LuntBuild - Build Automation and Management FAQ

LuntBuild - Build Automation and Management FAQ

Table of Contents

Chapter 1. LuntBuild FAQ

LuntBuild - Build Automation and Management FAQ

General FAQ

1.1.
How to configure several projects to use shared stuffs, for example shared jars?

1. Setup a project, for instance "common", and configure it to checkout your shared jars. And configure builder for this project to publish shared jars to the artifacts directory of this project.
2. Setup other projects, for instance: "projectA" and "projectB". These projects will use shared jars mentioned in step 1. When setting up these projects, you are not need to checkout shared jars.
3. Edit all schedules under projects created in step 2, and configure them to be dependent on related schedules under the "common" project. By doing this, Luntbuild will make sure the "common" project is up to date before build projectA and projectB.
4. Edit builders of projectA and projectB, pass in publishing directory of the shared jars as build properties:

```
sharedJarsDir=${system.project["common"].schedule[build.schedule.name].lastSuccessBuild.
```

5. In Ant script of projectA and projectB, access directory ${sharedJarsDir} to use your shared jars.

Of course for shared jars stuff, Maven(http://maven.apache.org) is a more suitable option. Luntbuild can work gracefully with Maven, such as instruct Maven to use Luntbuild generated version numbers, etc.

1.2.
Is it possible to control maximum number of concurrent builds on a particular Luntbuild server?

The property "Number of build threads" in the properties tab controls concurrent number of build threads used by Luntbuild.

1.3.
How do you define top location of your Ant target execution?

If you are using JNI or your application configuration depends on relative location of the files, you might have a problem running the application using Luntbuild. To resolve the issue, define environment variable with the top directory of your application, and use this variable to define location(s) in the configuration of your application. Then define the same environment variable in the appropriate Luntbuild builder.

1.4.
How do you create a new Project, Build, Schedule ...?

To create a new Project, Build, Schedule etc., look in upper right corner, just below the Tabs for an icon  . The icon is quite small and for first time user it is hard to find. But as soon as you get used to look for it in all the tabs/pages, the small icon makes for nice and compact user interface.

1.5.
My favorite VCS doesn't work with Luntbuild?

Make sure that external VCS application like cvs or p4 is in your default path when you are starting Luntbuild.

1.6.

How do I find pending builds?

To find pending builds, display builds tab and click on schedule. The schedule page will display, showing detailed information about that particular schedule, including pending build queue for that schedule.

1.7.

Can I include date and time in the build version?

Yes you can, for example, if you want the version be "foo-yyMMDD_HHMM", you can configure "next build version" property of a project as follows:

```
"foo-" + shortYear + numericMonth + dayOfMonth + "_" + hour + minute (For luntbuild-1.1.1)
```

or

```
foo-${shortYear}${numericMonth}${dayOfMonth}_${hour}${minute} (For latest luntbuild code i
```

Where "shortYear", "numericMonth", "dayOfMonth", "hour", "minute" are all OGNL expressions. OGNL expressions can be used in build version string, please refer to Luntbuild user manual.

1.8.

We are using Luntbuild for Continuous Integration and we set the continuous integration period to be 60 seconds. Sometimes for some reason my version control system goes down, and Luntbuild will send build failure notification every 60 seconds, which is a lot of emails. Is there any way to let Luntbuild send fail notification mail only once upon such failures? And send success notification email again when the version control system comes up.

You should consider using "notify when status changed" notification strategy to avoid multiple build failure emails in such a case. This notification strategy will only send out mail when the current build is successful and the last build has failed, or the current build has failed and the last build is successful.

1.9.

What is the initial username and password after Luntbuild installation?

The initial site admin username/password is luntbuild/luntbuild. It is stored at <tomcat>/webapps/luntbuild/WEB-INF/applicationContext.xml. The user manual explains this in more detail.

1.10.

I'm surprised to find that only recent VCS committers will get build result notifications. Is there any way to send build notification email to some designated person besides the developers?

By default Luntbuild only sends notifications to recent VCS committers. However you can configure the "notification users" property in projects/basic tab to add users to receive build result notifications.

1.11.

We are using Ant builder to run our build on Windows platform, and we set the "build success condition" to be "result==0". But Luntbuild still signals the build success while actually the Ant builder fails. What gives?

Luntbuild's Ant builder calls /path/to/ant.bat(provided by ant distribution) to perform build on Windows platform. The ant.bat will internally call java.exe to do its job. When there is a build error, java.exe will

fail. However, the error code will not get propagated to the caller of ant.bat(Luntbuild in this case). This is an known behavior of Windows batch file. To overcome that, you can set "build success condition" as:

```
result==0 and logContainsLine("BUILD SUCCESSFUL")
```

1.12.

I am currently using Luntbuild to kick off a nightly build process. After the build, I would like to have Luntbuild run one script which kills a few running services and then another script which restarts these services using the codebase I just built. The problem I run into is that the second script (let's call it start.sh) launches a bunch of java processes and then exits, but luntbuild won't continue with the build until those child processes terminate as well. I need to leave those processes up and running long after the build completes. Do I have any options for this?

Great response from Derek Chiles: [While I'm sure you have several options, I suggest the excellent Java Service Wrapper. [http://wrapper.sf.net/] You'll need to modify your code a little, but you'll get a clean, cross-platform way to start/stop/restart your services. On Unix, you'll get an init-style script, and on Windows you'll be able to integrate your app as a system service if you like.]

1.13.

Starting with version 1.3 Luntbuild supports anonymous access to view builds and other Luntbuild artifacts.

To disable anonymous login you will have to:

Edit <server>/webapps/luntbuild/luntbuild-login.jsp and remove line:

```
<a href="j_acegi_security_check.do?j_username=anonymous&j_password=anonymous">Login
```

Edit <server>/webapps/luntbuild/WEB-INF/applicationContext.xml and remove line:

```
anonymous=anonymous,ROLE_AUTHENTICATED
```

Application Server FAQ

1.1.

I am running Luntbuild on tomcat4.1.31 on Linux. But after installing Luntbuild and accessing it in the browser, I get the following stack trace starting with:

* com.luntsys.luntbuild.utility.Luntbuild.getDao(Luntbuild.java:118)
* com.luntsys.luntbuild.web.BuildsTab.loadData(BuildsTab.java:63)
* com.luntsys.luntbuild.web.Home.activateExternalPage(Home.java:67)
What's wrong?

Full Stack Trace:

```
* com.luntsys.luntbuild.utility.Luntbuild.getDao(Luntbuild.java:118)
* com.luntsys.luntbuild.web.BuildsTab.loadData(BuildsTab.java:63)
* com.luntsys.luntbuild.web.Home.activateExternalPage(Home.java:67)
* com.luntsys.luntbuild.web.Home.pageActivated(Home.java:99)
* org.apache.tapestry.engine.HomeService.service(HomeService.java:64)
* org.apache.tapestry.engine.AbstractEngine.service(AbstractEngine.java:872)
* org.apache.tapestry.ApplicationServlet.doService(ApplicationServlet.java:197)
* org.apache.tapestry.ApplicationServlet.doGet(ApplicationServlet.java:158)
* javax.servlet.http.HttpServlet.service(HttpServlet.java:696)
```

```
* javax.servlet.http.HttpServlet.service(HttpServlet.java:809)
* org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:200)
* org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:146)
* org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:209)
* org.apache.catalina.core.StandardPipeline$StandardPipelineValveContext.invokeNext(StandardPipeline
* org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:433)
* org.apache.catalina.core.ContainerBase.invoke(ContainerBase.java:948)
* org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:144)
* org.apache.catalina.core.StandardPipeline$StandardPipelineValveContext.invokeNext(StandardPipeline
* org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:433)
* org.apache.catalina.core.ContainerBase.invoke(ContainerBase.java:948)
* org.apache.catalina.core.StandardContext.invoke(StandardContext.java:2358)
* org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:133)
* org.apache.catalina.core.StandardPipeline$StandardPipelineValveContext.invokeNext(StandardPipeline
* org.apache.catalina.valves.ErrorDispatcherValve.invoke(ErrorDispatcherValve.java:118)
* org.apache.catalina.core.StandardPipeline$StandardPipelineValveContext.invokeNext(StandardPipeline
* org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:116)
* org.apache.catalina.core.StandardPipeline$StandardPipelineValveContext.invokeNext(StandardPipeline
* org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:433)
* org.apache.catalina.core.ContainerBase.invoke(ContainerBase.java:948)
* org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:127)
* org.apache.catalina.core.StandardPipeline$StandardPipelineValveContext.invokeNext(StandardPipeline
* org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:433)
* org.apache.catalina.core.ContainerBase.invoke(ContainerBase.java:948)
* org.apache.coyote.tomcat4.CoyoteAdapter.service(CoyoteAdapter.java:152)
* org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:799)
* org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.processConnection(Http11Protocol.j
* org.apache.tomcat.util.net.TcpWorkerThread.runIt(PoolTcpEndpoint.java:577)
* org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(ThreadPool.java:683)
* java.lang.Thread.run(Thread.java:534)
```

Check your web.xml file and verify that "installDir" parameter is correctly pointing to the Luntbuild installation directory (not the same location as the luntbuild webapp directory under Tomcat). The Luntbuild installation directory is the directory you extract luntbuild zip file to, and there are several subdirectories inside it, such as db, logs, work, publish, etc.

1.2.

After installation, Luntbuild gives me the "HTTP Status 500" error page with message "The server encountered an internal error () that prevented it from fulfilling this request." when I point my browser to Luntbuild. What's wrong?

```
HTTP Status 500 -

type Exception report

message

description The server encountered an internal error () that prevented it from fulfilling this reque

exception

javax.servlet.ServletException: Servlet.init() for servlet luntbuild threw exception
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:118)
org.apache.coyote.tomcat5.CoyoteAdapter.service(CoyoteAdapter.java:160)
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:799)
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.processConnection(Http11Protocol.jav
org.apache.tomcat.util.net.TcpWorkerThread.runIt(PoolTcpEndpoint.java:577)
org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(ThreadPool.java:683)
java.lang.Thread.run(Thread.java:567)

root cause

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'hibernateSes
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAuto
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:159)
org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(Defaul
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.ja
org.springframework.web.context.support.XmlWebApplicationContext.refresh(XmlWebApplicationContext.ja
com.luntsys.luntbuild.LuntBuildServlet.init(LuntBuildServlet.java:103)
javax.servlet.GenericServlet.init(GenericServlet.java:211)
org.apache.tapestry.ApplicationServlet.init(Unknown Source)
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:118)
org.apache.coyote.tomcat5.CoyoteAdapter.service(CoyoteAdapter.java:160)
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:799)
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.processConnection(Http11Protocol.jav
```

```
org.apache.tomcat.util.net.TcpWorkerThread.runIt(PoolTcpEndpoint.java:577)
org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(ThreadPool.java:683)
java.lang.Thread.run(Thread.java:567)
```

The full stack trace of the root cause is available in the Apache Tomcat/5.0.28 logs. This looks like permission issue. Does the user owning tomcat process has read/write access to the Luntbuild installation directory(the directory you extract luntbuild zip package to)?

1.3.

I'm running Luntbuild on Linux platform. During starting up, Tomcat gives the following error :

```
2005-06-22 13:49:53 StandardContext[/luntbuild]Exception starting filter Acegi Security System for S
java.lang.IllegalStateException: No WebApplicationContext found: no ContextLoaderListener registered
at org.springframework.web.context.support.WebApplicationContextUtils.getRequiredWebApplicationConte
```

Please make sure the process running tomcat has full access (rwx) to the Luntbuild installation directory (the directory in which you can find sub directories such as "db", "logs", etc.

Cvs FAQ

1.1.

I use luntbuild to perform build for my cvs repository. However the error "2004-10-14 08:41:15,657 [Thread-41] WARN com.luntsys.luntbuild.build.BuildGenerator - [log]log: invalid option -- S" occurs after I issue the build command. What's wrong?

The reason is, that the cvs version you are using is too old and does NOT support "-S" option. However, you can turn off "-S" option at cvs adaptor editing page, just set "disable -S option" property to "yes".

1.2.

I have a problem using Luntbuild with cygwin cvs, the automatically generated .cvspass file seems invalid to cygwin cvs. Is there any workaround?

For cygwin cvs, please goto cvs adaptor editing page, set "use cygwin cvs?" property to "yes".

1.3.

I use luntbuild to perform build for my cvs repository. However the error "2005-12-01 14:37:07,552 WARN com.luntsys.luntbuild.BuildGenerator - [history]cvs [history aborted]: Can't parse date/time: 2005-11-30 17:31:40" occurs after I issue the build command. What's wrong?

It seems that cvs 1.9 and erler versions have a y2k bug which makes commands with dates like "cvs history" to fail with dates above year 2000. Please upgrade to cvs 1.11 or higher.

Perforce FAQ

1.1.

How do you set up Perforce client spec if multiple users use Luntbuild?

- I set up Luntbuild work directory to /opt/builds.
- I name my project mybuilds-myproject.

- I specify client path in p4 clientspec to //myspec/..., and I got build going in / /opt/builds/mybuilds-myproject.
- I set up build file to build.xml.

1.2.

How do exclude files and directories in Perforce VCS using Luntbuild modules?

To exclude files or directories, create a separate module for each exclussion and precede the Depot path property with a minus (-) sign, as follows:

```
Depot path:  -//depot.side
Client path: //client.side
```

Visual Sourcesafe FAQ

1.1.

What Date format should I use with the Visual Sourcesafe Adapter in the English speaking countries with different date format like UK, Australia, Canada?

The default Datetime value is suitable for English language operating systems using US locale. For other English speaking countries with different date format like UK, Australia, and Canada the Visual Sourcesafe Date format to use (assuming you're using the appropriate locale setup as Visual Sourcesafe honors the local locale settings) should be as follows:

```
'd/M/yy;H:mm'
```

Database FAQ

1.1.

When and how are the project definitions and other Luntbuild data persisted to disk in the out-of-the-box Luntbuild distribution?

Luntbuild by default uses HSQL database to store project definitions. The data for the database is stored in <luntbuild-install-dir>/db. I recommend to backup the <luntbuild-install-dir> daily (most companies have some kind of autamated backup strategies), and if you need to restore the database, you can just:

1. Stop luntbuild (or the whole servlet container/application server).
2. Restore <luntbuild-install-dir>/db directory.
3. Start luntbuild (or the whole servlet container/application server).

The default database write delay time in Luntbuild-1.2 is 60 seconds. You can modify default write interval in the db/luntbuild.script file in your Luntbuild installation. Just modify the following line:

```
SET WRITE_DELAY 60
```

Actually Luntbuild-1.2.1 set this value to 0 to disable any write delay.

Development FAQ

1.1.
How do I build and debug Luntbuild?

Please see Luntbuild Wiki [http://www.javaforge.com/proj/doc/displayWikiDocument.do?doc_id=1433] for details.