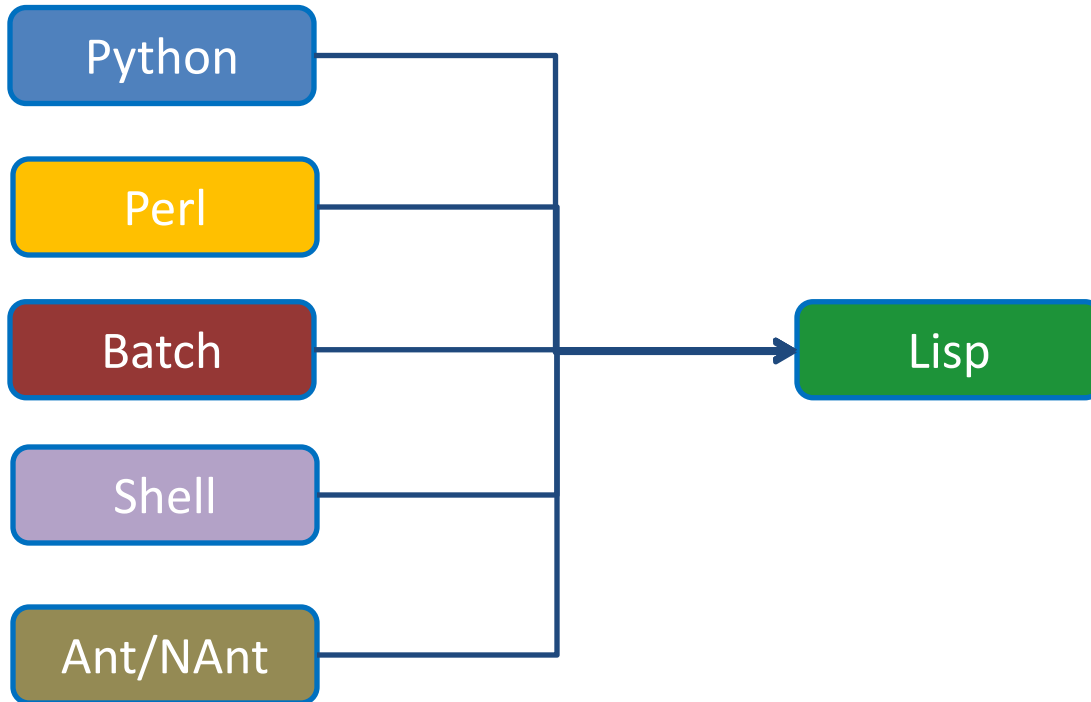


Build Scripts with Common Lisp

最爱素面 2011-11-27

Use Lisp instead of these scripts



Use Lisp instead of xml

```
<copy>  
  <toDir>../new/dir</toDir>  
  <fileset>  
    <dir>src_dir</dir>  
  </fileset>  
</copy>
```



```
(copy  
  (todir "../new/dir")  
  (fileset (dir "src_dir")))
```

Why choosing Lisp?

- Cross platform
- Less Code
- Extensibility
- Programmable programming language
- Interesting

Working environment



Simple Simulation of Build Forge

The screenshot displays the IBM Rational Build Forge web interface. The browser address bar shows the URL: `http://localhost:8083/fulcontrol/index.php?mod=projects&action=steps.view&bf_id=1`. The page title is "BuildForge - Projects". The main content area shows a project named "Web应用程序构建演示" (Web Application Build Demo) with the selector "Local Server" and environment "Web App Env". The user is logged in as "Build Engineer".

The interface features a left-hand navigation menu with options like Home, Projects, Adaptors, Classes, Log Filters, Templates, Libraries, Jobs, Schedules, Environments, Servers, Administrators, Help, Access Groups, and Build Engineer. The main area displays a list of 16 build steps, each with a checkbox, a step name, a selector, an environment, a result status, and an access level. The steps are:

#	Step Name	Selector	Environment	Result	Access
1	Create View for Source			Exit Code	Build Engineer
2	Set Config Specs for View			Exit Code	Build Engineer
3	Create Baseline			Exit Code	Build Engineer
4	Shutdown Tomcat Web Server			Exit Code	Build Engineer
5	Clean build area			Exit Code	Build Engineer
6	Checkout index.html			Exit Code	Build Engineer
7	Update index.html			Exit Code	Build Engineer
8	Compile Source			Exit Code	Build Engineer
9	Create WAR File			Exit Code	Build Engineer
10	Restart Tomcat			Exit Code	Build Engineer
11	Create Checkpoint			Exit Code	Build Engineer
12	Test Application			Exit Code	Build Engineer
13	Archive Test Results			Exit Code	Build Engineer
14	Checkout project.xml			Exit Code	Build Engineer
15	Export Build Configuration			Exit Code	Build Engineer
16	Checkout project.xml			Exit Code	Build Engineer

Below the list, there are buttons for "Save Step", "Delete Step", and "Add Note". The "Details" section for the selected step "Create View for Source" shows the following configuration:

Name:	Enabled	Access:
Directory:	Path:	Build Engineer
/	Absolute	

Scenario

- The scripts needs to change a lot if migrate to Build Forge.
- Build Forge is expensive, not all products can run on it.
- We can do nothing if build forge server down.

We need to found a way to be compatible with build forge functionalities.

Environment setting up

- LIB-PATH:"/*****/lib/"
- PROJECT-PATH:"/*****/"
- APPLICATION:"AppName"
- BUILD-TYPE:"Integrate"
- STAGE-DIR:"*****"
- DEVELOP-DIR:(STAGE-DIR "develop/")
- EMAIL-RECIPIENTS:"**.**@**.com;**@**.com"

Code usages on build forge simulation

- `(defstep setup ('("clisp" ($project-path "setup.lisp") $stage-dir $result-dir))`
- `(defstep sync ('(" ant" ($ANT_DIR "sync.xml")))`
- `(defstep compile ('("perl" ($BUILDSCRIPTS_DIR "\\Perl\\compile.pl")))`
- `(defproject test-proj *env-path* setup
 sync
 compile)`
- `(run-project test-proj)`

Usage for non-lispers

- Enable to setup environment
- The environment variable start with “\$”
- Know the corresponding functions/macros:
 - Defstep
 - Defproject
 - Run-project
 - Defbset
 - Defset
 - ...

Q & A