

Build Automation:

sviluppo e il testing del codice
Jug Roma - settembre 2006

Continuous Integration
Team Developing
Error Reporting
Real-time information
and more...

Valerio de Carolis
valerio.decarolis@gmail.com
<http://vdecarolis.blogspot.com>



Continuous Integration

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day.

Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly.

Martin Fowler

Continuous Integration

Esistono diversi sistemi opensource per poter praticare la continuous integration:

Lunbuild

(javaforge.com)

Continuum

(maven.apache.org)

CruiseControl

(sourceforge.net)

Hudson

(dev.java.net)

Ed anche sistemi di natura commerciale:

Quickbuild

(sviluppato dallo stesso team di Lunbuild)

Parabuild

I sistemi opensource rappresentano una valida alternativa per chi entra in contatto per la prima volta con la *CI*

In questo talk parleremo di **LunBuild**.

Luntbuild - Abstract

Luntbuild is a powerful build automation and management tool. Continuous Integration or nightly builds can be easily set using a clean web interface.

E' possibile sfruttare le features avanzate quali lo scheduling o il reporting per automatizzare la gestione.

Con Luntbuild si possono creare repository centrali e download area per il team di sviluppo e gli sviluppatori esterni.

You can set your Luntbuild system in less than thirty minutes.

Luntbuild - Technologies

Luntbuild è basato su diverse librerie e framework opensource, tra i quali:

Quartz
Tapestry
Velocity
Xstream
Hibernate
Ehcache



Luntbuild viene distribuito in formato **.zip** oppure sotto forma di **.jar** autoinstallante (creato con **IzPack**)

Per una lista completa delle dipendenze:

<http://luntbuild.javaforge.com/libraries-frameworks-list.html>

Luntbuild - Installation

I requisiti d'installazione sono minimi:

JDK 1.5

(supporta JDK 1.4)

Servlet Container

(compatibile con le spec. Servlet2.3 e JSP1.2)

Se non si dispone di un Servlet Container, è possibile avviare Luntbuild in modalità [“standalone”](#):

```
> cd <luntbuild-install-dir>
```

```
> java -jar luntbuild-standalone.jar localhost 8888
```

(viene avviato un server Jetty)

E' consigliata l'installazione tramite GUI, che autoconfigura il file web.xml e applicationContext.xml, oltre che a creare automaticamente tutte le directories necessarie.

I sorgenti sono rilasciati sotto: PMEase Public License

Luntbuild - Database

Luntbuild richiede la presenza di un database per memorizzare le informazioni riguardanti gli utenti, le build, e i progetti inseriti.

Luntbuild supporta ufficialmente nell'ultime release (1.3.2):

HSQldb
MySql
PostgreSQL
Oracle
H2
Derby
SQLServer

ORACLE



Apache Derby



Se non specificato diversamente, Luntbuild userà il database HSQL o H2 integrato (che per sistemi di testing e/o di piccole dimensioni risultano i più versatili).

Luntbuild – Build systems

Luntbuild è capace di gestire quasi tutti “script” di build esistenti, tra i quali:

Ant

Maven 1

Maven 2

Rake

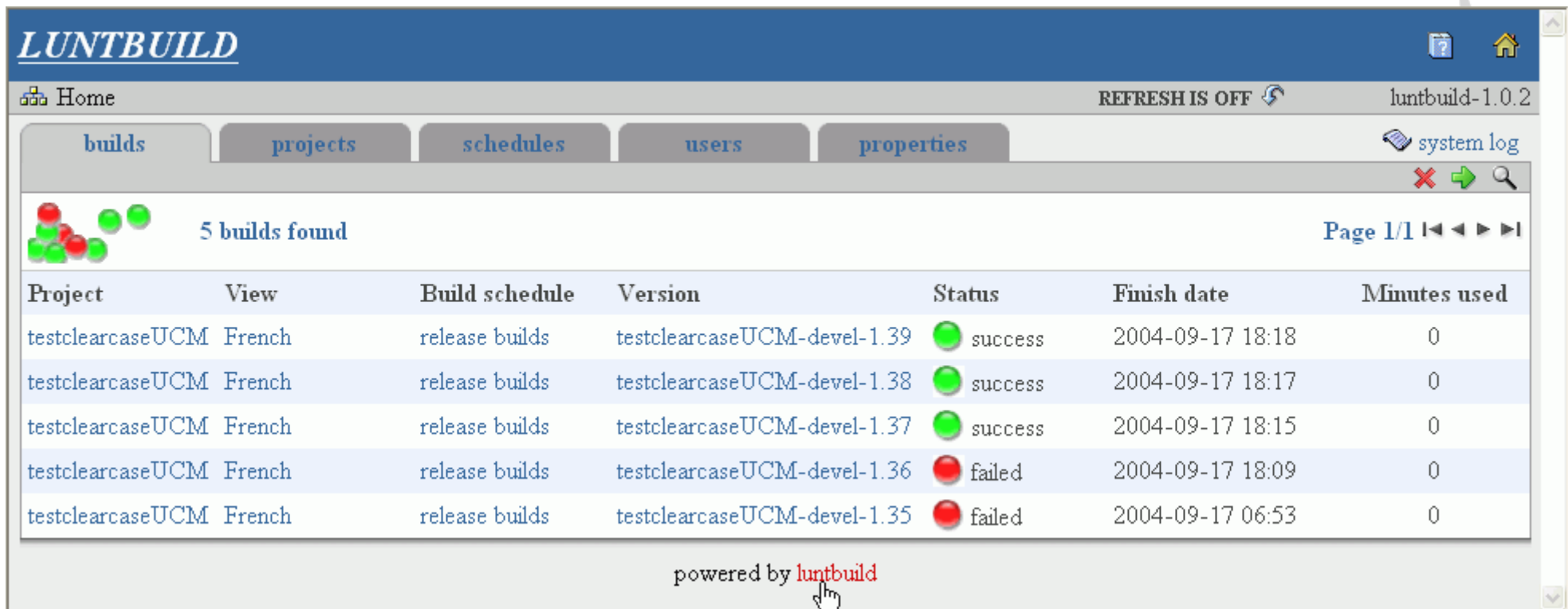


In oltre è possibile specificare un proprio “script” che Luntbuild interpreterà come classico file di comandi per la shell (.bat/.sh).

Specificando il tipo di script è possibile dichiarare tutte le variabili d'ambiente e di percorso.

E' possibile sfruttare espressioni OGNL e/o RegEx.

Luntbuild – Live Demo



The screenshot displays the Luntbuild web application interface. At the top, the title 'LUNTBUILD' is visible. Below it, there are navigation tabs for 'builds', 'projects', 'schedules', 'users', and 'properties'. The 'builds' tab is currently selected. The main content area shows a table of 5 builds found for the 'testclearcaseUCM' project. The table columns are Project, View, Build schedule, Version, Status, Finish date, and Minutes used. The builds are listed with their respective statuses: three successful (green circles) and two failed (red circles). The footer of the interface indicates it is powered by luntbuild.

Project	View	Build schedule	Version	Status	Finish date	Minutes used
testclearcaseUCM	French	release builds	testclearcaseUCM-devel-1.39	success	2004-09-17 18:18	0
testclearcaseUCM	French	release builds	testclearcaseUCM-devel-1.38	success	2004-09-17 18:17	0
testclearcaseUCM	French	release builds	testclearcaseUCM-devel-1.37	success	2004-09-17 18:15	0
testclearcaseUCM	French	release builds	testclearcaseUCM-devel-1.36	failed	2004-09-17 18:09	0
testclearcaseUCM	French	release builds	testclearcaseUCM-devel-1.35	failed	2004-09-17 06:53	0

Nella Live Demo vedremo come configurare Luntbuild per poter gestire una nostra applicazione basata su Ant.

References

Articles:

Martin Fowler: <http://www.martinfowler.com>

Continuous Build Systems:

Luntbuild: <http://luntbuild.javaforge.com>

Hudson: <https://hudson.dev.java.net/>

CruiseControl: <http://cruisecontrol.sourceforge.net/>

Continuum: <http://maven.apache.org/continuum/>