TomcatContext

Tomcat web application service (extended to support a custom context and an "external" property file)

Table of contents

1. Overview

TomcatContext: *Tomcat web application service (extended to support a custom context and an "external" property file)*

Use TomcatContext objects if your Tomcat instance requires more than one war file. Model the TomcatContext objects and the TomcatServer object with the same basedir. Attach all TomcatContext objects and the single TomcatServer object as child dependencies of TomcatSite. Then manage the service through the TomcatSite, deployed locally with the TomcatServer and TomcatContext objects. TomcatSites can be grouped into Sites in the same way as standalone TomcatServers.

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	TomcatContext

3. Constraints

3.1. Allowed Child Dependencies

- DefaultDeploymentType1
- DefaultPackageName1
- DefaultPackageType1
- DispatchChangeDependencies1
- TomcatContextFile 1
- TomcatImplicitContext 1
- TomcatProperty
- TomcatPropertyFile1
- TomcatRestartStrategy 1
- TomcatServer 1
- war1

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

3.2. Allowed Parent Dependencies

- Mediator
- Node

4. Attributes

4.1. Exported Attributes

Name	Property
`catalinaBase`	deployment-basedir
`catalinaHome`	deployment-install-root

4.2. Defaults for Imported Attributes

Name	Default
`contextFileName`	ROOT.xml
`defaultDeploymentType`	Deployment
`defaultPackageName`	^$
`defaultPackageType`	war
`dispatchChangeDependencies`	false
`tomcatImplicitContext`	false
`tomcatRestartStrategy`	none

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

generates all defined docs

Generate context file (if explicit context-- by default); generate property file if TomcatPropertyFile and TomcatProperty settings are set

Usage

```
Docs-Generate [-archivedir <${entity.instance.dir}/var>]
[-basedir <>]
```

Options

Option	Description
`archivedir`	*dir to store current output docs*
`basedir`	*tomcat instance dir*

5.2. Packages-Install

install package dependencies

Installs all package dependencies

Usage

```
Packages-Install [-buildstamp <>] [-packagetype <[^\.]*>]
[-restartStrategy <>]
```

Options

Option	Description
`buildstamp`	*build identifer. currently ignored*
`packagetype`	*type of packages to install*
`restartStrategy`	*determines the start strategy*

5.3. Configure

Configures the instance

Usage

```
Configure [-restartStrategy <>]
```

Options

Option	Description
`restartStrategy`	*determines the stop strategy*

5.4. War-Packages-Install

install war package dependencies

Install only war package dependencies

Usage

```
War-Packages-Install [-packagetype <war>]
```

5.4.1. Workflow

1. Packages-Install

Options

Option	Description
packagetype	*Platform packages to install*

5.5. stopService

executes the stop procedure

Usage

```
stopService [-restartStrategy <>]
```

Options

Option	Description
restartStrategy	*determines the stop strategy*

5.6. startService

executes the start procedure

Usage

```
startService [-restartStrategy <>]
```

Options

Option	Description
restartStrategy	*determines the start strategy*

5.7. assertServiceIsUp

asserts if service is up

Usage

```
assertServiceIsUp [-restartStrategy <>]
```

Options

Option	Description
`restartStrategy`	*determines the start strategy*

5.8. assertServiceIsDown

asserts if service is down

Usage

```
assertServiceIsDown [-restartStrategy <>]
```

Options

Option	Description
`restartStrategy`	*determines the stop strategy*

6. Related Types

The following types are defined for use with TomcatContext.

6.1. TomcatSetting

6.1.1. Overview

TomcatSetting: *Tomcat configuration setting*

6.1.2. Design

Super Type

Setting

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

6.2. TomcatContextFile

6.2.1. Overview

TomcatContextFile: *Tomcat application context path file name*

The file under conf/Catalina/localhost/ into which to write the context file. Default:

ROOT.xml

6.2.2. Design

Super Type

TomcatSetting

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

6.2.3. Constraints

6.2.3.1. Allowed Parent Dependencies

- TomcatContext
- TomcatServer

6.2.4. Attributes

6.2.4.1. Exported Attributes

Name	Property
`contextFileName`	settingValue

6.3. TomcatRestartStrategy

6.3.1. Overview

TomcatRestartStrategy: *Tomcat restart execution strategy*

Allowed values are none, context, and container. Default is none. Use *container* if you would like the server to be restarted when the TomcatContext is Deployed. For this case, attach the TomcatServer as a child object of TomcatContext.

6.3.2. Design

Super Type

TomcatSetting

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

6.3.3. Constraints

6.3.3.1. Allowed Parent Dependencies

- TomcatContext

6.3.4. Attributes

6.3.4.1. Exported Attributes

Name	Property
`tomcatRestartStrategy`	settingValue

6.4. TomcatImplicitContext

6.4.1. Overview

TomcatImplicitContext: *Tomcat Implicit Context (boolean)*

Default is false and a context file will be generated in conf/Catalina/localhost. If true, the context is an implicit context; i.e. the server already knows where it is and how to deploy it, and no context file is necessary. Usually this means that the application will be deployed in CATALINA_BASE/webapps. N.B. If you use an implicit context, the package type *must* be PackedWar or a subtype in order to avoid a race condition with the auto-unpacking functionality of Tomcat.

6.4.2. Design

Super Type
TomcatSetting

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

6.4.3. Constraints

6.4.3.1. Allowed Parent Dependencies

- TomcatContext

6.4.4. Attributes

6.4.4.1. Exported Attributes

Name	Property
`tomcatImplicitContext`	settingValue