

AtgDasServer

An ATG DAS application server

Table of contents

1 Overview.....	2
2 Design.....	2
3 Constraints.....	2
3.1 Allowed Child Dependencies.....	2
3.2 Allowed Parent Dependencies.....	2
3.3 Allowed Property Values.....	2
4 Attributes.....	3
4.1 Exported Attributes.....	3
4.2 Defaults for Imported Attributes.....	3
5 Commands.....	3
5.1 retryIsDownService.....	3
5.2 assertServiceIsUp.....	4
5.3 maybeKillService.....	5
5.4 invalidateCache.....	5
5.5 isNucleusRunning.....	5
5.6 runStopDynamo.....	6
5.7 mkdirStructure.....	6
5.8 stopService.....	7
5.9 startService.....	7
5.10 assertServiceIsDown.....	8
5.11 killService.....	8

1. Overview

[Open in Workbench](#)

AtgDasServer: *An ATG DAS application server*

2. Design

Super Type

[AtgAppServer](#)

Role	Abstract. (Objects cannot be created.)
Instance Names	Unique
Notification	false
Template Directory	<code>\${modules.dir}/AtgAppServer/templates</code>
Data View	Children, proximity: 1
Logger Name	AtgAppServer

3. Constraints

3.1. Allowed Child Dependencies

- AtgAppServername1
- AtgDirlist1
- AtgRmiPort1
- AtgSwitchingDataStore1

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

3.2. Allowed Parent Dependencies

- [AtgAppServerCollection](#)
- Node

3.3. Allowed Property Values

Property	Allowed Values	Default	Enforced
deployment-install-root	<ul style="list-style-type: none"> • /opt/Dynamo7/AT • /opt/Dynamo7/AT 		false

deployment-basedir	<ul style="list-style-type: none"> • /opt/Dynamo7/AT • /opt/Dynamo7/AT 	false
--------------------	--	-------

4. Attributes

4.1. Exported Attributes

Name	Property
dynamo_home	deployment-basedir
dynamo_root	deployment-install-root

4.2. Defaults for Imported Attributes

Name	Default
dirlist	data,j2ee/runtime,localconfig/atg,localconfig/atg/dynamo,logs/arch

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. retryIsDownService

checks status repeatedly up to status_retrymax sleeping status_waitinterval

Usage

```
retryIsDownService [-dynamo_home <>] [-dynamo_root <>]
[-name <>] [-status_retrymax <5>] [-status_waitinterval
<60>]
```

Execution	bash
Arguments	<pre> \${antdepo.home}/bin/ad-sh -M \${modules.dir}/AtgAppServer/lib/ad-sh -m init -c down -- -pidfile \${opts.dynamo_home}/servers/\${opts.name}/logs/dynamo.pid && { exit 0; } { cnt=1; while ["\$cnt" -le "\${opts.status_retrymax}"]; do echo "\${opts.name} not down. Will recheck in \${opts.status_waitinterval} seconds (retries: \$cnt</pre>

	<pre> of \${opts.status_retrymax}) ..."; sleep \${opts.status_waitinterval}; \${antdepo.home}/bin/ad-sh -M \${modules.dir}/AtgAppServer/lib/ad-sh -m init -c down -- -pidfile \${opts.dynamo_home}/servers/\${opts.name}/logs/dynamo.pid && { echo "dynamo DOWN"; exit 0; } { ["\$cnt" = "\${opts.status_retrymax}"] && { echo "\${opts.name} still not down after \$cnt retries. returning with exitcode 1"; exit 1; }; }; cnt=`expr \$cnt + 1`; done; exit 1; } </pre>
--	---

5.1.1. Options

Option	Description
dynamo_home	<i>the DYNAMO_HOME setting</i>
dynamo_root	<i>the DYNAMO_ROOT setting</i>
name	<i>the dynamo server name</i>
status_retrymax	<i>max times to retry check</i>
status_waitinterval	<i>seconds to wait before next check</i>

5.2. assertServiceIsUp

Checks if dynamo is running

Usage

```
assertServiceIsUp [-basedir <>] [-name <>]
```

Execution	bash
Arguments	<pre> \${antdepo.home}/bin/ad-sh -M \${modules.dir}/AtgAppServer/lib/ad-sh -m init -c up -- -pidfile \${opts.basedir}/logs/dynamo.pid </pre>

5.2.1. Options

Option	Description
basedir	<i>server base dir</i>
name	<i>server to check</i>

5.3. maybeKillService

calls retryIsDownService and if that fails calls killService

Usage

maybeKillService

5.3.1. Workflow

1. [retryIsDownService](#)

5.3.2. Error Handler

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

5.4. invalidateCache

tells server to invalidate its cache

Usage

invalidateCache [-rmiport <>]

5.4.1. Options

Option	Description
rmiport	<i>rmi control port</i>

5.5. isNucleusRunning

searches dynamo.log for "Nucleus running" text

Usage

isNucleusRunning [-name <>]

Execution	bash
Arguments	grep "Nucleus running" \${user.home}/atg/server/\${opts.name}/logs/dynamo.log { echo not running ; exit 1 ; }

5.5.1. Options

Option	Description
name	<i>atg instance name</i>

5.6. runStopDynamo

stop the service process

Usage

```
runStopDynamo [-dynamo_root <>] [-name <>]
```

Execution	bash
Arguments	<code>\${opts.dynamo_root}/home/bin/stopDynamo \${opts.name} -user admin -password admin</code>

5.6.1. Options

Option	Description
dynamo_root	<i>the DYNAMO_ROOT setting</i>
name	<i>the dynamo server name</i>

5.7. mkdirStructure

creates directories and makes necessary symlinks

Usage

```
mkdirStructure [-dirlist <>] [-dynamo_root <>] [-name <>]
```

Execution	bash
Arguments	<code>mkdir -p \${user.home}/atg/server/\${opts.name}; [-h \${opts.dynamo_root}/home/servers/\${opts.name}] { ln -s \${user.home}/atg/server/\${opts.name} \${opts.dynamo_root}/home/servers/\${opts.name}; }; IFS=, for subdir in \${opts.dirlist}; do mkdir -p \${user.home}/atg/server/\${opts.name}/\${subdir}; done</code>

5.7.1. Options

Option	Description
dirlist	<i>relative sub directory list</i>
dynamo_root	<i>dynamo install root</i>
name	<i>atg server name</i>

5.8. stopService

stops the dynamo instance

Usage

```
stopService [-dynamo_root <>] [-name <>]
```

5.8.1. Workflow

1. stopAcceptingNewSessions
2. [stopService](#)

5.8.2. Error Handler

Command	maybeKillService
---------	----------------------------------

5.8.3. Options

Option	Description
dynamo_root	<i>the dynamo install dir</i>
name	<i>server name</i>

5.9. startService

starts the dynamo instance

Usage

```
startService [-dynamo_root <>] [-java_home </usr/java>]
[-name <>]
```

Execution	bash
Arguments	<pre>export JAVA_HOME=\${opts.java_home}; \${antdepo.home}/bin/ad-sh -M \${modules.dir}/AtgAppServer/lib/ad-sh -m init -c nohupStart -- -x \${opts.dynamo_root}/home/bin/startDynamo \${opts.name}</pre>

5.9.1. Options

Option	Description
dynamo_root	<i>Dynamo install root</i>

java_home	<i>Defines JAVA_HOME environment variable</i>
name	<i>server name</i>

5.10. assertServiceIsDown

Checks if dynamo is down

Usage

```
assertServiceIsDown [-basedir <>] [-name <>]
```

Execution	bash
Arguments	<code>\${antdepo.home}/bin/ad-sh -M \${modules.dir}/AtgAppServer/lib/ad-sh -m init -c down -- -pidfile \${opts.basedir}/logs/dynamo.pid</code>

5.10.1. Options

Option	Description
basedir	<i>deployment basedir</i>
name	<i>server to check</i>

5.11. killService

kills the service process

Usage

```
killService [-dynamo_root <>] [-name <>]
```

Execution	bash
Arguments	<code>\${opts.dynamo_root}/home/bin/stopDynamo \${opts.name} -user admin -password admin -kill</code>

5.11.1. Options

Option	Description
dynamo_root	<i>the DYNAMO_ROOT setting</i>
name	<i>the dynamo server name</i>