

JBossServer

A JBoss application server

Table of contents

1 Overview.....	3
2 Design.....	3
3 Constraints.....	3
3.1 Allowed Child Dependencies.....	3
3.2 Allowed Parent Dependencies.....	4
3.3 Allowed Property Values.....	4
4 Attributes.....	4
4.1 Exported Attributes.....	4
4.2 Defaults for Imported Attributes.....	4
5 Commands.....	5
5.1 Docs-Generate.....	5
5.2 Docs-Register.....	7
5.3 Docs-Baseline.....	8
5.4 Docs-Verify.....	8
5.5 Configure.....	8
5.6 killService.....	9
5.7 packages-code-install.....	9
5.8 packages-config-install.....	9
5.9 Start.....	10
5.10 assertServiceIsDown.....	10
5.11 assertServiceIsUp.....	10
5.12 startServiceWrapper.....	11
5.13 startService.....	11

5.14 waitforStartEvent.....	12
5.15 Stop.....	12
5.16 stopServiceWrapper.....	12
5.17 stopService.....	13
5.18 waitforStopEvent.....	13
6 Related Types.....	14
6.1 JBossSetting.....	14
6.2 JBossExtractEar.....	14
6.3 JBossJavaHome.....	15
6.4 JBossJavaOpts.....	16
6.5 JBossPortConfig.....	16
6.6 JBBindAddr.....	17
6.7 JBConfiguration.....	18
6.8 JBInpPort.....	18
6.9 JBJrmpRmiObjectPort.....	19
6.10 JBNamingRmiPort.....	19
6.11 JBPooledServerBindPort.....	20
6.12 JBTcAjpPort.....	21
6.13 JBTcHttpPort.....	21
6.14 JBUIL2ServerBindPort.....	22
6.15 JBWSPort.....	22
6.16 JBossDocumentRoot.....	23
6.17 JVMMinHeap.....	23
6.18 JVMMaxHeap.....	24
6.19 JVMPermSize.....	24
6.20 JVMMaxPermSize.....	25
6.21 AppTimezone.....	26
6.22 JBossDefaultInstance.....	26

1. Overview

JBossServer: A JBoss application server

This type manages an instance of a JBoss Server.

A [JavaZip](#) object should be added as a child dependency to automate deployment of the jboss package distribution.

An example of managing JBoss server instances is here: [JBoss Library](#).

See: [JBoss Community](#).

2. Design

Super Type Service

Role	Concrete. (Objects can be created.)
Instance Names	Unique
Notification	false
Template Directory	<code>\${modules.dir}/JBossServer/templates</code>
Data View	Children, proximity: 1
Logger Name	JBossServer

3. Constraints

3.1. Allowed Child Dependencies

- [JBossEar](#) 1
- [JBossSetting](#)
- [JBossZip](#) 1
- [Rdb](#)

Support datasource generation within the jboss server instance's deploy directory.

1: These types have a *Singleton* constraint. Only one instance may be added as a resource.

Restrict JBossServer resources to JBossZip platform package, JBossEar packages, arbitrary jboss settings, and Rdb objects.

3.2. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

In support of windows java service capability.

- Node
- Site

Restrict JBossServer to Node, Site, and optionally a JavaServiceWrapper. for java/windows service support.

3.3. Allowed Property Values

Property	Allowed Values	Default	Enforced
deployment-basedir	• <code>\${entity.attri</code>	• <code>\${entity.attri</code>	false
deployment-install-root	• <code>\${user.home}/jb</code>	• <code>\${user.home}/jb</code>	false

4. Attributes

4.1. Exported Attributes

Name	Property	Description
jboss_basedir	deployment-basedir	associate the jboss_basedir attribute property with the standard deployment-basedir property.
jboss_install_root	deployment-install-root	associate the jboss_install_root attribute property with the standard deployment-install-root property.

4.2. Defaults for Imported Attributes

Name	Default	Description
documentRoot	<code>\${entity.attribute.jboss_basedir}/c</code>	identify the DocumentRoot based on the integrated jboss tomcat webapp for potential deployable static content.

java_home	\${env.JAVA_HOME}	set a reasonable JAVA_HOME by using the user's environment.
javaOpts		optional JAVA_OPTS to be passed to the java runtime at startup
jboss_extract_ear	false	set ear extraction behavior to false within the jboss instance deploy directory.
jbossDefaultInstance	default	set the jboss reference instance to the "default" instance.
jvm_maxheap	-Xmx512m	use a small maximum heap setting in order not to break modest virtual machines, etc
jvm_minheap	-Xms256m	use a small minimum heap setting.
port_config	ports-default	use the default port configuration in the jboss service bindings configuration.
timezone	US/Pacific	set the jboss timezone to US Pacific time

5. Commands

Note:

Commandline options displayed in square brackets "[]" are optional. If an option expects arguments, then angle brackets are shown after the option "<>". Any default value is shown within the brackets.

5.1. Docs-Generate

creates directories required for jboss operation

Auto-provisions the jboss instance using the reference instance identified by \${opts.jbossDefaultInstance}.

Note:

NOTE: exception is if \${opts.basedir} and \${opts.jbossDefaultInstance} are the same directory, where auto provisioning does not occur.

Also generates run.conf which contains java runtime configuration.

Usage

```
Docs-Generate [-basedir <>] [-install_root <>] [-java_home
<>] [-javaopts <>] [-jboss_version <>]
[-jbossDefaultInstance <default>] [-jvm_max_perm_size <>]
[-jvm_maxheap <>] [-jvm_minheap <>] [-jvm_perm_size <>]
[-overwrite <false>] [-port_config <>] [-timezone <>]
```

Options

Option	Description
basedir	<i>base directory of the jboss instance</i> Defaults to the value of the jboss_basedir attribute setting .
install_root	<i>the JBoss Installation root, i.e. \$JBOSS_HOME</i> Defaults to the value of the jboss_install_root attribute setting.
java_home	<i>JAVA_HOME</i> JBoss JAVA_HOME which defaults to the java_home attribute setting value
javaopts	<i>the java options</i> Arbitrary JAVA_OPTS passed to the jboss runtime, defaults to the javaOpts attribute setting value.
jboss_version	<i>version of the jboss server package</i> Defaults to the value of the jboss_version attribute setting and is used to validate versioned templates embedded within this module. See JBossZip regarding the jboss_version attribute.
jbossDefaultInstance	<i>specifies the jboss reference instance to copy from, if not specified the default server is assumed within the JBOSS_HOME/server directory</i> jboss default reference instance defaults to the jbossDefaultInstance attribute setting value
jvm_max_perm_size	<i>jre max_perm_size setting</i> JBoss java max perm size setting, defaults to the jvm_max_perm_size attribute setting value

jvm_maxheap	<i>JAVA_HOME</i> JBoss java maximum heap setting, defaults to the jvm_maxheap attribute setting value
jvm_minheap	<i>jre minheap setting</i> JBoss java minimum heap setting, defaults to the jvm_minheap attribute setting value
jvm_perm_size	<i>jre perm_size setting</i> JBoss java perm size setting, defaults to the jvm_perm_size attribute setting value
overwrite	<i>overwrite the configuration if true</i> Typically the first time jboss instance is provisioned, one time copying is executed prior to template expansion. The one time copying can be overridden since the default for this option is set to false. If true, it would be the equivalent of re-provisioning the jboss server instance.
port_config	<i>jboss binding port configuration</i> identifies the port configuration from the port_config attribute setting value used in the JBoss Service Bindings configuration.
timezone	<i>the time zone</i> JBoss timezone passed in at java/jboss startup, defaults to the timezone attribute setting value.

5.2. Docs-Register

Experimental: registers configuration documents required for jboss operation

Experimental command still requires documentation.

Usage

```
Docs-Register [-basedir <>] [-install_root <>]
               [-jboss_version <>]
```

Options

Option	Description
basedir	<i>base directory of the jboss instance</i>

install_root	
jboss_version	<i>version of the jboss server package</i>

5.3. Docs-Baseline

Experimental: Creates a baseline for the configuration documents required for jboss operation

Experimental command still requires documentation.

Usage

Docs-Baseline [-basedir <>] [-install_root <>] [-overwrite]

Options

Option	Description
basedir	<i>base directory of the jboss instance</i>
install_root	<i>the jboss installation root directory</i>
overwrite	<i>Overwrite existing baseline copies</i>

5.4. Docs-Verify

Experimental: Verifies the configuration documents required for jboss operation

Experimental command still requires documentation.

Usage

Docs-Verify [-basedir <>] [-install_root <>]

Options

Option	Description
basedir	<i>base directory of the jboss instance</i>
install_root	<i>the jboss installation root directory</i>

5.5. Configure

Run the configuration cycle for the deployment.

override of standard Configure Workflow to execute Docs-Generate, which bootstraps the jboss configuration instance and Docs-Baseline, which executes the experimental Docs-Baseline command.

Usage

Configure

5.5.1. Workflow

1. [Docs-Generate](#)
2. [Docs-Baseline](#)

5.6. killService

kills the jboss server process

Last resort command to kill the specific jboss server instance java process. Implicitly executed via a stopServiceWrapper exception, or can be executed on demand.

Usage

```
killService [-basedir <>] [-install_root <>] [-killwait <2>]
```

Options

Option	Description
basedir	<i>base directory of the jboss instance</i>
install_root	<i>installation root of the jboss server</i>
killwait	<i>number of seconds to wait to check if PIDs still exist after kill</i>

5.7. packages-code-install

installs all the code package dependencies

Usage

```
packages-code-install [-packagetype  
<[^\.]*(:(!Configuration))[\^\.]*>]
```

Options

Option	Description
packagetype	<i>regular expression of packages to include</i>

5.8. packages-config-install

installs all the Configuration package dependencies

Usage

```
packages-config-install [-packagetype
<[^\.]*Configuration[^\.]*>]
```

Options

Option	Description
packagetype	<i>regular expression of packages to include</i>

5.9. Start

Conditionally starts the JBoss Server

standard Start command, which idempotently starts the jboss service

Usage
Start

5.9.1. Workflow

1. [assertServiceIsUp](#)

5.9.2. Error Handler

Command	startServiceWrapper
---------	-------------------------------------

5.10. assertServiceIsDown

Check whether JBoss is down

inquires if the jboss service is down and will fail if it is up

Usage
assertServiceIsDown [-basedir <>] [-install_root <>]
[-port_config <>]

Options

Option	Description
basedir	<i>base directory of the jboss instance</i>
install_root	
port_config	<i>jboss binding port configuration</i>

5.11. assertServiceIsUp

Check whether JBoss is up

inquires if the jboss service is up and will fail if it is down

Usage

```
assertServiceIsUp [-basedir <>] [-install_root <>]
                  [-port_config <>]
```

Options

Option	Description
basedir	<i>base directory of the jboss instance</i>
install_root	
port_config	<i>jboss binding port configuration</i>

5.12. startServiceWrapper

wraps the jboss start procedure and waits for start event

Workflow to asynchronously execute the jboss startService command and subsequently executes a synchronous waitForStartEvent in order to verify successful startup of the service.

Usage

```
startServiceWrapper
```

5.12.1. Workflow

1. [startService](#)
2. [waitForStartEvent](#)

5.13. startService

asynchronously executes the jboss start command

standard startService command to asynchronously start the jboss server instance.

Usage

```
startService [-basedir <>] [-install_root <>] [-java_home
              <>] [-spawn <true>]
```

Options

Option	Description
basedir	<i>base directory of the jboss instance</i>
install_root	

java_home	JAVA_HOME
spawn	Whether or not to run in background

5.14. waitForStartEvent

waits for a startup even of an asynchronously started jboss service

polls the jboss service and returns successfully upon successful startup. Failure will occur from a configurable timeout, defaults to fifteen minutes in the case for lengthy app startups.

Usage

```
waitForStartEvent [-basedir <>] [-maxwait <900>]
```

Options

Option	Description
basedir	base directory of the jboss instance
maxwait	maximum time in seconds to wait for start event

5.15. Stop

Conditionally stops the JBoss Server

standard Stop command to idempotently stop the jboss server instance.

Usage

```
Stop
```

5.15.1. Workflow

1. [assertServiceIsDown](#)

5.15.2. Error Handler

Command	stopServiceWrapper
---------	------------------------------------

5.16. stopServiceWrapper

wraps the jboss stop procedure and waits for stop event

workflow to execute the asynchronously execute the standard stopService command and subsequently executes a synchronous waitForStopEvent. Upon exception, the killService comand is executed.

Usage

stopServiceWrapper

5.16.1. Workflow

1. [stopService](#)
2. [waitForStopEvent](#)

5.16.2. Error Handler

Command	killService
---------	-----------------------------

5.17. stopService*executes the jboss stop command*

executes the standard stopService command to stop the jboss server instance.

Usage

```
stopService [-basedir <>] [-install_root <>] [-maxwait <120>] [-port_config <>]
```

Options

Option	Description
basedir	
install_root	
maxwait	<i>timeout in seconds for standard jboss shutdown to complete</i>
port_config	<i>jboss port configuration</i>

5.18. waitForStopEvent*waits for a stop event of a jboss service*

executes synchronous waitForStopEvent command to poll for a successful jboss shutdown.

Usage

```
waitForStopEvent [-basedir <>] [-maxwait <60>] [-port_config <>]
```

Options

Option	Description
--------	-------------

basedir	<i>base directory of jboss instance</i>
maxwait	<i>maximum time in seconds to wait for stop event</i> default timeout of 60 seconds will trigger failure of this command.
port_config	<i>jboss service bindings port config</i> look up port configuration in service bindings to issue stop request.

6. Related Types

The following types are defined for use with JBossServer.

6.1. JBossSetting

6.1.1. Overview

JBossSetting: *A JBoss Setting*

6.1.1.1. Intent

Supertype to encapsulate related subtypes to support JBossServer service.

6.1.2. Design

Super Type
Setting

Role	Abstract. (Objects cannot be created.)
Instance Names	Unique

6.1.3. Constraints

6.1.3.1. Allowed Parent Dependencies

- [JBossServer](#)

Restrict setting objects of this type to be referred to by a JBossServer object.

6.2. JBossExtractEar

6.2.1. Overview

JBossExtractEar: *modify ear deployment behavior for the JBossServer to explode or not explode the ear within the jboss deploy directory*

6.2.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.2.3. Attributes

6.2.3.1. Exported Attributes

Name	Property	Description
jboss_extract_ear	settingValue	"true" or "false" to respectively explode or not explode the ear

6.3. JBossJavaHome

6.3.1. Overview

JBossJavaHome: *JAVA_HOME for JBoss*

Identifies the location of the java (jdk or jre) installation root typically referred to as JAVA_HOME.

6.3.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.3.3. Attributes

6.3.3.1. Exported Attributes

Name	Property	Description
java_home	settingValue	attribute setting containing the fully qualified path to the java installation root (i.e. JAVA_HOME).

6.4. JBossJavaOpts

6.4.1. Overview

JBossJavaOpts: *JAVA_OPTS for JBoss*

To manage the allowed options (standard and non standard) that need to be passed to the java runtime when starting the jboss server.

6.4.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.4.3. Attributes

6.4.3.1. Exported Attributes

Name	Property	Description
javaOpts	settingValue	attribute setting containing the allowed java runtime options when starting jboss.

6.5. JBossPortConfig

6.5.1. Overview

JBossPortConfig: *JBoss service bindings port configuration*

6.5.1.1. Intent

Since JBoss has many different configuration files that refer to the various ports required to

run a jboss server instance, ControlTier automates the JBossServer port configuration via the [Service Binding Manager](#). This approach allows centralized management of the ports (which there are many of them) that conform to a standard port numbering convention. This unique set of ports is then indexed by a configuration name (i.e ports-default, ports-01, etc).

6.5.1.2. Discussion

For example, the "ports-default" configuration setting refers to the standard JBoss HTTP Listen port, 8080, HTTPS Listen Port, 8443, and the AJP Port, 8009, etc.

Another example, the "ports-01" configuration setting refers to JBoss HTTP Listen port, 8180, HTTPS Listen Port, 8543, and the AJP Port, 8109, etc.

Therefore, by accepting the conventions that JBoss encourages via the use of their Service Binding Manager, managing sets of ports through a single setting is achieved.

6.5.2. Design

Super Type
[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.5.3. Attributes

6.5.3.1. Exported Attributes

Name	Property	Description
port_config	settingValue	attribute containing the desired port configuration within the Service Binding Manager.

6.6. JBBindAddr

6.6.1. Overview

JBBindAddr: *JBoss Bind Address*

Deprecated, see #JBossPortConfig

6.6.2. Design

Super Type
[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.6.3. Attributes

6.6.3.1. Exported Attributes

Name	Property
bindAddr	settingValue

6.7. JBossConfiguration

6.7.1. Overview

JBossConfiguration: *The name of a JBoss Configuration*

6.7.2. Design

Super Type
[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.7.3. Attributes

6.7.3.1. Exported Attributes

Name	Property
jboss_configuration	settingValue

6.8. JBJnpPort

6.8.1. Overview

JBJnpPort: *The name of a JBoss Configuration*

Deprecated, see #JBossPortConfig

6.8.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.8.3. Attributes

6.8.3.1. Exported Attributes

Name	Property
jnpPort	settingValue

6.9. JBJrmpRmiObjectPort

6.9.1. Overview

JBJrmpRmiObjectPort: *JBoss JRMP RMI Object Port*

Deprecated, see #JBossPortConfig

6.9.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.9.3. Attributes

6.9.3.1. Exported Attributes

Name	Property
jboss_jrmp_rmi_object_port	settingValue

6.10. JBNamingRmiPort

6.10.1. Overview

JBNamingRmiPort: *JBoss Naming RMI Object Port*

Deprecated, see #JBossPortConfig

6.10.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.10.3. Attributes

6.10.3.1. Exported Attributes

Name	Property
jboss_naming_rmi_port	settingValue

6.11. JBPooledServerBindPort

6.11.1. Overview

JBPooledServerBindPort: *JBoss Pooled Server Bind Port*

Deprecated, see #JBossPortConfig

6.11.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.11.3. Attributes

6.11.3.1. Exported Attributes

Name	Property
------	----------

jboss_pooled_server_bind_port	settingValue
-------------------------------	--------------

6.12. JBTcAjpPort

6.12.1. Overview

JBTcAjpPort: *Embedded Tomcat/JBoss AJP Port*

Deprecated, see #JBossPortConfig

6.12.2. Design

Super Type
[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.12.3. Attributes

6.12.3.1. Exported Attributes

Name	Property
ajpPort	settingValue

6.13. JBTcHttpPort

6.13.1. Overview

JBTcHttpPort: *JBoss HTTP Port*

Deprecated, see #JBossPortConfig

6.13.2. Design

Super Type
[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.13.3. Attributes

6.13.3.1. Exported Attributes

Name	Property
httpPort	settingValue

6.14. JBUIL2ServerBindPort**6.14.1. Overview**

JBUIL2ServerBindPort: *JBoss UIL2 ServerBind Port*

Deprecated, see #JBossPortConfig

6.14.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.14.3. Attributes**6.14.3.1. Exported Attributes**

Name	Property
jboss_uil2_server_bind_port	settingValue

6.15. JBWSPort**6.15.1. Overview**

JBWSPort: *JBoss WebServices Port*

Deprecated, see #JBossPortConfig

6.15.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
------	--

Instance Names	Unique
----------------	--------

6.15.3. Attributes

6.15.3.1. Exported Attributes

Name	Property
jboss_ws_port	settingValue

6.16. JBossDocumentRoot

6.16.1. Overview

JBossDocumentRoot: *JBoss Document Root*

6.16.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.16.3. Attributes

6.16.3.1. Exported Attributes

Name	Property
documentRoot	settingValue

6.17. JVMMinHeap

6.17.1. Overview

JVMMinHeap: *JVM minimum heap size*

If set, the java runtime minimum heap, i.e. -Xms256m, will be applied at Configure time

6.17.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.17.3. Attributes

attribute containing the java runtime minimum heap parameter.

6.17.3.1. Exported Attributes

Name	Property
jvm_minheap	settingValue

6.18. JVMMaxHeap

6.18.1. Overview

JVMMaxHeap: *JVM maximum heap size*

If set, the java runtime maximum heap, i.e. -Xmx512m, will be applied at Configure time

6.18.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.18.3. Attributes

attribute containing the java runtime maximum heap parameter.

6.18.3.1. Exported Attributes

Name	Property
jvm_maxheap	settingValue

6.19. JVMPermSize

6.19.1. Overview

JVMPermSize: *JVM Perm Size memory setting*

If set, the java runtime "Perm Size" setting is applied, i.e. -XX:PermSize=256m, at Configure time

6.19.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.19.3. Attributes

attribute containing the java runtime permsize parameter

6.19.3.1. Exported Attributes

Name	Property
jvm_perm_size	settingValue

6.20. JVMMaxPermSize

6.20.1. Overview

JVMMaxPermSize: *JVM Max Perm Size memory setting*

If set, the java runtime "Max Perm Size" setting is applied, i.e. -XX:MaxPermSize=256m, at Configure time

6.20.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.20.3. Attributes

attribute containing the java runtime max permsize parameter

6.20.3.1. Exported Attributes

Name	Property
jvm_max_perm_size	settingValue

6.21. AppTimezone

6.21.1. Overview

AppTimezone: *A appserver timezone*

if set, applies the -Duser.timezone=<value> java runtime property at Configure time.

6.21.2. Design

Super Type

[JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.21.3. Attributes

attribute containing the java runtime user.timezone property value

6.21.3.1. Exported Attributes

Name	Property
timezone	settingValue

6.22. JBossDefaultInstance

6.22.1. Overview

JBossDefaultInstance: *configure new jboss instance from a reference instance*

The jboss instance directory, i.e. the JBossServer object's basedir, is configured from scratch based on a reference jboss instance. By default, this reference instance is the "default" instance residing under the JBOSS_HOME/server directory. At Deploy time, the Configure

workflow will auto generate this server instance from the reference instance using recursive copy and filtering techniques. This assumes that the JBossServer base directory is not the same as the reference instance, otherwise this will not occur since we cannot copy a directory onto itself and in this case the server instance will be untouched. If the jboss "default" instance is not the desired reference instance, this can be overridden by this setting object. For example, this settingValue can be another instance such as "all" or "minimal".

6.22.2. Design

Super Type [JBossSetting](#)

Role	Concrete. (Objects can be created.)
Instance Names	Unique

6.22.3. Attributes

attribute setting identifying the jboss reference instance to copy from when provisioning at Configure time.

6.22.3.1. Exported Attributes

Name	Property
jbossDefaultInstance	settingValue