

Desktop Java En Action

**Romain Guy
Swing Team**

Sun Microsystems

Agenda

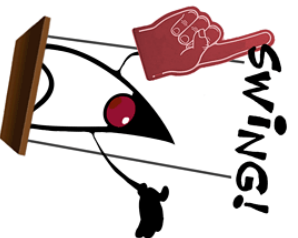
- Aperçu Desktop Java
- Fundamentals
 - > Databinding
 - > SwingX/JDIC
 - > Projets tierces parties
 - > Effets graphiques avec Swing
- Outils
- Q&R

Agenda

- **Aperçu Desktop Java**
- Les Fondamentaux
 - > Databinding
 - > SwingX/JDIC
 - > Projets tierces parties
 - > Effets graphiques avec Swing
- Outils
- Q&R

Citation de la semaine:

“Swing is the dominant
GUI toolkit”



Evans Data Report, Printemps 2005

<http://weblogs.java.net/blog/hansmuller>

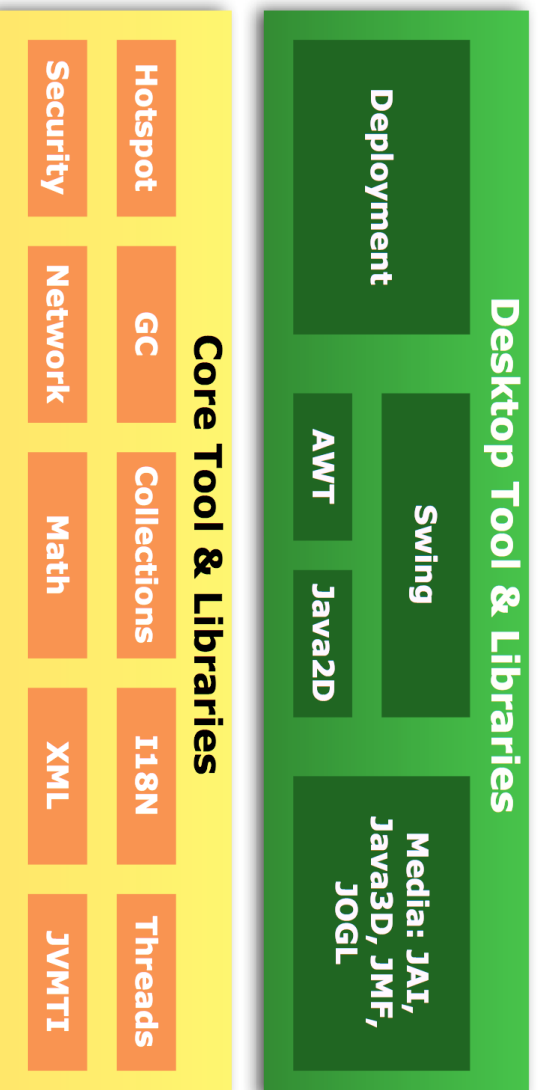
Qu'est-ce que Java pour Desktop ?

- GUI Puissant, Graphismes, et des outils de déploiement & APIs



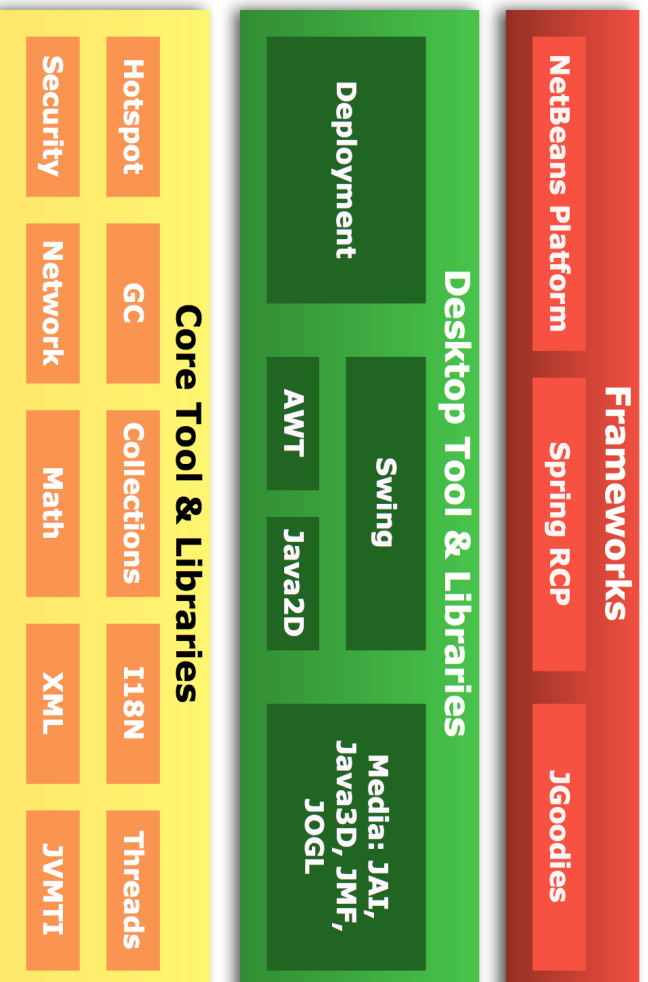
Qu'est-ce que Desktop Java?

- ... en combinaison avec le reste de la plateforme Java SE

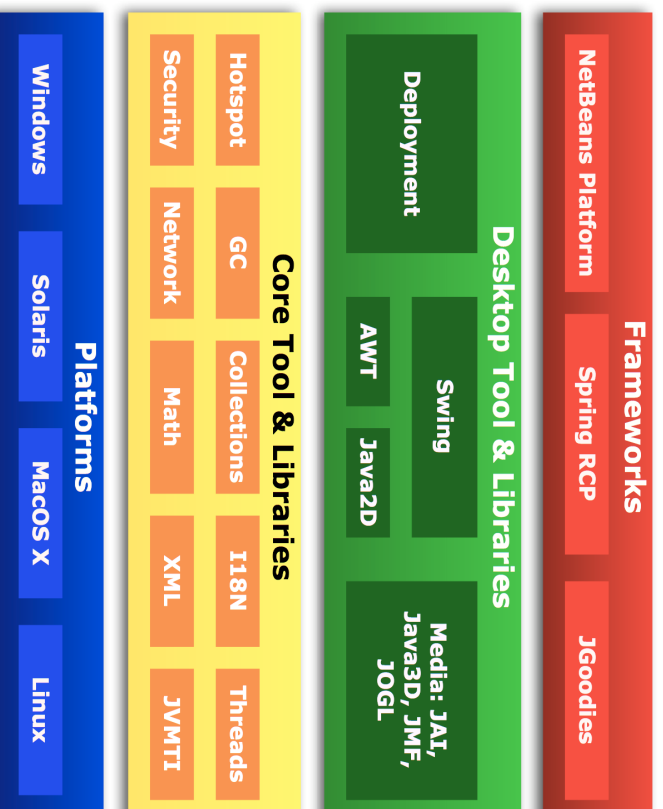


Qu'est ce que Desktop Java?

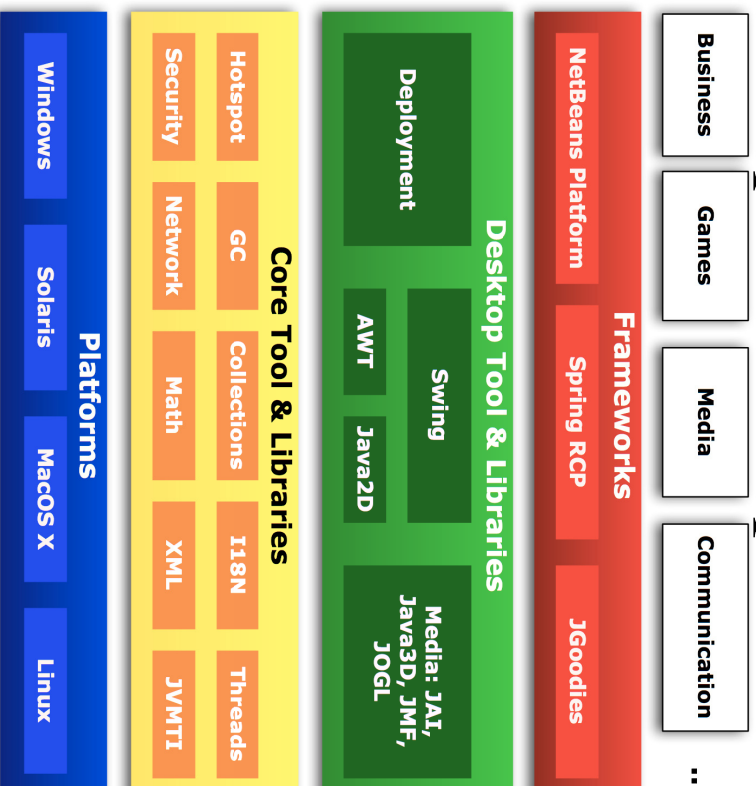
- ... Et des frameworks optionnels



- ## Qu'est ce que Desktop Java?
- ... tournant sur toutes les plateformes bureautiques importantes



Qu'est-ce que Desktop Java?



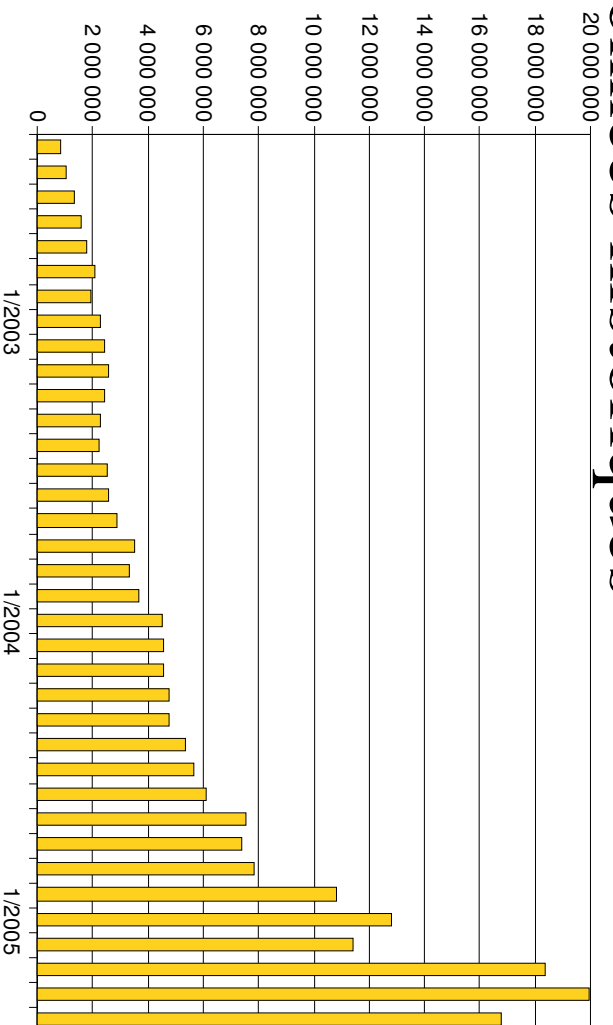
Desktop Java Aujourd'hui: Développeurs

- 4.5 millions de développeurs Java professionnels
- À quoi les développeurs Java passent leur temps (Evans Data, 2005):
 - > 43%: Applications bureautiques et applets
 - > 41%: Servlets, EJBs, JSPs
 - > 4%: Mobile
- "Swing is the dominant GUI toolkit"
 - > Evans Data, Printemps 2005
 - > Comparaison de Swing avec MFC, WinForms, et autres

Desktop Java Aujourd'hui: Utilisateurs

- 87% des PCs ont Java VM (de Sun ou Microsoft)
- >50% des PCs ont une Java VM de Sun
- PC OEMs: 60% des nouveaux PC sont couverts par des accords de distribution pour livrer les JVM de Sun
- téléchargement java.com:
 - > Quadruplé jusqu'à 15-20 millions de téléchargement/mois
 - > Taille téléchargement 7 Mo (mises à jour sont en-dessous de 2 Mo)
 - > Taux de succès d'installation >95% sur le marcher US

Téléchargements: Quelques Données historiques

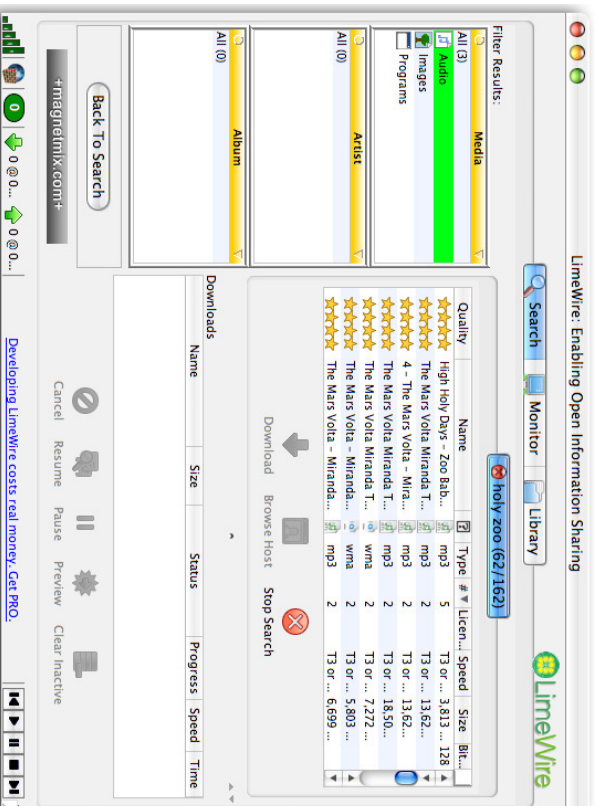


Téléchargement : Tailles

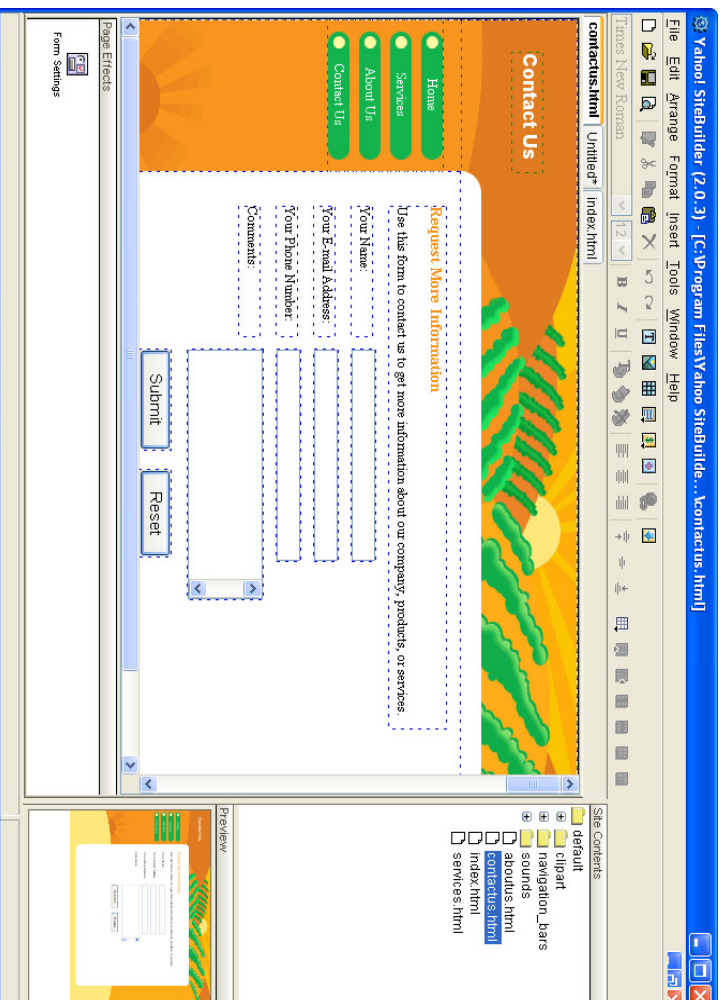
Application/Plateforme	Taille (Mo)	Application/Plateforme	Taille (Mo)
Flash	1.3	DivX Player	7.4
Google Desktop	1.4	Firefox 1.0.7 (Linux)	8.2
Limewire	2.2	AOL AIM	8.3
Ad-Aware 1.06	2.7	Trillian	8.6
WinZip	3.9	ZoneAlarm	8.9
ICQ 5.04	4.1	RealPlayer	9.6
Firefox 1.0.7 (Windows)	4.7	Mozilla 1.7.12	11.0
iMesh	5.0	Google Earth	11.2
Thunderbird 1.0.7	5.8	Quicktime (Windows)	11.7
Azureus	6.9	Windows Media Player	12.1
MSN Messenger	7.0	.NET Framework 2.0	23
J2SE JRE 1.5.0_05	7.1	iTunes 6	34.5

LimeWire: Java on the large scale

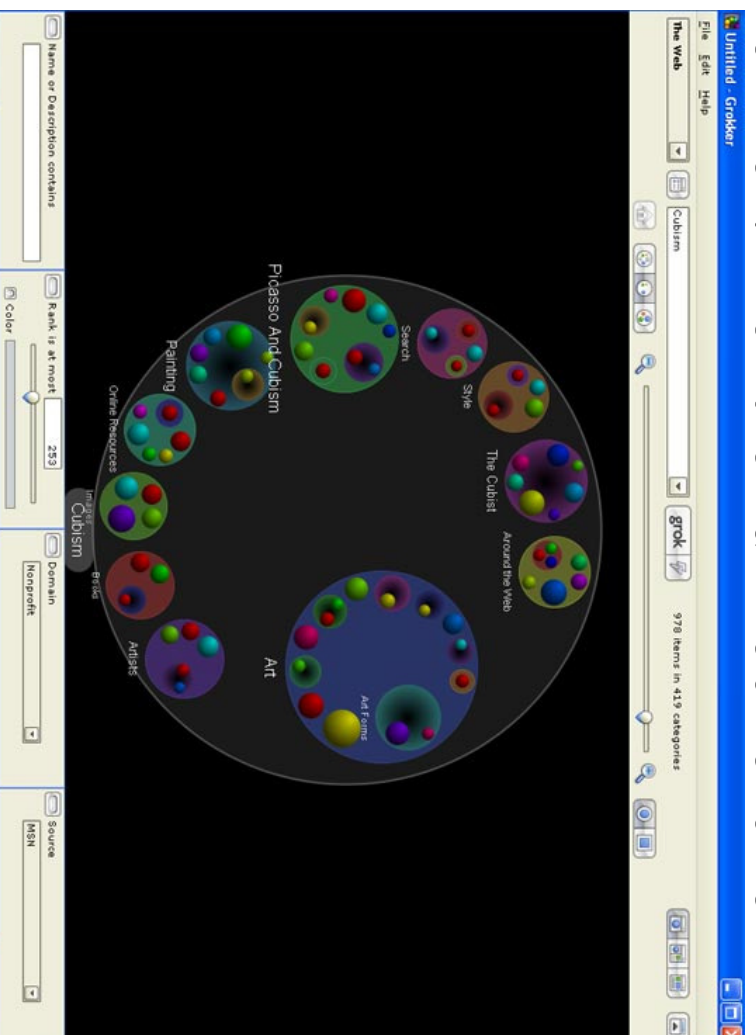
- Application de partage de fichiers très populaire, top 5 sur download.com depuis plus de 2 ans (total de 68m)



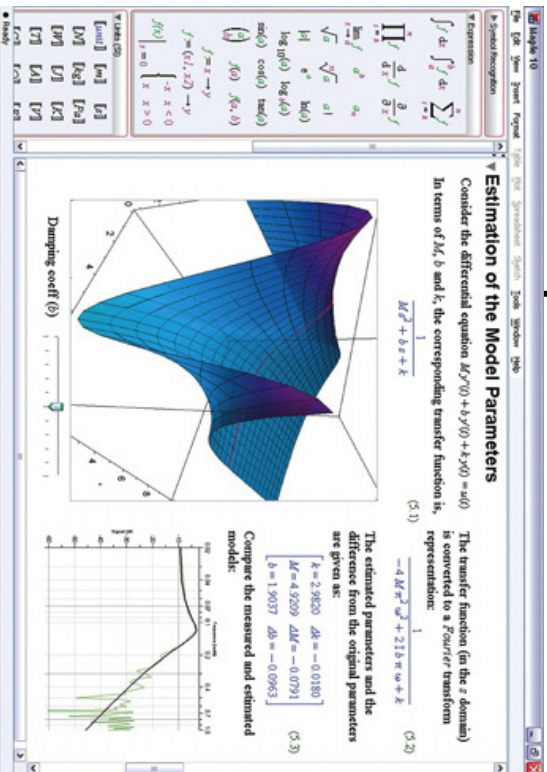
Yahoo! Sitebuilder: Éditeur HTML



Grokker: front-end Recherche

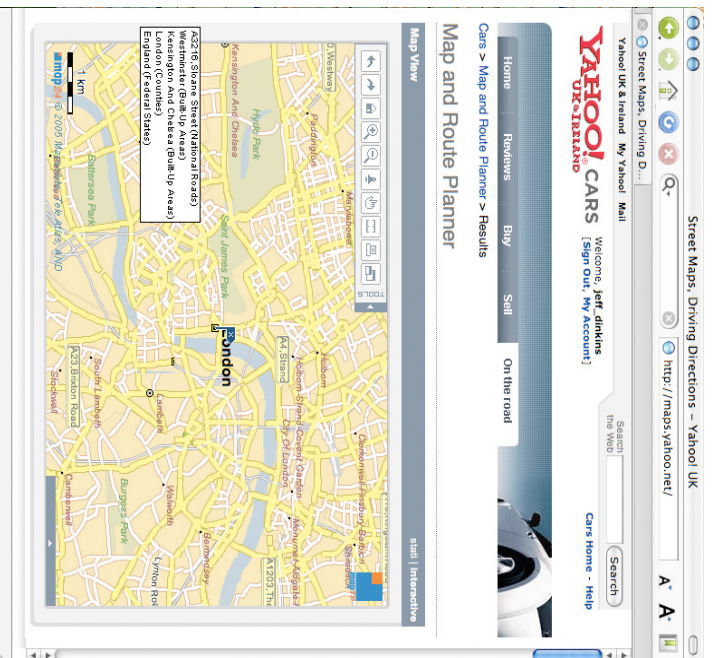


Maple: Java dans la communauté scientifique

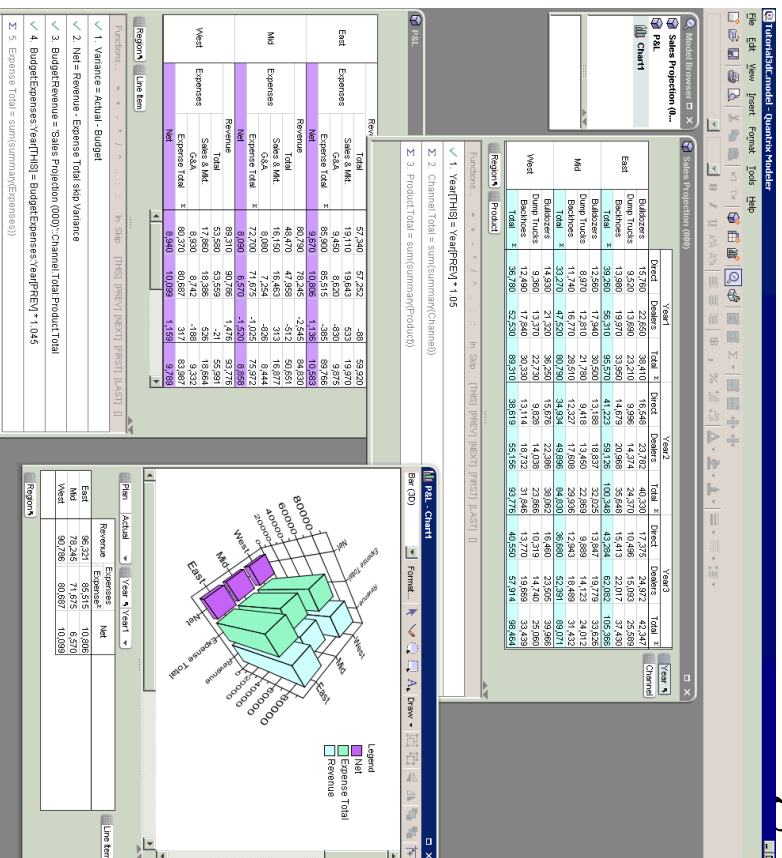
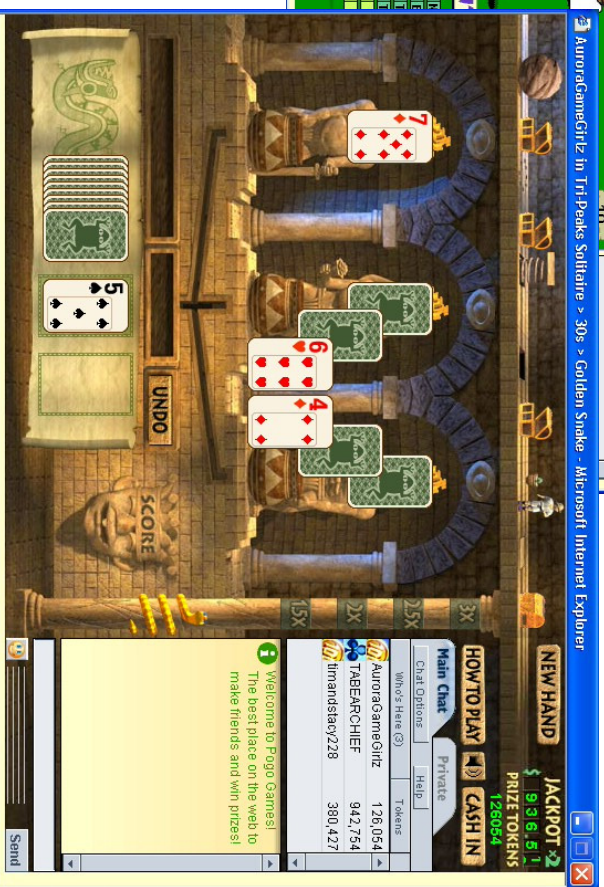


- Mathematical problem solver
- Interactive technical applications
- Utilisé dans 95% des universités
- Environs 3 millions d'utilisateurs

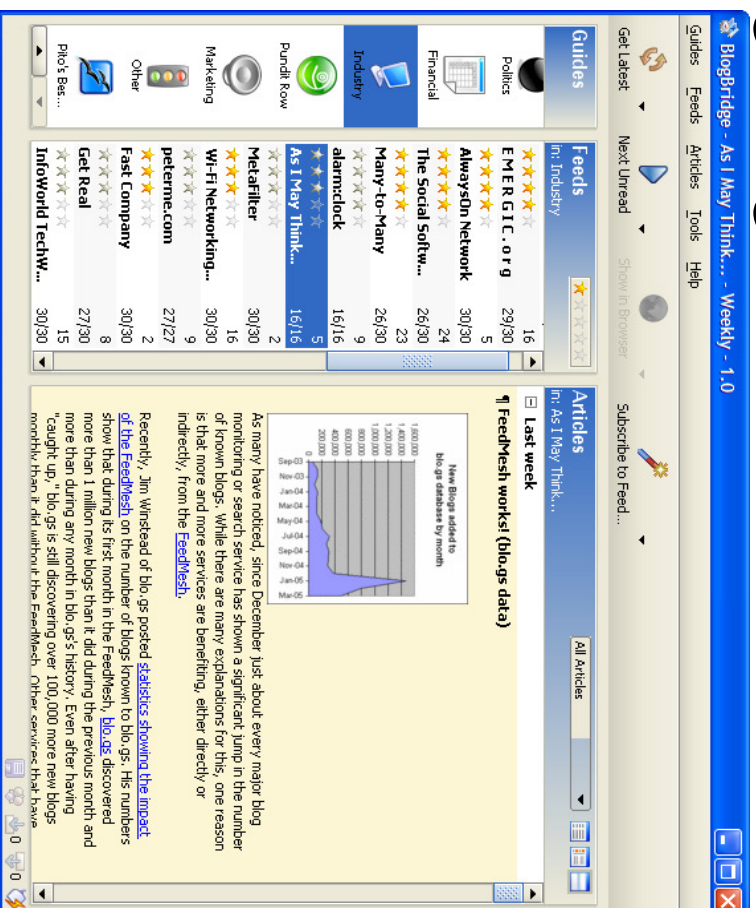
map24.com: Interactive mapplets



- Site de mapping par défaut pour Yahoo! Europe
- Jusqu'à 1 million d'applets Java délivrés chaque jour



BlogBridge: RSS feed reader



Performance

- Cours d'Histoire:

- > Les jours anciens (J2SE 1.2)

Nouvelle couche graphique (Java 2D, Swing)

+ runtime interprété

+ vitesse du matériel à cette époque

Performance peu satisfaisante

- > Depuis lors:

compilateur Hotspot (1.3)

+ accélération graphique matérielle (1.4)

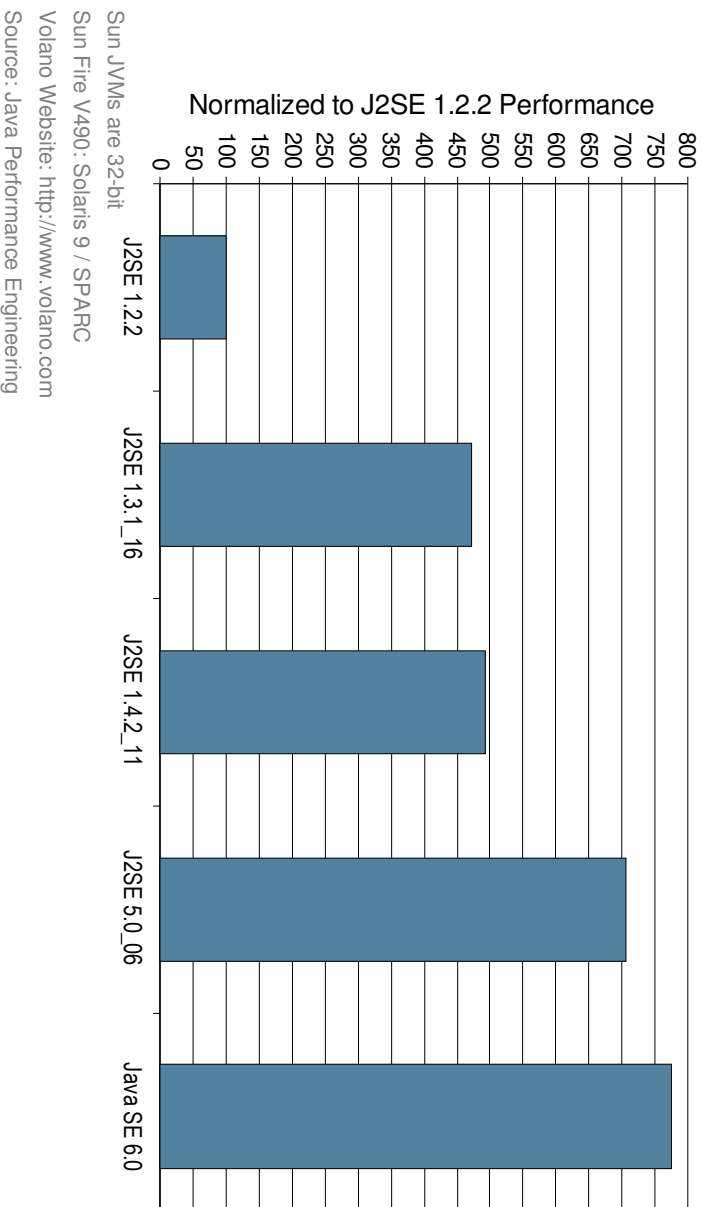
+ travail continu sur la performance (1.3, 1.4, 1.5, 6, ...)

+ matériel plus rapide

Performance excellente

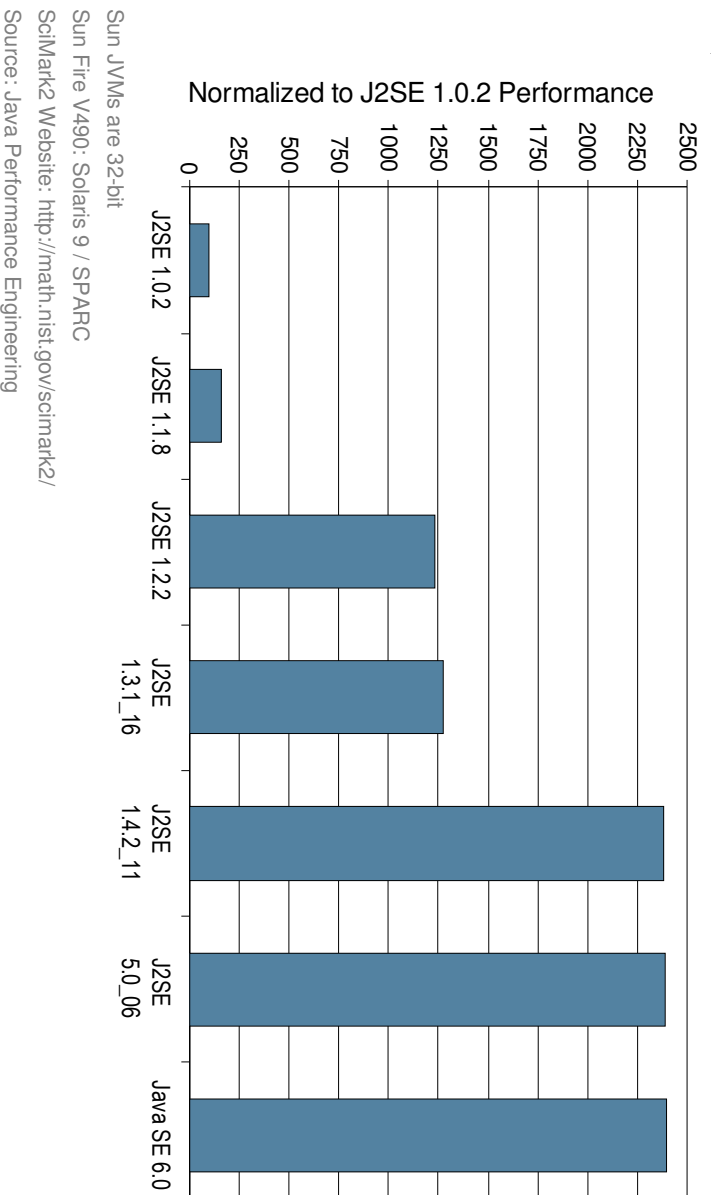
Histoire Performance Java SE

Server benchmark: Volano 2.5.0.9

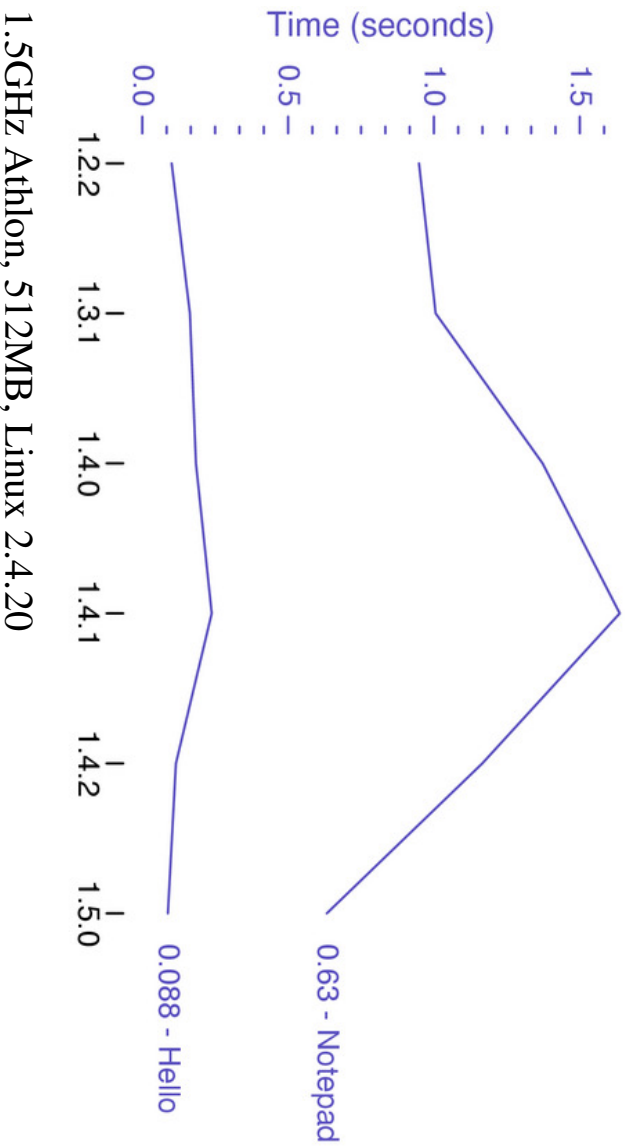


Histoire Performance Java SE

Server benchmark: SciMark2

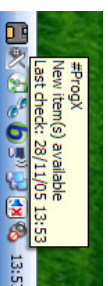
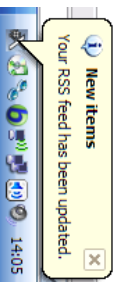


Histoire Performance Démarrage Java SE



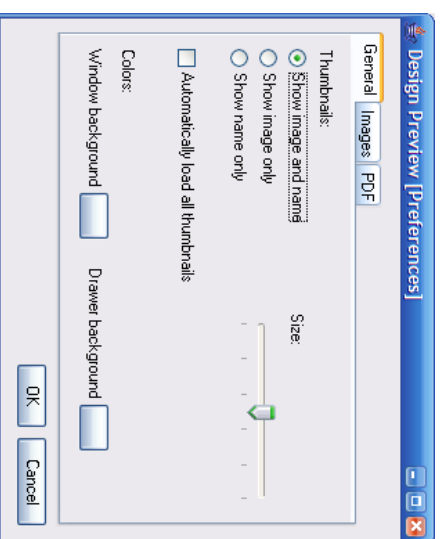
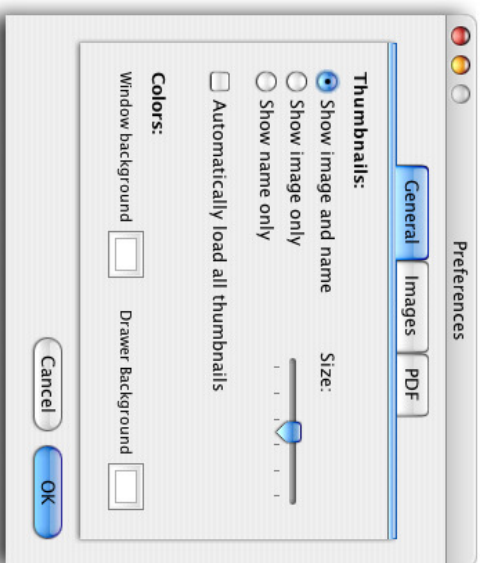
Intégration Desktop

- Les applications Java peuvent fonctionner et s'intégrer comme des applications natives:
 - > Icônes dans le menu démarrer, system tray (Windows)
 - > Dock integration (Macintosh)
 - > Splash screen au démarrage
 - > APIs pour s'intégrer dans les services de desktop standards (client mail par défaut, lecteurs documents, navigateurs, etc.)



Intégration Desktop : Devient Native

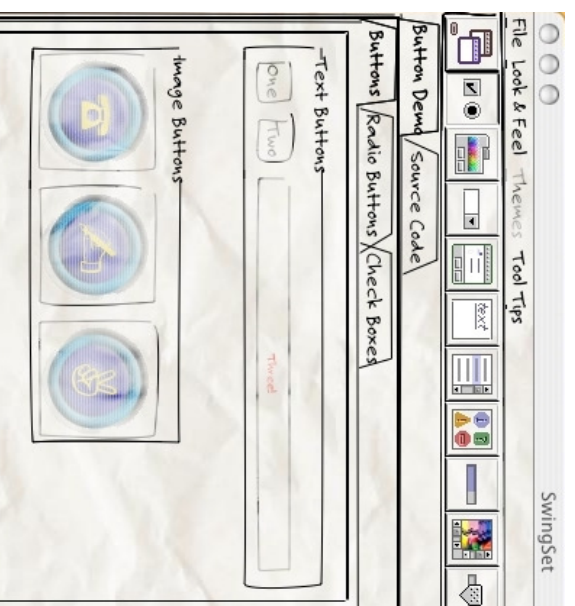
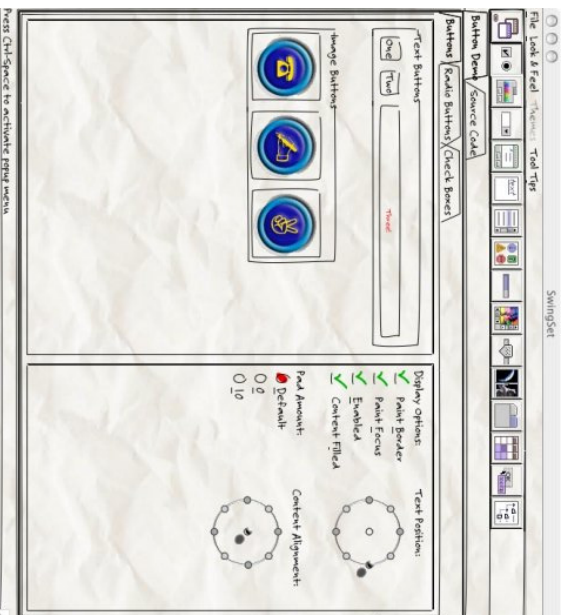
- Semble native quelque soit votre plateforme



Mais si vous devez la

~~petits boutons~~ petitsboutons.sourceforge.com:

> “semble comme si c’était élaboré sur le coin d’une nappe”



Déploiement d'Application

- 3 modèles de déploiement:
 - > Applet: s'exécute dans le conteneur du navigateur
 - > sécurisée (s'exécute dans le "bac à sable" de sécurité par défaut)
 - > UI familière du navigateur
 - > Application JavaWebStart:
 - > Se télécharge depuis le navigateur, s'exécute indépendamment
 - > Intégration avec le bureau
 - > sécurisée (s'exécute dans le "bac à sable" de sécurité par défaut)
 - > Mécanisme de mise à jour transparent
 - > Utilisation offline
 - > Application standalone:
 - > S'installe et s'exécute localement
 - > Aucune restriction de sécurité; comme une application native

JMX **BigDecimal** **z-ordering** **jstat**
RMI **dynamic proxies** **Generics** **Gnome** **Skins** **jps**
JDBC **Rowsets** **Autoboxing** **extended for loop**
faster startup **iconsole** **printf** **JVM sharing**
synth L&F **SAX 2.0** **Concurrency utilities**
Unicode **Surrogates** **importing constants** **OpenGL**
IP reachability **Ocean L&F** **SASL** **performance** **XAWT**
unsynchronized StringBuffer **apt** **DOM 3** **improved cookie support**
XML Schema **fatal error handlers** **JVM Monitoring**
Stack trace API **Remote JMX** **improved footprint**
varargs **swing printing** **AMD64** **Enumerated types**
XDnd **Packed JARs** **New Memory Model** **OCSP**
scanning **JVMPi Profiling**
performance ergonomics

Desktop Java dans Tiger

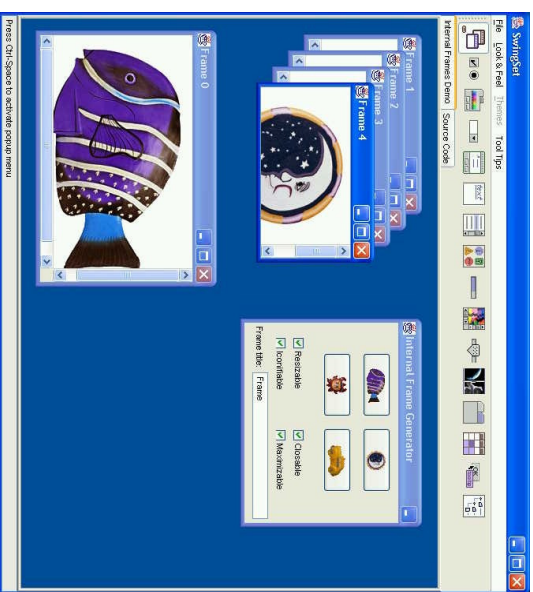
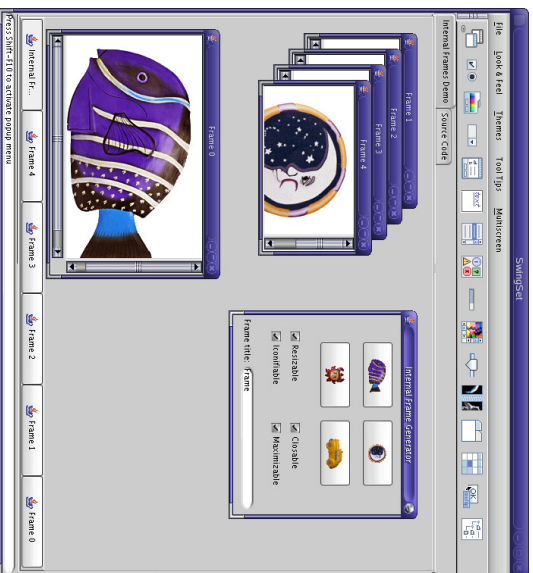
- Swing
 - > Look a Feels natif GTK, XP
 - > Thème Ocean pour Metal
 - > Synth
 - > JTable printing
 - > JFrame.add()
- AWT
 - > XAWT

Desktop Java dans Tiger

- Java 2D
 - > Managed Images Partout
 - > re-architecture des Polices
 - > Amélioration ImageIO
 - > Amélioration des Performances
- Déploiement
 - > Consolidation Plugin/Webstart
 - > Pack200
 - > Nombreuses amélioration webstart et plugin

Swing: Look & Feels

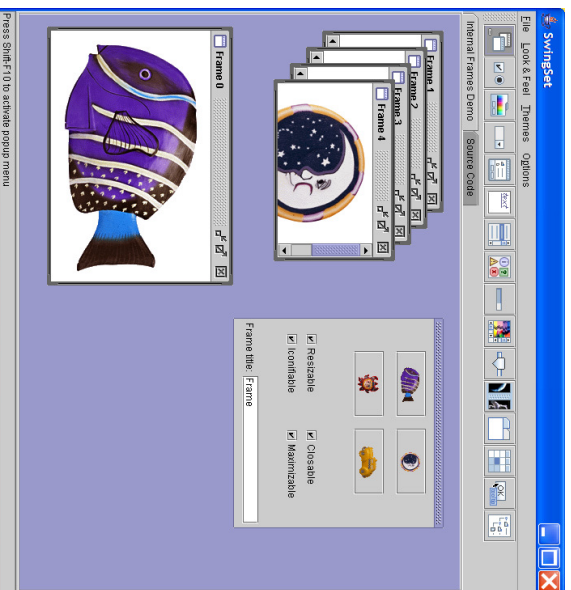
- Look & Feels Natifs: GTK et Windows XP



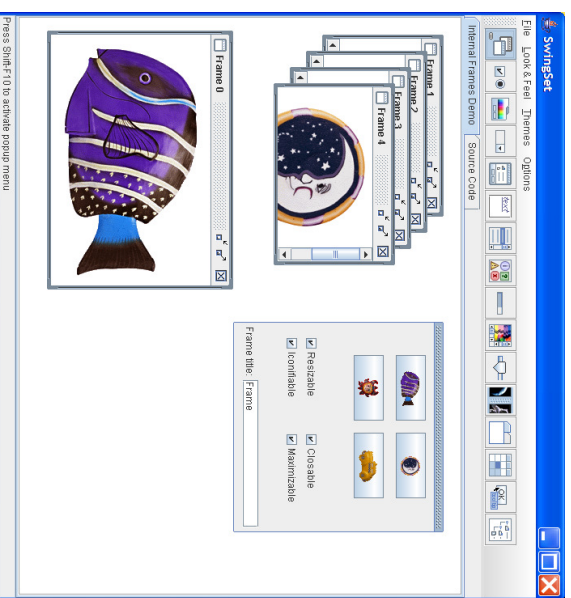
Swing: Look & Feels

- Nouveau Thème pour le L&F Multi-Plateforme

Ancien Thème “Steel”

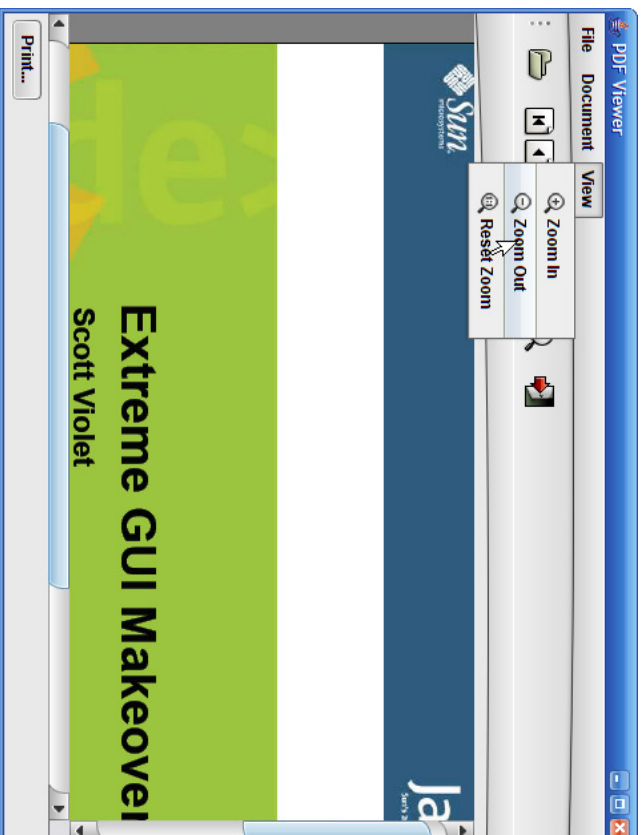


Nouveau thème “Ocean”



Swing: Look & Feels

- Look & Feel Synth



Compiler API

MBeans metadata
JTable upgrades
Longhorn Look & Feel

Splash screens

Split Verifier Windows system tray Unicode Normalizer Services

Parallelize Concurrent GC

Attach on CoreDump

chmod

Annotation properties parallel old-space GC

SwingWorker

JVM DTrace

Web Services Stack

Password prompting

Docs in JDBC 4.0

Javadoc Tags

LCD fonts
Free disk space

Chinese

More gfx acceleration

JVM & CLR Co-Existence

Improved OOM diagnosability
More desktop integration

Firefox support Pluggable Locales

Native L&F Fidelity

Scripting Languages

More GC Ergonomics

XAWT

HTTP cookie manager
JavaScript engine

XML digital signatures

Improved text rendering

Swing dans Mustang

- Plus de 
- Look & Feel Natif amélioré
 - > Meilleur sur les plateformes existantes, permet les plateformes futures (par exemple, Vista)
- Améliorations Layout
 - > Support Baseline
- Fixes au niveau Drag & Drop
- Tri & Filtrage JTable
- SwingWorker mis à jour et intégration

Swing dans Mustang

- Plus de rectangles glis
 - > double-buffering par fenêtre
- Look & Feel Natif amélioré
 - > Meilleur sur les plateformes existantes, permet les plateformes futures (par exemple, Vista)
- Améliorations Layout
 - > Support Baseline
- Fixes au niveau Drag & Drop
- Tri & Filtrage JTable
- SwingWorker mis à jour et intégration

Java 2D dans Mustang

- Texte LCD
 - > Utilisation de RVB des écrans LCD pour amélioration anti-aliasing du texte

Brazil
Brazil
Brazil

Java2D dans Mustang

- Qualité des Petits cercles
 - > Plus de “pneus plats”
- Rendu en un seul thread
 - > Évite la fragile interdépendance des threads
 - > Travaille mieux avec les drivers et matériels graphiques
- Meilleures performances
 - > amortizes API calls into larger batches of operations
 - > OpenGL (pas par défaut) rendering pipeline only this release



Mustang Snapshots

- Nouvelle version disponible chaque semaine
- **Binares** pour l'évaluation et les tests
 - > Par absolument n'importe qui
- **Sources** pour la collaboration de la communauté
 - > Par n'importe quel membre de la communauté JDK
 - > Requiert l'acceptation de la licence JRL

<http://mustang.dev.java.net>

Java Distribution License (JDL)



- Commercial redistribution license
 - > Principal successeur à la SCSL
 - > commercial use license simplifiée (plus lisible)
- Inclut les principaux termes commerciaux de la SCSL
 - > Requiert le passage du TCK
 - > Inclut la licence TCK
 - > Optional support agreement
 - > Requiert accord trademark
- Pas de surprises

Java Internal Use License (IUL)

Les Entreprises peuvent deployer leurs propres bugfixes EN INTERNE



- > Sun peut forcer la compatibilité mais ne requiert pas de passer le TCK
- > Honor system - "Aux propres risques de l'utilisateur"
- > Quelques recommandations: pas de modifications de l'API
- Pour usage interne, entre 4 murs
 - > Y compris les websites accessibles publiquement
 - > Exclut les distributions externes
- Permet le partage des bugfixes
 - > Encourage mais n'oblige pas à partager les fixes
- Paix de l'Esprit



Agenda

- Aperçu Desktop Java
- **Fondamentaux**
 - > Databinding
 - > SwingX/JDIC
 - > Projets tierces parties
 - > Effets graphiques Swing
- Outils
- Q&R

SwingLabs

- Laboratoire Open Source
- Exploration de nouvelles façons de créer des applications Swing
 - > Plus facilement
 - > Plus rapidement
 - > Prettier
- Projet Ombrelle
 - > SwingX
 - > Databinding
 - > Incubator

SwingLabs

Java™ Desktop Technology

Home Projects Screenshots Downloads Demos Documentation Contact Us

WHAT IS SWINGLABS?

SwingLabs is an Open Source laboratory for exploring new ways to make Swing applications easier to write, with improved performance and greater visual appeal. It is an umbrella project for various Open Source initiatives sponsored by Sun Microsystems and is part of the [java.net](#) community. Successful code and concepts may be migrated to future versions of the Java™ platform.

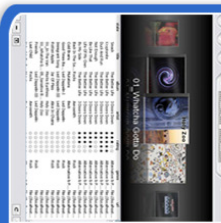
SwingLabs' SwingX provides a powerful suite of desktop components for all your Swing projects. Available under the LGPL license, the SwingX project is the perfect match for both your professional and personal projects.

Browse the [screenshot gallery](#) to discover the SwingLabs component set. Be sure to check out our [demos](#) to see what you can use in your application today!

The SwingLabs project is nearing its 1.0 release. We're working hard to stabilize the APIs and provide useful tutorials and documentation. We appreciate your help in making SwingLabs the premiere open source project for Java Desktop technologies!

Pick of the Month

JOOPLIN MP3 PLAYER



LOGIN PANEL

The SwingLabs JXLoginPanel provides a simplified Swing component for providing authentication in your rich client applications. It is simple window that defers to the classes in the [org.jdesktop.swingx.auth](#) package for providing the "guts" of the login procedure.

Need a login window for your application? Use the JXLoginPanel and in less than 3 lines of code provide a login dialog for your database applications. Feeling adventurous? Adapt a custom login dialog to use the SwingLabs Authentication Framework as the backend for custom login windows.



Read more about JXLoginPanel in the [JXLoginPanel Documentation](#), or check out the [JXLoginPanel API](#)

©2005 Sun Microsystems Inc. & Contributors

Java™ is a registered trademark of Sun Microsystems Inc.

Fundamental Teasers

- SwingLabs

SwingXDemo

Menu

Table of Contents

Extended Tables and Trees

Decorators

Task Panels

Tip Of The Day

Hyperlinks

Date Selection

More Date Selection

Autocomplete

Action Framework

Authentication Dialogs

Error Dialog

Drop Shadows

Glass Boxes

Translucency

Information :: JXTable/JXTreeTable

Read More...

With the SwingX JXTable and JXTreeTable controls your users can select displayed columns at runtime, rearrange columns as desired, highlight rows with plugable background highlighters, sort their data via detachable headers and more.

In this demo, try rearranging columns by dragging the column headers with your mouse. Select the columns shown in the table by clicking on the column control in the top right corner of the tables. Change background color highlighting using the highlighter combobox. Sort rows by clicking on the column headers, and change the sort characteristics using the Sorter radio buttons. Note that you can also re-size table columns automatically from the column control, as well as apply horizontal scrolling.

Demo

JXTable:

Type	Location	Size
MainWindow	(240, 100)	(800, 600)
JRootPane	(4, 25)	(792, 571)
JPanel	(0, 0)	(792, 571)
JLayeredPane	(0, 0)	(792, 571)
JMenuBar	(0, 0)	(792, 21)
JMenu	(0, 0)	(39, 19)
SwingDemo	(0, 21)	(792, 550)
JScrollPane	(6, 6)	(780, 538)
WindowScrollPane...	(200, 0)	(5, 538)
JXTreePanel	(0, 0)	(200, 538)
BasicTreePanelUI...	(8, 0)	(184, 22)
JLabel	(12, 4)	(160, 14)

Highlighter:

alienable linePrinter

▼

Sorter:

Lexical (default)

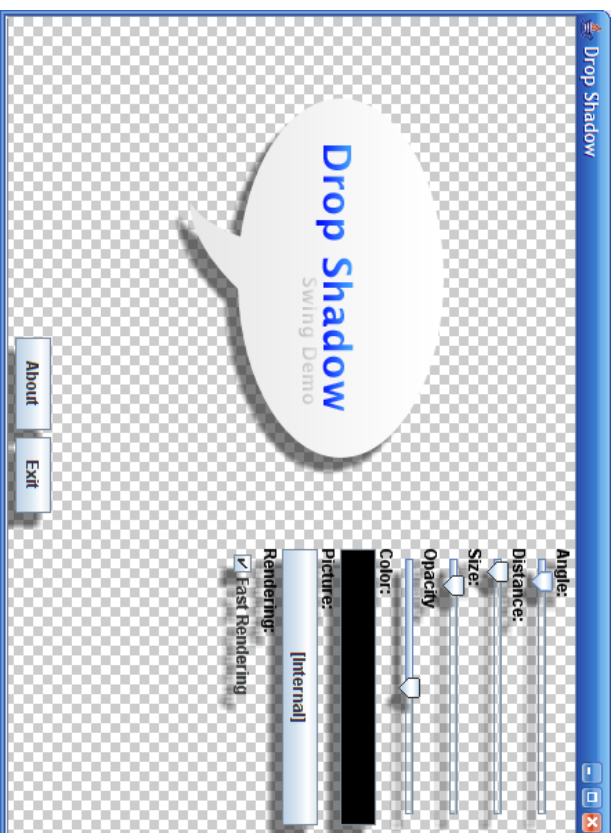
PerfDimension

JXTreeTable:

Type	Location	Size
MainWindow	(240, 100)	(800, 600)
JRootPane	(4, 25)	(792, 571)
JPanel	(0, 0)	(792, 571)
JLayeredPane	(0, 0)	(792, 571)
JMenuBar	(0, 0)	(792, 21)
JMenu	(0, 0)	(39, 19)
SwingDemo	(0, 21)	(792, 550)
JScrollPane	(6, 6)	(780, 538)
WindowScrollPane...	(200, 0)	(5, 538)
JXTreePanel	(0, 0)	(200, 538)
BasicTreePanelUI...	(8, 0)	(184, 22)
JLabel	(12, 4)	(160, 14)

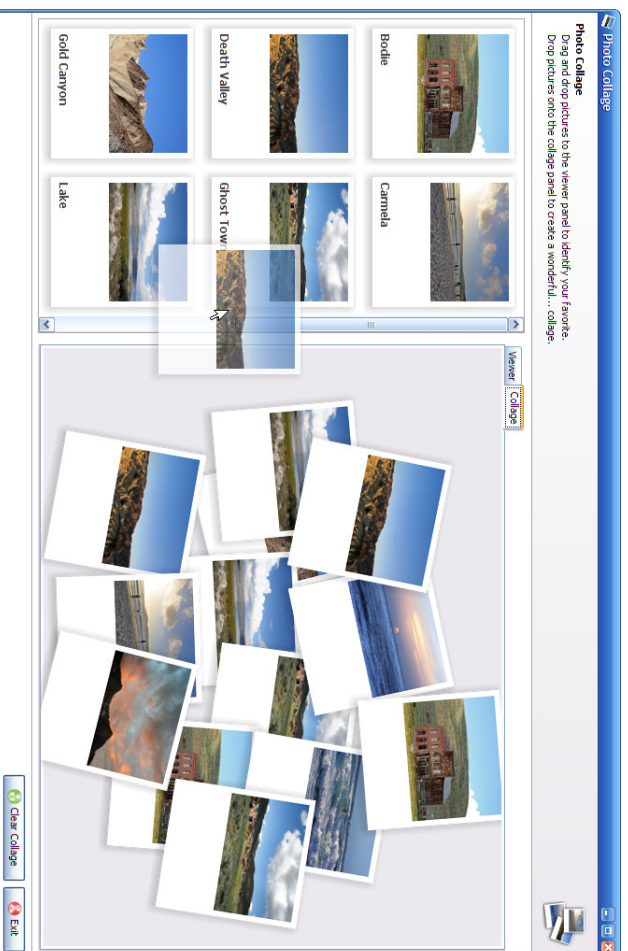
Fundamental Teasers

- Drop Shadow



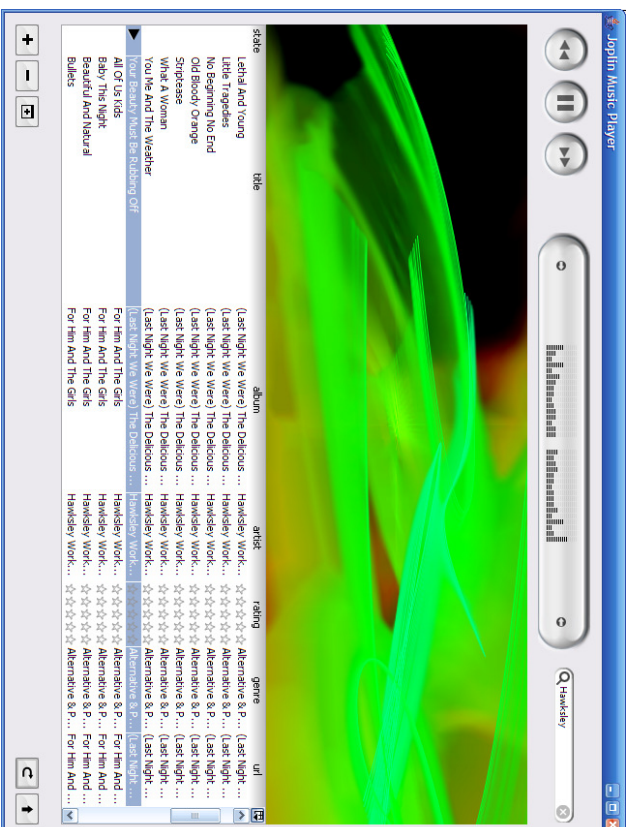
Fundamental Teasers

- Collage



Fundamental Teasers

- Joplin



Agenda Fondamentaux

FX

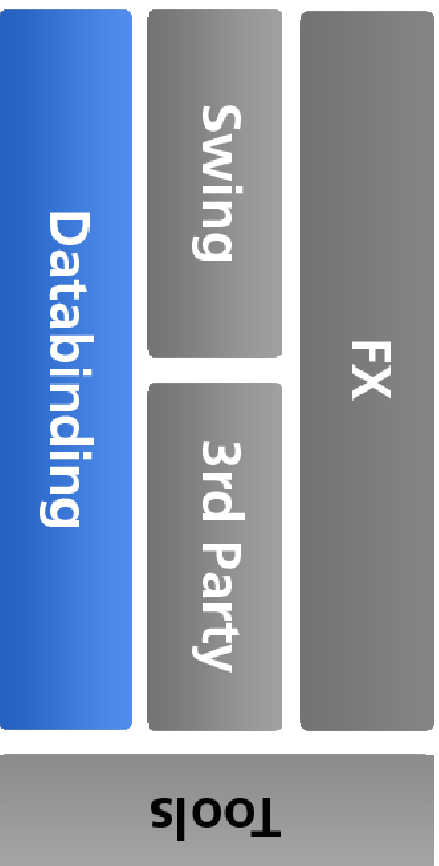
Swing

3rd Party

Databinding

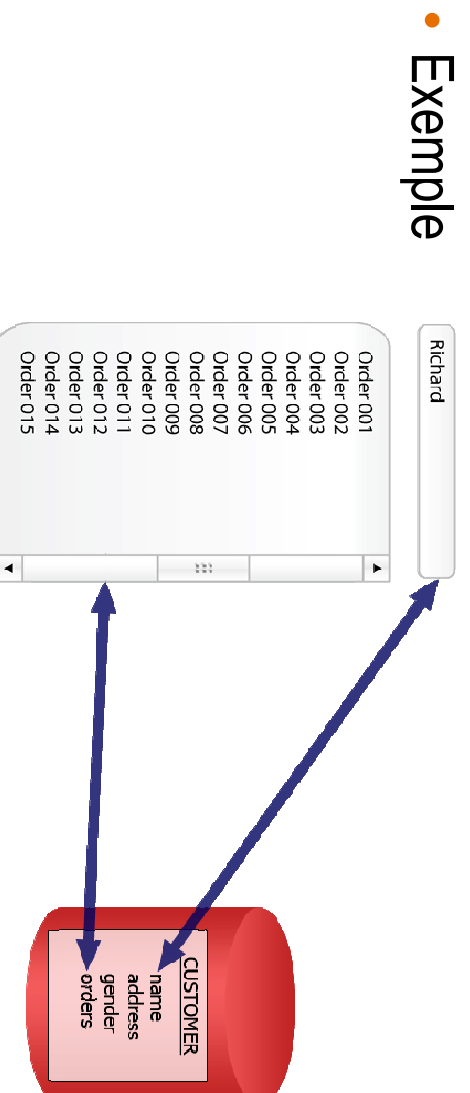
Tools

Agenda Fondamentaux



Databinding

- Définition
 - > Exposition automatique des modèles de données dans un interface utilisateur. La valeur du modèle et son GUI restent synchronisées.



Databinding: Tour de table

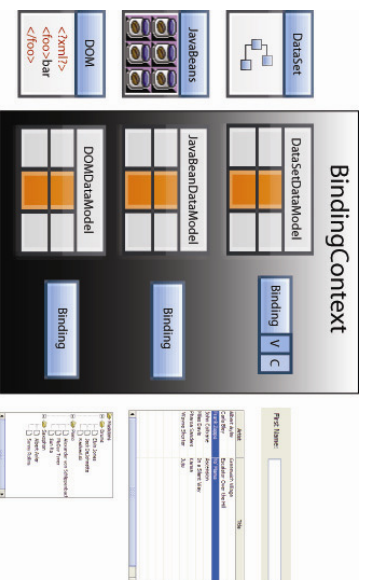
- Borland VCL (Delphi/C++ Builder)
- MS WinForms/ASP.NET (Visual Studio .NET)
- Braveheart (Java Studio Creator 2)
- JFace Data Binding (Eclipse v3.2)
- dbSwing/DataExpress (JBuilder)
- JGoodies
- Oracle ADF (JDeveloper)
- SwingLabs (NetBeans...)

Buts Design SwingLabs Binding

- Buts Principaux
 - > Simplification conception GUIs orienté données
 - > Par exemple, rendre plus trivial des front-ends de base de données de style “master/detail”
 - > Support pour les Outils
 - > Support Validation
 - > Travail avec de nombreux toolkits/couche (web vs client)
- Buts secondaires
 - > Extensible
 - > Testable

SwingLabs Binding 101

- Domain Data
 - > SwingLabs DataSet, JavaBeans, DOM
- Composants GUI
- “colle” Binding



Requirements des DataModels

- Encapsuler des structures de donnée arbitraires
 - > API commun pour tous les types de données
 - > Simplifier le travail des auteurs de composants “data-aware”
- > API vous permet d’accéder aux données sous-jacentes
- Notifier l’UI lorsque les données ont changées
- Les Modèles de données représentent
 - > Des tables (plusieurs colonnes, et plusieurs lignes)
 - > Des listes (une seule colonne, avec plusieurs lignes)
 - > Des records (plusieurs colonnes, et une seule ligne)
 - > Des scalaires (une seule colonne et une seule ligne)

Modèle de Données (Hiérarchie)

- Modèles de Données peuvent être imbriqués
 - > 0 ou 1 parent
 - > 0 ou plusieurs enfants
- Un enfant enveloppe le contenu d'une seule cellule

custid	age	address
1	23	com.myapp.Address@621653
2	37	com.myapp.Address@621654
3	26	com.myapp.Address@627643
4	15	com.myapp.Address@892932

street	city	zip code
123 Somewhl	Santa Aralc	95054

Requirements “Colle” Binding

- Adapter un Composant UI à un Modèle de Données
 - > Remplace le(s) modèle(s) des Composants là où c'est approprié
 - > Restaure le Composant à son état d'origine lorsque unbound
- Permet de nouvelles implémentations de *Binding*
 - > La plupart des applications ont besoin de composants personnalisés, requérant des bindings personnalisés
- Fournit des *hooks* pour la validation

Exemple “Colle” Binding

```
File f = new File("/usr/home");
JavaBeanDataModel dm = new JavaBeanDataModel(f);
JTextField tf = new JTextField();
JTextFieldBinding tfBinding = new JTextFieldBinding(tf, "name");
tfBinding.setDataModel(dm);

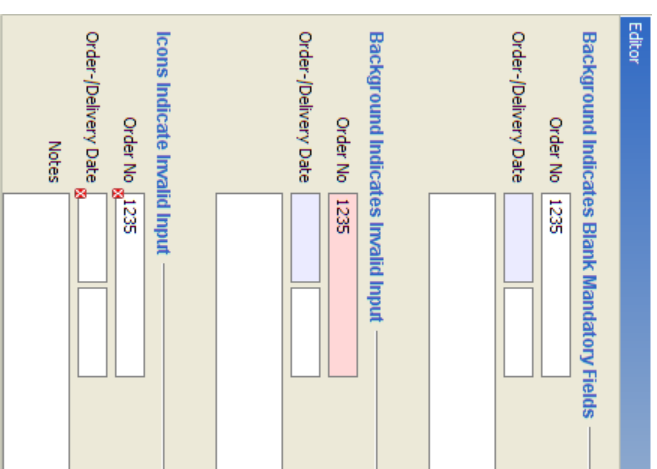
dm = new JavaBeanDataModel(f.listFiles());
JTable table = new JTable();
JTableBinding tBinding = new JTableBinding(table);
tBinding.setColumnNames("name", "path", "hidden");
tBinding.setDataModel(dm);
```

Requirements pour la Validation

- Flexible
 - > Pattern Observer
 - > Personnalisation de l’affichage des messages et erreurs
- Feedback riche
 - > Plusieurs messages de validation peuvent être générés depuis un seul événement de validation
- Plug into Binding framework
 - > Bindings va générer les événements de Validation automatiquement

Validation: GUI

- Différents moments de validation
 - > Touche pressée
 - > Focus perdu
 - > Commit
- Nombreux feedback visuels
- Marqueurs “Obligatoire”
- Conseils de Validation
 - > Utiles pour les humains



Editor

Background Indicates Blank Mandatory Fields

Order No

Order /Delivery Date

Background Indicates Invalid Input

Order No

Order /Delivery Date

Icons indicate Invalid Input

Order No

Order /Delivery Date

Notes

Validation

- Basé sur la Validation de JGoodies
- Bonne API, implémentation très utile
- Apache Jakarta Commons Validation
 - > Pas une option
 - > Trop orienté web
 - > Manque du feedback visuel
- > Pleins d'implémentations courantes très utiles
 - > EmailValidator
 - > URLValidator
 - > etc

Exemple Validation

```
Validator validator = new EmailValidator();//our impl
ValidationResult result = validator.validate("bad@email");
for (ValidationMessage msg : result.getMessages()) {
    System.out.println(msg.getSeverity() + " : ");
    System.out.println(msg.formattedText());
}

//a fictional component for displaying results
JXValidationPanel pnl = getValidationPanel();
pnl.setResult(result);
```

SwingX: Composants *Data Aware*

- SwingX fournit des composants *data aware*
- SwingX fournit des implémentation BindingContext
 - > JXFrame, JXDialog, JXRootPane
 - > Agissent comme des conteneurs de données
 - > Meilleur support par les outils
- Propriété dataPath spécifie la donnée à lier
 - > "cust.firstName"
 - > "cust[@selected].orders[@sel_orders].number"
 - > "package[@selected].packageActivtyList.activityList"

Exemple: Composants SwingX

```
public class MyPanel extends JPanel {
    BindingContext ctx;

    public void initComponents() {
        ctx = new BindingContext();
        newslst = new org.jdesktop.swingx.JXList();
        newslst.setDatapath("news");
        newslst.setDisplayName("subject");
        newslst.setBindingContext(ctx);

        bodyArea = new org.jdesktop.swingx.JXTextArea();
        bodyArea.setDatapath("news[@selected].body");
        bodyArea.setBindingContext(ctx);
        //...
    }

    public void bindComponents() {
        List<News> news = ...;
        ctx.addDomainData("news", news);
    }
}
```

Exemple: Composants SwingX

```
public class MyPanel extends JPanel {
    BindingContext ctx;

    public void initComponents() {
        ctx = new BindingContext();
        newslst = new org.jdesktop.swingx.JXList();
        newslst.setDatapath("news");
        newslst.setDisplayName("subject");
        newslst.setBindingContext(ctx);

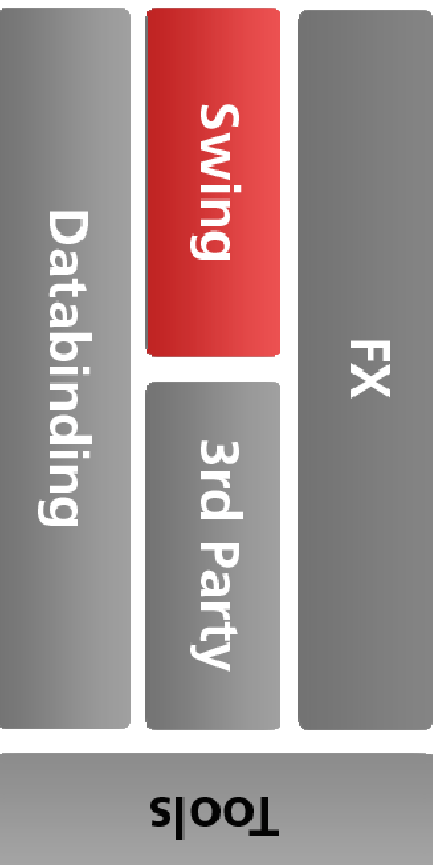
        bodyArea = new org.jdesktop.swingx.JXTextArea();
        bodyArea.setDatapath("news[@selected].body");
        bodyArea.setBindingContext(ctx);
        //...
    }

    public void bindComponents() {
        List<News> news = ...;
        ctx.addDomainData("news", news);
    }
}
```

Exemple: Composants SwingX

```
public class MyPanel extends JPanel {  
    public void initComponents() {  
        newList = new org.jdesktop.swingx.JXList();  
        newList.setDatapath("news");  
        newList.setDisplayName("subject");  
  
        bodyArea = new org.jdesktop.swingx.JXTextArea();  
        bodyArea.setDatapath("news[@selected].body");  
        //...  
    }  
  
    public void bindComponents() {  
        List<News> news = ...;  
        addDomainData("news", news);  
    }  
}
```

Agenda Fondamentaux



SwingX – Swing Extensions

- Composants Swing Améliorés
 - > Table/List highlighting, tri, et filtrage
 - > *Data aware*
 - > Auto completion
- Nouveaux Composants SwingX
 - > Conçu au dessus de Swing
 - > JXTreeTable, JXImagePanel, JXDatePicker
- Candidats possibles pour migrer vers le JDK
- Beaucoup de contributions d'autres projets
 - > L2FPROD, JGA, JGoodies

SwingX: JXTable

- Étends JTable pour fournir :
 - > Tri
 - > Filtre
 - > Mise en évidence (*Highlighting*) conditionnel
 - > Recherche
 - > Contrôle visibilité colonne
- Tri & Filtre ont migré vers Java 6.0

date	subject	author	replies
1098551943	PowerBook	Glx	95
1074288240	#ProgX Party	RICO	90
1110231617	ADSL Max chez Free	RICO	69
1098695910	PowerBook/OSX	Glx	47
1106754697	Diag n' Ghost	Glx	45
1077759720	Le son c'est bon (pour le moral) (éèèèèè moralééééèè)	RICO	43
1104704873	#ProgX en podcast	Glx	39
1110246822	Forsaken	Glx	36
1083096638	Looking Glass pour Windows	Glx	34
1101838627	24, Saison 4	Glx	34
11105973146	Vous repreniez bien un peu de GUI ?	Glx	33
1099317267	Bon anniversaire Rico !	Glx	32
1086229400	Suprise	Glx	31
1098350070	Quelle extension de fichier êtes-vous ?	Glx	31
1107291578	Login de février	Glx	30
1070363280	T610 Brotherz	RICO	29
1105474382	Apple Strikes Back	Glx	29
1069737400	Réactions	RICO	28
1068713880	Merci DVH	RICO	28
1079651433	RiseOfMilled 2.0 pour Winamp, iTunes et XMMS	Glx	27
1093480317	Projet Waterfool, jour 1	Glx	27
1109853609	Nintendo DS	Glx	26
1118006567	Des livres et des pages !	Glx	26
1068073900	The Matrix - Révolutions	Glx	25
1073763300	Mensonges, altérisme et calbal ...	RICO	25

```
newTable = new org.jdesktop.swingx.JTable();
newTable.setColumnControlVisible(true);
newTable.setShowHorizontalLines(false);
newTable.setShowVerticalLines(false);
newTable.setHighlighters(
    new HighlighterPipeline(
        new Highlighter[] {
            AlternateRowHighlighter.class,
            AlternateColumnHighlighter.class
        }
    )
);
newTable.packAll();
```

SwingX: JXTreeTable

- Un JTree avec plusieurs colonnes
- Display hierarchical multi property values
- Exemples Simples
 - > Courrier Entrant Email (threaded message view)
 - > Organizational Chart
- Tout comme JXTable, supporte
 - > Filtre
 - > Mise en évidence (*Highlighting*)
 - > Recherche

SwingX: Copie Écran JXTreeTable

JXTreeTable:

	Type	Location	Size
MainWindow		(-301, 115)	(1359, 634)
▼ JRootPane		(0, 22)	(1359, 612)
JPanel		(0, 0)	(1359, 612)
JLayeredPane		(0, 0)	(1359, 612)
JMenuBar		(0, 0)	(1359, 23)
JMenu		(0, 0)	(48, 22)
SwingXDemo		(0, 23)	(1359, 589)
JSplitPane		(12, 12)	(1336, 560)
AquaSplitPaneDivider		(194, 0)	(9, 560)
JSplitPane		(203, 0)	(1133, 560)
AquaSplitPaneDivider		(0, 107)	(1133, 9)
JXTitledPanel		(0, 0)	(1133, 107)
BasicTitledPanelUI\$J...		(8, 0)	(1117, 24)
JXPanel		(8, 24)	(1117, 75)
JScrollPane		(0, 0)	(1117, 75)

SwingX: Code Exemple

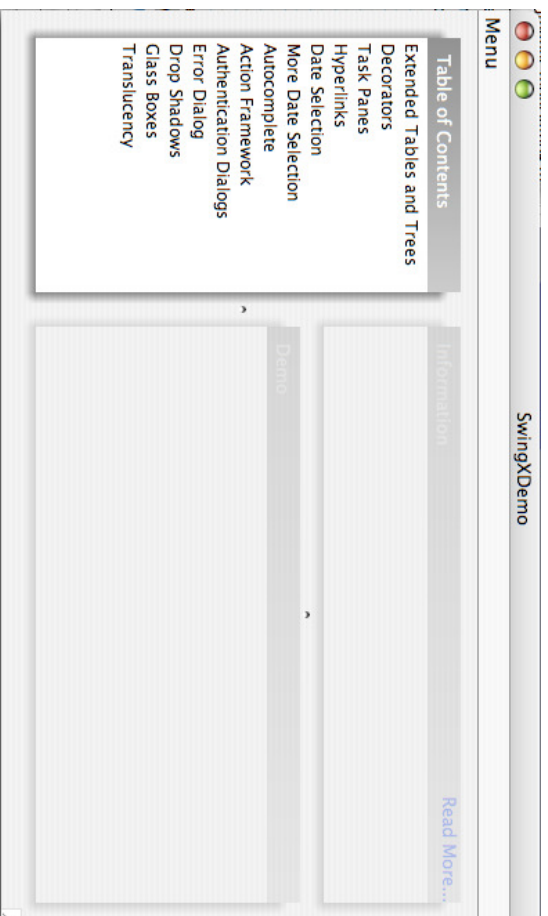
`JXTreeTable` `treeTableHeader().setToolTipText("<html>"`

```
        "<b> Sorting: </b><p> " +
        "Not supported <p>" +
        "<b> Auto-Resizing: </b><p>" +
        "Double-Click in resize region" +
        "</html>");
treeTable.setRootVisible(true);
treeTable.setShowsRootHandles(false);
treeTable.setHighlighters(
    new HighlighterPipeline(
        new Highlighter[] {
            AlternateRowHighlighter.classicLinePrinter
        }
    )
);
treeTable.packAll();
```

SwingX: JXPanel

- Fournit API translucide
 - > Tous les composants d'un JXPanel héritent de la translucidité
 - > JXPanel peut surcharger la translucidité d'un JXPanel parent
 - > Un bon remplaçant pour JPanel
- Également d'autres sous-classes utiles
 - > JXImagePanel
 - > JXLoginPanel
 - > JXC collapsiblePanel
 - > JXTitledPanel
 - > JXGlassBox

SwingX: Copie Écran JXPanel



SwingX: Copie Écran JXPanel



SwingX: Code Exemple JXPanel

```
Border shadowBorder = /* shareable border */
    new DropShadowBorder(UIManager.getColor("Control"),
        0, 8, .5f, 12, false, true, true, true);

JXtitledPanel toc = new JXtitledPanel("Table of Contents");
JXtitledPanel info = new JXtitledPanel("Information");
JXtitledPanel demo = new JXtitledPanel("Demo");

toc.setBorder(shadowBorder);
info.setBorder(shadowBorder);
demoPanel.setBorder(shadowBorder);

// Set transparency to 30% until user clicks on TOC
info.setAlpha(.3f);
demo.setAlpha(.3f);
```

SwingX: Copie Écran JXImagePanel



SwingX: JXImagePanel Code Sample

```
imagePanel = new org.desktop.swingx.JXImagePanel();  
try {  
    Image image = ImageIO.read(  
        getClass().getResource("aerith_logo.png"));  
    imagePanel.setImage(image);  
} catch (IOException ex) {  
}  
}
```

SwingX: JXTitledPanel



SwingX: JXTitledPanel

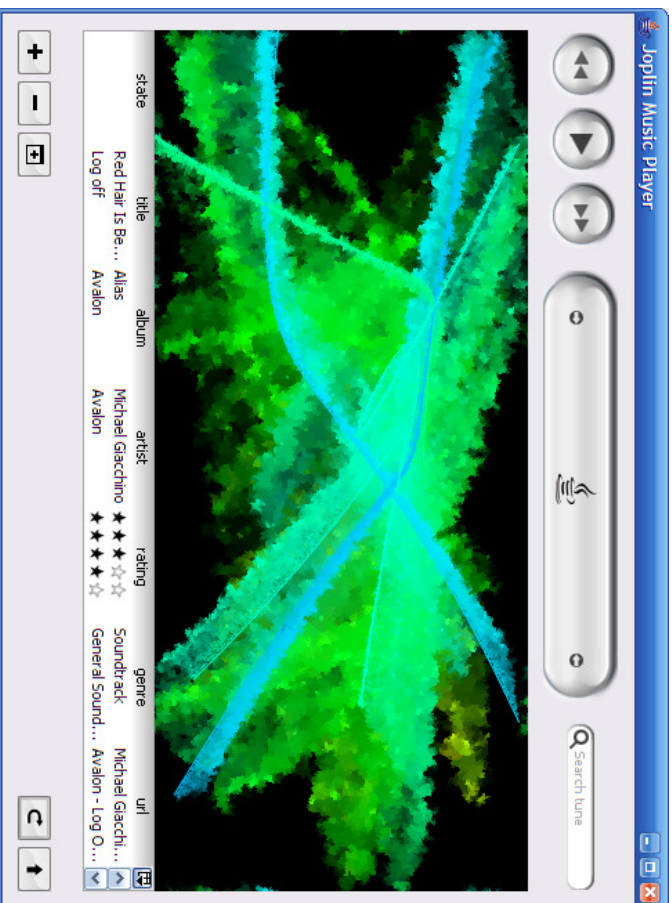
```
JXTitledPanel toc = new JXTitledPanel("Table of Contents");
JXTitledPanel info = new JXTitledPanel("Information");
JXTitledPanel demo = new JXTitledPanel("Demo");

JXPanel content = new JXPanel();
//Configure the content panel here. Point is, this
//is just a Container
//...
demo.setContentContainer(content);
```

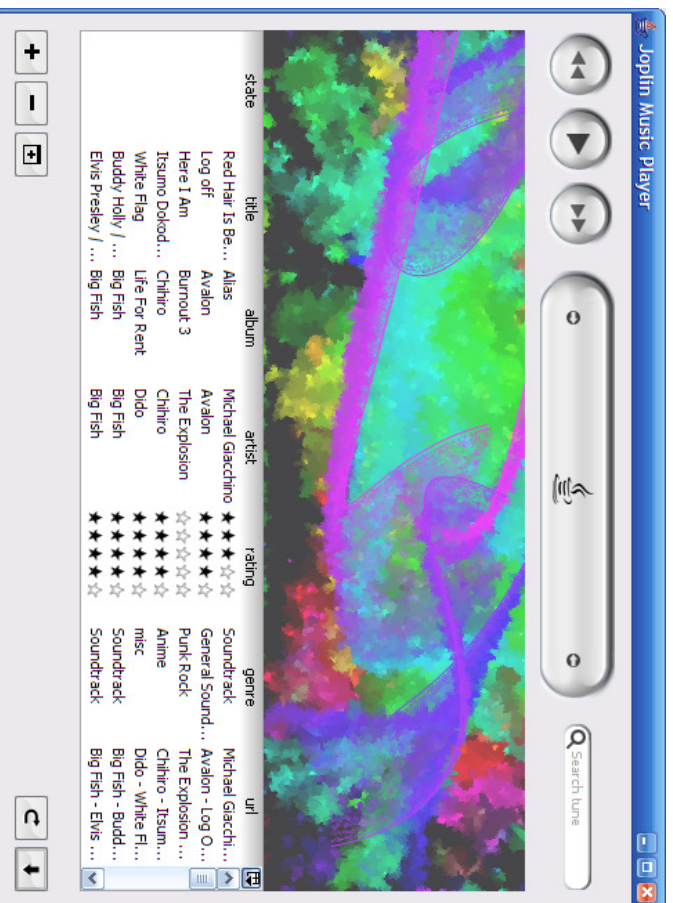
SwingX: JXCCollapsiblePane

- Une extension de JPanel qui peut étendre ou refermer son contenu avec une animation et des effets de fade in/fade out.
- Utilisé par JXTaskPane
- JXCCollapsiblePane a une action par défaut enregistrée sous le nom de TOGGLE_ACTION

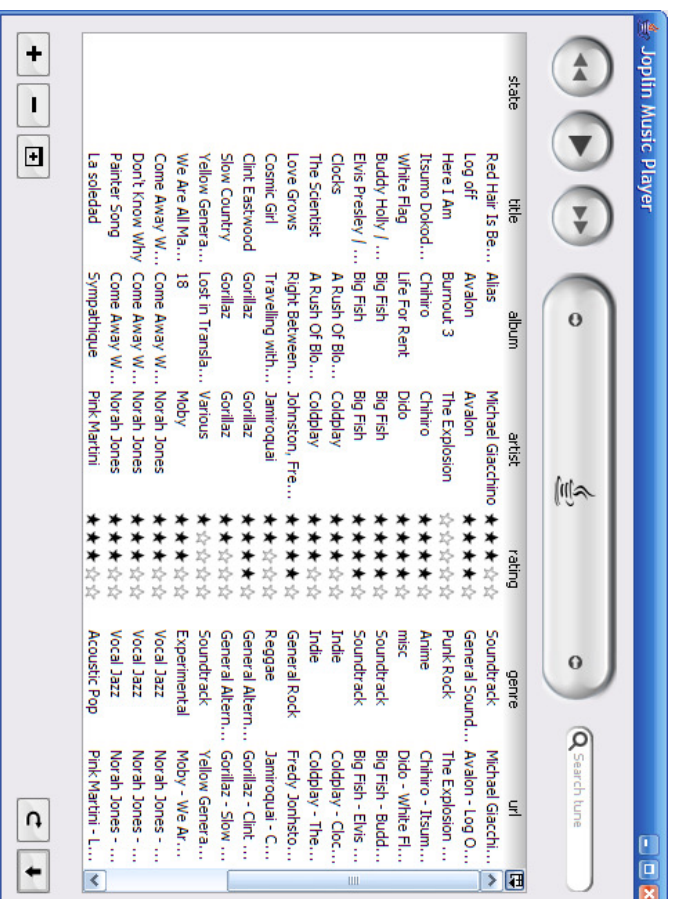
SwingX: JXCollapsePane



SwingX: JXCollapsePane



SwingX: JXCollapsiblePane



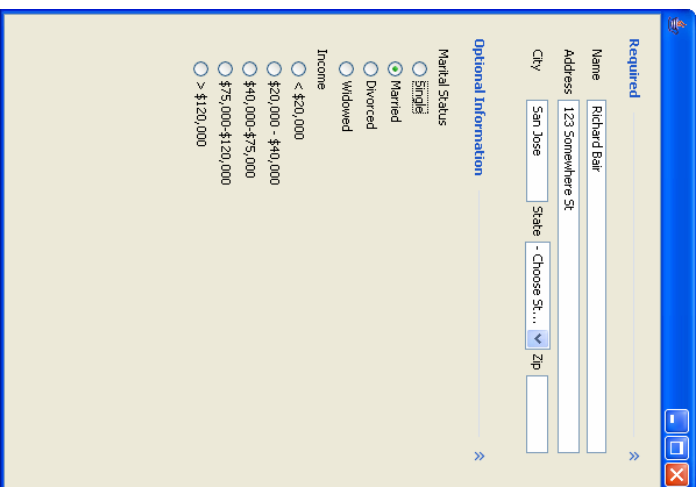
SwingX: JXCollapsiblePane (code)

```
JXCollapsiblePane pane = new JXCollapsiblePane();  
ActionMap map = pane.getActionMap();  
Action a = map.get(JXCollapsiblePane.TOGGLE_ACTION);  
JButton button = new JButton(a);
```

SwingX: JXTaskPane

- Conteneur pour des tâches (représentés par des JXHyperlinks) ou d'autres composants arbitraires
- Habituellement regroupés dans un JXTaskPaneContainer
- Conçu en collaboration avec Frédéric Lavigne (L2FPROD)

SwingX: Copie Écran JXTaskPane



Required

Name: Richard Blair

Address: 123 Somewhere St

City: San Jose State: - Choose St... ZIP:

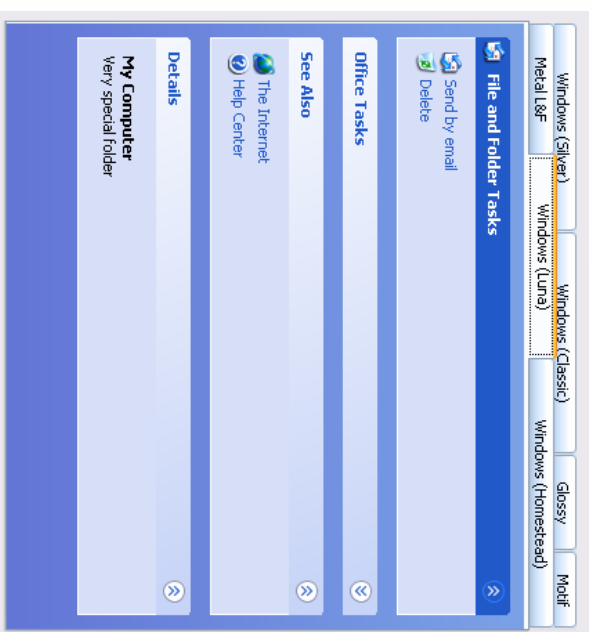
Optional Information

Marital Status:

- ☒ Single
- ☐ Married
- ☐ Divorced
- ☐ Widowed

Income:

- ☐ < \$20,000
- ☐ \$20,000 - \$40,000
- ☐ \$40,000-\$75,000
- ☐ \$75,000-\$120,000
- ☐ > \$120,000



SwingX: JXTaskPane (code)

```
// a container to put all JXTaskPane together
JXTaskPaneContainer taskPaneContainer = new
JXTaskPaneContainer();

// create a first taskPane with common actions
JXTaskPane actionPane = new JXTaskPane();
actionPane.setTitle("Files and Folders");
actionPane.setSpecial(true);

// actions can be added, an hyperlink will be created
Action renameSelectedFile = createRenameFileAction();
actionPane.add(renameSelectedFile);
actionPane.add(createDeleteFileAction());

// add this taskPane to the taskPaneContainer
taskPaneContainer.add(actionPane);
```

SwingLabs Incubator

- Place pour partager du code prototype en relation avec SwingLabs
- Tous les membres de la communauté qui a signé le JCA peut recevoir l'accès en écriture (commit)
- Lorsque l'idée est mature, elle peut être "promouvée"
- Incubateur
 - > JIconFile
 - > DropShadowPanel & ShadowFactory
 - > JComponentComboBox
- <http://jinc-incubator.dev.java.net>

DEMO

Drop shadow Panel

Incubateur: DropShadowPanel

(code)

```
dropShadowPanel = new DropShadowPanel(new BorderLayout());  
dropShadowPanel.setDistance(10);  
dropShadowPanel.setAngle(45.0f);  
dropShadowPanel.add(picture);
```

```
JPanel buttons = new JPanel();  
buttons.setOpaque(false);  
buttons.add(new JButton("About"));  
buttons.add(new JButton("Exit"));  
dropShadowPanel.add(buttons, BorderLayout.SOUTH);
```

```
JPanel debugPane = buildDebugPane();  
debugPane.setOpaque(false);  
dropShadowPanel.add(debugPane, BorderLayout.EAST);
```

SwingLabs: JDIC

- JDesktop Integration Components
- Sous-projet dédié à l'intégration native
 - > WebBrowser
 - > TrayIcon
- Inclut incubateur avec des composants expérimentaux
 - > Icon Service
 - > FileUtil
 - > getFreeSpace(File file)
 - > recycle(File file)
 - > Music Player Control API

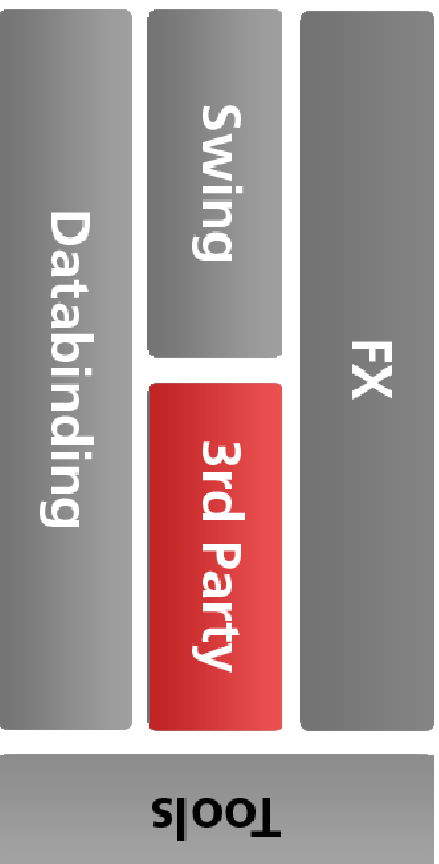
DEMO

JDIC WebBrowser

JDIC: WebBrowser (code)

```
public BrowserWindow() {  
    initComponents();  
    try {  
        browser = new WebBrowser();  
        browser.setURL(new URL("http://swinglabs.org"));  
    } catch (MalformedURLException ex) {  
        ex.printStackTrace();  
    }  
    browserPanel.add(browser);  
    getRootPane().setDefaultButton(goButton);  
}
```

Agenda Fondamentaux



Vous ne pourrez plus pirater les logiciels Sun!

- Java Enterprise System
 - > Suite complète de middleware
 - > Intégré et multi-plate-formes
- Vente a l'employé
 - > Vente à la CPU disponible
- Accès libre au logiciel, y compris en production, sans support de tout Java ES
 - > Si besoin de support, retour au cas ci-dessus

Roadmap Java

- J2SE 5.0 (Tiger) : sept 2004
- Java SE 6 (Mustang) : printemps 2006
- Java EE 5 : printemps 2006
 - > EJB 3 (POJO + persistence)
 - > JSF 1.2

<http://www.jcp.org>

GlassFish

- Le serveur Java EE 5 Open Source de Sun
- Disponible aujourd'hui (quasi toutes les fonctionnalités)
- Pour apprendre Java EE 5 (blueprints)
- Innovations
 - > Grizzly, Managenet WS, Toplink, outils admin, load on-demand
- Outils disponibles aujourd'hui (NetBeans, ...)

- <http://glassfish.dev.java.net>

Outils de développement

- GRATUITS
- NetBeans.org (Open Source)
- Sun Java Studio Creator 1 & 2 (JSF RAD)
- Sun Java Studio Enterprise 8 (UML)
- Sun Studio 11

<http://developer.sun.com>

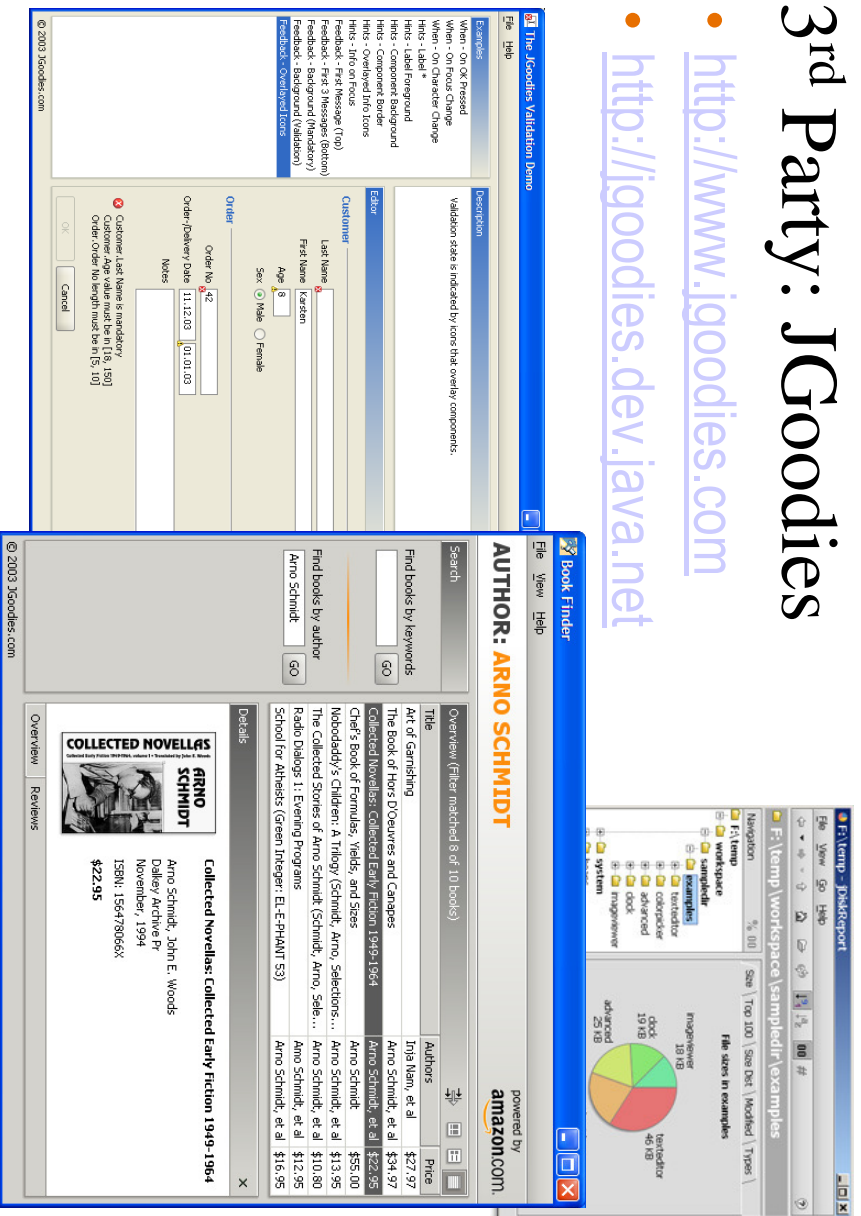
Bibliothèques Tièrces Parties Très

UJFreebies

- > Looks
- > Forms
- > Binding
- > Validation
- > Animation
- JFreeChart
- JFreeReport & JasperReport
- ... et pleins d'autres encore \o/

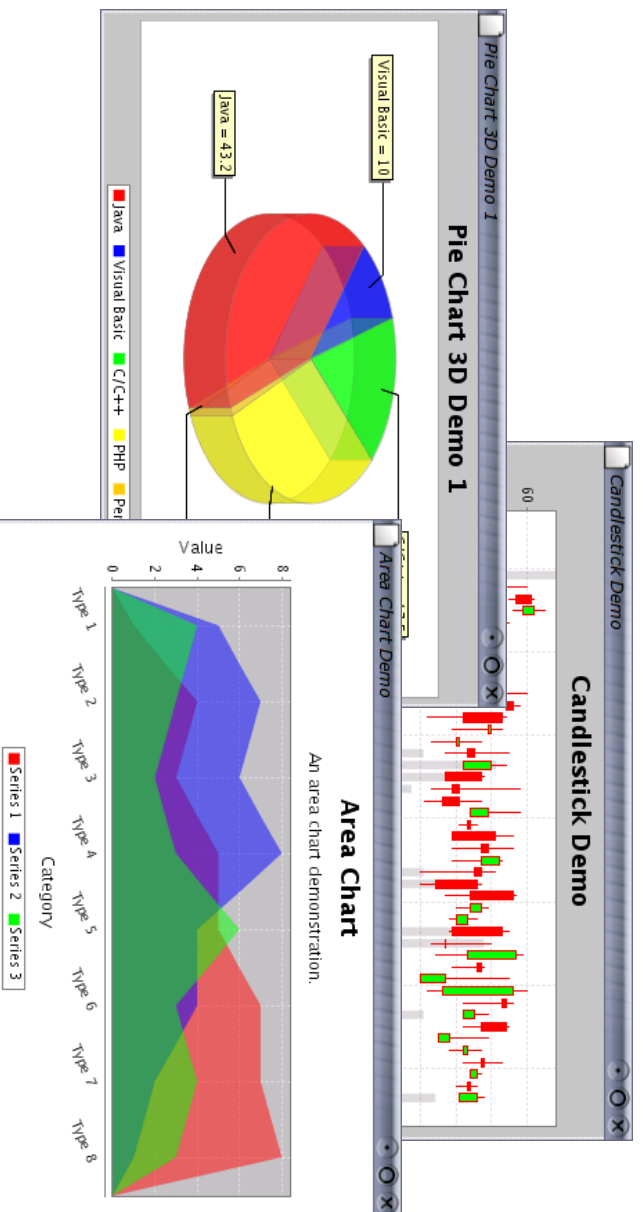
3rd Party: JGoodies

- <http://www.jgoodies.com>
- <http://jgoodies.dev.java.net>



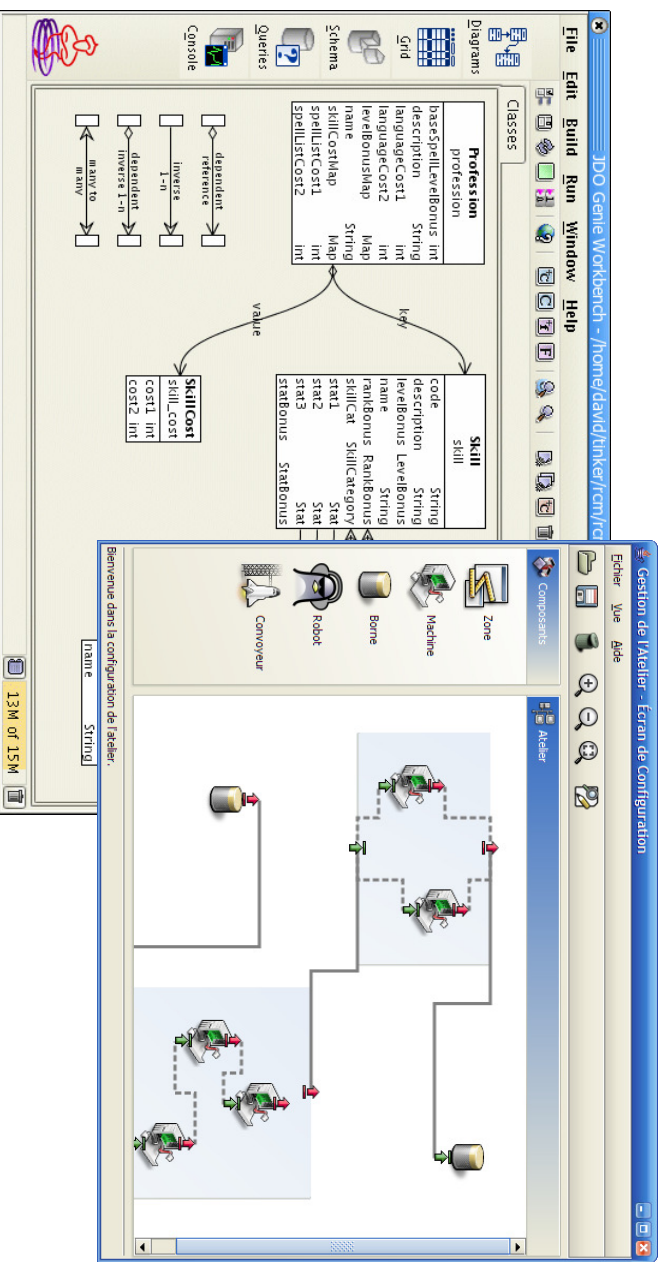
3rd Party: JFreeChart

- <http://www.jfree.org/jfreechart>



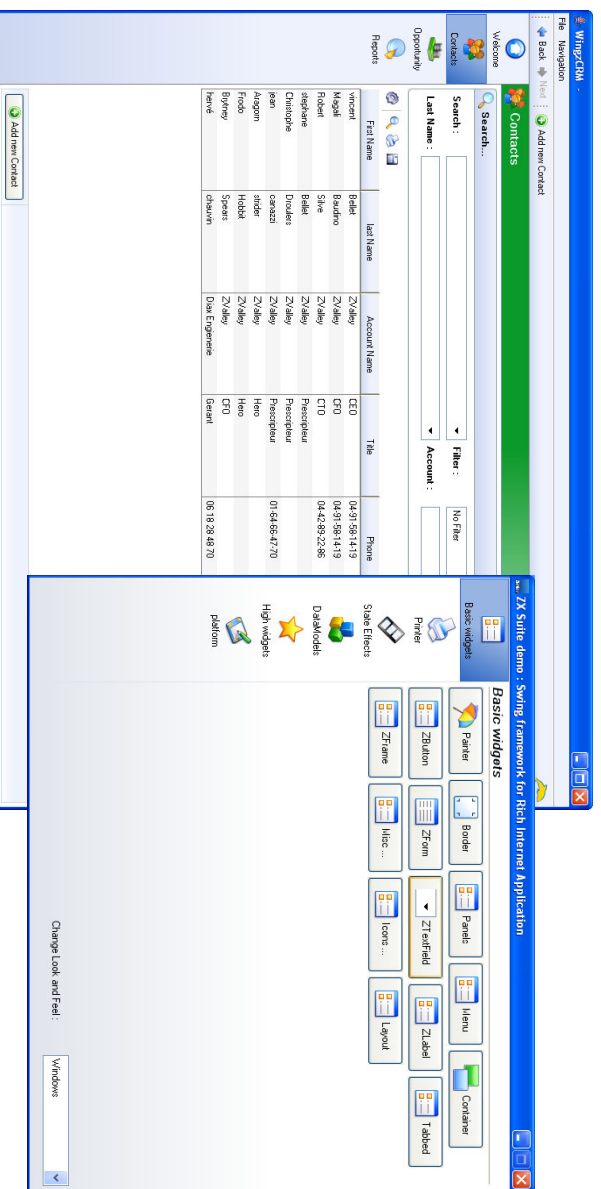
3rd Party: JGraph

- <http://www.jgraph.com>



3rd Party: WingZ

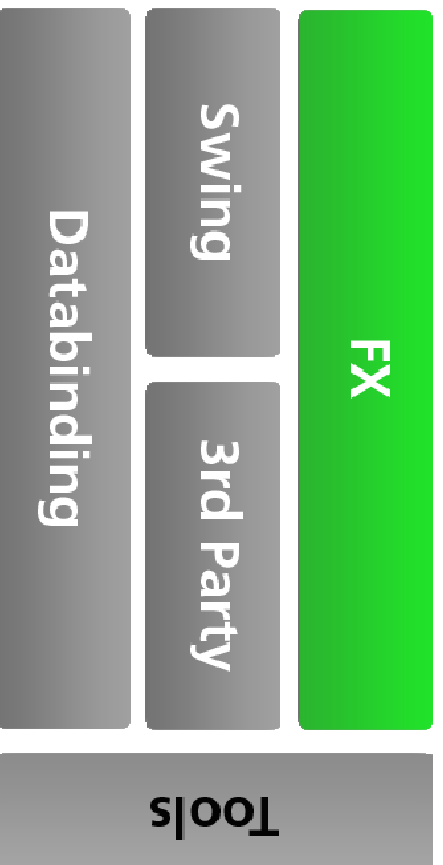
- <http://www.zvalley.com>



3rd Party: Plein d'autres!

- Regardez le site web Swing Depot
 - > <http://www.javadesktop.org/rollups/components>
- Reprend des composants Swing de Haute-Qualité
- Offres Open Source et commerciales
- Utiles pour tout types d'applications
 - > Général
 - > Graphes, diagrammes
 - > Sélection de Dates

Agenda Fondamentaux



Agenda FX

- Introduction
- Glass pane magic
- Animation
- Photo Collage
 - > Combinaison animations et glass pane magic
 - > Astuces de Performance
- Aller plus loin

Agenda FX

- **Introduction**
- Glass pane magic
- Animation
- Photo Collage
 - > Combinaison animations et glass pane magic
 - > Astuces de Performance
- Aller plus loin

FX: Pourquoi S'En Soucier?

- Meilleure expérience utilisateur
 - > Eye candy sells...
 - > ...and brings better ergonomics
 - > Professional polish
- Demandez-vous: "Pourquoi j'utilise JGoodies?"
- C'est beaucoup de plaisir
- Ce n'est pas si difficile
- I mean it!

FX: Le Ciel est la Limite

- Swing toolkit
 - > Hautement flexible
 - > Chaque composant peut être personnalisé
 - > Conçu au-dessus de Java2D
- Java2D
 - > API de dessin de haute qualité, et compréhensible
 - > Architecture extensible
 - > Abstraction du Device
 - > Cela peut se passer à l'écran, à l'impression, dans un PDF...
 - > Accélération matérielle

Agenda FX

- Introduction
- **Glass pane magic**
- Animation
- Photo Collage
 - > Combinaison animations et glass pane magic
 - > Astuces de Performance
- Aller plus loin

FX: Glass Pane Magic

- Applications exécutent des tâches de durée inconnue
- Solutions Possibles
 - > Montrer le sablier
 - > Utiliser une JProgressBar indéterminée (damn Cylon!)
 - > Désactiver les composants
- Cela fonctionne et *they're not pretty*
- La solution de l'homme moderne
 - > Utilise le glass pane
 - > Utilise une forme circulaire (après tout, Apple l'a fait)

FX: Glass Pane Magic

- Qu'est ce que le glass pane?



DEMO Infinite Progress Panel

FX: Glass Pane Magic

- Premièrement, créez votre propre glass pane

> Extends JComponent, JPanel...

```
public class InfiniteProgressPanel extends JComponent {  
    @Override  
    protected void paintComponent(Graphics g) {  
        g.setColor(new Color(255, 255, 255, alphaLevel));  
        g.fillRect(0, 0, getWidth(), getHeight());  
        // paint the rest  
    }  
}
```

> Et setOpaque()?

FX: Glass Pane Magic

- Ensuite, bloquez les entrées

```
public class InfiniteProgressPanel extends JComponent {  
    public void setVisible(boolean visible) {  
        if (visible) {  
            setFocusTraversalKeysEnabled(false);  
            addMouseListener(new MouseAdapter() { });  
            addMouseMotionListener(new MouseMotionAdapter() {  
                });  
            addKeyListener(new MouseAdapter() { });  
            super.setVisible(true);  
            requestFocusInWindow();  
        } else {  
            // undo  
        }  
    }  
}
```

> Focus ne peut être requis que si visible

FX: Glass Pane Magic

- Finalement, installer votre glass pane

```
public WootDemo extends JFrame implements ActionListener {
    public WootDemo() {
        super("Woott");
        setGlassPane(new InfiniteProgressPanel());
    }

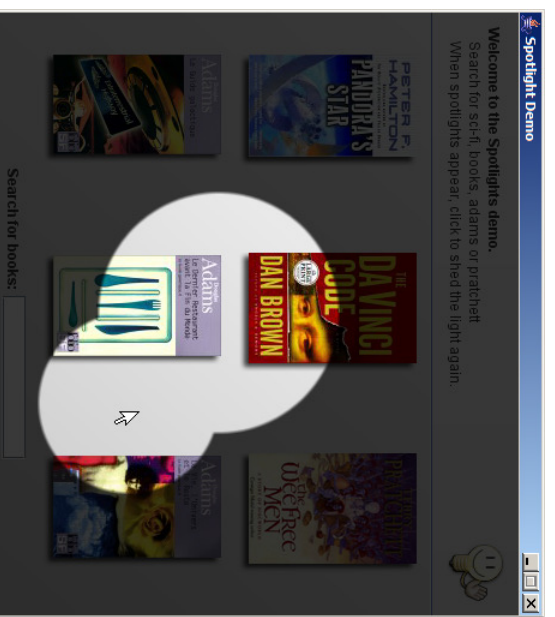
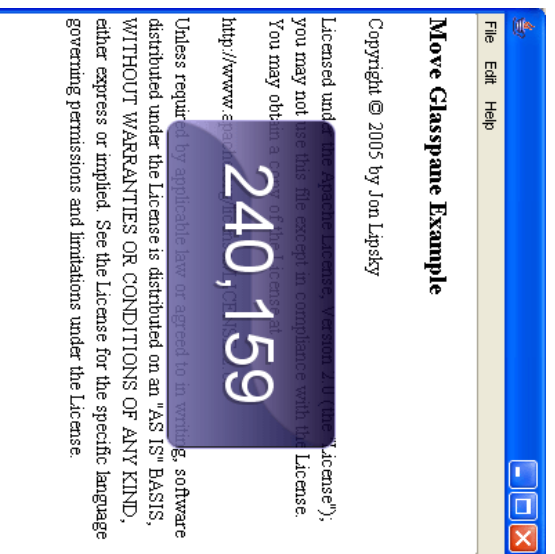
    public void actionPerformed(ActionEvent evt) {
        getGlassPane().setVisible(true);
    }
}
```

- > Souvenez-vous, un glass pane est invisible lorsqu'il est installé

- <http://swingfx.dev.java.net>

FX: Plus sur Glass Pane Magic

- <http://www.jroller.com/page/gfx/Archives>
- Jon Lipsky: <http://blog.elevenworks.com/?p=7>



Agenda FX

- Introduction
- Glass pane magic
- **Animation**
- Photo Collage
 - > Combinaison animations et glass pane magic
 - > Astuces de Performance
- Aller plus loin

FX: Animation

- Pourquoi les animations sont-elles utiles?
 - > Le GUI apparait avoir plus de répondant
 - > Meilleur capture visuelle de ce qui se passe
 - > Transitions sont plus réalistes
- Comment créer des animations?
 - > javax.swing.Timer
 - > Chet's Timing Framework
 - > Karsten's JGoodies Animation library

DEMO

Drop in Motion



FX: Animation Timer

- Créer un timer
 - > This one runs at 60 fps

```
private void startAnimation() {  
    Timer animation = new Timer(1000 / 60, new Animation());  
    animation.start();  
}
```

- Créer le handler

```
private class Animation implements ActionListener {  
    public void actionPerformed(ActionEvent evt) {  
    }  
}
```

FX: Animation Handler

- Timing n'est pas garanti
 - > Timer est dépendant de la plateforme
 - > Utilisation Massive du CPU peut affecter le timer
 - > **Ne Jouez pas** une frame d'animation par événement reçu
- Savoir où vous situez dans le temps
 - > f(elapsed time) -> [0, 1]
 - > Très souvent, un cycle, pas de répétition
 - > f(elapsed time) = elapsed time / delay

FX: Animation Handler

```
private long start;
private boolean isInitialized;

private Animation() {
    isInitialized = false;
}

public void actionPerformed(ActionEvent e) {
    if (!isInitialized) {
        start = System.currentTimeMillis();
        isInitialized = true;
    }

    long elapsed = System.currentTimeMillis() - start;
    if (elapsed > DELAY_ANIMATION) {
        ((Timer) e.getSource()).stop();
    } else {
        double time = (double) elapsed / (double) DELAY_ANIMATION;
        // ...
        repaint();
    }
}
```

FX: Comportement Animation

- La valeur du temps ne fonctionne pas bien pour le mouvement

- > Nous avons besoin de notre position dans l'espace
- > $g(f(\text{elapsed time})) \rightarrow [0, 1]$

$$y = X m . e^{-\frac{K}{m} \cdot x}$$

- Maintenant, nous pouvons dessiner
 - > Tailles des icônes sont réadaptées
 - > Taille de l'ombre est réadaptée
 - > Modification distance pénombre
 - > Modification opacité de la pénombre

DEMO

Drop in Motion, Equation

FX: Animation Behavior

- Pour dessiner avec notre équation “cool”

```
private void drawItem(Graphics2D g2) {  
    double position = equation.compute(time);
```

```
    float alpha = 1.0f - (0.5f * (float) position);
```

```
    Composite old = g2.getComposite();
```

```
    Composite c = AlphaComposite.getInstance(AlphaComposite.SRC_OVER,  
                                              alpha);
```

```
    g2.setComposite(c);  
    // draw shadow first  
    g2.setComposite(old);
```

```
    int width = (int) (image.getWidth() / 2 * (1.0 + position));  
    int height = (int) (image.getHeight() / 2 * (1.0 + position));
```

```
    g2.drawImage(image, dropX, dropY, width, height, null);  
}
```

FX: Animation Timer Drawbacks

- A lot of boilerplate
- Les animations complexes sont difficiles
 - > Beaucoup de cycles
 - > Va et Viens
 - > Répétition
- Timing Framework de Chet
 - > Gère tout pour vous
- JGoodies Animation
 - > Simplifie les animations simples

Agenda FX

- Introduction
- Glass pane magic
- Animation
- **Photo Collage**
 - > Combinaison animations et glass pane magic
 - > Astuces de Performance
- Aller plus loin

FX: Aperçu Photo Collage

- Un simple photo viewer
- Vous pouvez créer des collages
 - > Comme dans Google Picasa
- Une application Swing avec :
 - > Aucun curseurs de souris Tirer-Lâcher affreux
 - > Des Animations
 - > Des vignettes de Hautes qualités

DEMO

Photo Collage

FX: Photo Collage Drag & Drop

- Tout est fait sur le glass pane
 - > Un JPanel non opaque
- Le glass pane doit
 - > Dessiner une image spécifique
 - > À une position donnée
- Drag source et drop targets
 - > Définit l'image avec `glassPane.setImage()`
 - > Définit la position `glassPane.setPosition()`

FX: Photo Collage Drag & Drop

- Problèmes de Performance peuvent être *tricky*
 - > Appels à repaint() sont chers

```
public void setPoint(Point location) {  
    this.oldlocation = this.location;  
    this.location = location;  
}
```

```
public Rectangle getRepaintRect() {  
    int x = location.getX();  
    int y = location.getY();  
    int x2 = oldlocation.getX();  
    int y2 = oldlocation.getY();
```

```
    Rectangle oldPic = new Rectangle(x2, y2, width, height);  
    return new Rectangle(x, y, width, height).union(oldPic);  
}
```

FX: Photo Collage Drag & Drop

- Lorsque la souris bouge, appelez repaint(Rectangle)

```
public void dragMouseMoved(DragSourceDragEvent dsde) {  
    JRootPane root = SwingUtilities.getRootPane(this);  
    GhostGlassPane glassPane = (GhostGlassPane) root.getClassPane();
```

```
    Point p = (Point) dsde.getLocation().clone();  
    SwingUtilities.convertPointFromScreen(p, glassPane);  
    glassPane.setPoint(p);
```

```
    glassPane.repaint(glassPane.getRepaintRect());  
}
```

FX: Photo Collage Astuces

- Utilisez des images compatibles

- > Images JPEG peuvent ne pas être compatibles
- > Performance améliorée par un facteur de 20

```
public static BufferedImage createCompatibleImage(BufferedImage image) {  
    GraphicsEnvironment e = GraphicsEnvironment.getLocalGraphicsEnvironment();  
    GraphicsDevice d = e.getDefaultScreenDevice();  
    GraphicsConfiguration c = d.getDefaultConfiguration();
```

```
    BufferedImage compatibleImage = c.createCompatibleImage(image.getWidth(),  
    image.getHeight());
```

```
    Graphics g = compatibleImage.getGraphics();  
    g.drawImage(image, 0, 0, null);  
    g.dispose();
```

```
    return compatibleImage;  
}
```

FX: Photo Collage Astuces

- Ne jamais appeler `getScaledInstance()`, utilisez `drawImage()`

- > Logiciel vs Matériel

```
double ratio = (double) image.getWidth() /  
                (double) image.getHeight();
```

```
int width = ITEM_WIDTH;  
int height = (int) (width / ratio);  
g2.drawImage(image, x, y, width, height, null);
```

FX: Photo Collage Astuce

- Bilinear et bicubic resizing ne sont pas parfait
 - > Lorsqu'on réduit de plus de la moitié
 - > Procéder étapes par étapes
 - > Technique similaire au mipmapping dans les jeux
- Resizing demo
 - > L'une utilise drawImage() et bilinear interpolation
 - > L'autre utilise une méthode personnalisée

DEMO
Photo Collage, Resizing

FX: Astuces Photo Collage

```
public static BufferedImage createThumbnail(BufferedImage image,
                                             int thumbWidth) {
    int width = image.getWidth();
    float ratio = (float) width / (float) image.getHeight();
    BufferedImage thumb = image;
    do {
        width /= 2;
        // ...
        BufferedImage temp = new BufferedImage(width,
                                                (int) (width / ratio),
                                                BufferedImage.TYPE_INT_ARGB);
        g2.setRenderingHint(...);
        g2.drawImage(thumb, 0, 0, temp.getWidth(), temp.getHeight(), null);
        g2.dispose();
        thumb = temp;
    } while (width != thumbWidth);
    return thumb;
}
```

FX: Astuces Photo Collage

- Ce n'est pas plus lent!
 - > En fait, pas trop
 - > ~ = 15ms vs ~ = 0ms for 640x480
 - > Vous pouvez mettre le résultat en cache
- Beaucoup plus rapide que getScaledInstance()
- Semble meilleur qu'un simple drawImage()

Agenda FX

- Introduction
- Glass pane magic
- Animation
- Photo Collage
 - > Combinaison animations et glass pane magic
 - > Astuces de Performance
- **Aller plus loin**

DEMO
Music Shelf

DEMO Twinkle



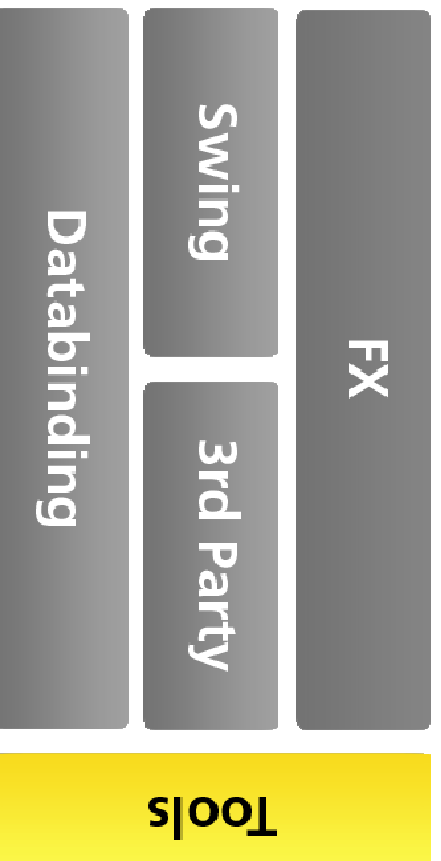
FX: Aller Plus Loin

- Extreme GUI Makeover, JavaOne 2005 TS-3605
 - > <http://developers.sun.com/learning/javaoneonline/>
- Jon Lipsky's blog
 - > <http://blog.elevenworks.com/>
- Romain Guy's blogs
 - > <http://iroller.com/page/gfx>
 - > <http://weblogs.java.net/blog/gfx>
- Santhosh Kumar's blog
 - > <http://www.iroller.com/page/santhosh>

FX: Aller Plus Loin

- SwingX
 - > <http://www.swinglabs.org>
- SwingLabs Incubator, FX module
 - > <http://jdnc-incubator.dev.java.net>
- SwingFX project
 - > <http://swingfx.dev.java.net>
- JGoodies (and Karsten's session on Thursday)
 - > <http://jgoodies.dev.java.net>
- ClientJava
 - > <http://www.clientjava.com>

Agenda Fondamentaux



Outils: Utilisez-les et soyez heureux

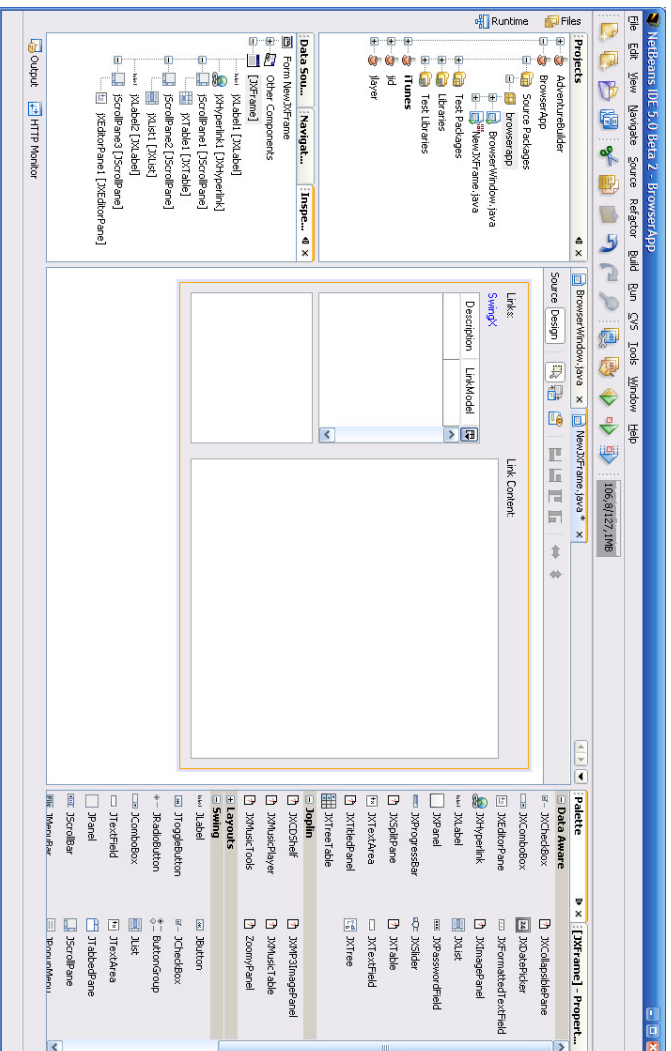
- Outils tendent à nous rendre plu productifs
 - > Integration avec un contrôle de code source
 - > Débogueur intégré
 - > Assistance à l'encodage
 - > Support Refactoring
- Les Outils de GUI ne sont pas des exceptions
 - > La Conception de GUI est une tâche visuelle. Utilisez un visual builder
 - > Si vous ne pouvez pas (ou ne voulez pas) utiliser un *visual builder*, utiliser un *non-visual builder*. Mais utilisez un outils!
 - > Si vous ne pouvez pas utiliser un outils, allez chercher de la caféine— cela risque d'être une longue nuit

Préférez-vous voir ceci...

```
private void build() {
    FormLayout formLayout = new FormLayout(
        "f:d:n, t:7dlu:n, f:d:g", // columns
        "c:d:n, t:4dlu:n, c:d:n, t:7dlu:n, f:d:g, t:7dlu:n, c:d:n"); // rows
    PanelBuilder builder = new PanelBuilder(this, formLayout);
    builder.setDefaultDialogBorder();
    CellConstraints cc = new CellConstraints();
    CellConstraints cl = new CellConstraints();

    builder.addLabel("Links: ", cl.xywh(1, 1, 1, 1),
        textOnlyHyperlink, cc.xywh(1, 3, 1, 1));
    builder.addLabel("Link Content:", cl.xywh(3, 1, 1, 1),
        new JScrollPane(LinkContent.getOutputComponent()), cc.xywh(3, 3, 1,
5));
    builder.add(new JScrollPane(table), cc.xywh(1, 5, 1, 1) );
    builder.add(new JScrollPane(list), cc.xywh(1, 7, 1, 1));
}
```

... Ou ceci?



DEMO

Joplin: Concevoir un
Music Player avec
Netbeans

Q&R

romain.guy@mac.com