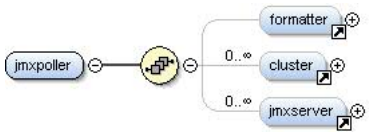
Namespace	No namespace
Properties	attribute form default: `unqualified` element form default: `qualified`

Element(s)

Element `jmxpoller`

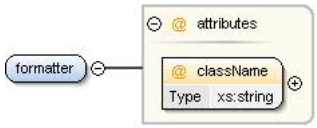
Namespace	No namespace

Annotations	`Root element of the configuration file.`
	`This configuration file is where you specify local and remote JMX`
	`servers to connect to across`
	`your enterprise and extract whatever MBean attributes you have declared`
	`to query.`
	`The result will then be written to STDOUT for SPLUNK indexing.`

Diagram	
Properties	content: `complex`
Model	formatter{0,1} , cluster* , jmxserver*

Children	cluster, formatter, jmxserver
Instance	<jmxpoller> <formatter className="">{0,1}</formatter> <cluster description="" name="">{0,unbounded}</cluster> <jmxserver host="" jmxpass="" jmxport="" jmxuser="" jvmDescription="" pid="" pidCommand="" pidFile="">jmxserver> </jmxpoller>
Source	<xs:element name="jmxpoller"> <xs:annotation> <xs:documentation>Root element of the configuration file. This configuration file is where you specify local and remote JMX servers to connect to across your enterprise and extract whatever MBean attributes you have declared to query. The result will then be written to STDOUT for SPLUNK indexing.</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element minOccurs="0" maxOccurs="1" ref="formatter"/> <xs:element minOccurs="0" maxOccurs="unbounded" ref="cluster"/> <xs:element minOccurs="0" maxOccurs="unbounded" ref="jmxserver"/> </xs:sequence> </xs:complexType> </xs:element>

Element formatter

Namespace	No namespace
Annotations	Custom formatter declaration allows you to override the default STDOUT format
Diagram	
Properties	content: complex
Used by	Element: jmxpoller

Attributes

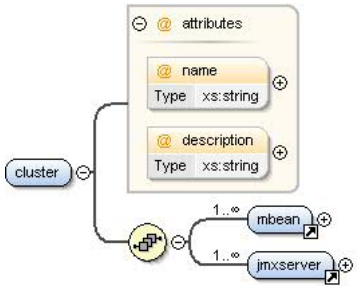
QName	Type	Fixed	Default	Use
className	xs:string			required
	Fully qualified Java class name of the formatter implementation, implements the com.dtdsoftware.splunk.formatter.Formatter interface			

Source

```
<xs:element name="formatter">
  <xs:annotation>
    <xs:documentation>Custom formatter declaration allows you to override the default
 STDOUT format</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="className" type="xs:string" use="required">
      <xs:annotation>
        <xs:documentation>Fully qualified Java class name of the formatter implementation,
 implements the com.dtdsoftware.splunk.formatter.Formatter interface</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Element cluster

Namespace	No namespace
Annotations	For JVMs with the same MBeans, you can group them under this element so you only have to declare the common beans to query once. You can still declare additional mbeans specfic to each jmxserver within the jmxserver elements.

Diagram	
Properties	content: complex
Used by	Element jmxpoller
Model	mbean+ , jmxserver+
Children	jmxserver, mbean

Instance	<pre><cluster description="" name="">
	  <mbean domain="" properties="">{1,unbounded}</mbean>
	  <jmxserver host="" jmxpass="" jmxport="" jmxuser="" jvmDescription="" pid="" pidCommand="" pidFile="">
	jmxserver>
	</cluster></pre>

Attributes

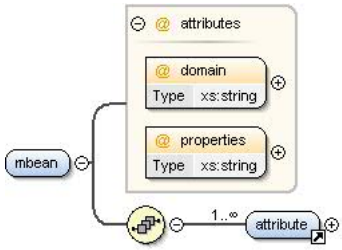
QName	Type	Fixed	Default	Use
description	xs:string			optional
	Description of this cluster			
name	xs:string			optional
	Name for this cluster			

Source

```
<xs:element name="cluster">
  <xs:annotation>
    <xs:documentation>For JVMs with the same MBeans, you can group them under this
 element so you only have to declare the common beans to query once. You can still
 declare additional mbeans specfic to each jmxserver within the jmxserver elements.</
xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="mbean"/>
      <xs:element maxOccurs="unbounded" ref="jmxserver"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string">
      <xs:annotation>
        <xs:documentation>Name for this cluster</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="description" type="xs:string">
      <xs:annotation>
        <xs:documentation>Description of this cluster</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Element mbean

Namespace	No namespace
Annotations	An MBean to query Standard JMX object name wildcard patterns * and ? are supported If no values are specified for the "domain" and "properties" attributes , the value will default to the * wildcard

Diagram	
Properties	content: complex
Used by	Elements cluster, jmxserver
Model	attribute+
Children	attribute

Instance	`<mbean domain="" properties="">`
	`  <attribute name="" outputname="">{1,unbounded}</attribute>`
	`</mbean>`

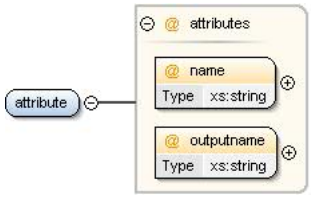
Attributes

QName	Type	Fixed	Default	Use
domain	xs:string			required
	The MBean domain			
properties	xs:string			required
	The MBean properties string in "key=value,key2=value2" format			

Source

```
<xs:element name="mbean">
  <xs:annotation>
    <xs:documentation>An MBean to query Standard JMX object name wildcard patterns * and ?
 are supported If no values are specified for the "domain" and "properties" attributes ,
 the value will default to the * wildcard</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="attribute"/>
    </xs:sequence>
    <xs:attribute name="domain" use="required" type="xs:string">
      <xs:annotation>
        <xs:documentation>The MBean domain</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="properties" use="required" type="xs:string">
      <xs:annotation>
        <xs:documentation>The MBean properties string in "key=value,key2=value2" format</
xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Element `attribute`

Namespace	No namespace
Annotations	An MBean attribute
Diagram	
Properties	content: complex
Used by	Element mbean

Attributes

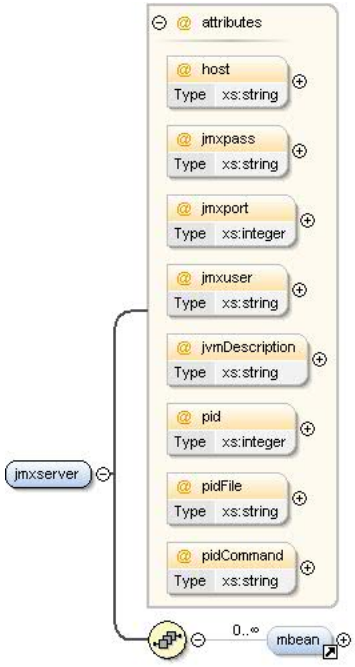
QName	Type	Fixed	Default	Use
name	xs:string			required
	The attribute name For attributes that are multi			

QName	Type	Fixed	Default	Use
	level ie: composite and tabular attributes , then you can use a ":" delimited notation for specifying the attribute name. ie: foo:goo:myattribute			
outputname	xs:string			required
	The attribute key that is output to STDOUT for SPLUNK indexing			

Source

```
<xs:element name="attribute">
  <xs:annotation>
    <xs:documentation>An MBean attribute</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="name" use="required" type="xs:string">
      <xs:annotation>
        <xs:documentation>The attribute name For attributes that are multi level ie:
 composite and tabular attributes , then you can use a ":" delimited notation for
 specifying the attribute name. ie: foo:goo:myattribute</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="outputname" use="required" type="xs:string">
      <xs:annotation>
        <xs:documentation>The attribute key that is output to STDOUT for SPLUNK indexing</
xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Element `jmxserver`

Namespace	No namespace
Annotations	A local or remote JMX Server to connect to
Diagram	
Properties	content: complex
Used by	Elements: cluster, jmxpoller
Model	mbean*
Children	mbean

Instance	`<jmxserver host="" jmxpass="" jmxport="" jmxuser="" jvmDescription="" pid="" pidCommand="" pidFile="">`
	`  <mbean domain="" properties="">{0,unbounded}</mbean>`
	`</jmxserver>`

Attributes

QName	Type	Fixed	Default	Use
host	xs:string			optional
	IP Address, Hostname or DNS Alias.			
jmxpass	xs:string			optional
	JMX Password			
jmxport	xs:integer			optional
	JMX Port			
jmxuser	xs:string			optional
	JMX Username			
jvmDescription	xs:string			optional
	A description of this JVM			
pid	xs:integer			optional
	Process ID for attaching directly to a locally running JVM			
pidCommand	xs:string			optional
	Command/Script that outputs to STDOUT the Process ID for attaching directly to a locally running JVM			
pidFile	xs:string			optional
	File containing the Process ID for attaching directly to a locally running JVM.The only file contents should be the PID on the first line of the file.			

Source

```
<xs:element name="jmxserver">
  <xs:annotation>
    <xs:documentation>A local or remote JMX Server to connect to</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="mbean"/>
    </xs:sequence>
    <xs:attribute name="host" type="xs:string">
      <xs:annotation>
        <xs:documentation>IP Address, Hostname or DNS Alias.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="jmxpass" type="xs:string">
      <xs:annotation>
        <xs:documentation>JMX Password</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="jmxport" type="xs:integer">
      <xs:annotation>
        <xs:documentation>JMX Port</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="jmxuser" type="xs:string">
      <xs:annotation>
        <xs:documentation>JMX Username</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="jvmDescription" type="xs:string">
      <xs:annotation>
        <xs:documentation>A description of this JVM</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="pid" type="xs:integer">
      <xs:annotation>
        <xs:documentation>Process ID for attaching directly to a locally running JVM</
xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="pidFile" type="xs:string">
      <xs:annotation>
        <xs:documentation>File containing the Process ID for attaching directly to a
 locally running JVM.The only file contents should be the PID on the first line of the
 file.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="pidCommand" type="xs:string">
      <xs:annotation>
        <xs:documentation>Command/Script that outputs to STDOUT the Process ID for
 attaching directly to a locally running JVM</xs:documentation>
```

```
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Attribute(s)

Attribute `formatter / @className`

Namespace	No namespace
Annotations	Fully qualified Java class name of the formatter implementation, implements the com.dtdsoftware.splunk.formatter.Formatter interface
Type	xs:string
Properties	use: required
Used by	Element formatter
Source	<xs:attribute name="className" type="xs:string" use="required"> <xs:annotation> <xs:documentation>Fully qualified Java class name of the formatter implementation, implements the com.dtdsoftware.splunk.formatter.Formatter interface</xs:documentation> </xs:annotation> </xs:attribute>

Attribute `attribute / @name`

Namespace	No namespace
Annotations	The attribute name For attributes that are multi level ie: composite and tabular attributes , then you can use a ":" delimited notation for specifying the attribute name. ie: foo:goo:myattribute
Type	xs:string
Properties	use: required
Used by	Element attribute
Source	<xs:attribute name="name" use="required" type="xs:string"> <xs:annotation> <xs:documentation>The attribute name For attributes that are multi level ie: composite and tabular attributes , then you can use a ":" delimited notation for specifying the attribute name. ie: foo:goo:myattribute</xs:documentation> </xs:annotation> </xs:attribute>

Attribute `attribute / @outputname`

Namespace	No namespace
Annotations	The attribute key that is output to STDOUT for SPLUNK indexing
Type	xs:string
Properties	use: required
Used by	Element attribute
Source	<xs:attribute name="outputname" use="required" type="xs:string"> <xs:annotation> <xs:documentation>The attribute key that is output to STDOUT for SPLUNK indexing</xs:documentation> </xs:annotation> </xs:attribute>

Attribute `mbean / @domain`

Namespace	No namespace
Annotations	The MBean domain

Type	xs:string
Properties	use: required
Used by	Element mbean
Source	<xs:attribute name="domain" use="required" type="xs:string"> <xs:annotation> <xs:documentation>The MBean domain</xs:documentation> </xs:annotation> </xs:attribute>

Attribute mbean / @properties

Namespace	No namespace
Annotations	The MBean properties string in "key=value,key2=value2" format
Type	xs:string
Properties	use: required
Used by	Element mbean
Source	<xs:attribute name="properties" use="required" type="xs:string"> <xs:annotation> <xs:documentation>The MBean properties string in "key=value,key2=value2" format</xs:documentation> </xs:annotation> </xs:attribute>

Attribute jmxserver / @host

Namespace	No namespace
Annotations	IP Address, Hostname or DNS Alias.
Type	xs:string
Properties	content: simple
Used by	Element jmxserver
Source	<xs:attribute name="host" type="xs:string"> <xs:annotation> <xs:documentation>IP Address, Hostname or DNS Alias.</xs:documentation> </xs:annotation> </xs:attribute>

Attribute jmxserver / @jmxpass

Namespace	No namespace
Annotations	JMX Password
Type	xs:string
Properties	content: simple
Used by	Element jmxserver
Source	<xs:attribute name="jmxpass" type="xs:string"> <xs:annotation> <xs:documentation>JMX Password</xs:documentation> </xs:annotation> </xs:attribute>

Attribute jmxserver / @jmxport

Namespace	No namespace
Annotations	JMX Port
Type	xs:integer
Properties	content: simple
Used by	Element jmxserver
Source	<xs:attribute name="jmxport" type="xs:integer">

```
  <xs:annotation>
    <xs:documentation>JMX Port</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Attribute `jmxserver / @jmxuser`

Namespace	No namespace
Annotations	`JMX Username`
Type	xs:string
Properties	content: `simple`
Used by	Element jmxserver
`Source`	

```
<xs:attribute name="jmxuser" type="xs:string">
  <xs:annotation>
    <xs:documentation>JMX Username</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Attribute `jmxserver / @jvmDescription`

Namespace	No namespace
Annotations	`A description of this JVM`
Type	xs:string
Properties	content: `simple`
Used by	Element jmxserver
`Source`	

```
<xs:attribute name="jvmDescription" type="xs:string">
  <xs:annotation>
    <xs:documentation>A description of this JVM</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Attribute `jmxserver / @pid`

Namespace	No namespace
Annotations	`Process ID for attaching directly to a locally running JVM`
Type	xs:integer
Properties	content: `simple`
Used by	Element jmxserver
`Source`	

```
<xs:attribute name="pid" type="xs:integer">
  <xs:annotation>
    <xs:documentation>Process ID for attaching directly to a locally running JVM</
xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Attribute `jmxserver / @pidFile`

Namespace	No namespace
Annotations	`File containing the Process ID for attaching directly to a locally running JVM.The only file contents should be the PID on the first line of the file.`
Type	xs:string
Properties	content: `simple`
Used by	Element jmxserver
`Source`	

```
<xs:attribute name="pidFile" type="xs:string">
  <xs:annotation>
    <xs:documentation>File containing the Process ID for attaching directly to a locally
 running JVM.The only file contents should be the PID on the first line of the file.</
xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Attribute `jmxserver / @pidCommand`

Namespace	No namespace
Annotations	`Command/Script that outputs to STDOUT the Process ID for attaching directly to a locally running JVM`
Type	xs:string
Properties	content: `simple`
Used by	Element jmxserver
Source	<xs:attribute name="pidCommand" type="xs:string"> <xs:annotation> <xs:documentation>Command/Script that outputs to STDOUT the Process ID for attaching directly to a locally running JVM</xs:documentation> </xs:annotation> </xs:attribute>

Attribute `cluster / @name`

Namespace	No namespace
Annotations	`Name for this cluster`
Type	xs:string
Properties	content: `simple`
Used by	Element cluster
Source	<xs:attribute name="name" type="xs:string"> <xs:annotation> <xs:documentation>Name for this cluster</xs:documentation> </xs:annotation> </xs:attribute>

Attribute `cluster / @description`

Namespace	No namespace
Annotations	`Description of this cluster`
Type	xs:string
Properties	content: `simple`
Used by	Element cluster
Source	<xs:attribute name="description" type="xs:string"> <xs:annotation> <xs:documentation>Description of this cluster</xs:documentation> </xs:annotation> </xs:attribute>