Testsuite

Exported from JBoss Community Documentation Editor at 2017-06-19 14:19:29 EDT

Table of Contents

1 JBoss AS 7 Testsuite

Where to go next?

- WildFly Integration Testsuite User Guide if you changed JBoss AS 7 code and want to check for regressions.
- AS 7 Testsuite Harness Developer Guide to learn how the testsuite works (shell scripts, Ant scripts, pom.xml files)
- AS 7 Testsuite Test Developer Guide if you want to add a new test case to the testsuite (to increase code coverage or to reproduce a bug)

2 WildFly Testsuite Overview

This document will detail the implementation of the testsuite Integration submodule as it guides you on adding your own test cases.

The WildFly integration test suite has been designed with the following goals:

- support execution of all identified test case use cases
- employ a design/organization which is scalable and maintainable
- provide support for the automated measurement of test suite quality (generation of feature coverage reports, code coverage reports)

In addition, these requirements were considered:

- identifying distinct test case runs of the same test case with a different set of client side parameters and server side parameters
- separately maintaining server side execution results (e.g. logs, the original server configuration) for post-execution debugging
- running the testsuite in conjunction with a debugger
- the execution of a single test (for debugging purposes)
- running test cases against different container modes (managed in the main, but also remote and embedded)
- configuring client and server JVMs separately (e.g., IPv6 testing)

2.1 Test Suite Organization

The testsuite module has a small number of submodules:

- **benchmark** - holds all benchmark tests intended to assess relative performance of specific feature
- **domain** - holds all domain management tests
- **integration** - holds all integration tests
- **stress** - holds all stress tests

It is expected that test contributions fit into one of these categories.

The pom.xml file located in the testsuite module is inherited by all submodules and is used to do the following:

- set defaults for common testsuite system properties (which can then be overridden on the command line)
- define dependencies common to all tests (Arquillian, junit or testng, and container type)
- provide a workaround for @Resource(lookup=...) which requires libraries in jbossas/endorsed

It should not:

- define module-specific server configuration build steps
- define module-specific surefire executions

These elements should be defined in logical profiles associated with each logical grouping of tests; e.g., in the pom for the module which contains the tests. The submodule poms contain additional details of their function and purpose as well as expanded information as shown in this document.

2.2 Profiles

You should not activate the abovementioned profiles by -P, because that disables other profiles which are activated by default.

Instead, you should always use activating properties, which are in parenthesis in the lists below.

Testsuite profiles are used to group tests into logical groups.

- all-modules.module.profile (all-modules)
- integration.module.profile (integration.module)
- compat.module.profile (compat.module)
- domain.module.profile (domain.module)
- benchmark.module.profile (benchmark.module)
- stress.module.profile (stress.module)

They also pprepare WildFly instances and resources for respective testsuite submodules.

- jpda.profile - sets surefire.jpda.args (debug)
- ds.profile -sets database properties and prepares the datasource (ds=<db id>)
    - Has related database-specific profiles, like mysql51.profile etc.

Integration testsuite profiles configure surefire executions.

- smoke.integration.tests.profile
- basic.integration.tests.profile
- clustering.integration.tests.profile

2.3 Integration tests

2.3.1 Smoke -Dts.smoke, -Dts.noSmoke

Contains smoke tests.

Runs by default; use -Dts.noSmoke to prevent running.

Tests should execute quickly.

Divided into two Surefire executions:

- One with full profile
- Second with web profile (majority of tests).

2.3.2 Basic -Dts.basic

Basic integration tests - those which do not need special configuration like cluster.

Divided into three Surefire executions:

- One with full profile,
- Second with web profile (majority of tests).
- Third with web profile, but needs to be run after server restart to check whether persistent data are really persisted.

2.3.3 Cluster -Dts.clust

Sets up a cluster of two nodes.

2.3.4 IIOP -Dts.iiop

2.3.5 XTS -Dts.XTS

2.3.6 Multinode -Dts.multinode

3 WildFly Integration Testsuite User Guide

See also: WildFly Testsuite Test Developer Guide

Target Audience: Those interested in running the testsuite or a subset thereof, with various configuration options.

3.1 Running the testsuite

The tests can be run using:

- `build.sh` or `build.bat`, as a part of WildFly build.
  - By default, only smoke tests are run. To run all tests, run build.sh install -DallTests.
- `integration-tests.sh` or `integration-tests.bat`, a convenience script which uses bundled Maven (currently 3.0.3), and runs all parent testsuite modules (which configure the AS server).
- pure maven run, using `mvn install`.

The scripts are wrappers around Maven-based build. Their arguments are passed to Maven (with few exceptions described below). This means you can use:

- build.sh (defaults to install)
- build.sh install
- build.sh clean install
- integration-tests.sh install
- ...etc.

3.1.1 Supported Maven phases

Testsuite actions are bounds to various Maven phases up to `verify`. Running the build with earlier phases may fail in the submodules due to missed configuration steps. Therefore, the only Maven phases you may safely run, are:

- clean
- install
- site

The `test` phase is not recommended to be used for scripted jobs as we are planning to switch to the `failsafe` plugin bound to the `integration-test` and `verify` phases. See WFLY-625 and WFLY-228.

3.1.2 Testsuite structure

testsuite
- integration
  - smoke
  - basic
  - clust
  - iiop
  - multinode
  - xts
- compat
- domain
- mixed-domain
- stress
- benchmark

3.1.3 Test groups

To define groups of tests to be run, these properties are available:

- `-DallTests` - Runs all subgroups.
- `-DallInteg` - Runs all integration tests. Same as `cd testsuite/integration; mvn clean install -DallTests`
- `-Dts.integ` - Basic integration + clustering tests.
- `-Dts.clust` - Clustering tests.
- `-Dts.iiop` - IIOP tests.
- `-Dts.multinode` - Tests with many nodes.
- `-Dts.manualmode` - Tests with manual mode Arquillian containers.
- `-Dts.bench` - Benchmark tests.
- `-Dts.stress` - Stress tests.
- `-Dts.domain` - Domain mode tests.
- `-Dts.compat` - Compatibility tests.

3.2 Examples

- `integration-tests.sh [install]` -- Runs smoke tests.
- `integration-tests.sh clean install` -- Cleans the target directory, then runs smoke tests.
- `integration-tests.sh install -Dts.smoke` -- Same as above.
- `integration-tests.sh install -DallTests` -- Runs all testsuite tests.
- `integration-tests.sh install -Dts.stress` -- Runs smoke tests and stress tests.
- `integration-tests.sh install -Dts.stress -Dts.noSmoke` -- Runs stress tests only.

Pure maven - if you prefer not to use scripts, you may achieve the same result with:

- `mvn ... -rf testsuite`

The `-rf ...` parameter stands for "resume from" and causes Maven to run the specified module *and all successive*.

It's possible to run only a single module (provided the ancestor modules were already run to create the AS copies) :

- `mvn ... -pl testsuite/integration/cluster`

The `-pl ...` parameter stands for "project list" and causes Maven to run the specified module *only*.

3.2.1

3.2.2 Output to console

```
-DtestLogToFile
```

3.2.3 Other options

`-DnoWebProfile` - Run all tests with the *full* profile (`standalone-full.xml`). By default, most tests are run under *web* profile (`standalone.xml`).

`-Dts.skipTests` - Skip testsuite's tests. Defaults to the value of `-DskipTests`, which defaults to `false`. To build AS, skip unit tests and run testsuite, use `-DskipTests -Dts.skipTests=false`.

3.2.4 Timeouts

Surefire execution timeout

Unfortunatelly, no math can be done in Maven, so instead of applying a timeout ratio, you need to specify timeout manually for Surefire.

```
-Dsurefire.forked.process.timeout=900
```

test timeout ratios

Ratio in prercent - 100 = default, 200 = two times longer timeouts for given category.

Currently we have five different ratios. Later, it could be replaced with just one generic, one for database and one for deployment operations.

```
-Dtimeout.ratio.fsio=100
-Dtimeout.ratio.netio=100
-Dtimeout.ratio.memio=100
-Dtimeout.ratio.proc=100
-Dtimeout.ratio.db=100
```

3.2.5 Running a single test (or specified tests)

Single test is run using **-Dtest=...** . Examples:

- `./integration-tests.sh install -Dtest='*Clustered*' -Dintegration.module -Dts.clust`
- `./integration-tests.sh clean install -Dtest=org/jboss/as/test/integration/ejb/async/*TestCase.java -Dintegration.module -Dts.basic`
- `cd testsuite; mvn install -Dtest='*Clustered*' -Dts.basic  # No need for -Dintegration.module - integration module is active by default.`

The same shortcuts listed in "Test groups" may be used to activate the module and group profile.

Note that `-Dtest=` overrides `<includes>` and `<excludes>` defined in pom.xml, so do not rely on them when using wildcards - all compiled test classes matching the wildcard will be run.

Which Surefire execution is used?

Due to Surefire's design flaw, tests run multiple times if there are multiple surefire executions.
To prevent this, if `-Dtest=`... is specified, non-default executions are disabled, and standalone-full is used for all tests.
If you need it other way, you can overcome that need:

- `basic-integration-web.surefire with standalone.xml - Configure standalone.xml to be used as server config.`
- `basic-integration-non-web.surefire - For tests included here, technically nothing changes.`
- `basic-integration-2nd.surefire - Simply run the second test in another invocation of Maven.`

3.2.6 Running against existing AS copy (not the one from build/target/jboss-as-*)

-Djboss.dist=<path/to/jboss-as> will tell the testsuite to copy that AS into submodules to run the tests against.

For example, you might want to run the testsuite against AS located in `/opt/wildfly-8` :

```
./integration-tests.sh -DallTests -Djboss.dist=/opt/wildfly-8
```

The difference between jboss.dist and jboss.home:

jboss.dist is the location of the tested binaries. It gets copied to testsuite submodules.

jboss.home is internally used and points to those copied AS instances (for multinode tests, may be even different for each AS started by Arquillian).

Running against a running JBoss AS instance

Arquillian's WildFly 8 container adapter allows specifying `allowConnectingToRunningServer` in `arquillian.xml`, which makes it check whether AS is listening at `managementAddress:managementPort`, and if so, it uses that server instead of launching a new one, and doesn't shut it down at the end.

All arquillian.xml's in the testsuite specify this parameter. Thus, if you have a server already running, it will be re-used.

Running against JBoss Enterprise Application Platform (EAP) 6.0

To run the testsuite against AS included JBoss Enterprise Application Platform 6.x (EAP), special steps are needed.

Assuming you already have the sources available, and the distributed EAP maven repository unzipped in e.g. `/opt/jboss/eap6-maven-repo/` :

1) Configure maven in settings.xml to use only the EAP repository. This repo contains all artifacts necessary for building EAP, including maven plugins.
The build (unlike running testsuite) may be done offline.
The recommended way of configuring is to use special settings.xml, not your local one (typically in .m2/settings.xml).

```
   <mirror>
      <id>eap6-mirror-setting</id>
      <mirrorOf>
         *,!central-eap6,!central-eap6-plugins,!jboss-public-eap6,!jboss-public-eap6-plugins
      </mirrorOf>
      <name>Mirror Settings for EAP 6 build</name>
      <url>file:///opt/jboss/eap6-maven-repo</url>
    </mirror>
  </mirrors>
```

2) Build EAP. You won't use the resulting EAP build, though. The purpose is to get the artifacts which the testsuite depends on.

```
mvn clean install -s settings.xml -Dmaven.repo.local=local-repo-eap
```

3) Run the testsuite. Assuming that EAP is located in `/opt/eap6`, you would run:

```
./integration-tests.sh -DallTests -Djboss.dist=/opt/eap6
```

For further information on building EAP and running the testsuite against it, see the official EAP documentation (link to be added).

How-to for EAP QA can be found here (Red Hat internal only).

3.2.7 Running with a debugger

Argument	What will start with debugger	Default port	Port change arg.
-Ddebug	AS instances run by Arquillian	8787	-Ddebug.port.as=...
-Djpda	alias for -Ddebug		
-DdebugClient	Test JVMs (currently Surefire)	5050	-Ddebug.port.surefire=...
-DdebugCLI	AS CLI	5051	-Ddebug.port.cli=...

Examples

```
./integration-tests.sh install -DdebugClient -Ddebug.port.surefire=4040

...

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Listening for transport dt_socket at address: 4040
```

```
./integration-tests.sh install -DdebugClient -Ddebug.port.surefire

...

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Listening for transport dt_socket at address: 5050
```

```
./integration-tests.sh install -Ddebug
```

```
./integration-tests.sh install -Ddebug -Ddebug.port.as=5005
```

JBoss AS is started by Arquillian, when the first test which requires given instance is run. Unless you pass **-DtestLogToFile=false, there's (currently) no challenge text in the console**; it will look like the first test is stuck. This is being solved in http://jira.codehaus.org/browse/SUREFIRE-781.

Depending on which test group(s) you run, multiple AS instances may be started. In that case, you need to attach the debugger multiple times.

3.2.8 Running tests with custom database

To run with different database, specify the `-Dds` and use these properties (with the following defaults):

```
-Dds.jdbc.driver=
-Dds.jdbc.driver.version=
-Dds.jdbc.url=
-Dds.jdbc.user=test
-Dds.jdbc.pass=test
-Dds.jdbc.driver.jar=${ds.db}-jdbc-driver.jar
```

`driver` is JDBC driver class. JDBC `url`, `user` and `pass` is as expected.

`driver.version` is used for automated JDBC driver downloading. Users can set up internal Maven repository hosting JDBC drivers, with artifacts with

GAV = `jdbcdrivers:${ds.db}:${ds.jdbc.driver.version}`

Internally, JBoss has such repo at http://nexus.qa.jboss.com:8081/nexus/content/repositories/thirdparty/jdbcdrivers/ .

The `ds.db` value is set depending on ds. E.g. `-Dds=mssql2005` sets `ds.db=mssql` (since they have the same driver). `-Dds.db` may be overriden to use different driver.

~~In case you don't want to use such driver, set just -Dds.db= (empty) and provide the driver to the AS manually.~~
Not supported; work in progress on parameter to provide JDBC Driver jar.

Default values

For WildFly continuous integration, there are some predefined values for some of databases, which can be set using:

```
-Dds.db=<database-identifier>
```

Where database-identifier is one of: `h2`, `mysql51`

3.2.9 Running tests with IPv6

`-Dipv6` - Runs AS with `-Djava.net.preferIPv4Stack=false -Djava.net.preferIPv6Addresses=true`

and the following defaults, overridable by respective parameter:

Parameter	IPv4 default	IPv6 default	
-Dnode0	127.0.0.1	::1	Single-node tests.
-Dnode1	127.0.0.1	::1	Two-node tests (e.g. cluster) use this for the 2nd node.
-Dmcast	230.0.0.4	ff01::1	ff01::1 is IPv6 Node-Local scope mcast addr.
-Dmcast.jgroupsDiag	224.0.75.75	ff01::2	JGroups diagnostics multicast address.
-Dmcast.modcluster	224.0.1.105	ff01::3	mod_cluster multicast address.

Values are set in AS configuration XML, replaced in resources (like ejb-jar.xml) and used in tests.

3.2.10 Running tests with security manager / custom security policy

`-Dsecurity.manager` - Run with default policy.

`-Dsecurity.policy=<path>` - Run with the given policy.

`-Dsecurity.manager.other=<set of Java properties>` - Run with the given properties. Whole set is included in all server startup parameters.

Example:

```
./integration-tests.sh clean install -Dintegration.module -DallTests \
\"-Dsecurity.manager.other=-Djava.security.manager \
-Djava.security.policy==$(pwd)/testsuite/shared/src/main/resources/secman/permitt_all.policy \
-Djava.security.debug=access:failure \"
```

Notice the \" quotes delimiting the whole `-Dsecurity.manager.other` property.

3.2.11 Creating test reports

Test reports are created in the form known from EAP 5. To create them, simply run the testsuite, which will create Surefire XML files.

Creation of the reports is bound to the `site` Maven phase, so it must be run separatedly afterwards. Use one of these:

```
./integration-tests.sh site
```

```
cd testsuite; mvn site
```

```
mvn -pl testsuite site
```

Note that it will take all test results under `testsuite/integration/` - the pattern is `**/*TestCase.xml` , without need to specify `-DallTests`.

3.2.12 Creating coverage reports

Jira: https://issues.jboss.org/browse/WFLY-585

Coverage reports are created by JaCoCo.

During the integration tests, Arquillian is passed a JVM argument which makes it run with JaCoCo agent, which records the executions into `${basedir}/target/jacoco` .

In the `site` phase, a HTML, XML and CSV reports are generated. That is done using `jacoco:report` Ant task in `maven-ant-plugin` since JaCoCo's maven report goal doesn't support getting classes outside target/classes.

Usage

```
./build.sh clean install -DskipTests
./integration-tests.sh clean install -DallTests -Dcoverage
./integration-tests.sh site -DallTests -Dcoverage ## Must run in separatedly.
```

Alternative:

```
mvn clean install -DskipTests
mvn -rf testsuite clean install -DallTests -Dcoverage
mvn -rf testsuite site -DallTests -Dcoverage
```

3.2.13 Cleaning the project

To have most stable build process, it should start with:

- clean target directories
- only central Maven repo configured
- clean local repository or at least:
  - free of artefacts to be built
  - free of dependencies to be used (especially snapshots)

To use , you may use these commands:

```
mvn clean install -DskipTests -DallTests  ## ...to clean all testsuite modules.
mvn dependency:purge-local-repository build-helper:remove-project-artifact
-Dbuildhelper.removeAll
```

In case the build happens in a shared environment (e.g. network disk), it's recommended to use local repository:

```
cp /home/hudson/.m2/settings.xml .
sed
"s|<settings>|<settings><localRepository>/home/ozizka/hudson-repos/$JOBNAME</localRepository>|"
-i settings.xml
```

Or:

```
mvn clean install ... -Dmaven.repo.local=localrepo
```

See also https://issues.jboss.org/browse/WFLY-628.

3.3 Troubleshooting Common Issues

3.3.1 Timeouts

May happen especially on slower computers. Try setting a different timeout (in seconds) :

-Dsurefire.forked.process.timeout=9999

3.3.2 "Server already running"

Known issue: JBPAPP-8368 Arquillian should wait until a port is free after AS JVM process ends to prevent "port in use".

~~Currently, the only solution is to re-run.~~ This has been fixed in 7.1.2, see pull request https://github.com/jbossas/jboss-as/pull/1999 .

3.3.3 Database failures

3.3.4 Build gets stuck at first test of a module

If you use NFS (Network file system), it might be a file locking issue.

Try running using a local disk.

4 WildFly Testsuite Harness Developer Guide

Audience: Whoever wants to change the testsuite harness

JIRA: WFLY-576

4.1 Testsuite requirements

http://community.jboss.org/wiki/ASTestsuiteRequirements will probably be merged here later.

4.2 Adding a new maven plugin

The plugin version needs to be specified in jboss-parent. See https://github.com/jboss/jboss-parent-pom/blob/master/pom.xml#L65 .

4.3 Shortened Maven run overview

See Shortened Maven Run Overview.

4.4 How the AS instance is built

See How the JBoss AS instance is built and configured for testsuite modules.

4.5 Properties and their propagation

Propagated to tests through arquillian.xml:

```
<property name="javaVmArguments">${server.jvm.args}</property>
```

TBD: https://issues.jboss.org/browse/ARQ-647

4.5.1 JBoss AS instance dir

integration/pom.xml

(currently nothing)

***-arquillian.xml**

```
<container qualifier="jboss" default="true">
    <configuration>
        <property name="jbossHome">${basedir}/target/jbossas</property>
```

4.5.2 Server JVM arguments

```
<surefire.memory.args>-Xmx512m -XX:MaxPermSize=256m</surefire.memory.args>
    <surefire.jpda.args></surefire.jpda.args>
    <surefire.system.args>${surefire.memory.args} ${surefire.jpda.args}</surefire.system.args>
```

4.5.3 IP settings

- `${ip.server.stack}` - used in `<systemPropertyVariables> / <server.jvm.args>` which is used in `*-arquillian.xml`.

4.5.4 Testsuite directories

- `${jbossas.ts.integ.dir}`
- `${jbossas.ts.dir}`
- `${jbossas.project.dir}`

4.5.5 Clustering properties

- node0
- node1

4.6 Debug parameters propagation

```
<surefire.jpda.args></surefire.jpda.args>       - default

<surefire.jpda.args>-Xrunjdwp:transport=dt_socket,address=${as.debug.port},server=y,suspend=y</sur
- activated by -Ddebug or -Djpda


testsuite/pom.xml:        <surefire.system.args>... ${surefire.jpda.args}
...</surefire.system.args>
testsuite/pom.xml:                        <jboss.options>${surefire.system.args}</jboss.options>

testsuite/integration/pom.xml:     <server.jvm.args>${surefire.system.args}
${jvm.args.ip.server} ${jvm.args.security} ${jvm.args.timeouts} -Dnode0=${node0} -Dnode1=


integration/pom.xml:
<server.jvm.args>${surefire.system.args} ${jvm.args.ip.server} ${jvm.args.security}
${jvm.args.timeouts} -Dnode0=${node0} -Dnode1=${node1} -DudpGroup=${udpGroup}
${jvm.args.dirs}</server.jvm.args>

arquillian.xml:
<property name="javaVmArguments">${server.jvm.args}
-Djboss.inst=${basedir}/target/jbossas</property>
```

4.7 How the JBoss AS instance is built and configured for testsuite modules.

Refer to Shortened Maven Run Overview to see the mentioned build steps.

1) AS instance is copied from `${jboss.dist}` to `testsuite/target/jbossas`.
   Defaults to AS which is built by the project (`build/target/jboss-as-*`).

2)

testsuite/pom.xml:

from ${jboss.home} to ${basedir}/target/jbossas
phase generate-test-resources: resource-plugin, goal copy-resources

testsuite/integration/pom.xml:

phase process-test-resources: antrun-plugin:

```
<ant antfile="$\{basedir}/src/test/scripts/basic-integration-build.xml">
    <target name="build-basic-integration"/>
    <target name="build-basic-integration-jts"/>
</ant>
```

Which invokes

```
<target name="build-basic-integration" description="Builds server configuration for
basic-integration tests">
      <build-server-config name="jbossas"/>
```

Which invokes

```
<!-- Copy the base distribution. -->
<!-- We exclude modules and bundles as they are read-only and we locate the via sys props. -->
<copy todir="@{output.dir}/@{name}">
    <fileset dir="@{jboss.dist}">
        <exclude name="**/modules/**"/>
        <exclude name="**/bundles/**"/>
    </fileset>
</copy>

<!-- overwrite with configs from test-configs and apply property filtering -->
<copy todir="@{output.dir}/@{name}" overwrite="true" failonerror="false">
    <fileset dir="@{test.configs.dir}/@{name}"/>
    <filterset begintoken="${" endtoken="}">
        <filter token="node0" value="${node0}"/>
        <filter token="node1" value="${node1}"/>
        <filter token="udpGroup" value="${udpGroup}"/>
        <filter-elements/>
    </filterset>
</copy>
```

4.7.1 Arquillian config file location

```
-Darquillian.xml=some-file-or-classpath-resource.xml
```

4.8 Plugin executions matrix

x - runs in this module
xx - runs in this and all successive modules
x! - runs but should not.

initialize									

	TS	integ	smoke	basic	clust	iiop	comp	domain	be
maven-help-plugin	xx	x	x	x	x	x	x	x	x
properties-maven-plugin:write-project-properties	x								
maven-antrun-plugin:1.6:run (banner)									
process-resources									
maven-resources-plugin:2.5:resources (default-resources)	xx								
maven-dependency-plugin:2.3:copy (copy-annotations-endorsed)	xx!								
compile									
maven-compiler-plugin:2.3.2:compile (default-compile)	xx								
generate-test-resources									
maven-resources-plugin:2.5:copy-resources (build-jbossas.server)	xx!								
Should be:	x								
maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas)		x							
maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas.groups)		x	x	x	?	?	?	!	
Should be:		xx	x	x	x	x	x	x	x
process-test-resources									
maven-resources-plugin:2.5:testResources (default-testResources)	xx								
maven-antrun-plugin:1.6:run (build-smoke.server)			x						
maven-antrun-plugin:1.6:run (prepare-jars-basic-integration.server)				x					
maven-antrun-plugin:1.6:run (build-clustering.server)					x	x!?			
test-compile									
maven-compiler-plugin:2.3.2:testCompile (default-testCompile)	xx								

xml-maven-plugin:1.0:transform (update-ip-addresses-jbossas.server)	x								
maven-antrun-plugin:1.6:run (build-jars)							x		
test									
maven-surefire-plugin:2.10:test (smoke-full.surefire)									
maven-surefire-plugin:2.10:test (smoke-web.surefire)									
maven-surefire-plugin:2.10:test (default-test)					x	x	x	x	
maven-surefire-plugin:2.10:test (basic-integration-default-full.surefire)				x					
maven-surefire-plugin:2.10:test (basic-integration-default-web.surefire)				x					
maven-surefire-plugin:2.10:test (basic-integration-2nd.surefire)				x					
maven-surefire-plugin:2.10:test (tests-clust-multi-node-unm...surefire)					x				
maven-surefire-plugin:2.10:test (tests-clustering-single-node.surefire)					x				
maven-surefire-plugin:2.10:test (tests-clustering-multi-node.surefire)					x				
maven-surefire-plugin:2.10:test (tests-iiop-multi-node.surefire)						x			
package									
maven-jar-plugin:2.3.1:jar (default-jar)	xx!								
maven-source-plugin:2.1.2:jar-no-fork (attach-sources)	x								
install									
maven-install-plugin:2.3.1:install (default-install)	xx!								
	TS	integ	smoke	basic	clust	iiop	comp	domain	be

4.9 Shortened Maven Run Overview

4.9.1 How to get it

```
./integration-tests.sh clean install -DallTests | tee TS.txt | testsuite/tools/runSummary.sh
```

4.9.2 How it's done

Run this script on the output of the AS7 testsuite run:

```
##  Cat the file or stdin if no args,
##  filter only interesting lines - plugin executions and modules separators,
##  plus Test runs summaries,
##  and remove the boring plugins like enforcer etc.

cat $1 \
 | egrep ' --- |Building| ---------|Tests run: | T E S T S' \
 | grep -v 'Time elapsed'
 | sed 's|Tests run:|               Tests run:|' \
 | grep -v maven-clean-plugin \
 | grep -v maven-enforcer-plugin \
 | grep -v buildnumber-maven-plugin \
 | grep -v maven-help-plugin \
 | grep -v properties-maven-plugin:.*:write-project-properties \
;
```

You'll get an overview of the run.

4.9.3 Example output with comments.

```
ondra@ondra-redhat: ~/work/AS7/ozizka-as7 $  ./integration-tests.sh clean install -DallTests |
tee TS.txt | testsuite/tools/runSummary.sh
[INFO] ------------------------------------------------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Aggregator 7.1.0.CR1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @ jboss-as-testsuite ---
              Copies org.jboss.spec.javax.annotation:jboss-annotations-api_1.1_spec to
${project.build.directory}/endorsed .
              Inherited - needed for compilation of all submodules.

[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @ jboss-as-testsuite
---
              Copies ${jboss.home} to target/jbossas .  TODO: Should be jboss.dist.
```

```
[INFO] --- xml-maven-plugin:1.0:transform (update-ip-addresses-jbossas.server) @
jboss-as-testsuite ---
              Changes IP addresses used in server config files -
              applies ${xslt.scripts.dir}/changeIPAddresses.xsl on
${basedir}/target/jbossas/standalone/configuration/standalone-*.xml
              Currently inherited, IMO should not be.

[INFO] --- maven-source-plugin:2.1.2:jar-no-fork (attach-sources) @ jboss-as-testsuite ---
              TODO: Remove

[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ jboss-as-testsuite ---

[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Integration Aggregator 7.1.0.CR1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-integration-agg ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-integration-agg ---
              TODO: Remove
[INFO] --- maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas) @
jboss-as-testsuite-integration-agg ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas.groups) @
jboss-as-testsuite-integration-agg ---
[INFO] --- xml-maven-plugin:1.0:transform (update-ip-addresses-jbossas.server) @
jboss-as-testsuite-integration-agg ---
              TODO: Remove
[INFO] --- maven-source-plugin:2.1.2:jar-no-fork (attach-sources) @
jboss-as-testsuite-integration-agg ---
              TODO: Remove
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @
jboss-as-testsuite-integration-agg ---
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss AS Test Suite: Integration - Smoke 7.1.0.CR1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @
jboss-as-testsuite-integration-smoke ---
              TODO: Remove
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas.groups) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- xml-maven-plugin:1.0:transform (update-ip-addresses-jbossas.server) @
jboss-as-testsuite-integration-smoke ---
              TODO: Remove

[INFO] --- maven-antrun-plugin:1.6:run (build-smoke.server) @
jboss-as-testsuite-integration-smoke ---
     [echo] Building AS instance "smoke" from /home/ondra/work/EAP/EAP6-DR9 to
/home/ondra/work/AS7/ozizka-as7/testsuite/integration/smoke/target
              TODO: Should be running one level above!
```

```
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-surefire-plugin:2.10:test (smoke-full.surefire) @
jboss-as-testsuite-integration-smoke ---
 T E S T S
                Tests run: 4, Failures: 0, Errors: 4, Skipped: 0
```

4.9.4 Example output, unchanged

```
ondra@lenovo:~/work/AS7/ozizka-git$ ./integration-tests.sh clean install -DallTests | tee TS.txt
| testsuite/tools/runSummary.sh
SSCmeetingWestfordJan     [copy] Warning:
/home/ondra/work/AS7/ozizka-git/testsuite/integration/src/test/resources/test-configs/smoke does
not exist.
     [copy] Warning:
/home/ondra/work/AS7/ozizka-git/testsuite/integration/src/test/resources/test-configs/clustering-u
does not exist.
     [copy] Warning:
/home/ondra/work/AS7/ozizka-git/testsuite/integration/src/test/resources/test-configs/clustering-u
does not exist.
     [copy] Warning:
/home/ondra/work/AS7/ozizka-git/testsuite/integration/src/test/resources/test-configs/iiop-client
does not exist.
     [copy] Warning:
/home/ondra/work/AS7/ozizka-git/testsuite/integration/src/test/resources/test-configs/iiop-server
does not exist.
[INFO] ------------------------------------------------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Aggregator 7.1.0.Final-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-antrun-plugin:1.6:run (banner) @ jboss-as-testsuite ---
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @ jboss-as-testsuite ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @ jboss-as-testsuite
---
[INFO] --- xml-maven-plugin:1.0:transform (update-ip-addresses-jbossas.server) @
jboss-as-testsuite ---
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ jboss-as-testsuite ---
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Integration 7.1.0.Final-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-integration-agg ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-integration-agg ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas) @
jboss-as-testsuite-integration-agg ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas.groups) @
jboss-as-testsuite-integration-agg ---
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @
jboss-as-testsuite-integration-agg ---
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Integration - Smoke 7.1.0.Final-SNAPSHOT
[INFO] ------------------------------------------------------------------------
```

```
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas.groups) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-antrun-plugin:1.6:run (build-smoke.server) @
jboss-as-testsuite-integration-smoke ---
     [echo] Building AS instance "smoke" from
/home/ondra/work/AS7/ozizka-git/testsuite/integration/smoke/../../../build/target/jboss-as-7.1.0.F
to /home/ondra/work/AS7/ozizka-git/testsuite/integration/smoke/target
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @
jboss-as-testsuite-integration-smoke ---
[INFO] --- maven-surefire-plugin:2.10:test (smoke-full.surefire) @
jboss-as-testsuite-integration-smoke ---
 T E S T S
               Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- maven-surefire-plugin:2.10:test (smoke-web.surefire) @
jboss-as-testsuite-integration-smoke ---
 T E S T S
               Tests run: 116, Failures: 0, Errors: 0, Skipped: 6
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ jboss-as-testsuite-integration-smoke ---
[INFO] Building jar:
/home/ondra/work/AS7/ozizka-git/testsuite/integration/smoke/target/jboss-as-testsuite-integration-
--- maven-install-plugin:2.3.1:install (default-install) @ jboss-as-testsuite-integration-smoke
---
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Integration - Basic 7.1.0.Final-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-integration-basic ---
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @
jboss-as-testsuite-integration-basic ---
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @
jboss-as-testsuite-integration-basic ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-integration-basic ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas.groups) @
jboss-as-testsuite-integration-basic ---
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @
jboss-as-testsuite-integration-basic ---
[INFO] --- maven-antrun-plugin:1.6:run (prepare-jars-basic-integration.server) @
jboss-as-testsuite-integration-basic ---
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @
jboss-as-testsuite-integration-basic ---
[INFO] --- maven-surefire-plugin:2.10:test (basic-integration-default-full.surefire) @
jboss-as-testsuite-integration-basic ---
 T E S T S
               Tests run: 323, Failures: 0, Errors: 4, Skipped: 30
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Integration - Clustering
7.1.0.Final-SNAPSHOT
```

```
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-integration-clust ---
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @
jboss-as-testsuite-integration-clust ---
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @
jboss-as-testsuite-integration-clust ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-integration-clust ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas.groups) @
jboss-as-testsuite-integration-clust ---
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @
jboss-as-testsuite-integration-clust ---
[INFO] --- maven-antrun-plugin:1.6:run (build-clustering.server) @
jboss-as-testsuite-integration-clust ---
     [echo] Building config clustering-udp-0
     [echo] Building AS instance "clustering-udp-0" from
/home/ondra/work/AS7/ozizka-git/testsuite/integration/clust/../../../build/target/jboss-as-7.1.0.F
to /home/ondra/work/AS7/ozizka-git/testsuite/integration/clust/target
     [echo] Building config clustering-udp-1
     [echo] Building AS instance "clustering-udp-1" from
/home/ondra/work/AS7/ozizka-git/testsuite/integration/clust/../../../build/target/jboss-as-7.1.0.F
to /home/ondra/work/AS7/ozizka-git/testsuite/integration/clust/target
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @
jboss-as-testsuite-integration-clust ---
[INFO] --- maven-surefire-plugin:2.10:test (tests-clustering-multi-node-unmanaged.surefire) @
jboss-as-testsuite-integration-clust ---
 T E S T S
               Tests run: 9, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- maven-surefire-plugin:2.10:test (tests-clustering-single-node.surefire) @
jboss-as-testsuite-integration-clust ---
 T E S T S
               Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- maven-surefire-plugin:2.10:test (tests-clustering-multi-node.surefire) @
jboss-as-testsuite-integration-clust ---
 T E S T S
               Tests run: 8, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ jboss-as-testsuite-integration-clust ---
[INFO] Building jar:
/home/ondra/work/AS7/ozizka-git/testsuite/integration/clust/target/jboss-as-testsuite-integration-
--- maven-install-plugin:2.3.1:install (default-install) @ jboss-as-testsuite-integration-clust
---
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Integration - IIOP 7.1.0.Final-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-integration-iiop ---
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @
jboss-as-testsuite-integration-iiop ---
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @
jboss-as-testsuite-integration-iiop ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-integration-iiop ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (ts.copy-jbossas.groups) @
jboss-as-testsuite-integration-iiop ---
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @
jboss-as-testsuite-integration-iiop ---
[INFO] --- maven-antrun-plugin:1.6:run (build-clustering.server) @
```

```
jboss-as-testsuite-integration-iiop ---
     [echo] Building config iiop-client
     [echo] Building AS instance "iiop-client" from
/home/ondra/work/AS7/ozizka-git/testsuite/integration/iiop/../../../build/target/jboss-as-7.1.0.Fi
to /home/ondra/work/AS7/ozizka-git/testsuite/integration/iiop/target
     [echo] Building config iiop-server
     [echo] Building AS instance "iiop-server" from
/home/ondra/work/AS7/ozizka-git/testsuite/integration/iiop/../../../build/target/jboss-as-7.1.0.Fi
to /home/ondra/work/AS7/ozizka-git/testsuite/integration/iiop/target
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @
jboss-as-testsuite-integration-iiop ---
[INFO] --- maven-surefire-plugin:2.10:test (tests-iiop-multi-node.surefire) @
jboss-as-testsuite-integration-iiop ---
 T E S T S
               Tests run: 12, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ jboss-as-testsuite-integration-iiop ---
[INFO] Building jar:
/home/ondra/work/AS7/ozizka-git/testsuite/integration/iiop/target/jboss-as-testsuite-integration-i
--- maven-install-plugin:2.3.1:install (default-install) @ jboss-as-testsuite-integration-iiop
---
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Compatibility Tests 7.1.0.Final-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-integration-compat ---
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @
jboss-as-testsuite-integration-compat ---
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @
jboss-as-testsuite-integration-compat ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-integration-compat ---
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @
jboss-as-testsuite-integration-compat ---
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @
jboss-as-testsuite-integration-compat ---
[INFO] --- maven-antrun-plugin:1.6:run (build-jars) @ jboss-as-testsuite-integration-compat ---
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @
jboss-as-testsuite-integration-compat ---
 T E S T S
               Tests run: 7, Failures: 0, Errors: 4, Skipped: 3
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Domain Mode Integration Tests
7.1.0.Final-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-integration-domain ---
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @
jboss-as-testsuite-integration-domain ---
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @
jboss-as-testsuite-integration-domain ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-integration-domain ---
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @
jboss-as-testsuite-integration-domain ---
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @
jboss-as-testsuite-integration-domain ---
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @
jboss-as-testsuite-integration-domain ---
```

```
 T E S T S
               Tests run: 89, Failures: 0, Errors: 0, Skipped: 4
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ jboss-as-testsuite-integration-domain ---
[INFO] Building jar:
/home/ondra/work/AS7/ozizka-git/testsuite/domain/target/jboss-as-testsuite-integration-domain-7.1.
--- maven-install-plugin:2.3.1:install (default-install) @ jboss-as-testsuite-integration-domain
---
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Benchmark Tests 7.1.0.Final-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-benchmark ---
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @
jboss-as-testsuite-benchmark ---
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ jboss-as-testsuite-benchmark
---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-benchmark ---
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @
jboss-as-testsuite-benchmark ---
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @
jboss-as-testsuite-benchmark ---
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @ jboss-as-testsuite-benchmark ---
 T E S T S
               Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ jboss-as-testsuite-benchmark ---
[INFO] Building jar:
/home/ondra/work/AS7/ozizka-git/testsuite/benchmark/target/jboss-as-testsuite-benchmark-7.1.0.Fina
--- maven-install-plugin:2.3.1:install (default-install) @ jboss-as-testsuite-benchmark ---
[INFO] ------------------------------------------------------------------------
[INFO] Building JBoss Application Server Test Suite: Stress Tests 7.1.0.Final-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO] --- maven-dependency-plugin:2.3:copy (copy-annotations-endorsed) @
jboss-as-testsuite-stress ---
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ jboss-as-testsuite-stress
---
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ jboss-as-testsuite-stress ---
[INFO] --- maven-resources-plugin:2.5:copy-resources (build-jbossas.server) @
jboss-as-testsuite-stress ---
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @
jboss-as-testsuite-stress ---
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @
jboss-as-testsuite-stress ---
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @ jboss-as-testsuite-stress ---
 T E S T S
               Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ jboss-as-testsuite-stress ---
[INFO] Building jar:
/home/ondra/work/AS7/ozizka-git/testsuite/stress/target/jboss-as-testsuite-stress-7.1.0.Final-SNAP
--- maven-install-plugin:2.3.1:install (default-install) @ jboss-as-testsuite-stress ---
[INFO] ------------------------------------------------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] ------------------------------------------------------------------------
```

5 WildFly Testsuite Test Developer Guide

See also: WildFly Integration Testsuite User Guide

5.1 requisites

Please be sure to read Pre-requisites - test quality standards and follow those guidelines.

5.2 Arquillian container configuration

See AS 7.1 managed container adapter refrence.

5.3 ManagementClient and ModelNode usage example

```
final ModelNode operation = new ModelNode();
operation.get(ModelDescriptionConstants.OP).set(ModelDescriptionConstants.READ_RESOURCE_OPERATION);
ModelNode result = managementClient.getControllerClient().execute(operation);
Assert.assertEquals(ModelDescriptionConstants.SUCCESS,
result.get(ModelDescriptionConstants.OUTCOME).asString());
```

ManagementClient can be obtained as described below.

5.4 Arquillian features available in tests

@ServerSetup

TBD

```
@ContainerResource private ManagementClient managementClient;
final ModelNode result = managementClient.getControllerClient().execute(operation);
```

TBD

```
@ArquillianResource private ManagementClient managementClient;
ModelControllerClient client = managementClient.getControllerClient();
```

```
@ArquillianResource ContainerController cc;

@Test
public void test() {
    cc.setup("test", ...properties..)
    cc.start("test")
}
```

```
<arquillian>
    <container qualifier="test" mode="manual" />
</arquillian>
```

```
// Targeted containers HTTP context.
@ArquillianResource URL url;
```

```
// Targeted containers HTTP context where servlet is located.
@ArquillianResource(SomeServlet.class) URL url;
```

```
// Targeted containers initial context.
@ArquillianResource InitialContext|Context context;
```

```
// The manual deployer.
@ArquillianResource Deployer deployer;
```

See Arquillian's Resource Injection docs for more info, https://github.com/arquillian/arquillian-examples for examples.

See also Arquillian Reference.

Note to @ServerSetup annotation: It works as expected only on non-manual containers. In case of manual mode containers it calls setup() method after each server start up which is right (or actually before deployment), but the tearDown() method is called only at AfterClass event, i.e. usually after your manual shutdown of the server. Which limits you on the ability to revert some configuration changes on the server and so on. I cloned the annotation and changed it to fit the manual mode, but it is still in my github branch :)

5.5 Properties available in tests

5.5.1 Directories

- jbosssa.project.dir - Project's root dir (where ./build.sh is).
- jbossas.ts.dir - Testsuite dir.
- jbossas.ts.integ.dir - Testsuite's integration module dir.
- jboss.dist - Path to AS distribution, either built (build/target/jboss-as-...) or user-provided via -Djboss.dist
- jboss.inst - (Arquillian in-container only) Path to the AS instance in which the test is running (until ARQ-650 is possibly done)

- ~~jboss.home~~ - Deprecated as it's name is unclear and confusing. Use jboss.dist or jboss.inst.

5.5.2 Networking

- node0
- node1
- 230.0.0.4

5.5.3 Time-related coefficients (ratios)

In case some of the following causes timeouts, you may prolong the timeouts by setting value >= 100:

100 = leave as is,
150 = 50 % longer, etc.

- timeout.ratio.gen - General ratio - can be used to adjust all timeouts.When this and specific are defined, both apply.

- timeout.ratio.fs- Filesystem IO
- timeout.ratio.net - Network IO
- timeout.ratio.mem - Memory IO
- timeout.ratio.cpu - Processor
- timeout.ratio.db - Database

Time ratios will soon be provided by `org.jboss.as.test.shared.time.TimeRatio.for*()` methods.

5.6 Negative tests

To test invalid deployment handling: @ShouldThrowException

Currently doesn't work due to WFLY-673.

optionally you might be able to catch it using the manual deployer

```
@Deployment(name = "X", managed = false) ...

@Test
public void shouldFail(@ArquillianResource Deployer deployer) throws Exception {
  try {
    deployer.deploy("X")
  }
  catch(Exception e) {
   // do something
  }
}
```

5.7 Clustering tests (WFLY-616)

You need to deploy the same thing twice, so two deployment methods that just return the same thing. And then you have tests that run against each.

```
@Deployment(name = "deplA", testable = false)
    @TargetsContainer("serverB")
    public static Archive<?> deployment()

    @Deployment(name = "deplB", testable = false)
    @TargetsContainer("serverA")
    public static Archive<?> deployment(){ ... }

    @Test
    @OperateOnDeployment("deplA")
    public void testA(){ ... }

    @Test
    @OperateOnDeployment("deplA")
    public void testA() {...}
```

5.8 How to get the tests to master

- First of all, **be sure to read the "Before you add a test" section**.
- **Fetch** the newest mater: `git fetch upstream  # Provided you have the jbossas/jbossas GitHub repo as a remote called 'upstream'.`
- **Rebase** your branch: git checkout WFLY-1234-your-branch; git rebase upstream/master
- **Run *whole* testsuite** (integration-tests -DallTests). You may use https://jenkins.mw.lab.eng.bos.redhat.com/hudson/job/wildfly-as-testsuite-RHEL-matrix-openJDK7/last .
    - If any tests fail and they do not fail in master, fix it and go back to the "Fetch" step.
- **Push** to a new branch in your GitHub repo: `git push origin WFLY-1234-new-XY-tests`
- **Create a pull-request** on GitHub. Go to your branch and click on "Pull Request".
    - If you have a jira, start the title with it, like - WFLY-1234 New tests for XYZ.
    - If you don't, write some apposite title. In the description, describe in detail what was done and why should it be merged. Keep in mind that the diff will be visible under your description.
- **Keep the branch rebased daily** until it's merged (see the Fetch step). If you don't, you're dramatically decreasing chance to get it merged.
- There's a mailing list, jbossas-pull-requests, which is notified of every pull-request.
- You might have someone with merge privileges to cooperate with you, so they know what you're doing, and expect your pull request.
- When your pull request is reviewed and merged, you'll be notified by mail from GitHub.
- You may also check if it was merged by the following: `git fetch upstream; git cherry <branch>  ## Or` git branch --contains{{<branch> - see}} `here`
- Your commits will appear in master. They will have the same hash as in your branch.
    - You are now safe to delete both your local and remote branches: `git branch -D WFLY-1234-your-branch; git push origin :WFLY-1234-your-branch`

5.9 How to Add a Test Case

(Please don't (re)move - this is a landing page from a Jira link.)

Thank you for finding time to contribute to WildFly 8 quality.
Covering corner cases found by community users with tests is very important to increase stability.
If you're providing a test case to support your bug report, it's very likely that your bug will be fixed much sooner.

5.9.1 1) Create a test case.

It's quite easy - a simple use case may even consist of one short .java file.

Check WildFly 8 test suite test cases for examples.

For more information, see WildFly Testsuite Test Developer Guide. Check the requirements for a test to be included in the testsuite.

Ask for help at WildFly 8 forum or at IRC - #wildfly @ FreeNode.

5.9.2 2) Push your test case to GitHub and create a pull request.

For information on how to create a GitHub account and push your code therein, see Hacking on WildFly.

If you're not into Git, send a diff file to JBoss forums, someone might pick it up.

5.9.3 3) Wait for the outcome.

Your test case will be reviewed and eventually added. It may take few days.

When something happens, you'll receive a notification e-mail.

5.10 Before you add a test

Every added test, whether ported or new should follow the same guidelines:

- **Verify the test belongs in WildFly 8**

AS6 has a lot of tests for things that are discontinued. For example the legacy JBoss Transaction Manager which was replaced by Arjuna. Also we had tests for SPIs that no longer exist. None of these things should be migrated.

- **Only add CORRECT and UNDERSTANDABLE tests**

If you don't understand what a test is doing (perhaps too complex), or it's going about things in a strange way that might not be correct, THEN DO NOT PORT IT. Instead we should have a simpler, understandable, and correct test. Write a new one, ping the author, or skip it altogether.

- **Do not add duplicate tests**

Always check that the test you are adding doesn't have coverage elsewhere (try using "git grep"). As mentioned above we have some overlap between 6 and 7. The 7 test version will likely be better.

- **Don't name tests after JIRAs**

A JIRA number is useless without an internet connection, and they are hard to read. If I get a test failure thats XDGR-843534578 I have to dig to figure out the context. It's perfectly fine though to link to a JIRA number in the comments of the test. Also the commit log is always available.

- **Tests should contain javadoc that explains what is being tested**

This is especially critical if the test is non-trivial

- **Prefer expanding an EXISTING test over a new test class**

If you are looking at migrating or creating a test with similar functionality to an exiting test, it is better to expand upon the existing one by adding more test methods, rather than creating a whole new test. In general each new test class adds at least 300ms to execution time, so as long as it makes sense it is better to add it to an existing test case.

- **Organize tests by subsystem**

Integration tests should be packaged in subpackages under the relevant subsystem (e.g org.jboss.as.test.integration.ejb.async). When a test impacts multiple subsystems this is a bit of a judgement call, but in general the tests should go into the package of the spec that defines the functionality (e.g. CDI based constructor injection into an EJB, even though this involves CDI and EJB, the CDI spec defines this behaviour)

- **Explain non-obvious spec behavior in comments**

The EE spec is full of odd requirements. If the test is covering behavior that is not obvious then please add something like "Verifies EE X.T.Z - The widget can't have a foobar if it is declared like blah"

- **Put integration test resources in the source directory of the test**

At the moment there is not real organization of these files. It makes sense for most apps to have this separation, however the testsuite is different. e.g. most apps will have a single deployment descriptor of a given type, for the testsuite will have hundreds, and maintaining mirroring package structures is error prone.
This also makes the tests easier to understand, as all the artifacts in the deployment are in one place, and that place tends to be small (only a handful of files).

- **Do not hard-code values likely to need configuration (URLs, ports, ...)**

URLs hardcoded to certain address (localhost) or port (like the default 8080 for web) prevent running the test against different address or with IPv6 adress.
Always use the configurable values provided by Arquillian or as a system property.
If you come across a value which is not configurable but you think it should be, file an WildFly 8 jira issue with component "Test suite".
See @ArquillianResourrce usage example.

- **Follow best committing practices**
  - Only do changes related to the topic of the jira/pull request.
  - Do not clutter your pull request with e.g. reformatting, fixing typos spotted along the way - do another pull request for such.
  - Prefer smaller changes in more pull request over one big pull request which are difficult to merge.
  - Keep the code consistent across commits - e.g. when renaming something, be sure to update all references to it.
  - Describe your commits properly as they will appear in master's linear history.
  - If you're working on a jira issue, include it's ID in the commit message(s).

- **Do not use blind timeouts**

Do not use Thread.sleep() without checking for the actual condition you need to be fulfilled.
You may use active waiting with a timeout, but prefer using timeouts of the API/SPI you test where available.

Make the timeouts configurable: For a group of similar test, use a configurable timeout value with a default if not set.

- **Provide messages in assert*() and fail() calls**

Definitely, it's better to see "File x/y/z.xml not found" instead of:

```
junit.framework.AssertionFailedError
     at junit.framework.Assert.fail(Assert.java:48) [arquillian-service:]
     at junit.framework.Assert.assertTrue(Assert.java:20) [arquillian-service:]
     at junit.framework.Assert.assertTrue(Assert.java:27) [arquillian-service:]
     at
org.jboss.as.test.smoke.embedded.parse.ParseAndMarshalModelsTestCase.getOriginalStandaloneXml(Pars
[bogus.jar:]
```

- **Provide configuration properties hints in exceptions**

If your test uses some configuration property and it fails possibly due to misconfiguration, note the property and it's value in the exception:

```
File jdbcJar = new File( System.getProperty("jbossas.ts.dir", "."),
       "integration/src/test/resources/mysql-connector-java-5.1.15.jar");
    if( !jdbcJar.exists() )
        throw new IllegalStateException("Can't find " + jdbcJar + " using $\{jbossas.ts.dir} ==
" + System.getProperty("jbossas.ts.dir") );
```

- **Clean up**

- - Close sockets, connections, file descriptors;
  - Don't put much data to static fields, or clean them in a finaly {...} block.
  - Don't alter AS config (unless you are absolutely sure that it will reload in a final {...} block or an @After* method)

- **Keep the tests configurable**

  Keep these things in properties, set them at the beginning of the test:

  - Timeouts
  - Paths
  - URLs
  - Numbers (of whatever)

They either will be or already are provided in form of system properties, or a simple testsuite until API (soon to come).

5.11 Shared Test Classes and Resources

5.11.1 Among Testsuite Modules

Use the `testsuite/shared` module.

Classes and resources in this module are available in all testsuite modules - i.e. in testsuite/* .

Only use it if necessary - don't put things "for future use" in there.

Don't split packages across modules. Make sure the java package is unique in the WildFly project.

Document your util classes (javadoc) so they can be easily found and reused! A generated list will be put here.

5.11.2 Between Components and Testsuite Modules

To share component's test classes with some module in testsuite, you don't need to split to submodules. You can create a jar with classifier using this:

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-jar-plugin</artifactId>
    <executions> <execution>
        <goals> <goal>test-jar</goal> </goals>
    </execution></executions>
</plugin>
```

This creates a jar with classifier "tests", so you can add it as dependency to a testsuite module:

```
<dependency>
   <groupId>org.jboss.as</groupId>
   <artifactId>jboss-as-clustering-common</artifactId>
   <classifier>tests</classifier>
   <version>${project.version}</version>
  <scope>test</scope>
</dependency>
```