

Environnements de développement Java sous Ubuntu



Abdelmonam Kouka

<http://geek-tounsi.blogspot.com/>



This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License.





Plan

- EDI
 - Définition
 - Buts
 - Alternatives
 - Les EDI aujourd'hui
- Focus sur Eclipse
- Focus sur NetBeans
- Comparaison entre Eclipse & NetBeans
- Démo



EDI : Définition

- Un environnement de développement intégré (EDI, ou Integrated Development Environment – IDE) est un logiciel regroupant un ensemble d'outils nécessaires au développement logiciel dans un (ou plusieurs) langage(s) de programmation
- Un EDI c'est :
 - un éditeur de texte spécialisé (avec coloration syntaxique, indentation automatique, complétion automatique, . . .),
 - un compilateur (ou au moins l'intégration d'un compilateur existant),
 - un débogueur (ou au moins l'intégration d'un débogueur existant),
 - des outils d'automatisation de la compilation et de gestion de projets

une image vaut 1000 mots...



```
<td>Birthday: </td>
<td>
  <input type="text" value="" />
</td>
</tr>
</tbody>
</table>
<div id="callContainer">
  <input type="button" name="button" value="Calendar" />
</div>
</form>

<script type="text/javascript">
<script type="text/javascript">
<script type="text/javascript">
<script type="text/javascript">
<script src="../javascripts/yahoo/widget/calendar-event.js"></script>

<script type="text/javascript">
var bday = $("bday");
YAHOO.util.Event.addListener(bday, "click", function(e) {
  var cal = new YAHOO.widget.Calendar(cal, callContainer, {navigator:true});
  var cal2 = new YAHOO.widget.Calendar(cal, callContainer, {navigator:true});
  function handleSelect(type, args) {
    var dates = args[0];
    var date = dates[0];
    var year = date[0], month = date[1];
    bday.value = month + "/" + year;
    cal.hide();
  }
  function renderCal() {
    cal.selectEvent.subscribe(renderCal);
    cal.render();
    cal.show();
  }
</script>
</body>
</html>
```

YAHOO.widget
Calendar(String id, containerId, Object config)
From calendar-debug.js

Summary

Calendar is the base class for the Calendar widget. In its most basic implementation, it has the ability to render a calendar widget on the page that can be manipulated to select a single date, move back and forth between months and years.

To construct the placeholder for the calendar widget, the code is as follows:

```
<div id="calContainer"></div>
```

NOTE: As of 2.4.0, the constructor's ID argument is optional. The Calendar can be constructed by simply providing a container ID string, or a reference to a container DIV

Some results omitted. Type more to narrow search.

- Axis()
- BarChart(containerId, dataSource, attributes)
- BarSeries()
- BaseCellEditor(sType, oConfigs)
- Button(p_oElement, p_oAttributes)
- ButtonGroup(Object p_oElement, Object p_oAttributes)
- Calendar()
- Calendar(String id, containerId, Object config)**
- Calendar2up(id, containerId, config)
- CalendarGroup(String id, containerId, Object config)
- CalendarGroup()
- CalendarNavigator()
- CalendarNavigator(Calendar|CalendarGroup cal)
- Carousel(el, cfg)
- CartesianChart(type, containerId, dataSource, attributes)
- CartesianSeries()



une image vaut 1000 mots...

Extrait du code de Linux avec son indentation originale

```
if (unlikely(prev->policy == SCHED_RR)) {
    if (!prev->counter) {
        prev->counter = NICE_TO_TICKS(prev->nice);
        move_last_runqueue(prev);
    }
}
switch (prev->state) {
case TASK_INTERRUPTIBLE:
    if (signal_pending(prev)) {
        prev->state = TASK_RUNNING;
        break;
    }
default:
    del_from_runqueue(prev);
case TASK_RUNNING:;
}
prev->need_resched = 0;
```

Le même code après suppression de son indentation

```
if (unlikely(prev->policy == SCHED_RR))
    if (!prev->counter) {
        prev->counter = NICE_TO_TICKS(prev->nice);
        move_last_runqueue(prev);
    }
switch (prev->state) {
case TASK_INTERRUPTIBLE:
    if (signal_pending(prev)) {
        prev->state = TASK_RUNNING;
        break;
    }
default:
    del_from_runqueue(prev);
case TASK_RUNNING:;
}
prev->need_resched = 0;
```



EDI : Définition

- Des outils complémentaires :
 - un système de gestion de versions (eg CVS ou Subversion),
 - conception d'interface graphique
 - un navigateur de classes (pour explorer la hiérarchie des classes),
 - des outils de tests unitaires et de couverture du code,
 - des outils de maintenance/remaniement du code (refactoring),
 - un générateur de documentation (eg Javadoc ou Doxygen).



EDI : buts

- Augmenter significativement la productivité du développeur :
 - en minimisant le temps passé à basculer entre les différentes tâches intervenant dans le cycle de développement logiciel (édition, compilation, exécution, débogage, test, documentation, import/export vers le dépôt de versions, . . .),
 - en minimisant le temps d'apprentissage requis par les différents outils intervenant dans le cycle de développement : pas de syntaxe (e.g. celles des Makefile), ou de commandes/instructions (e.g. jdb/gdb) à apprendre.
- l'utilisation d'un EDI ne dispense pas complètement d'une certaine familiarité avec chacune des étapes du cycle de développement. . .



EDI : alternatives

- Le cycle édition/compilation/exécution a lieu dans un éditeur de texte avancé qui (en général) gère les fonctionnalités suivantes (entre autres) :
 - coloration syntaxique
 - indentation automatique
 - complétion automatique
 - intégration avec le compilateur
- Exemples classiques :
 - Emacs : libre, multiplateforme, classique linux
 - Vim : libre, multiplateforme, mode commande/édition



EDI : alternatives

- Ce type d'approche du développement logiciel est limitée :
 - la complétion automatique des éditeurs ne dépend pas (assez) du contexte,
 - il n'y a pas ou peu d'intégration avec le débogueur,
 - il n'y a pas ou peu d'intégration avec les outils de gestion de version,
 - des tâches laborieuses comme la maintenance/remaniement de code restent entièrement manuelles,
 - requiert une expertise importante dans chacune des phases du cycle de développement (configuration de l'éditeur de texte, syntaxe des Makefile, commandes du débogueur, ...)

EDI : état des lieux

- Logiciels propriétaires :
 - C++ Builder (Borland) : C/C++, gratuit/payant, Windows uniquement,
 - JBuilder (Borland) : Java (gratuit/payant, multiplateforme),
 - Xcode (Apple) : C/C++, Objective C, Java (payant, Mac OS X seulement),
 - Visual Studio (Microsoft) : C/C++, C#, Web ASP (gratuit/payant, Windows uniquement),
 - Rational Application Developer IBM : Java JEE, payant, multiplateforme

EDI : état des lieux

- Logiciels « libres » :
 - KDevelop (KDE) : C/C++, Java, base sur les outils GNU (GCC, make, GDB) et sur d'autres outils répandus (CVS, Doxygen), Unix uniquement,
 - Anjuta (GNOME) : C/C++ uniquement, aussi base sur les outils GNU (multiplateforme),
 - Netbeans (Sun) : initialement Java uniquement, maintenant C/C++, Java Micro Edition, Ruby, JavaScript, . . . (multiplateforme),
 - Eclipse : Java, C/C++, Java Micro Edition, PHP, . . . (multiplateforme)
 - ...



EDI : focus sur Eclipse

- Initialement un simple EDI Java (IBM/OTI VisualAge for Java), Eclipse est devenu un framework
 - Offrir une plateforme ouverte pour le développement d'applications :
 - non-dédiée à un langage ou un SE
 - facile à comprendre mais surtout facile à étendre,
 - paramétrable selon les besoins/goûts du développeur,
 - capable d'automatiser les tâches lourdes du développement,
 - open source,
 - ouvert et **EXTENSIBLE** par plugins



Eclipse : un framework

- Eclipse = plateforme + greffons (plugins)
- plateforme :
 - un exécutif indépendant du SE (JVM),
 - un ensemble basique de greffons extensibles,
 - des mécanismes (API), règles et outils pour construire des greffons,
 - un moteur pour découvrir, charger et exécuter des greffons
- greffon/plugin :
 - se connecte a des points précis de la plateforme,
 - remplit une tâche (pas forcément exécutable),
 - offre des points d'extension,
 - coexiste avec d'autres greffons,
 - instance : ensemble de greffons qui coopèrent pour offrir un EDI



Eclipse : installation

- Méthode générique pour tout SE
 - Installer un JDK, Sun JDK 5 (ou 6)
 - Télécharger UN zip d'Eclipse [version, SE]
 - Extraire l'archive [conseil : pas d'espace dans le chemin]
 - Lancer l'exécutable eclipse
- Pour ubuntu la vie est simple :)
 - `apt-get install sun-java6-jdk`
 - `apt-get install eclipse`



EDI : focus sur NetBeans

- NetBeans, créé à l'initiative de Sun Microsystems (Noyau de Forte4J/SunOne), présente toutes les caractéristiques indispensables à un EDI de qualité
 - Multi-language (Java, Ruby, C/C++, PHP...)
 - OpenSource...bref tout ce qui est dit pour eclipse :) en plus plain d'autres choses
 - NetBeans rajoute, à coté du support pour CVS et SubVersion, un support pour ClearCase, mais aussi pour Mercurial.



Autres atouts de NetBeans

- un système de projets basé sur Ant, permettant une meilleure indépendance et portabilité de vos travaux ;
- un éditeur visuel pour les fichiers CSS ;
- un support fortement amélioré pour tout ce qui est Ajax, avec entre autre un support pour jMaki et GWT ;
- des éditeurs pour les langages Groovy, Grails, Ruby, Rails, Python, C/C++, PHP, etc. ;
- un support pour importer des projets Eclipse.
- Une palette graphique complète intégrée pour SWING
- ...



NetBeans : installation

- Méthode générique pour tout SE
 - Installer un JDK, Sun JDK 5 (ou 6)
 - Télécharger UN package de NetBeans [version, SE] (ex: netbeans-6.8-ml-linux.sh)
 - Exécuter: `chmod +x netbeans-6.8-ml-linux.sh`
 - Exécuter: `sudo ./netbeans-6.8-ml-linux.sh`
 - Lancer netbeans du menu « Programming »
- Pour ubuntu la vie est toujours simple :)
 - `apt-get install sun-java6-jdk`
 - `apt-get install netbeans`



Enfin qui est le meilleur: Eclipse ou NetBeans?

- Installation: Sans grandes difficultés, Eclipse reporte cette manche, car il suffit de décompresser le ZIP d'eclipse et puis l'exécuter
- Démarrage : C'est un petit avantage à Eclipse, mais si infime qu'il ne peut être compté.
- Premier abord: Pour ce qui est de ce que l'on voit, l'avantage va sans contexte à Netbeans, les menus ont une très jolie police, et les icônes sont ravissantes.
- Contenu: On retrouve ensuite pour les deux un contenu similaire, les projets se font de la même manière, il n'y a pas vraiment de différences entre les deux sur ce point de vue la.



Enfin qui est le meilleur: Eclipse ou NetBeans?

- Utilisation: Pour l'utilisation, je préfère clairement Netbeans, pour son aspect visuel bien sur, mais aussi car comme de Sun, on a accès facilement à l'aide, à savoir, lorsque que l'on utilise le fameux raccourci <espace>+<ctrl>, marchant sur les deux, on a une description de ce que l'on utilise, qui peut être très intéressante dans beaucoup de cas.

L'avantage d'Eclipse dans l'utilisation est un visuel plus simple, donc une facilité de lecture, car on se fatigue moins les yeux, enfin c'est ce que je pense, Netbeans fait plaisir a ouvrir, mais à la longue on aime bien avoir Eclipse.

- Greffons: Je n'utilise pas trop les greffons, je ne serai pas donc très pointu sur mon jugement, par contre ce que j'en ai à dire, c'est que c'est très facile pour Eclipse d'en installer la plupart, voir tous.



Conclusion

- Si on fait la conclusion, d'un point de vu mathématique, les scores sont égaux, ce qui confirme ce que j'avais pu dire avant, maintenant pour ma propre expérience, j'avais commencé sous Netbeans lors de mes études, et j'utilise Eclipse actuellement dans ma vie professionnelle et je l'apprécie bien.
- Je pense que quelque soit votre choix, vous ne pourrez vous tromper :)
- Je pense aussi que dans le monde du libre, c'est à chacun de faire son choix, et je vous encourage à installer les deux pour pouvoir les tester.

Démos et puis vos questions



