# Schema documentation for config.xsd

july 16, 2011

# **Table of Contents**

1
na(s)
Main schema config.xsd
ent(s)
Element jmxpoller
Element formatter
Element cluster
Element mbean
Element operation
Element parameter
Element attribute
Element jmxserver
oute(s)
Attribute formatter / @className
Attribute parameter / @value
Attribute parameter / @type
Attribute operation / @name
Attribute operation / @outputname
Attribute attribute / @name
Attribute attribute / @outputname1
Attribute mbean / @domain
Attribute mbean / @properties
Attribute mbean / @dumpAllAttributes
Attribute jmxserver / @host
Attribute jmxserver / @jmxpass
Attribute jmxserver / @jmxport
Attribute jmxserver / @jmxuser
Attribute jmxserver / @protocol
Attribute jmxserver / @lookupPath
Attribute jmxserver / @stubSource
Attribute jmxserver / @encodedStub
Attribute jmxserver / @jmxServiceURL
Attribute jmxserver / @jvmDescription
Attribute jmxserver / @pid
Attribute jmxserver / @pidFile
Attribute jmxserver / @pidCommand
Attribute cluster / @name
Attribute cluster / @description

# 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

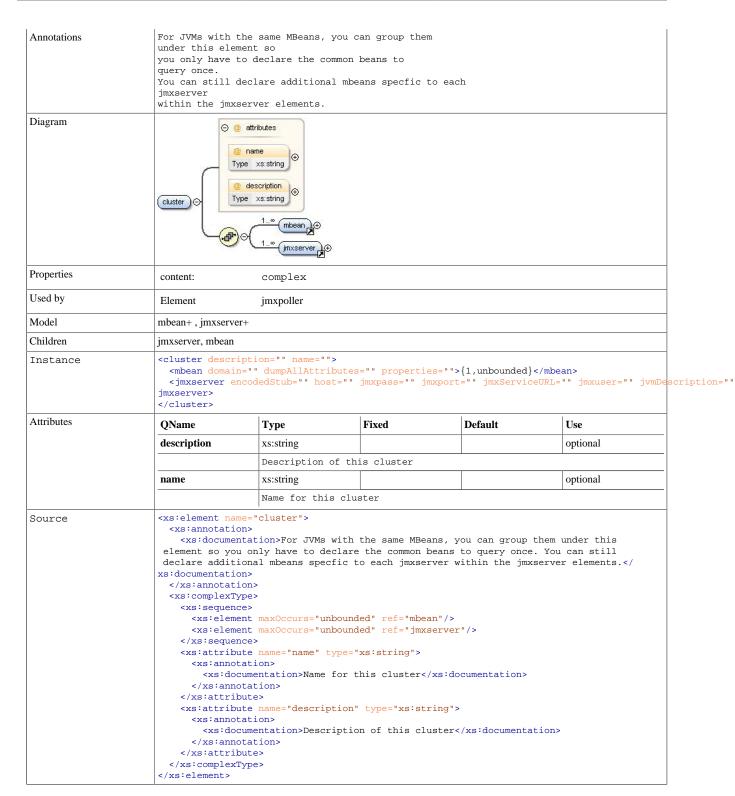
	to query. The result will then be written to STDOUT for SPLUNK indexing.	
Diagram	formatter    O * cluster   O * inxserver   O * inxserver   O * inxserver	
Properties	content: complex	
Model	formatter{0,1}, cluster*, jmxserver*	
Children	cluster, formatter, jmxserver	
Instance	<pre><jmxpoller>     <formatter classname="">{0,1}</formatter>     <cluster description="" name="">{0,unbounded}</cluster>     <jmxserver encodedstub="" host="" jmxpass="" jmxport="" jmxserver="" jmxserviceurl="" jmxuser="" jvmdescrip="">     </jmxserver></jmxpoller></pre>	ptior
Source	<pre><xs:element name="jmxpoller"></xs:element></pre>	

### **Element** formatter

Namespace	No namespace	No namespace					
Annotations		Custom formatter declaration allows you to override the default STDOUT format					
Diagram	formatter 🗇 🥝	formatter					
Properties	content:	complex					
Used by	Element	Element jmxpoller					
Attributes	QName	Туре	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	<pre><xs:annotation>   <xs:documental <="" <<="" <xs:attribute="" <xs:complextype:="" <xs:documental="" format<="" stdout="" td="" xs="" xs:annotation:=""><td colspan="4"><pre><xs:element name="formatter"></xs:element></pre></td></xs:documental></xs:annotation></pre>	<pre><xs:element name="formatter"></xs:element></pre>					

### Element cluster

Namespace	No namespace
-----------	--------------



#### 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	@ pro Type :	nain ⊕ xs:string				
Properties	content:	complex				
Used by	Elements	cluster, jmxserver				
Model	operation*, attribute*					
Children	attribute, operation					
Instance	<pre><operation name="&lt;/pre"></operation></pre>	dumpAllAttributes="" outputname="">{ "" outputname="">{	0,unbounded} <th>eration&gt;</th> <th></th>	eration>		
Attributes	QName	Type	Fixed	Default	Use	
	domain	xs:string			required	
		The MBean domain				
	dumpAllAttributes	xs:boolean			optional	
	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.					
	properties	xs:string			required	
	The MBean properties string in  "key=value,key2=value2" format					
Source <pre> <xs:element "prontation="" "required"="" 0"="" and="" butes"="" ccurs="unbounded ccurs=" ddcard<="" maxominoccurs="0&lt;/td&gt;&lt;td&gt;diffied for the " name="mbear&lt;/td&gt;&lt;td&gt;minOccurs=" properties="" strin="" t="" type="xs: true will dump a&lt;/td&gt;&lt;td&gt;omain" unbounded="" use="required" xs:docume="">  " ref="operation" ref="attribut" "xs:string"&gt; entation&gt;  ype="xs:string" g in "key=value"  boolean"&gt;  ll of the attri</xs:element></pre>		on"/> e"/> >> .,key2=value2" format </td				

### **Element** operation

Namespace	No namespace
Annotations	An MBean operation

Diagram		attributes  name ype xs:string  outputname ype xs:string  outputname ype xs:string	⊕			
Properties	content:	complex				
Used by	Element	mbean				
Model	parameter*					
Children	parameter					
Instance	<pre><operation <="" <parameter="" name:="" operation="" typ=""></operation></pre>		"> 0,unbounded} <td>meter&gt;</td> <td></td>	meter>		
Attributes	QName	Туре	Fixed	Default	Use	
	name	xs:string			required	
		The operation name.  For overloaded operations, the operation signature is inferred from the paramaters list.				
	outputname	xs:string			optional	
		The operation result key that is output to STDOUT for SPLUNK indexing.Optional, some operations may not return values.				
Source	<pre><xs:annotation <="" <xs:attribut="" <xs:complextyy="" <xs:document="" <xs:element="" <xs:sequence="" <xs<="" td="" xs:annotation="" xs:sequence=""><td colspan="3"><pre><xs:element name="operation"></xs:element></pre></td><td>ions, the operation o STDOUT for SPLUNK</td></xs:annotation></pre>	<pre><xs:element name="operation"></xs:element></pre>			ions, the operation o STDOUT for SPLUNK	

# Element parameter

Namespace	No namespace	No namespace				
Annotations	An MBean opera	tion parameter				
Diagram	parameter	parameter    a attributes  a value Type xs:string  type Type restriction of 'xs:string'				
Properties	content:	complex				
Used by	Element	operation				
Attributes	QName	Туре	Fixed	Default	Use	
	type	restriction of xs:string			required	
The parameter type						

	QName	Type	Fixed	Default	Use		
	value	xs:string			required		
		The parameter value					
Source	<pre><xs:annotati< td=""><td>me="parameter"&gt; on&gt; on&gt; ontation&gt;An MBean ope ion&gt; ype&gt; oute name="value" use otation&gt; totation&gt; bute&gt; oute name="type" use otation&gt; cumentation&gt;The para otation&gt; leType&gt; striction base="xs: enumeration value="tenumeration val</td><td>eration parameter  ="required" type  meter value  meter type  nt"/&gt; byte"/&gt; chort"/&gt; cloat"/&gt; cloable"/&gt; coolean"/&gt; char"/&gt;</td><td>e="xs:string"&gt; documentation&gt;</td><td>n&gt;</td></xs:annotati<></pre>	me="parameter"> on> on> ontation>An MBean ope ion> ype> oute name="value" use otation> totation> bute> oute name="type" use otation> cumentation>The para otation> leType> striction base="xs: enumeration value="tenumeration val	eration parameter  ="required" type  meter value  meter type  nt"/> byte"/> chort"/> cloat"/> cloable"/> coolean"/> char"/>	e="xs:string"> documentation>	n>		
	<pre><xs:enumeration value="string"></xs:enumeration></pre>						
	<td></td> <td></td> <td></td> <td></td>						

### **Element** attribute

Namespace	No namespace					
Annotations	An MBean attribute					
Diagram	attribute 🗇	attributes  name ype xs:string  outputname ype xs:string				
Properties	content:	complex				
Used by	Element	mbean				
Attributes	QName	Туре	Fixed	Default	Use	
	name	xs:string			required	
		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	
		The attribut SPLUNK index	e key that is outp	ut to STDOUT for		
Source	<pre><td>n&gt; utation&gt;An MBean on&gt; pe&gt; ute name="name" uation&gt; umentation&gt;The atabular attribut</td><td>attributeuse="required" type attribute name For tes , then you can . ie: foo:goo:myatt</td><td>e="xs:string"&gt; attributes that ar use a ":" delimite</td><td>ed notation for</td></pre>	n> utation>An MBean on> pe> ute name="name" uation> umentation>The atabular attribut	attributeuse="required" type attribute name For tes , then you can . ie: foo:goo:myatt	e="xs:string"> attributes that ar use a ":" delimite	ed notation for	

### Element jmxserver

Namespace	No namespace	No namespace				
Annotations	A local or remot	A local or remote JMX Server to connect to				
Diagram	inxserver (a)	@ attributes  @ host Type   xs:string    @ jmxpass Type   xs:string    @ jmxport Type   xs:integer     @ jmxuser Type   xs:string     @ protocol Type   restriction of 'xs:string'     @ lookupPath Type   xs:string     @ stubSource Type   restriction of 'xs:string'     @ stubSource Type   restriction of 'xs:string'     @ oncodedStub Type   xs:string     @ jmxServiceURL Type   xs:string     @ jwmDescription Type   xs:string     @ pidFile   Type   xs:string     @ pidCommand     @ pidCommand     Type   xs:string     @ pidCommand     #### Type   Type				
Properties Used by	content:	complex				
	Elements mbean*	cluster, jmxpoller				
Model Children						_
Instance	<pre><jmxserver encod<="" pre=""></jmxserver></pre>	<pre>mbean  <jmxserver encodedstub="" host="" jmxpass="" jmxport="" jmxserviceurl="" jmxuser="" jvmdescrip<="" td=""></jmxserver></pre>				
Attributes	QName	Туре	Fixed	Default	Use	
	encodedStub	xs:string			optional	—
			 tub value for stub	Source types of		
l .	1	Dabcos carossis.	.uu varac ron	JOULLE CIPCE .	TOT arra Scare	1
	host	xs:string	_		optional	

	QName	Туре	Fixed	Default	Use	
	jmxServiceURL	xs:string			optional	
		A raw jmx service	e url in format "se rmi:///jndi/rmi:// precedence over o	myhost:9909/jm	xrmi	
	jmxpass	xs:string			optional	
	jmxport	JMX Password xs:integer			optional	
		JMX Port				
	jmxuser	xs:string  JMX Username			optional	
	jvmDescription	xs:string			optional	
		A description of	this JVM			
	lookupPath	xs:string			optional	
	pid	The url path of t	the service.		optional	
			taching directly	to a locally		
	pidCommand	xs:string			optional	
			nat outputs to STD ly to a locally ru		s ID for	
	pidFile	xs:string			optional	
		directly to a loc	the Process ID for cally running JVM.' e first line of the	The only file	contents should	
	protocol	restriction of xs:string	5		optional	
		Note: the "local MX4J to be used	as the JMX	an" and "burla	". p" protocols require ss of the connection	
	stubSource	restriction of xs:string	e remote stub.Will	1.6.2	optional	
ource	<pre><xs:element <xs:annotatior<="" name="" th=""><th>n&gt; cation&gt;A local or re on&gt; oe&gt;</th><th>mote JMX Server to</th><th>connect to<th>s:documentation&gt;</th></th></xs:element></pre>	n> cation>A local or re on> oe>	mote JMX Server to	connect to <th>s:documentation&gt;</th>	s:documentation>	
	<pre></pre>	nt minOccurs="0" max ce> ce name="host" type= ation> umentation>IP Addres	"xs:string">			
	<xs:annota <xs:docu <td>e name="jmxpass" ty ation&gt; mentation&gt;JMX Passw ation&gt;</td><td>_</td><td>ion&gt;</td><td></td></xs:docu </xs:annota 	e name="jmxpass" ty ation> mentation>JMX Passw ation>	_	ion>		
	<pre><xs:annote< td=""><td colspan="5"><pre></pre></td></xs:annote<></pre>	<pre></pre>				
	<pre><xs:annota< td=""><td>ation&gt; amentation&gt;JMX Usern ation&gt;</td><td></td><td>ion&gt;</td><td></td></xs:annota<></pre>	ation> amentation>JMX Usern ation>		ion>		
	<pre><xs:annota "local",="" "soap'<="" <xs:docu="" pre=""></xs:annota></pre>	ation> mentation>The servi ', "hessian" and "bu	rlap" protocols re	quire MX4J to	to "rmi". Note : the be used as the JMX ion.	
	<td>cation&gt;</td> <td></td> <td></td> <td></td>	cation>				

```
<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:documentation>The url path of the service.</xs:documentation>
    </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: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>
        \verb|\color| < xs: \verb|\documentation| > \verb|\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>
      </r></r></r>
    </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
Туре	xs:string

Properties	use:	required
Used by	Element	formatter
Source	<pre><xs:annotation>   <xs:documentat< pre=""></xs:documentat<></xs:annotation></pre>	e="className" type="xs:string" use="required"> cion>Fully qualified Java class name of the formatter implementation, om.dtdsoftware.splunk.formatter.Formatter interface /xs:documentation

### Attribute parameter / @value

Namespace	No namespace
Annotations	The parameter value
Туре	xs:string
Properties	use: required
Used by	Element parameter
Source	<pre><xs:attribute name="value" type="xs:string" use="required">     <xs:annotation>         <xs:documentation>The parameter value</xs:documentation>         </xs:annotation>         </xs:attribute></pre>

# Attribute parameter / @type

Namespace	No namespace		
Annotations	The parameter type		
Туре	restriction of xs:string		
Properties	use:	required	
Facets	enumeration	int	
	enumeration	byte	
	enumeration	short	
	enumeration	long	
	enumeration	float	
	enumeration	double	
	enumeration	boolean	
	enumeration	char	
	enumeration	string	
Used by	Element	parameter	
Source	<pre><xs:annotation>   <xs:documentat <="" xs:annotation="">   <xs:simpletype>   <xs:restrictic <xs:enumerat="" <xs:enumerat<="" pre=""></xs:restrictic></xs:simpletype></xs:documentat></xs:annotation></pre>	<pre>ion base="xs:string"&gt; ion value="int"/&gt; ion value="byte"/&gt; ion value="short"/&gt; ion value="long"/&gt; ion value="float"/&gt; ion value="double"/&gt; ion value="double"/&gt; ion value="boolean"/&gt; ion value="char"/&gt; ion value="string"/&gt; ion value="string"/&gt; ion value="string"/&gt; ion value="string"/&gt; ion value="string"/&gt; ion value="string"/&gt; ion</pre>	

### Attribute operation / @name

Namespace	No namespace
Annotations	The operation name.

	For overloaded operations, the operation signature is inferred from the paramaters list.
Туре	xs:string
Properties	use: required
Used by	Element operation
Source	<pre><xs:attribute name="name" type="xs:string" use="required">     <xs:annotation></xs:annotation></xs:attribute></pre>

### Attribute operation / @outputname

Namespace	No namespace
Annotations	The operation result key that is output to STDOUT for SPLUNK indexing.Optional, some operations may not return values.
Туре	xs:string
Properties	content: simple
Used by	Element operation
Source	<pre><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></pre>

### 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	<pre><xs:attribute name="name" type="xs:string" use="required">     <xs:annotation></xs:annotation></xs:attribute></pre>

# Attribute attribute / @outputname

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

### Attribute mbean / @domain

Namespace	No namespace	
Annotations	The MBean domain	
Туре	xs:string	
Properties	use: required	
Used by	Element mbean	
Source	<pre><xs:attribute name="domain" type="xs:string" use="required">   <xs:annotation>     <xs:documentation>The MBean domain</xs:documentation>     </xs:annotation> </xs:attribute></pre>	

### Attribute mbean / @properties

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

### Attribute mbean / @dumpAllAttributes

Namespace	No namespace	
Annotations	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.	
Туре	xs:boolean	
Properties	content: simple	
Used by	Element mbean	
Source	<pre><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:documentation></xs:annotation>     </xs:attribute></pre>	

### Attribute jmxserver / @host

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

### Attribute jmxserver / @jmxpass

	No namespace
--	--------------

Annotations	JMX Password	
Туре	xs:string	
Properties	content: simp	le
Used by	Element jmxse	erver
Source	<pre><xs:attribute name="jmxpass" type="xs:string">     <xs:annotation></xs:annotation></xs:attribute></pre>	

# Attribute jmxserver / @jmxport

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

### Attribute jmxserver / @jmxuser

Namespace	No namespace	
Annotations	JMX Username	
Туре	xs:string	
Properties	content: simple	
Used by	Element jmxserver	
Source	<pre><xs:attribute name="jmxuser" type="xs:string">   <xs:annotation>    <xs:documentation>JMX Username</xs:documentation>    </xs:annotation>    </xs:attribute></pre>	

# Attribute jmxserver / @protocol

Namespace	No namespace	No namespace	
Annotations	Note: the "loo	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.	
Type	restriction of xs:str	ing	
Properties	content:	simple	
Facets	enumeration	rmi	
	enumeration	iiop	
	enumeration	local	
	enumeration	soap	
	enumeration	hessian	
	enumeration	burlap	
	enumeration	soap+ssl	
	enumeration	hessian+ssl	
	enumeration	burlap+ssl	
Used by	Element	jmxserver	
Source		<pre><xs:attribute name="protocol">   <xs:annotation></xs:annotation></xs:attribute></pre>	

```
<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

Namespace	No namespace	
Annotations	The url path of the service.	
Туре	xs:string	
Properties	content: simple	
Used by	Element jmxserver	
Source	<pre><xs:attribute name="lookupPath" type="xs:string">     <xs:annotation></xs:annotation></xs:attribute></pre>	

#### Attribute jmxserver / @stubSource

Namespace	No namespace	
Annotations	The source of the remote stub.Will default to "jndi"	
Туре	restriction of xs:string	
Properties	content: simple	
Facets	enumeration jndi	
	enumeration stub	
	enumeration ior	
Used by	Element jmxserver	
Source	<pre><xs:attribute name="stubSource">     <xs:annotation></xs:annotation></xs:attribute></pre>	

#### Attribute jmxserver / @encodedStub

Namespace	No namespace	
Annotations	Base64 encoded stub value for stubSource types of "ior" and "stub"	
Туре	xs:string	
Properties	content: simple	
Used by	Element jmxserver	
Source	<pre><xs:attribute name="encodedStub" type="xs:string"></xs:attribute></pre>	

### Attribute jmxserver / @jmxServiceURL

Namespace	No namespace
Annotations	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
Туре	xs:string
Properties	content: simple
Used by	Element jmxserver
Source	<pre><xs:attribute name="jmxServiceURL" type="xs:string"></xs:attribute></pre>

### Attribute jmxserver / @jvmDescription

Namespace	No namespace
Annotations	A description of this JVM
Туре	xs:string
Properties	content: simple
Used by	Element jmxserver
Source	<pre><xs:attribute name="jvmDescription" type="xs:string">   <xs:annotation>    <xs:documentation>A description of this JVM</xs:documentation>   </xs:annotation> </xs:attribute></pre>

#### Attribute jmxserver / @pid

Namespace	No namespace
Annotations	Process ID for attaching directly to a locally running JVM
Туре	xs:integer
Properties	content: simple
Used by	Element jmxserver
Source	<pre><xs:attribute name="pid" type="xs:integer"></xs:attribute></pre>

### 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.
Туре	xs:string
Properties	content: simple
Used by	Element jmxserver
Source	<pre><xs:attribute name="pidFile" type="xs:string">   <xs:annotation></xs:annotation></xs:attribute></pre>

### Attribute jmxserver / @pidCommand

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

#### Attribute cluster / @name

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

### Attribute cluster / @description

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