

JavaServiceWrapper

Windows platform integration of Tanuki Software's Java Wrapper Service facility

Table of contents

1 Overview.....	2
2 Design.....	2
3 Constraints.....	2
4 Attributes.....	3
5 Commands.....	4
6 Related Types.....	4

1. Overview

JavaServiceWrapper: *Windows platform integration of Tanuki Software's Java Wrapper Service facility*

This type manages the Tanuki Software Java Wrapper Service, and allows a dependent Service object to be automatically deployed.

A [JavaServiceWrapperZip](#) object should be added as a child dependency.

A Service object should also be added as a child dependency.

Appropriate [Settings](#) should be added to JavaServiceWrapper object to configure the appropriate Java based configuration to run the underlying Service.

An example of using JavaServiceWrapper for managing the [JBossServer](#) type is here: [Windows service example](#).

See: [Tanuki Software Ltd. - Java Service Wrapper](#).

2. Design

Super Type Service

Role	Concrete. (Objects can be created.)
Instance Names	Unique
Notification	false
Template Directory	
Data View	Children, proximity: 1
Logger Name	

3. Constraints

3.1. Allowed Child Dependencies

- [JavaServiceWrapperAppParameters](#) 1
- [JavaServiceWrapperConsoleTitle](#) 1
- [JavaServiceWrapperJavaAdditional](#) 1
- [JavaServiceWrapperJavaClassPath](#) 1

- [JavaServiceWrapperJavaHome](#) 1
- [JavaServiceWrapperJavaInitMemory](#) 1
- [JavaServiceWrapperJavaMainClass](#) 1
- [JavaServiceWrapperJavaMaxMemory](#) 1
- [JavaServiceWrapperNtServiceDescription](#) 1
- [JavaServiceWrapperNtServiceDisplayName](#) 1
- [JavaServiceWrapperNtServiceInteractive](#) 1
- [JavaServiceWrapperNtServiceName](#) 1
- [JavaServiceWrapperNtServiceStartType](#) 1
- [JavaServiceWrapperZip](#) 1
- Service1

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

3.2. Allowed Parent Dependencies

- Node
- Site

4. Attributes

4.1. Exported Attributes

Name	Property	Description
installRoot	deployment-install-root	The deployment-install-root is exported as the "installRoot" attribute. It is used to default the -installroot option in many commands.

4.2. Defaults for Imported Attributes

Name	Default
javaServiceWrapperConsoleTitle	Test Wrapper Sample Application
javaServiceWrapperJavaHome	\${env.JAVA_HOME}
javaServiceWrapperJavaInitMemory	3
javaServiceWrapperJavaMainClass	org.tanukisoftware.wrapper.WrapperSimpleApp
javaServiceWrapperJavaMaxMemory	64

javaServiceWrapperNtServiceDescription	Test Wrapper Sample Application Description
javaServiceWrapperNtServiceDisplayName	Test Wrapper Sample Application
javaServiceWrapperNtServiceInteractive	false
javaServiceWrapperNtServiceName	testwrapper
javaServiceWrapperNtServiceStartType	AUTO_START

5. Commands

View the available commands in the [User Reference](#).

6. Related Types

The following types are defined for use with JavaServiceWrapper.

6.1. JavaServiceWrapperAppParameters

6.1.1. Overview

JavaServiceWrapperAppParameters: *Application commandline parameters for the Java Service Wrapper, space separated.*

The value should be a space separated list of parameters to pass to the main class defined with a [JavaServiceWrapperJavaMainClass](#) Setting object.

When the JavaServiceWrapperMainClass value is "org.tanukisoftwares.wrapper.WrapperSimpleApp", then the value of this setting should be the Main class name of the target Java application, followed by the commandline parameters separated by spaces.

For example:

```
org.myapp.Main -param value
```

See: [Java Service Wrapper - Integration Methods](#).

See: [wrapper.java.mainclass Property](#).

6.1.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.1.3. Constraints

6.1.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.1.4. Attributes

6.1.4.1. Exported Attributes

Name	Property
javaServiceWrapperAppParameters	settingValue

6.2. JavaServiceWrapperConsoleTitle

6.2.1. Overview

JavaServiceWrapperConsoleTitle: *Title to use when running the Java Service Wrapper as a console*

If configured to run in a console window, this will be the title given to the window.

See: [wrapper.console.title Property](#).

6.2.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.2.3. Constraints

6.2.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.2.4. Attributes

6.2.4.1. Exported Attributes

Name	Property
javaServiceWrapperConsoleTitle	settingValue

6.3. JavaServiceWrapperJavaAdditional

6.3.1. Overview

JavaServiceWrapperJavaAdditional: *A list of additional JAVA_OPTS for the Java Service Wrapper, space separated*

The values are split at the space characters and generated into the wrapper.conf file as separate numbered properties.

See: [wrapper.java.additional<n> Property](#).

6.3.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.3.3. Constraints

6.3.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.3.4. Attributes

6.3.4.1. Exported Attributes

Name	Property
javaServiceWrapperJavaAdditional	settingValue

6.4. JavaServiceWrapperJavaClassPath

6.4.1. Overview

JavaServiceWrapperJavaClassPath: *A list of additional CLASSPATH elements for the Java Service Wrapper, comma separated*

The comma-separated java class paths for this Setting will be added to a default set which includes the necessary jars for the Tanuki service wrapper software.

6.4.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.4.3. Constraints

6.4.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.4.4. Attributes

6.4.4.1. Exported Attributes

Name	Property
javaServiceWrapperJavaClassPath	settingValue

6.5. JavaServiceWrapperJavaHome

6.5.1. Overview

JavaServiceWrapperJavaHome: *Value of JAVA_HOME for the Java Service Wrapper*

6.5.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.5.3. Constraints

6.5.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.5.4. Attributes

6.5.4.1. Exported Attributes

Name	Property
javaServiceWrapperJavaHome	settingValue

6.6. JavaServiceWrapperJavaInitMemory

6.6.1. Overview

JavaServiceWrapperJavaInitMemory: *Initial Java heap size in MB for the Java Service Wrapper*

Used in the wrapper.conf configuration file.

See: [wrapper.java.initmemory Property](#).

6.6.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.6.3. Constraints

6.6.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.6.4. Attributes

6.6.4.1. Exported Attributes

Name	Property
javaServiceWrapperJavaInitMemory	settingValue

6.7. JavaServiceWrapperJavaMainClass

6.7.1. Overview

JavaServiceWrapperJavaMainClass: *Class implementing the Java Service Wrapper WrapperListener interface*

The main class to set with this Setting is most often one provided by the Tanuki software, such as "org.tanukisoftware.wrapper.WrapperSimpleApp", unless target Java software to manage has its own implementation.

See: [Java Service Wrapper - Integration Methods](#).

6.7.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.7.3. Constraints

6.7.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.7.4. Attributes

6.7.4.1. Exported Attributes

Name	Property
javaServiceWrapperJavaMainClass	settingValue

6.8. JavaServiceWrapperJavaMaxMemory

6.8.1. Overview

JavaServiceWrapperJavaMaxMemory: *Maximum Java heap size in MB for the Java Service Wrapper*

Used in the wrapper.conf configuration file.

See: [wrapper.java.maxmemory Property](#).

6.8.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.8.3. Constraints

6.8.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.8.4. Attributes

6.8.4.1. Exported Attributes

Name	Property
javaServiceWrapperJavaMaxMemory	settingValue

6.9. JavaServiceWrapperNtServiceAccount

6.9.1. Overview

JavaServiceWrapperNtServiceAccount: *Account to use with the Java Service Wrapper Windows service.*

Note:

The usage of this setting is not currently enabled in this implementation.

See: [wrapper.ntservice.account Property](#).

6.9.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.9.3. Constraints

6.9.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.9.4. Attributes

6.9.4.1. Exported Attributes

Name	Property
javaServiceWrapperNtServiceAccount	settingValue

6.10. JavaServiceWrapperNtServiceDescription

6.10.1. Overview

JavaServiceWrapperNtServiceDescription: *Description of the Java Service Wrapper Windows service*

See: [wrapper.ntservice.description Property](#).

6.10.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.10.3. Constraints

6.10.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.10.4. Attributes

6.10.4.1. Exported Attributes

Name	Property
javaServiceWrapperNtServiceDescriptio	settingValue

6.11. JavaServiceWrapperNtServiceDisplayName

6.11.1. Overview

JavaServiceWrapperNtServiceDisplayName: *Displayed name of the Java Service Wrapper Windows service*

See: [wrapper.ntservice.displayname Property](#).

6.11.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.11.3. Constraints

6.11.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.11.4. Attributes

6.11.4.1. Exported Attributes

Name	Property
javaServiceWrapperNtServiceDisplayName	settingValue

6.12. JavaServiceWrapperNtServiceInteractive

6.12.1. Overview

JavaServiceWrapperNtServiceInteractive: *Whether or not to allow the Java Service Wrapper Windows service to interact with the desktop. (true/false)*

See: [wrapper.ntservice.interactive Property](#).

6.12.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.12.3. Constraints

6.12.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.12.3.2. Allowed Property Values

Property	Allowed Values	Default	Enforced
settingValue	<ul style="list-style-type: none"> • false • true 	<ul style="list-style-type: none"> • true 	true

6.12.4. Attributes

6.12.4.1. Exported Attributes

Name	Property
javaServiceWrapperNtServiceInteractiv	settingValue

6.13. JavaServiceWrapperNtServiceName

6.13.1. Overview

JavaServiceWrapperNtServiceName: *Identifying Name of the Java Service Wrapper Windows service*

This is the identifier used by Windows, and should be unique among all Windows Service. Make sure that all JavaServiceWrapper objects have different values for this setting.

See: [wrapper.ntservice.name Property](#).

6.13.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.13.3. Constraints

6.13.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.13.4. Attributes

6.13.4.1. Exported Attributes

Name	Property	Description
javaServiceWrapperNtServ	settingValue	This attribute is used by the lifecycle commands of JavaServiceWrapper to uniquely identify the Windows Service that is being managed.

6.14. JavaServiceWrapperNtServicePassword

6.14.1. Overview

JavaServiceWrapperNtServicePassword: *Password to use with the Java Service Wrapper Windows service.*

Note:

The usage of this setting is not currently enabled in this implementation.

See: [wrapper.ntservice.password Property](#).

6.14.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.14.3. Constraints

6.14.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.14.4. Attributes

6.14.4.1. Exported Attributes

Name	Property
javaServiceWrapperNtServicePassword	settingValue

6.15. JavaServiceWrapperNtServiceStartType

6.15.1. Overview

JavaServiceWrapperNtServiceStartType: *Mode in which the Java Service Wrapper Windows service is installed. AUTO_START or DEMAND_START.*

See: [wrapper.ntservice.starttype Property](#).

6.15.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.15.3. Constraints

6.15.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.15.3.2. Allowed Property Values

Property	Allowed Values	Default	Enforced
settingValue	<ul style="list-style-type: none"> • AUTO_START • DEMAND_START 	<ul style="list-style-type: none"> • AUTO_START 	true

6.15.4. Attributes

6.15.4.1. Exported Attributes

Name	Property
javaServiceWrapperNtServiceStartType	settingValue

6.16. JavaServiceWrapperSetting

6.16.1. Overview

JavaServiceWrapperSetting: *Java Service Wrapper configuration setting*

This abstract Setting subtype is the supertype for all JavaServiceWrapper Setting types.

6.16.2. Design

Super Type
Setting

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