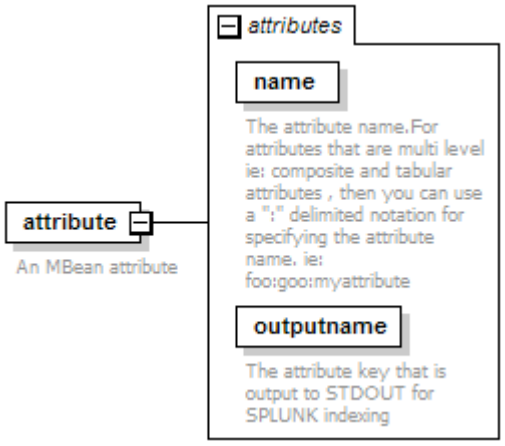
Schema **config.xsd**

Elements
attribute
cluster
formatter
jmxpoller
jmxserver
mbean
notificationListener
operation
param
parameter

element **attribute**

diagram	
properties	content complex
used by	element **mbean**

attributes

Name	Type	Use	Default	Fixed	Annotation
name	**xs:string**	required			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
outputname	**xs:string**	required			documentation The attribute key that is output to STDOUT for SPLUNK indexing

annotation	documentation An MBean attribute
source	<xs:element name="attribute"> <xs:annotation>

```
    <xs:documentation>An MBean attribute</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="name" type="xs:string" use="required">
      <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" type="xs:string" use="required">
      <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>
```

attribute **attribute/@name**

type	**xs:string**
properties	use required
annotation	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

source	`<xs:attribute name="name" type="xs:string" use="required">`
	`  <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>`

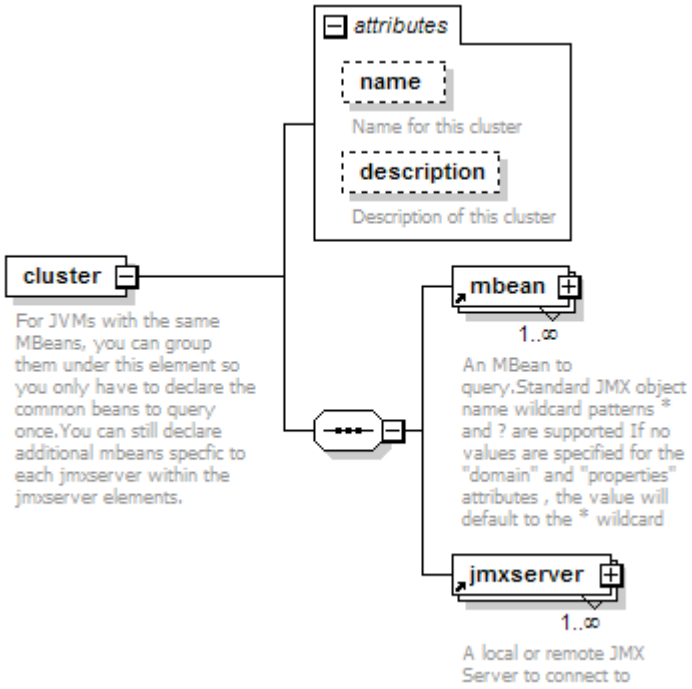
attribute **attribute/@outputname**

type	**xs:string**
properties	use required
annotation	documentation The attribute key that is output to STDOUT for SPLUNK indexing

source	`<xs:attribute name="outputname" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation>The attribute key that is output to STDOUT for SPLUNK indexing</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

element **cluster**

diagram



properties

content complex

children **mbean** **jmxserver**

used by

element **jmxpoller**

attributes

Name	Type	Use	Default	Fixed	Annotation
name	**xs:string**				documentation Name for this cluster
description	**xs:string**				documentation Description of this cluster

annotation

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.

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 ref="mbean" maxOccurs="unbounded"/>
   <xs:element ref="jmxserver" maxOccurs="unbounded"/>
  </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>
```

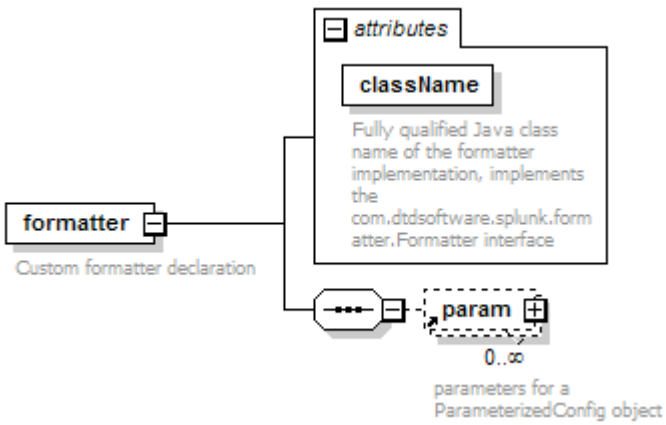
attribute **cluster/@name**

type	**xs:string**
annotation	documentation Name for this 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**

type	**xs:string**
annotation	documentation Description of this cluster
source	<xs:attribute name="description" type="xs:string"> <xs:annotation> <xs:documentation>Description of this cluster</xs:documentation> </xs:annotation> </xs:attribute>

element **formatter**

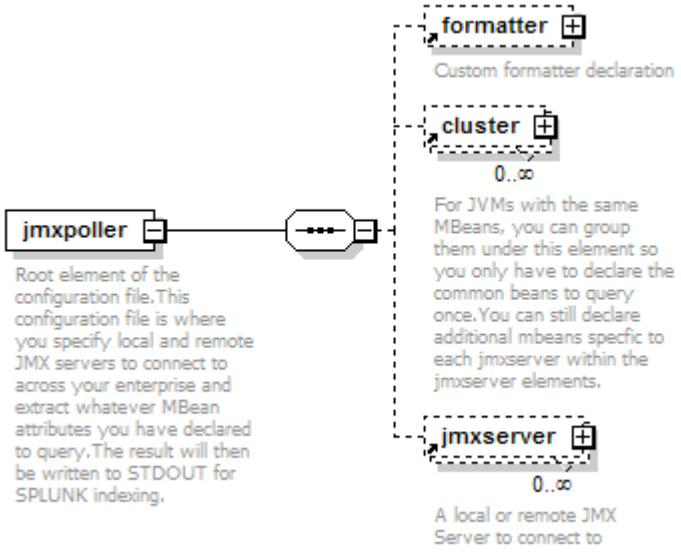
diagram	
properties	content complex
children	**param**
used by	element **jmxpoller**
attributes	Name: className; Type: **xs:string**; Use: required; Default: ; Fixed: ; Annotation: documentation — Fully qualified Java class name of the formatter implementation, implements the

	com.dtdsoftware.splunk.formatter.Formatter interface
annotation	documentation Custom formatter declaration
source	<xs:element name="formatter"> <xs:annotation> <xs:documentation>Custom formatter declaration</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="param" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> <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>

attribute **formatter/@className**

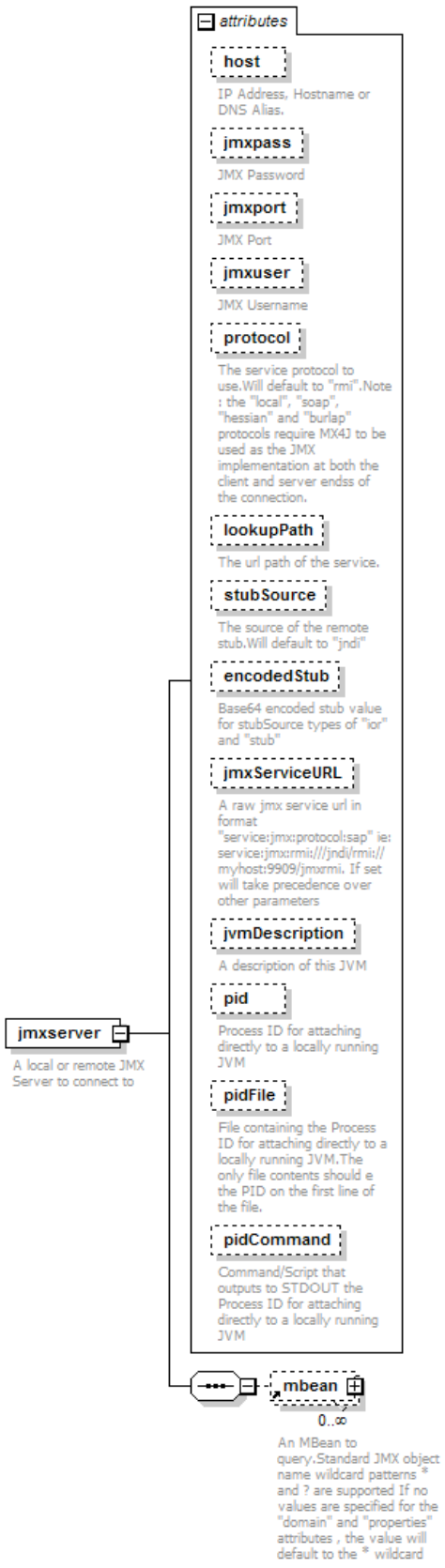
type	**xs:string**
properties	use required
annotation	documentation Fully qualified Java class name of the formatter implementation, implements the com.dtdsoftware.splunk.formatter.Formatter interface
source	<xs:attribute name="className" type="xs:string" use="required"> <xs:annotation> <xs:documentation>Fully qualified Java class name of the formatter implementation, implements the com.dtdsoftware.splunk.formatter.Formatter interface</xs:documentation> </xs:annotation> </xs:attribute>

element **jmxpoller**

diagram	
properties	content complex
children	**formatter** **cluster** **jmxserver**
annotation	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.
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 ref="formatter" minOccurs="0" maxOccurs="1"/> <xs:element ref="cluster" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="jmxserver" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element>

element **jmxserver**

diagram



properties
content complex

children
mbean

used by
elements **cluster jmxpoller**

attributes

Name	Type	Use	Default	Fixed	Annotation
host	**xs:string**				documentation IP Address, Hostname or DNS Alias.
jmxpass	**xs:string**				documentation JMX Password
jmxport	**xs:integer**				documentation JMX Port
jmxuser	**xs:string**				documentation JMX Username
protocol	**derived by: xs:string**				documentation The service protocol to use.Will default to "rmi".Note : the "local", "soap", "hessian" and "burlap" protocols require MX4J to be used as the JMX implementation at both the client and server endss of the connection.
lookupPath	**xs:string**				documentation The url path of the service.
stubSource	**derived by: xs:string**				documentation The source of the remote stub.Will default to "jndi"
encodedStub	**xs:string**				documentation Base64 encoded stub value for stubSource types of "ior" and "stub"
jmxServiceURL	**xs:string**				documentation A raw jmx service url in format "service:jmx:protocol:sap" ie: service:jmx:rmi:///jndi/rmi://myhost:9909/jmxrmi. If set will take precedence over other parameters
jvmDescription	**xs:string**				documentation A description of this JVM
pid	**xs:integer**				documentation Process ID for attaching directly to a locally running JVM
pidFile	**xs:string**				documentation File containing the Process ID for attaching directly to a locally running JVM.The only file contents should e the PID on the first line of the file.
pidCommand	**xs:string**				documentation Command/Script that outputs to STDOUT the Process ID for attaching directly to a locally running JVM

annotation
documentation
A local or remote JMX Server to connect to

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 ref="mbean" minOccurs="0" maxOccurs="unbounded"/>
  </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="protocol">
      <xs:annotation>
        <xs:documentation>The service protocol to use.Will default to "rmi".Note : the "local", "soap",
"hessian" and "burlap" protocols require MX4J to be used as the JMX implementation at both the
client and server endss of the connection.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="rmi"/>
          <xs:enumeration value="iiop"/>
          <xs:enumeration value="local"/>
          <xs:enumeration value="soap"/>
          <xs:enumeration value="hessian"/>
          <xs:enumeration value="burlap"/>
          <xs:enumeration value="soap+ssl"/>
          <xs:enumeration value="hessian+ssl"/>
          <xs:enumeration value="burlap+ssl"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="lookupPath" type="xs:string">
      <xs:annotation>
        <xs:documentation>The url path of the service.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="stubSource">
      <xs:annotation>
        <xs:documentation>The source of the remote stub.Will default to "jndi"</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="jndi"/>
          <xs:enumeration value="stub"/>
          <xs:enumeration value="ior"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="encodedStub" type="xs:string">
      <xs:annotation>
        <xs:documentation>Base64 encoded stub value for stubSource types of "ior" and
"stub"</xs:documentation>
      </xs:annotation>
    </xs:attribute>
```

```
    <xs:attribute name="jmxServiceURL" type="xs:string">
      <xs:annotation>
        <xs:documentation>A raw jmx service url in format "service:jmx:protocol:sap" ie:
service:jmx:rmi:///jndi/rmi://myhost:9909/jmxrmi. If set will take precedence over other
parameters</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 e 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 **jmxserver/@host**

type	**xs:string**
annotation	documentation IP Address, Hostname or DNS Alias.

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**

type	**xs:string**
annotation	documentation JMX Password

source	`<xs:attribute name="jmxpass" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>JMX Password</xs:documentation>`
	`  </xs:annotation>`

	</xs:attribute>

attribute **jmxserver/@jmxport**

type	**xs:integer**
annotation	documentation JMX Port
source	<xs:attribute name="jmxport" type="xs:integer"> <xs:annotation> <xs:documentation>JMX Port</xs:documentation> </xs:annotation> </xs:attribute>

attribute **jmxserver/@jmxuser**

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

attribute **jmxserver/@protocol**

type	restriction of **xs:string**
facets	Kind / Value / Annotation enumeration rmi enumeration iiop enumeration local enumeration soap enumeration hessian enumeration burlap enumeration soap+ssl enumeration hessian+ssl enumeration burlap+ssl
annotation	documentation The service protocol to use.Will default to "rmi".Note : the "local", "soap", "hessian" and "burlap" protocols require MX4J to be used as the JMX implementation at both the client and server endss of the connection.
source	<xs:attribute name="protocol"> <xs:annotation> <xs:documentation>The service protocol to use.Will default to "rmi".Note : the "local", "soap", "hessian" and "burlap" protocols require MX4J to be used as the JMX implementation at both the client and server endss of the connection.</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="rmi"/> <xs:enumeration value="iiop"/>

```
      <xs:enumeration value="local"/>
      <xs:enumeration value="soap"/>
      <xs:enumeration value="hessian"/>
      <xs:enumeration value="burlap"/>
      <xs:enumeration value="soap+ssl"/>
      <xs:enumeration value="hessian+ssl"/>
      <xs:enumeration value="burlap+ssl"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

attribute **jmxserver/@lookupPath**

type	**xs:string**
annotation	documentation The url path of the service.
source	<xs:attribute name="lookupPath" type="xs:string"> <xs:annotation> <xs:documentation>The url path of the service.</xs:documentation> </xs:annotation> </xs:attribute>

attribute **jmxserver/@stubSource**

type	restriction of **xs:string**
facets	Kind / Value / Annotation enumeration jndi enumeration stub enumeration ior
annotation	documentation The source of the remote stub.Will default to "jndi"
source	<xs:attribute name="stubSource"> <xs:annotation> <xs:documentation>The source of the remote stub.Will default to "jndi"</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="jndi"/> <xs:enumeration value="stub"/> <xs:enumeration value="ior"/> </xs:restriction> </xs:simpleType> </xs:attribute>

attribute **jmxserver/@encodedStub**

type	**xs:string**
annotation	documentation Base64 encoded stub value for stubSource types of "ior" and "stub"
source	<xs:attribute name="encodedStub" type="xs:string"> <xs:annotation>

	<xs:documentation>Base64 encoded stub value for stubSource types of "ior" and "stub"</xs:documentation> </xs:annotation> </xs:attribute>

attribute **jmxserver/@jmxServiceURL**

type	**xs:string**
annotation	documentation A raw jmx service url in format "service:jmx:protocol:sap" ie: service:jmx:rmi:///jndi/rmi://myhost:9909/jmxrmi. If set will take precedence over other parameters
source	<xs:attribute name="jmxServiceURL" type="xs:string"> <xs:annotation> <xs:documentation>A raw jmx service url in format "service:jmx:protocol:sap" ie: service:jmx:rmi:///jndi/rmi://myhost:9909/jmxrmi. If set will take precedence over other parameters</xs:documentation> </xs:annotation> </xs:attribute>

attribute **jmxserver/@jvmDescription**

type	**xs:string**
annotation	documentation A description of this JVM
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**

type	**xs:integer**
annotation	documentation Process ID for attaching directly to a locally running JVM
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**

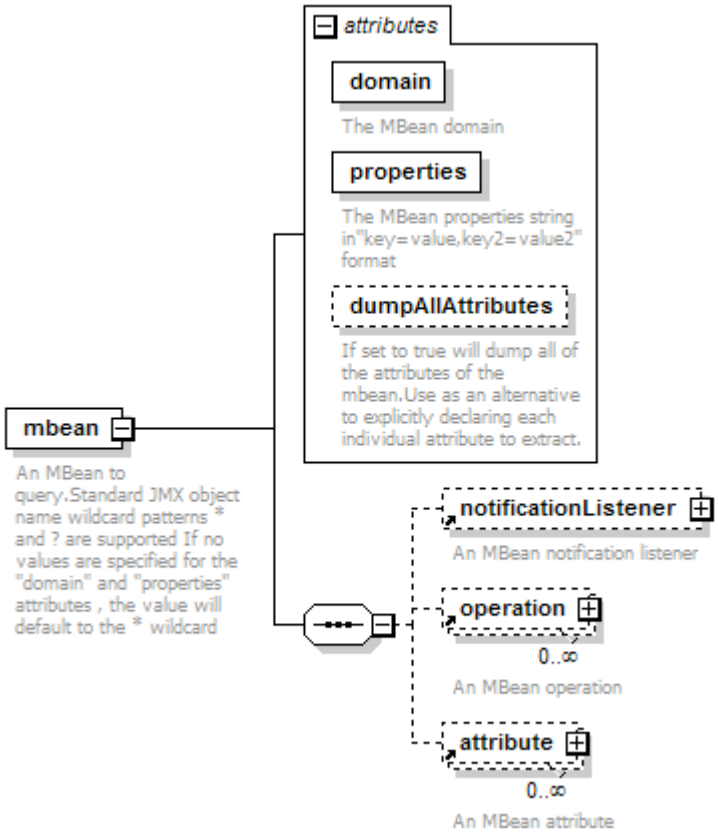
type	**xs:string**
annotation	documentation File containing the Process ID for attaching directly to a locally running JVM.The only file contents should e the PID on the first line of the file.
source	<xs:attribute name="pidFile" type="xs:string"> <xs:annotation> <xs:documentation>File containing the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

	JVM.The only file contents should e the PID on the first line of the file.</xs:documentation> </xs:annotation> </xs:attribute>

attribute **jmxserver/@pidCommand**

type	**xs:string**
annotation	documentation Command/Script that outputs to STDOUT the Process ID for attaching directly to a locally running JVM
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>

element **mbean**

diagram	
properties	content complex
children	**notificationListener operation attribute**
used by	elements **cluster jmxserver**
attributes	Name **domain** Type **xs:string** Use required Default Fixed Annotation documentation The MBean domain

	properties	**xs:string**	required	documentation The MBean properties string in"key=value,key2=value2" format
	dumpAllAttributes	**xs:boolean**		documentation If set to true will dump all of the attributes of the mbean.Use as an alternative to explicitly declaring each individual attribute to extract.
annotation	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			

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 ref="notificationListener" minOccurs="0" maxOccurs="1"/>
   <xs:element ref="operation" minOccurs="0" maxOccurs="unbounded"/>
   <xs:element ref="attribute" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="domain" type="xs:string" use="required">
   <xs:annotation>
    <xs:documentation>The MBean domain</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="properties" type="xs:string" use="required">
   <xs:annotation>
    <xs:documentation>The MBean properties string in"key=value,key2=value2" format</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="dumpAllAttributes" type="xs:boolean">
   <xs:annotation>
    <xs:documentation>If set to true will dump all of the attributes of the mbean.Use as an alternative to explicitly declaring each individual attribute to extract.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
 </xs:complexType>
</xs:element>
```

attribute **mbean/@domain**

type	**xs:string**
properties	use required
annotation	documentation The MBean domain

source

```
<xs:attribute name="domain" type="xs:string" use="required">
 <xs:annotation>
  <xs:documentation>The MBean domain</xs:documentation>
 </xs:annotation>
```

	</xs:attribute>

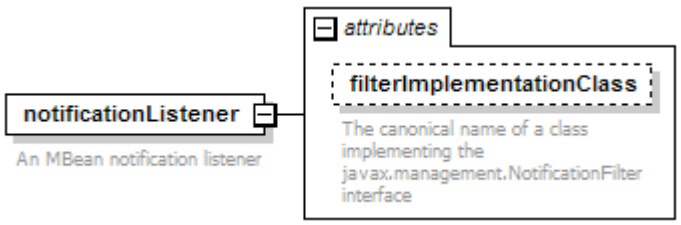
attribute **mbean/@properties**

type	**xs:string**
properties	use required
annotation	documentation The MBean properties string in"key=value,key2=value2" format
source	<xs:attribute name="properties" type="xs:string" use="required"> <xs:annotation> <xs:documentation>The MBean properties string in"key=value,key2=value2" format</xs:documentation> </xs:annotation> </xs:attribute>

attribute **mbean/@dumpAllAttributes**

type	**xs:boolean**
annotation	documentation If set to true will dump all of the attributes of the mbean.Use as an alternative to explicitly declaring each individual attribute to extract.
source	<xs:attribute name="dumpAllAttributes" type="xs:boolean"> <xs:annotation> <xs:documentation>If set to true will dump all of the attributes of the mbean.Use as an alternative to explicitly declaring each individual attribute to extract.</xs:documentation> </xs:annotation> </xs:attribute>

element **notificationListener**

diagram	
properties	content complex
used by	element **mbean**
attributes	Name: filterImplementationClass; Type: **xs:string**; Use: ; Default: ; Fixed: ; Annotation: documentation: The canonical name of a class implementing the javax.management.NotificationFilter interface
annotation	documentation An MBean notification listener
source	<xs:element name="notificationListener"> <xs:annotation> <xs:documentation>An MBean notification listener</xs:documentation> </xs:annotation>

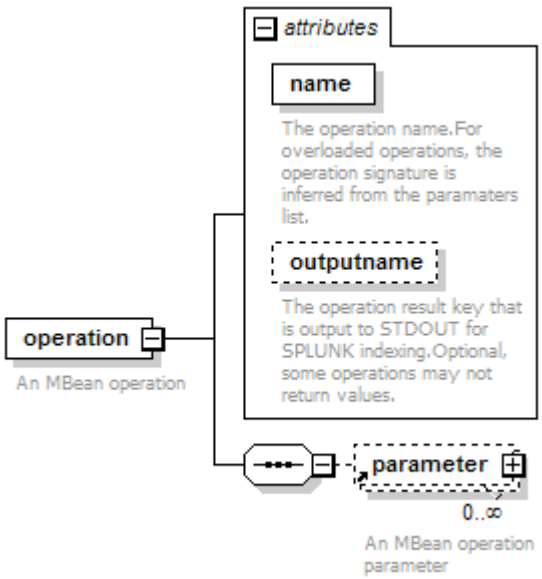
```
  <xs:complexType>
   <xs:attribute name="filterImplementationClass" type="xs:string">
    <xs:annotation>
     <xs:documentation>The canonical name of a class implementing the
javax.management.NotificationFilter interface</xs:documentation>
    </xs:annotation>
   </xs:attribute>
  </xs:complexType>
</xs:element>
```

attribute **notificationListener/@filterImplementationClass**

type	**xs:string**
annotation	documentation The canonical name of a class implementing the javax.management.NotificationFilter interface

source	`<xs:attribute name="filterImplementationClass" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>The canonical name of a class implementing the javax.management.NotificationFilter interface</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

element **operation**

diagram	
properties	content complex
children	**parameter**
used by	element **mbean**

attributes	Name	Type	Use	Default	Fixed	Annotation
	name	**xs:string**	required			documentation The operation name.For overloaded operations, the operation

	outputname **xs:string**	signature is inferred from the paramaters list. documentation The operation result key that is output to STDOUT for SPLUNK indexing.Optional, some operations may not return values.
annotation	documentation An MBean operation	
source	<xs:element name="operation"> <xs:annotation> <xs:documentation>An MBean operation</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="parameter" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute name="name" type="xs:string" use="required"> <xs:annotation> <xs:documentation>The operation name.For overloaded operations, the operation signature is inferred from the paramaters list.</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="outputname" type="xs:string"> <xs:annotation> <xs:documentation>The operation result key that is output to STDOUT for SPLUNK indexing.Optional, some operations may not return values.</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType> </xs:element>	

attribute **operation/@name**

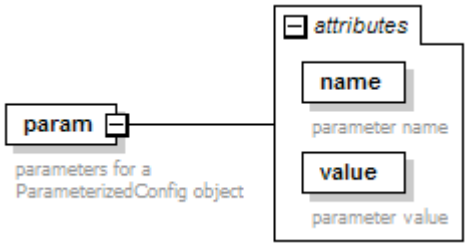
type	**xs:string**
properties	use required
annotation	documentation The operation name.For overloaded operations, the operation signature is inferred from the paramaters list.
source	<xs:attribute name="name" type="xs:string" use="required"> <xs:annotation> <xs:documentation>The operation name.For overloaded operations, the operation signature is inferred from the paramaters list.</xs:documentation> </xs:annotation> </xs:attribute>

attribute **operation/@outputname**

type	**xs:string**
annotation	documentation The operation result key that is output to STDOUT for SPLUNK indexing.Optional, some operations may not return

	values.
source	<xs:attribute name="outputname" type="xs:string"> <xs:annotation> <xs:documentation>The operation result key that is output to STDOUT for SPLUNK indexing.Optional, some operations may not return values.</xs:documentation> </xs:annotation> </xs:attribute>

element **param**

diagram	
properties	content complex
used by	element **formatter**

attributes

Name	Type	Use	Default	Fixed	Annotation
name	**xs:string**	required			documentation parameter name
value	**xs:string**	required			documentation parameter value

annotation	documentation parameters for a ParameterizedConfig object
source	<xs:element name="param"> <xs:annotation> <xs:documentation xml:lang="en">parameters for a ParameterizedConfig object</xs:documentation> </xs:annotation> <xs:complexType> <xs:attribute name="name" type="xs:string" use="required"> <xs:annotation> <xs:documentation xml:lang="en">parameter name</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="value" type="xs:string" use="required"> <xs:annotation> <xs:documentation xml:lang="en">parameter value</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType> </xs:element>

attribute **param/@name**

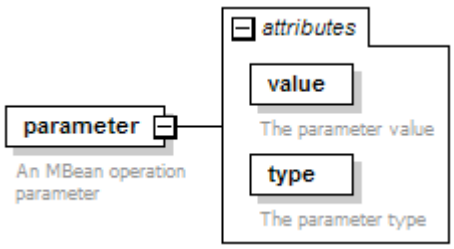
type	**xs:string**
properties	use required

annotation	documentation parameter name
source	<xs:attribute name="name" type="xs:string" use="required"> <xs:annotation> <xs:documentation xml:lang="en">parameter name</xs:documentation> </xs:annotation> </xs:attribute>

attribute **param/@value**

type	**xs:string**
properties	use required
annotation	documentation parameter value
source	<xs:attribute name="value" type="xs:string" use="required"> <xs:annotation> <xs:documentation xml:lang="en">parameter value</xs:documentation> </xs:annotation> </xs:attribute>

element **parameter**

diagram	
properties	content complex
used by	element **operation**

attributes	Name	Type	Use	Default	Fixed	Annotation
	value	**xs:string**	required			documentation The parameter value
	type	**derived by: xs:string**	required			documentation The parameter type

annotation	documentation An MBean operation parameter
source	<xs:element name="parameter"> <xs:annotation> <xs:documentation>An MBean operation parameter</xs:documentation> </xs:annotation> <xs:complexType> <xs:attribute name="value" type="xs:string" use="required"> <xs:annotation> <xs:documentation>The parameter value</xs:documentation> </xs:annotation> </xs:attribute> <xs:attribute name="type" use="required"> <xs:annotation>

```
      <xs:documentation>The parameter type</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="int"/>
        <xs:enumeration value="byte"/>
        <xs:enumeration value="short"/>
        <xs:enumeration value="long"/>
        <xs:enumeration value="float"/>
        <xs:enumeration value="double"/>
        <xs:enumeration value="boolean"/>
        <xs:enumeration value="char"/>
        <xs:enumeration value="string"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
</xs:element>
```

attribute **parameter/@value**

type	**xs:string**
properties	use required
annotation	documentation The parameter value
source	<xs:attribute name="value" type="xs:string" use="required"> <xs:annotation> <xs:documentation>The parameter value</xs:documentation> </xs:annotation> </xs:attribute>

attribute **parameter/@type**

type	restriction of **xs:string**
properties	use required
facets	Kind Value Annotation enumeration int enumeration byte enumeration short enumeration long enumeration float enumeration double enumeration boolean enumeration char enumeration string
annotation	documentation The parameter type
source	<xs:attribute name="type" use="required"> <xs:annotation> <xs:documentation>The parameter type</xs:documentation> </xs:annotation>

```
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="int"/>
      <xs:enumeration value="byte"/>
      <xs:enumeration value="short"/>
      <xs:enumeration value="long"/>
      <xs:enumeration value="float"/>
      <xs:enumeration value="double"/>
      <xs:enumeration value="boolean"/>
      <xs:enumeration value="char"/>
      <xs:enumeration value="string"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```