

JavaServiceWrapper

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

Table of contents

1 Overview.....	3
2 Design.....	3
3 Constraints.....	3
3.1 Allowed Child Dependencies.....	3
3.2 Allowed Parent Dependencies.....	4
4 Attributes.....	4
4.1 Exported Attributes.....	4
4.2 Defaults for Imported Attributes.....	4
5 Commands.....	4
5.1 Deploy.....	5
5.2 Add.....	5
5.3 Remove.....	5
5.4 Docs-Generate.....	6
5.5 addService.....	6
5.6 removeService.....	6
5.7 assertServiceIsUp.....	7
5.8 assertServiceIsDown.....	7
5.9 assertServiceIsInstalled.....	7
5.10 assertServiceIsNotInstalled.....	8
5.11 startService.....	8
5.12 stopService.....	9
6 Related Types.....	9

6.1 JavaServiceWrapperSetting.....	9
6.2 JavaServiceWrapperJavaHome.....	9
6.3 JavaServiceWrapperJavaMainClass.....	10
6.4 JavaServiceWrapperJavaClassPath.....	11
6.5 JavaServiceWrapperJavaAdditional.....	11
6.6 JavaServiceWrapperJavaInitMemory.....	12
6.7 JavaServiceWrapperJavaMaxMemory.....	13
6.8 JavaServiceWrapperAppParameters.....	13
6.9 JavaServiceWrapperConsoleTitle.....	14
6.10 JavaServiceWrapperNtServiceName.....	15
6.11 JavaServiceWrapperNtServiceDisplayName.....	16
6.12 JavaServiceWrapperNtServiceDescription.....	16
6.13 JavaServiceWrapperNtServiceStartType.....	17
6.14 JavaServiceWrapperNtServiceInteractive.....	18
6.15 JavaServiceWrapperNtServiceAccount.....	18
6.16 JavaServiceWrapperNtServicePassword.....	19

1. Overview

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

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

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
installRoot	deployment-install-root

4.2. Defaults for Imported Attributes

Name	Default
javaServiceWrapperAppParameters	
javaServiceWrapperConsoleTitle	Test Wrapper Sample Application
javaServiceWrapperJavaAdditional	
javaServiceWrapperJavaClassPath	
javaServiceWrapperJavaHome	\${env.JAVA_HOME}
javaServiceWrapperJavaInitMemory	3
javaServiceWrapperJavaMainClass	org.tanukisoftware.wrapper.test.Main
javaServiceWrapperJavaMaxMemory	64
javaServiceWrapperNtServiceDescription	Test Wrapper Sample Application Description
javaServiceWrapperNtServiceDisplayName	Test Wrapper Sample Application
javaServiceWrapperNtServiceInteractive	false
javaServiceWrapperNtServiceName	testwrapper
javaServiceWrapperNtServiceStartType	AUTO_START

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. Deploy

Run the service deployment cycle, stopping, unconfiguring, installing package dependencies, configuring and the starting the service. .

Usage
Deploy

5.1.1. Workflow

1. Stop
2. [Remove](#)
3. Packages-Install
4. Configure
5. [Add](#)
6. Start

5.2. Add

Add the Java Service Wrapper Windows service if it is not already installed.

Usage
Add

5.2.1. Workflow

1. [assertServiceIsInstalled](#)

5.2.2. Error Handler

Command	addService
---------	----------------------------

5.3. Remove

Remove the Java Service Wrapper Windows service if it is installed.

Usage
Remove

5.3.1. Workflow

1. [assertServiceIsNotInstalled](#)

5.3.2. Error Handler

Command	removeService
---------	-------------------------------

5.4. Docs-Generate

Generate the server properties file

Usage

Docs-Generate [-installroot <>] [-packagebase <>]

Options

Option	Description
installroot	Java wrapper service wrapper home directory
packagebase	Java wrapper service wrapper package installation directory

5.5. addService

Add the Java Service Wrapper Windows service to the system

Usage

addService [-installroot <>] [-packagebase <>]

Options

Option	Description
installroot	Java wrapper service wrapper home directory
packagebase	Java wrapper service wrapper package installation directory

5.6. removeService

Remove the Java Service Wrapper Windows service from the system

Usage

removeService [-installroot <>] [-packagebase <>]

Options

Option	Description
installroot	Java wrapper service wrapper home directory

packagebase	<i>Java wrapper service wrapper package installation directory</i>
-------------	--

5.7. assertServiceIsUp

Check whether the Java Service Wrapper Windows service is up

Usage

```
assertServiceIsUp [-installroot <>] [-packagebase <>]
                  [-servicename <>]
```

Options

Option	Description
installroot	<i>Java wrapper service wrapper home directory</i>
packagebase	<i>Java wrapper service wrapper package installation directory</i>
servicename	<i>Java wrapper service wrapper Windows service name</i>

5.8. assertServiceIsDown

Check whether the Java Service Wrapper Windows service is down

Usage

```
assertServiceIsDown [-installroot <>] [-packagebase <>]
                    [-servicename <>]
```

Options

Option	Description
installroot	<i>Java wrapper service wrapper home directory</i>
packagebase	<i>Java wrapper service wrapper package installation directory</i>
servicename	<i>Java wrapper service wrapper Windows service name</i>

5.9. assertServiceIsInstalled

Check whether the Java Service Wrapper Windows service is installed

Usage

```
assertServiceIsInstalled [-installroot <>] [-packagebase <>]
[-servicename <>]
```

Options

Option	Description
installroot	<i>Java wrapper service wrapper home directory</i>
packagebase	<i>Java wrapper service wrapper package installation directory</i>
servicename	<i>Java wrapper service wrapper Windows service name</i>

5.10. assertServiceIsNotInstalled

Check whether the Java Service Wrapper Windows service is not installed

Usage

```
assertServiceIsNotInstalled [-installroot <>] [-packagebase
<>] [-servicename <>]
```

Options

Option	Description
installroot	<i>Java wrapper service wrapper home directory</i>
packagebase	<i>Java wrapper service wrapper package installation directory</i>
servicename	<i>Java wrapper service wrapper Windows service name</i>

5.11. startService

Start the Java Service Wrapper Windows service

Usage

```
startService [-installroot <>] [-packagebase <>]
[-servicename <>]
```

Options

Option	Description
installroot	<i>Java wrapper service wrapper home directory</i>
packagebase	<i>Java wrapper service wrapper package</i>

	<i>installation directory</i>
servicename	<i>Java wrapper service wrapper Windows service name</i>

5.12. stopService

Stop the Java Service Wrapper Windows service

Usage

```
stopService [-installroot <>] [-packagebase <>]
[-servicename <>]
```

Options

Option	Description
installroot	<i>Java wrapper service wrapper home directory</i>
packagebase	<i>Java wrapper service wrapper package installation directory</i>
servicename	<i>Java wrapper service wrapper Windows service name</i>

6. Related Types

The following types are defined for use with JavaServiceWrapper.

6.1. JavaServiceWrapperSetting

6.1.1. Overview

JavaServiceWrapperSetting: *Java Service Wrapper configuration setting*

6.1.2. Design

Super Type Setting

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

6.2. JavaServiceWrapperJavaHome

6.2.1. Overview

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

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
javaServiceWrapperJavaHome	settingValue

6.3. JavaServiceWrapperJavaMainClass

6.3.1. Overview

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

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
javaServiceWrapperJavaMainClass	settingValue

6.4. JavaServiceWrapperJavaClassPath**6.4.1. Overview**

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

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. JavaServiceWrapperJavaAdditional

6.5.1. Overview

JavaServiceWrapperJavaAdditional: *Additional parameters 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
javaServiceWrapperJavaAdditional	settingValue

6.6. JavaServiceWrapperJavaInitMemory

6.6.1. Overview

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

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. JavaServiceWrapperJavaMaxMemory**6.7.1. Overview**

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

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
javaServiceWrapperJavaMaxMemory	settingValue

6.8. JavaServiceWrapperAppParameters

6.8.1. Overview

JavaServiceWrapperAppParameters: *Application parameters for the Java Service Wrapper*

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
javaServiceWrapperAppParameters	settingValue

6.9. JavaServiceWrapperConsoleTitle

6.9.1. Overview

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

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
javaServiceWrapperConsoleTitle	settingValue

6.10. JavaServiceWrapperNtServiceName

6.10.1. Overview

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

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
javaServiceWrapperNtServiceName	settingValue

6.11. JavaServiceWrapperNtServiceDisplayName

6.11.1. Overview

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

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. JavaServiceWrapperNtServiceDescription

6.12.1. Overview

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

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](#)

6.12.4. Attributes

6.12.4.1. Exported Attributes

Name	Property
javaServiceWrapperNtServiceDescriptio	settingValue

6.13. JavaServiceWrapperNtServiceStartType

6.13.1. Overview

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

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
------	----------

javaServiceWrapperNtServiceStartType	settingValue
--------------------------------------	--------------

6.14. JavaServiceWrapperNtServiceInteractive

6.14.1. Overview

JavaServiceWrapperNtServiceInteractive: *Whether or not to allow the Java Service Wrapper Windows service to interact with the desktop.*

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
javaServiceWrapperNtServiceInteractiv	settingValue

6.15. JavaServiceWrapperNtServiceAccount

6.15.1. Overview

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

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](#)

6.15.4. Attributes

6.15.4.1. Exported Attributes

Name	Property
javaServiceWrapperNtServiceAccount	settingValue

6.16. JavaServiceWrapperNtServicePassword

6.16.1. Overview

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

6.16.2. Design

Super Type

[JavaServiceWrapperSetting](#)

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

6.16.3. Constraints

6.16.3.1. Allowed Parent Dependencies

- [JavaServiceWrapper](#)

6.16.4. Attributes

6.16.4.1. Exported Attributes

Name	Property
javaServiceWrapperNtServicePassword	settingValue