

Nouveautés de Swing dans Mustang (Java 6.0)



Date de publication : 21/09/2005

Dernière mise à jour :

Compte rendu de la séance technique de chat sur : Les nouveautés de Swing dans Java 6.0 (Mustang)

- I - Sujet, participants et déroulement
- II - Compte rendu de la discussion
- III - Remerciements et témoignages
- IV - Téléchargements et liens

I - Sujet, participants et déroulement

Cette première séance technique organisée par l'équipe Java de developpez.com s'est déroulée sur notre [chat](#) le 15/09/2005. La discussion s'est faite pendant plus de trois heures, avec une douzaine de participants, autour des nouveautés de l'API **Swing** dans **Java 6.0** (nom de code Mustang).

Nous avons eu la chance de pouvoir poser nos questions directement à un des développeurs de l'API Swing : [Romain Guy](#). Ayant eu le temps, durant cette discussion de parler de choses aussi diverses que l'API Swing, Mustang, les équipes travaillant sur le JDK de Sun, NetBeans, Niagara (devinez ce que c'est ...) et le menu du Japonais dans lequel Romain était en train de manger, ce compte rendu n'est qu'une version abrégée de la séance ;)

Néanmoins, vous pourrez en apprendre beaucoup sur les évolutions prévues dans Java 6.0 pour Swing, Java2D, les L&F, Synth, JWS, etc.

Ce compte rendu reprends le plus fidèlement possible la discussion du chat : questions des participants et réponses de Romain. Bonne lecture :)

II - Compte rendu de la discussion

* Pourrais tu tout d'abord nous résumer les nouveautés qu'on verra dans Swing pour Java 6 ? *vbrabant*

Bien sûr, mais avant de commencer je tiens à préciser que Mustang est une petite release comparée à Tiger (1.5) ou Merlin (1.4). Le cycle de développement est beaucoup plus court. Tiger est disponible depuis un an environ et nous sommes en ce moment même en train de finaliser la première beta de Mustang. Donc Swing propose deux nouveautés majeures, dont une est plutôt un correctif. La première est le support, tant attendu, du sub pixel anti-aliasing également appelé ClearType par Microsoft et la seconde, le correctif donc, est l'implémentation d'un véritable double buffering au coeur de Swing/AWT/Java2D. Cette implémentation supprime la fameux effet de rectangle gris que l'on observe en déplaçant une fenêtre par dessus une application Swing. Il y a d'autres nouveautés mais moins importantes.

* Une idée de la date de la première bêta ? *vbrabant*

C'est une question de semaines. En fait nous avons fini le code, les équipes Qualité sont en train de tester nos correctifs. Elle sera sûrement là début Octobre.

* Des nouveaux layouts manager au programme ? *bulbo*

Pas pour le moment, Mustang améliore surtout des fonctionnalités existantes. Cela étant dit Scott Violet, l'architecte de Swing, a fait une présentation sur les Layout Managers à JavaOne, il connaît très bien le sujet. On va commencer à planifier Dolphin, si vous sentez qu'un nouveau Layout Manager est indispensable n'hésitez pas à nous envoyer des mails à ce sujet.

* Même pas celui de Matisse ? *vbrabant*

Le layout manager de Matisse est un cas particulier, il ne sera pas intégré au JDK. En revanche, pour le développer des nouveautés ont été introduites. Les layout managers peuvent par exemple aligner les composants sur leur baselines.

* Cela ne risque pas d'être un peu pénalisant de sortir une version de Java tout les ans ? *bulbo*

Comme je l'ai dit Mustang sort très vite, et la version finale n'est pas attendue avant l'été prochain, soit 2 ans après Tiger. Dolphin sera beaucoup plus important, le cycle de développement sera de l'ordre de deux fois celui de Mustang. Il est vrai que ça va vite et on aimerait voir plus de gens passer sous Tiger.

* Pour en revenir à Swing: Le drag'n drop va t'il pouvoir interagir avec la plateforme désormais ou reste t'il toujours limité à la VM ? *bulbo*

De quelle interaction parle-tu ? Il a toujours été possible, par exemple, de drag n'dropper des fichiers depuis l'explorer vers une application Swing. En parlant de drag n' drop, le bug des JTree a été corrigé. En outre un drag n' drop dans un JTree, une JList ou une JTable permet maintenant d'afficher un indicateur visuel, pour dire où sera placé l'élément.

* Qu'en est-il des Look and Feel ? Celui de Vista par exemple ? *vbrabant*

Bonne question. L'équipe de Java2D a bossé sur le sujet et on a quelque chose de similaire à GTK. Au lieu d'écrire un look and feel Vista, on a un moteur qui extrait les ressources de l'OS. De fait, quelle que soit le look and feel final de Vista... Swing a le même. Joshua Marinacci a également bossé sur le look and feel Windows XP pour corriger les dernières imperfections. Et je travaille pour ma part sur Synth, j'ai corrigé plusieurs bugs importants et

ajouté quelques fonctionnalités bien utiles.

*** Qu'en est-il des nouveaux composants Swing ? Je pense notamment aux éléments récupérés à partir de projets comme JDNC et JDIC. Des détails sur ce point précis ? Et y aura t il d'autres api natives du style system tray ? (browser natif, lancement d'applications etc...) vedaer, lunatix**

Oui, Mustang intègre le TrayIcon de JDIC et en même temps les fonctionnalités de JXTable du projet SwingX ont été portées dans Swing. On peut donc maintenant trier et filtre les JTable en standard (enfin !). C'est d'ailleurs le rôle du projet SwingX, tout ce qui est fait dedans pourrait un jour finir dans le JDK. Je vous invite d'ailleurs à contribuer.

*** Donc pas de nouveaux composants Swing ? vbrabant**

Non, peut être dans Dolphin. Cela dit on en trouve de très intéressants dans SwingX et Richard Bair, leader du projet SwingLabs/SwingX, travaille à temps complet dans les mêmes locaux que l'équipe Swing.

*** Même pas le joglpanel ? lunatix**

Le joglpanel est fourni avec JOGL. Il y a un début d'intégration entre OpenGL et Swing mais on a quelques idées pour aller encore plus loin.

*** Pas de docking framework de prévu pour swing pour la suite (une DockJframe qui par défaut pourrait stocker ses sous fenetres dans des barres laterales) ? lunatix**

Le docking ça va d'abord atterir dans SwingX : rien d'officiel ni de définitif ni de prévu.

*** Et du point de vue des performances de Swing ? Est-ce amélioré dans JDK 6 ? vbrabant**

En utilisant le pipeline OpenGL oui. On se rend également compte que dans de nombreux cas le pipeline software va plus vite que le pipeline DirectX sous Windows. Il faut donc tester vos applications au cas par cas. Mais de manière générale les performances de Swing sont surtout menacées par une mauvaise utilisation de l'EDT. C'est d'ailleurs pour cela que le SwingWorker est intégré à Mustang.

*** SwingWorker ? vbrabant**

Le SwingWorker est une vieeeiiiille classe proposée par Sun qui simplifie l'usage de l'EDT. Elle crée pour vous un thread dans lequel vous placez votre code de traitement et propose des méthodes "avant" et "après" qui sont exécutées dans l'EDT. Elle est très simple à utiliser et je vous recommande d'y jeter un oeil. Scott Delap (<http://www.clientjava.com>) a un excellent tutorial là dessus dans son livre Desktop Java Live. Cette classe est indépendante des versions du JDK. On la trouve en téléchargement ça et là pour les JDK 1.5, 1.4, 1.3...

*** Et il y a des chances de voir les composants SwingX dans l'API standard ? adiGuba**

C'est le but de SwingX. Les fonctionnalités de la JXTable sont maintenant dans Mustang, avec une API un peu différente pour mieux s'intégrer au JDK cela dit. Mais vous pouvez contribuer à SwingX ****et**** à Mustang. SwingX a un nombre raisonnable de contributeurs mais beaucoup d'observateurs (50+ enregistrés). Mustang n'a pas encore beaucoup de contributeurs mais nous sommes là pour vous guider, en particulier Joshua Marinacci.

*** Des supports d'autres L&F pour Unix/Linux ? vbrabant**

Pas pour le moment. En fait le plus simple sera d'utiliser Synth pour cela.

*** Pourrais-tu expliquer un peu l'apport de Synth pour développer de nouveaux L&F ?** *vbrabant*

Créer un LaF demande malheureusement beaucoup de temps et nous ne sommes pas nombreux. Synth est une technologie similaire à GTK pour les thèmes. Un descripteur (ici un fichier XML) permet d'appliquer des images aux composants suivant leur état. Il suffit par exemple d'écrire cela pour skinner un bouton :

```
<state>
    <imagePainter method="buttonBackground" path="images/button.png" sourceInsets="4 4 4 4"
  />
</state>
```

Donc dans Mustang nous avons introduit plusieurs nouveautés pour Synth. Ma préférée est le chargement des thèmes via une URL. Dans mes tests j'ai par exemple chargé des skins depuis des serveurs HTTP, FTP, des JAR, des ZIP, etc. Synth permet également d'empiler des couches, un peu comme les calques sous Photoshop.

*** L'integration de rhino script (java script) a mustang peut elle servir a swing ? si oui comment ?** *lunatix*

Pour Rhino cela peut en effet servir à Swing mais indirectement. J'ai par exemple développé un éditeur de code source, www.jext.org, dans lequel toutes les actions des menus sont écrites en Python. L'utilisation d'un langage de scripting dans une UI a un avantage génial : on peut changer le code sans relancer/compiler le logiciel.

*** Y aura t'il un langage de script genre BeanShell ou Groovy un jour dans le JDK ?** *bulbo*

J'en doute, Rhino est proposé pour pouvoir utiliser l'interface de scripting, mais rien ne vous empêche d'utiliser un autre langage avec cette interface. Le cas de Groovy est plus délicat puisque c'est une JSR non finalisée.

*** Utiliser Swing dans une application J2ME, c'est possible ?** *adiGuba*

Et la réponse est... oui. J'ai développé pour JavaOne une application Swing/Java2D tournant sur un téléphone mobile. Il faut cependant faire attention, cela ne marche que pour les appareils J2ME CDC et pas CLDC MIDP (la plupart des téléphones mobiles sont CLDC MIDP). Plus d'infos ici : <http://www.jcp.org/en/jsr/detail?id=209>

*** Et Java dans les GSM ? C'est un Swing light, ou le swing complet ?** *vbrabant*

Swing complet, avec Java 2D. Encore une fois, ce n'est pas du CLDC MIDP.

*** Il existe déjà de nombreux outils pour générer des interfaces graphiques Swing a partir de fichiers XML ("à la XUL"). Par exemple JDNC Markup Language. Est-ce que quelque chose de standardisé est prévu pour Swing ? Que penses tu de ce genre de techno ?** *vedaer*

On y pense sérieusement, Hans Muller a fait une présentation à ce sujet à JavaOne. On va essayer de faire quelque chose, c'est certain, mais quand et comment je ne sais pas du tout. Personnellement je n'ai pas été convaincu par ces technologies (j'ai fait du XUL notamment). Disons qu'elles sont parfaitement adaptées à la réalisation d'applications "normales" mais pour sortir du cadre par défaut... XAML semble plus approprié pour cela.

*** Peut on imaginer l'integration de swixml (ou un equivalent) dans dolphin et java web start (swixml est un langage xml de description de gui pour swing, a la xul) ?** *lunatix*

L'intégration d'un projet open source tiers est toujours délicat. Il y en a quelques uns dans le JDK mais ce n'est pas facile pour des raisons légales. Le mieux serait que vous en parliez sur les forums javadesktop.org. Vous pouvez aussi rédiger une RFE dans la Bug Parade et contacter directement Sun, notamment Hans Muller qui connaît bien ce sujet.

*** Mais si Java était également open source, cela serait plus facile pour intégrer d'autres projets open source. Que pense tu personnellement des licences Java ?** *vbrabant*

Pas dans la distribution officielle de Sun. Sun a annoncé les nouvelles licences lors de JavaOne 2005 et elles sont pour moi très satisfaisantes, notamment la licence pour utilisation interne à une entreprise. Je n'ai personnellement jamais ressenti le besoin d'une autre licence et puis nous avons aujourd'hui le projet PeaBody, qui permet à quiconque de contribuer au JDK. Et vous savez quoi ? Vos contributions sont prioritaires à notre travail. Si je soumetts un correctif et que vous faites la même chose au même moment, le vôtre sera examiné en premier.

*** Je reviens sur synth. tu travailles sur synth studio, ce projet doit il devenir un application complete ou est ce juste un test/ passe temps / proof concept ?** *lunatix*

C'est bien là la question que je me pose. J'aimerais que cela devienne quelque chose de complet, mais cela demande du temps. Plusieurs personnes m'ont contacté et disposent déjà d'applications semblables. Je vais essayer une collaboration entre ces personnes et moi-même pour parvenir à quelque chose.

*** A quand un rendering JSF qui générerait des applications Swing au lieu de HTML ?** *vbrabant*

Les équipes de Studio Creator réfléchissent à ça d'après ce que j'ai compris. Quoi qu'il en soit ce n'est pas l'équipe Swing qui s'en chargera, même si on intégrera sûrement des changements pour leur simplifier la tâche, s'ils le font.

*** Pourrait tu nous en dire plus sur l'équipe Swing ? Qui et combien êtes vous ? Quels outils, EDI, méthodes utilisez vous lors du développement ? L'ambiance est bonne ?** *vedaer*

Bien sûr que je peux en parler. Dans l'équipe Swing elle-même nous sommes une grosse dizaine de personnes et nous travaillons étroitement avec Java2D et AWT. L'équipe Swing (Java2D également) est à Santa Clara en Californie, ainsi qu'au Canada pour un membre et AWT est en Russie. L'ambiance est excellente car Sun nous laisse une grande liberté : horaires flexibles, pas de codes vestimentaires, un bureau par personne, etc. Pour le développement cela dépend de chacun. Je suis le seul sous Eclipse (mais chut ;), les autres travaillent sous Emacs, jEdit ou NetBeans. Ensuite nous avons tout un tas d'outils internes, notre gestionnaire de version n'est pas CVS, SourceSafe ou autre, mais TeamWare.

*** Comment s'est passé l'intégration dans l'équipe Swing ? Facile ?** *vbrabant*

Ah oui aucun problème, moins d'un mois après mon arrivée je bossais sur des démos pour JavaOne et ils m'ont donné une session à JavaOne.

*** Quel va être ton apport (de spécialiste gui) chez Sun pour la suite ?** *lunatix*

Principalement finir un look and feel Synth et ensuite bosser sur Swing comme les autres. Mais je vais continuer à faire plein de démos visuelles pour voir ce qu'il pourrait être intéressant d'avoir dans Swing, notamment pour les animations. Je discute pas mal avec les membres de Java2D. J'ai besoin d'avoir des flous rapides et ils pensaient tenter ça avec des pixel shaders.

*** Tu as écrit un article dans Login sur Wicket, qui se rapproche du développement Swing pour le client-serveur. Qu'en penses-tu ?** *vbrabant*

Oui j'ai écrit au sujet de Wicket, j'ai été emballé. Je ne suis pas fan du développement Web, même si j'ai touché à pas mal de choses de ce côté là, mais Wicket m'a donné envie d'en faire. C'est simple et élégant je trouve, et en effet ça parle aux développeurs AWT/Swing.

*** Que penser de l'engouement des entreprises à faire de plus en plus d'applications web, même en interne**

? *vbrabant*

Je comprends les motivations en terme de développement et de maintenance, mais pour les utilisateurs, mon dieu quelle plaie. Bien sûr, c'est justifié dans certains cas, mais bien souvent je trouve cela inadapté et maladroit, d'autant plus qu'avec des technologies comme JNLP ou XUL, il est possible d'avoir les avantages des deux solutions. J'ai fait beaucoup d'AJAX récemment, qui rapproche le web du desktop, mais cela reste compliqué et surtout très brouillon à développer.

*** Parlant de JNLP, quels arguments utiliser pour convaincre l'utilisation de JNLP et ses avantages ?**
vbrabant

C'est facile, ça ne coûte rien si on a déjà un serveur web, le développement n'est (presque) pas changé et le déploiement est automatique. Je suis déçu par la faible adoption de WebStart, je m'en sers énormément, et chez Sun nous avons plusieurs outils en WebStart, et c'est tellement pratique ... Cela dit si le client WebStart ne plaît pas il existe des alternatives, il ne s'agit que d'UNE implémentation de JNLP.

*** Sans doute la peur du certificat.... Il faut dire que le message n'est pas fait pour rassurer l'utilisateur.**
christopheJ

Ah c'est corrigé dans Mustang ça. Cela dit on peut se passer de ces messages alarmants en utilisant l'API JNLP. Un développeur Swing a récemment écrit deux articles intéressants sur WebStart :

<http://today.java.net/pub/a/today/2005/08/11/webstart.html> et

<http://today.java.net/pub/a/today/2005/09/01/webstart.html>

*** Tu traduais pas un livre sur Swing ? Est-ce que ta présence à Javapolis est confirmée ?** *vbrabant*

Je n'ai pas traduit un livre sur Swing, j'ai écrit dedans : "Swing Hacks" aux éditions O'Reilly. Et je serais en effet à JavaPolis pour deux sessions. Je travaille d'ailleurs sur des trucs sympas que j'espère avoir le temps de finir pour JavaPolis.

*** Tu peux dévoiler un peu ?** *vbrabant*

Bien sûr, j'ai une démo presque terminée sur l'utilisation d'une physique réaliste pour un drag n' drop. Elle utilise du motion blur et de la physique des oscillateurs pour animer une icône. Je réfléchis pas mal à l'utilisation de la physique des dessins animés en me basant sur un vieux projet de Sun créé pour le langage Self. Et depuis hier je bosse sur l'adaptation des performances des rendus Java2D à l'aide d'un moteur de logique floue.

*** A part Swing, quelle est pour toi la raison pour laquelle il faudrait adopter au plus vite Java 6 à sa sortie ? Quelle est la killer feature ?** *vbrabant*

L'anti-aliasing. Pour le reste l'intégration des API des WebServices JAX-WS et JAXB.

*** Les éléments deprecated seront-ils supprimés un jour ? Ne serait-ce que pour simplifier l'API ?** *adiGuba*

Non, en fait, pour des raisons de compatibilité les éléments dépréciés contiennent le code, les nouvelles méthodes appellent en fait les méthodes dépréciées. S'il est vrai que cela simplifierait l'API, le gain serait minime pour le JDK. On réfléchit plutôt à une nouvelle JavaDoc qui permettra de filtrer les dépréciés en cliquant un lien ou un bouton.

*** La classe Vector va t'elle enfin disparaître de l'API Swing ou au moins les ArrayList vont-elles apparaître**

? Depuis le temps que Sun deconseille le Vector. *bulbo*

Vector ne disparaîtra jamais, on doit malheureusement soutenir notre passé. Gérer un Vector et une List implique d'énormes modifications dans JTable et crois moi tu ne veux pas voir le code de la JTable. Cela dit c'est possible.

*** Que pense l'équipe de Swing au sujet de SWT ?** *vbrabant*

Je ne peux pas parler pour les autres au sujet de SWT mais je n'aime pas personnellement. Je ne parle pas de la technologie elle-même, juste de l'API. Je trouve que c'est un pas en arrière par rapport à Swing ou même QT.

*** Tu nous disais un peu plus tôt que Mustang n'apporterait pas autant de "boulversements" à Java que Tiger, par exemple. Mais qu'en est-il du futur ? Y'a t'il déjà des choses à raconter sur Dolphin ?** *vedaer*

On a des idées pour Dolphin, par exemple intégrer la gestion de XML au langage comme le fait ECMAScript ou C# 3.0 (quoique le cas de C# est plus compliqué). En fait XML serait un type de données. Par exemple, imagine que ta variable element est un noeud XML, tu pourrais alors faire :

```
element.nomAttribut = valeur
```

Cela dit rien n'est décidé. On va également essayer de proposer des outils pour réaliser plus facilement des effets dans Swing, par exemple avoir des classes pour animer plus facilement.

III - Remerciements et témoignages

Nous tenons à remercier tout particulièrement Romain pour sa disponibilité et pour tous les éclaircissements qu'il nous a apportés. Merci aussi aux participants de cette première séance de chat qui en on fait un moment particulièrement convivial et enrichissant.

Vous pouvez lire ci dessous quelques avis des membres ayant participé à la discussion.

adiGuba

Instructif et prometteur quand à l'avenir de Java dans les applications Desktop

bulbo

Je trouve cette première expérience de chat technique vraiment réussie et je reparticiperai avec plaisir. Outre le coté technique qui était au rendez-vous et qui en a fait une discussion très instructive, le ton en était convivial et détendu, ce qui explique surement l'heure tardive ou cela c'est terminé, personne n'ayant envie de mettre fin a ce bon moment.

vedaer

C'était vraiment une discussion très intéressante qui m'a permis personnellement d'avoir les idées plus claires sur ce que l'on doit attendre de Mustang en général et des évolutions de Swing en particulier. Merci encore à Romain pour sa gentillesse et sa disponibilité.

IV - Téléchargements et liens

Liens

- [Mustang](#) : site principal dédié à Java 6.0 nom de code Mustang.
- [JDK 6.0 Snapshot Releases](#) : téléchargements des binaires de previews de Mustang
- [Documentation du JDK 6.0](#)
- [Java Development Kit Community](#)
- <http://gfx.developpez.com/> : site de Romain Guy sur www.developpez.com
- <http://weblogs.java.net/blog/gfx/> : blog de Romain Guy sur java.net
- <http://jroller.com/page/gfx> : blog de Romain dédié à Swing

Vous pouvez télécharger une version PDF de ce compte rendu [ici](#) ou encore [ici](#).