

Environnements
de
développement intégré
(IDE):
Netbeans
et
les pratiques associées

<http://www.projet-plume.org/fr/fiche/netbeans>

pascal.dayre@irit.fr

AGENDA

1. Introduction aux IDEs
2. Utilisation de base de Netbeans
3. Utilisation avancée de Netbeans
4. Cas d'étude d'un projet avec WS

1. Introduction aux IDEs

- a) Les IDEs, support des pratiques
- b) L'IDE, le tout en un
- c) Quelques IDE

a) Les IDEs, support des pratiques

- Pratiques pour le développement
- IDE = un socle + différentes extensions (plugins)
- Environnement normalisé pour les projets collaboratifs
- la modélisation
- la génération et le développement de code
- le test (unitaire, profiling, recette, ...)
- la maintenance (debogage, suivi de ticket, ...)
- le support (documentation, ...)

b) L'IDE, le tout en un

- IDE : Environnement de développement intégré
 - ★ Application fournissant aux développeurs de logiciels différents outils
 - ➔ Éditeur de code source
 - ➔ Un compilateur et/ou un interpréteur
 - ➔ Des outils de génération automatique
 - ➔ Un debugger
 - ➔ Un système de gestion de version
 - ➔ Différents outils pour faciliter la construction d'IHM
 - ➔ Pour les développements orientés objet:Un class browser, Un class inspector,Un diagramme hiérarchique de classes
 - ★ Pourquoi les utiliser ?
 - ➔ Ils offrent la majorité des fonctionnalités listées
 - ➔ Auto complétion du code et le surlignage de la syntaxe
 - ➔ Accès plus simple aux différents fichiers du projet
 - ➔ Commentaires contextuels si le projet est commenté comme il faut (javadoc par ex...)
 - ➔ Aide/doc intégrée à l'ide
 - ➔ Création et intégration/réutilisation de composants/assemblages
 - ➔ Indépendance par rapport à l'OS de la machine de développement
 - ➔ Outils de création d'IHM
 - ➔ Refactoring

c) Quelques IDEs

★ Quelques IDE

★ Lié à un langage ou une famille de langages, ou généraliste

★ Licences opensource

→ Eclipse <http://www.eclipse.org/>

→ NetBeans <http://www.netbeans.org/>

→ Kdevelop <http://www.kdevelop.org/>

→ RDE Ruby Development Environment <http://sourceforge.net/projects/rubyde/>

★ Licences commerciales/free

→ JBuilder <http://www.codegear.com/products/jbuilder> (CodeGear)

→ Jcreator <http://www.jcreator.com/> (XINOX software)

→ IntelliJ Idea <http://www.jetbrains.com/idea/> (JetBrains)

III.3 L'utilisation avancée des IDEs par l'exemple

1. Gestion de la documentation
2. Listes des tâches à effectuer
3. Personnalisation de son IDE : Installation d'un plugin
4. Utilisation du plugin Visual Editor dans Eclipse
5. Importation de librairies et de projets
6. Tests unitaires avec Junits
7. Profiling
8. Gestion de version
9. Reverse engineering
10. Déploiement (WS, J2EE, mobile, ...)
11. Les raccourcis clavier pour un usage intensif

2. L'utilisation de base de NetBeans

a) Présentation de l'interface de Netbeans

b) Création d'un projet

c) Edition d'un projet

d) Exécution d'un projet

e) Débogage d'un projet

COMPIL

Présentation de l'interface de Netbeans

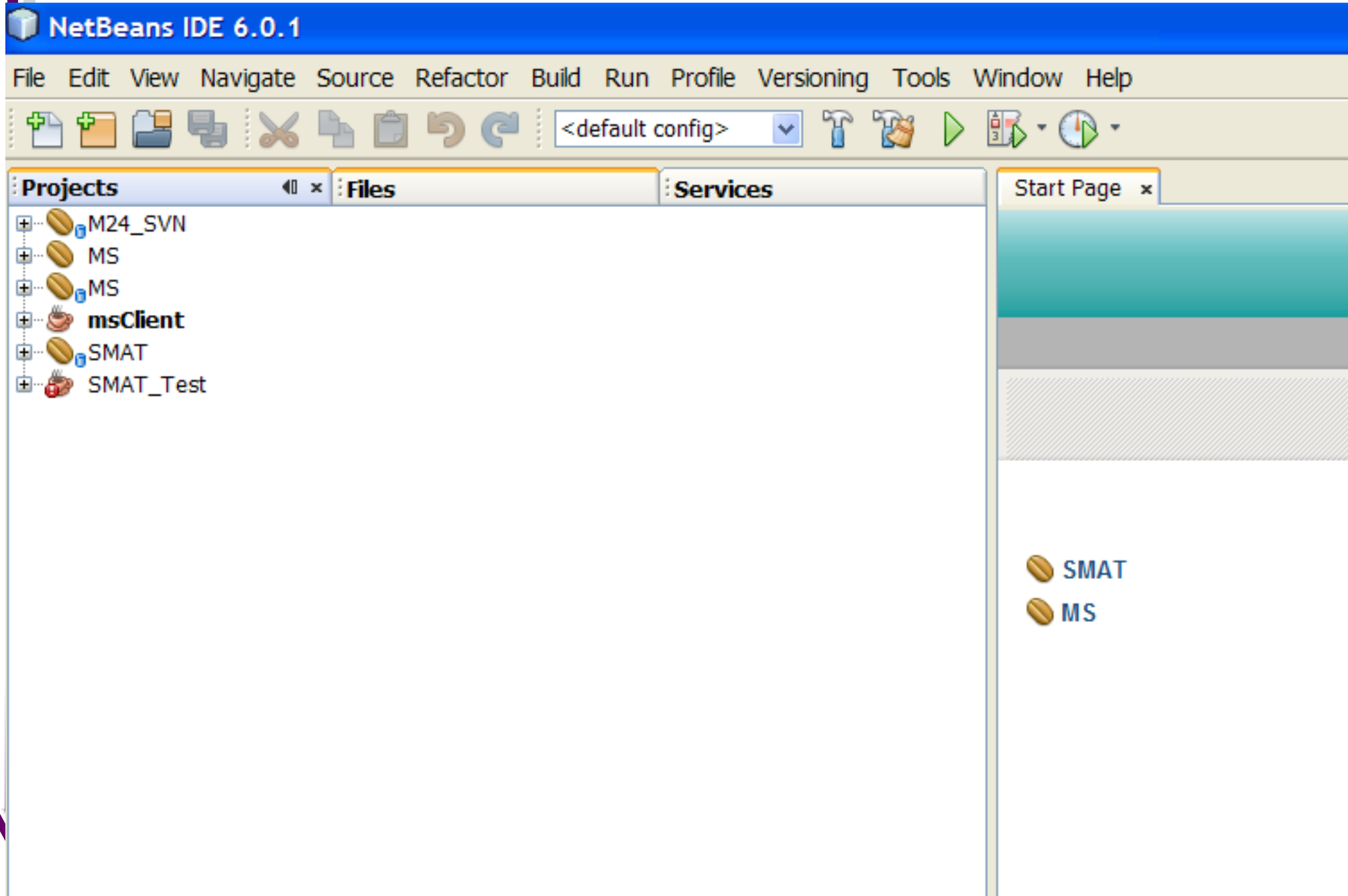
The screenshot displays the NetBeans IDE 6.0.1 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, and Help. The left sidebar contains the Projects, Files, and Services views. The main workspace shows the Start Page with the following sections:

- Recent Projects:** Lists SMAT and MS.
- News & Tutorials:** A list of articles with dates, including "Setting Up NetBeans IDE, JDK 6 and JavaFX on 32-bit Intel Macs" (08/10/08), "Nominate NetBeans IDE for Developer.com 2009 Product Awards!" (08/10/08), "Using Page Fragments in Web Applications" (06/10/08), and "Enhancing a Structured Document Using jQuery" (03/10/08). Links for "ALL NEWS >>" and "ALL ARTICLES >>" are provided.
- Featured Demo:** Titled "Using JPA Support with EclipseLink in the NetBeans IDE", it includes a screenshot of a dialog box for configuring EclipseLink. The dialog shows "Project: EclipseLinkDemo2", "Location: Source Packages", and "Package:" with a text input field. A checkbox "Generate Named Query Annotations for Persistence" is checked. A warning message states "The project does not have a persistence unit. You can create one." with a "Create Persistence Unit..." button. A link "ALL DEMOS >>" is also present.
- Blogs:** A list of blog entries with dates, including "Wen's Blog: 2009 Developer.com Awards - Nominations Open!" (08/10/08), "Vadiraj's Blog: NetBeans Debugging tricks #1211" (08/10/08), "NetBeans Zone - The social network for developers: Counting Ruby Objects on the JRuby Heap" (08/10/08), "James' Blog: Changing the MySQL Root Password" (08/10/08), and "APIDesign - Blogs: Define 'Friday Morning'" (08/10/08). A link "ALL BLOGS >>" is provided.

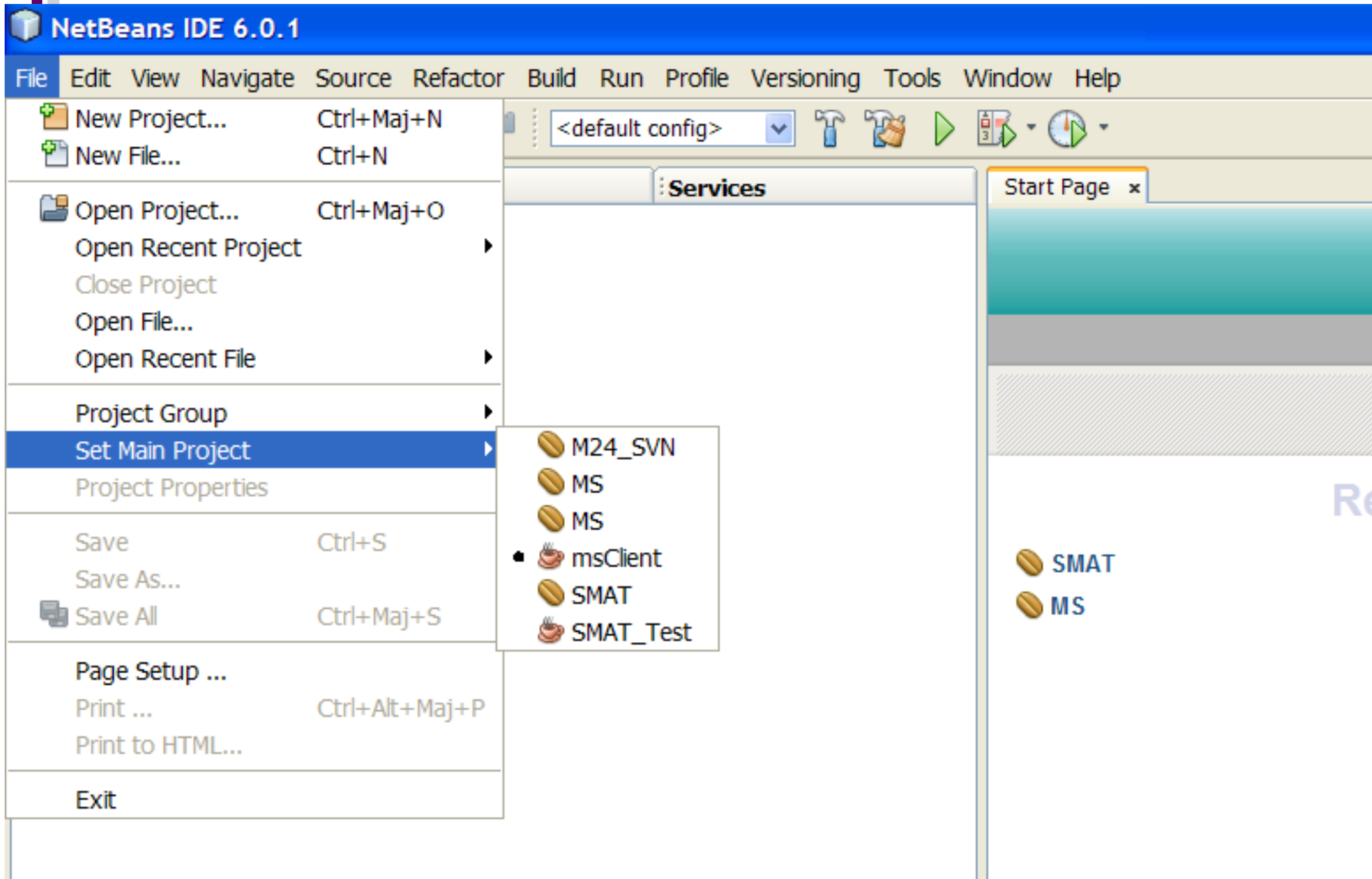
At the bottom of the Start Page, there is a "Show On Startup" checkbox which is checked, and the Sun Microsystems logo.

COMPIL

Présentation de l'interface de Netbeans



Présentation de l'interface de Netbeans



COMPIL

Présentation de l'interface de Netbeans

The screenshot displays the NetBeans IDE 6.0.1 interface. The title bar reads "MultimediaAnnotationServices - NetBeans IDE 6.0.1". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, and Help. The Edit menu is open, showing various editing actions with their respective keyboard shortcuts. The main editor window displays a Java file named "ContentSegmentComparedByOrder.java" with the following code:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package org.kplab.smat.comparator;

import java.util.Comparator;
import org.kplab.smat.domainModel.ContentSegment;

/**
 *
 * @author Morpheus
 */
public class ContentSegmentComparedByOrder implements Comparator<ContentSegment>

    public int compare(ContentSegment o1, ContentSegment o2) {
        return o1.getOrder() - o2.getOrder();
    }

}
```

The left sidebar shows a project tree with the following structure:

- org.kplab.smat.ws.launcher.AnnotationTool
- org.kplab.smat.ws.smatException
- org.kplab.smat.ws.structureMultimediaContent
- org.kplab.smat.ws.userPdf
- org.kplab.smat.ws.userProfiling
- Test Packages
- Libraries
- Test Libraries
- SMAT
- SMAT_Test
- SMATServer

Présentation de l'interface de Netbeans

The screenshot displays the NetBeans IDE 6.0.1 interface. The title bar reads "MultimediaAnnotationServices - NetBeans IDE 6.0.1". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, and Help. The "View" menu is open, showing options like Editors, Code Folds, Web Browser, IDE Log File, Toolbars, Show Editor Toolbar, Show Line Numbers, Show Diff Sidebar, Show Versioning Labels, and Full Screen (Alt+Maj+Entrée). The Project Explorer on the left shows a project structure with packages like org.kplab.smat.comparator and org.kplab.smat.controller, and a file named ContentSegmentComparedByOrder.java. The Services window is empty. The code editor on the right shows a Java file with package declarations, imports, and a class definition.

```

/*
 * To change this
 * and open the te
 */

package org.kplab.

import java.util.C
import org.kplab.s

/**
 *
 * @author Morpheu
 */
public class Conte

public int com
return o1.
}

```

COMPIL

Présentation de l'interface de Netbeans

MultimediaAnnotationServices - NetBeans IDE 6.0.1

File Edit View **Navigate** Source Refactor Build Run Profile Versioning Tools Window Help

Go to File... Alt+Maj+O
Go to Type... Ctrl+O
Go to Function or Variable...
Go to Test Ctrl+Maj+T
Go to Rails Action or View Ctrl+Maj+A
Go to Previous Document
Go to Source Ctrl+Maj+B
Go to Declaration Ctrl+B
Go to Super Implementation Ctrl+Maj+P
Inspect
Last Edit Location Ctrl+Q
Back Alt+Gauche
Forward Alt+Droite
Go to Line... Ctrl+G
Toggle Bookmark Ctrl+Maj+M
Next Bookmark Ctrl+Maj+Period
Previous Bookmark Ctrl+Maj+Comma
Next Error Ctrl+Period
Previous Error Ctrl+Comma
Select in Projects Ctrl+Maj+1
Select in Files Ctrl+Maj+2
Select in Classes Ctrl+Maj+9
Select in Favorites Ctrl+Maj+3
Select in Model Ctrl+Maj+5

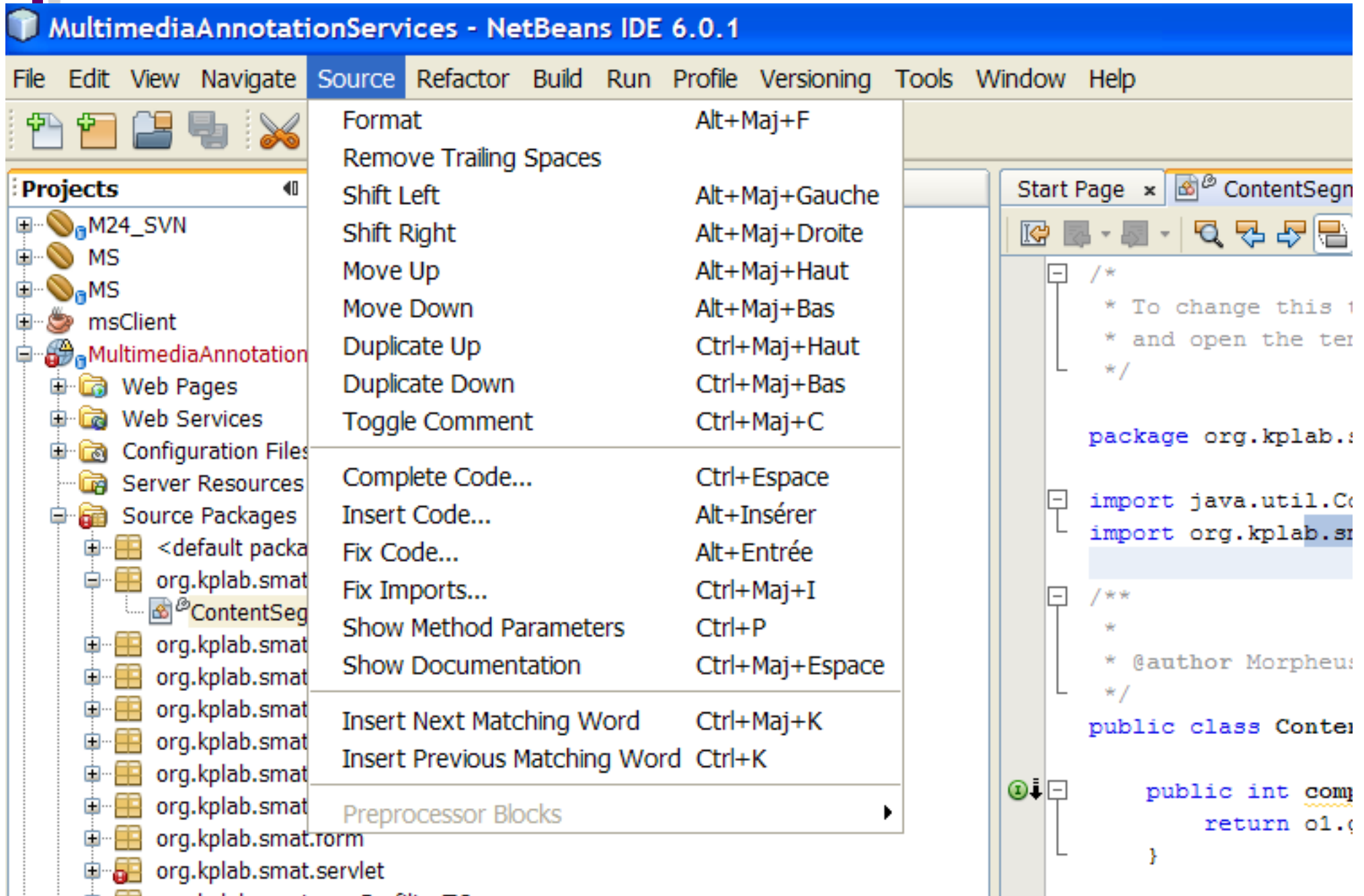
Start Page x ContentSegmentComparedByOrder.java x

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package org.kplab.smat.comparator;  
  
import java.util.Comparator;  
import org.kplab.smat.domainModel.ContentSegment;  
  
/**  
 *  
 * @author Morpheus  
 */  
public class ContentSegmentComparedByOrder implements Comparator  
  
    public int compare(ContentSegment o1, ContentSegment o2) {  
        return o1.getOrder() - o2.getOrder();  
    }  
}
```

Présentation de l'interface de Netbeans

MultimediaAnnotationServices - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help



Format	Alt+Maj+F
Remove Trailing Spaces	
Shift Left	Alt+Maj+Gauche
Shift Right	Alt+Maj+Droite
Move Up	Alt+Maj+Haut
Move Down	Alt+Maj+Bas
Duplicate Up	Ctrl+Maj+Haut
Duplicate Down	Ctrl+Maj+Bas
Toggle Comment	Ctrl+Maj+C
Complete Code...	Ctrl+Espace
Insert Code...	Alt+Insérer
Fix Code...	Alt+Entrée
Fix Imports...	Ctrl+Maj+I
Show Method Parameters	Ctrl+P
Show Documentation	Ctrl+Maj+Espace
Insert Next Matching Word	Ctrl+Maj+K
Insert Previous Matching Word	Ctrl+K
Preprocessor Blocks	

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package org.kplab.smat;

import java.util.Collections;
import org.kplab.smat.smat;

/**
 *
 * @author Morpheus
 */
public class ContentSeg...

public int compute...
    return 0;
}

```

Présentation de l'interface de Netbeans

MultimediaAnnotationServices - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Rename... Ctrl+R
 Move...
 Copy...
 Safe Delete...
 Change Method Parameters...
 Encapsulate Fields...
 Pull Up...
 Push Down...
 Extract Interface...
 Extract Superclass...
 Use Supertype Where Possible...
 Move Inner to Outer Level...
 Introduce Variable...
 Introduce Constant...
 Introduce Field...
 Introduce Method...
 Convert Anonymous to Inner
 Undo
 Redo

Projects Files

- M24_SVN
- MS
- MS
- msClient
- MultimediaAnnotationServices
 - Web Pages
 - Web Services
 - Configuration Files
 - Server Resources
 - Source Packages
 - <default package>
 - org.kplab.smat.compara
 - ContentSegmentCom
 - org.kplab.smat.controlle
 - org.kplab.smat.dao
 - org.kplab.smat.dao.impl
 - org.kplab.smat.dao.impl
 - org.kplab.smat.dao.impl
 - org.kplab.smat.domainM
 - org.kplab.smat.form
 - org.kplab.smat.servlet
 - org.kplab.smat.userProfilingTO

Start Page x ContentSegn

```

/*
 * To change this 1
 * and open the ter
 */

package org.kplab.s

import java.util.Co
import org.kplab.s

/**
 *
 * @author Morpheus
 */
public class Conte

public int comp
return o1.c
}
  
```


COMPI



MultimediaAnnotationServices - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help



Projects Files

- M24_SVN
- MS
- MS
- msClient
- MultimediaAnnotationServices**
 - Web Pages
 - Web Services
 - Configuration Files
 - Server Resources
 - Source Packages
 - <default package>
 - org.kplab.smat.comparator
 - ContentSegmentComparedByOrder.java
 - org.kplab.smat.controller
 - org.kplab.smat.dao
 - org.kplab.smat.dao.impl.kr
 - org.kplab.smat.dao.impl.mysql
 - org.kplab.smat.dao.impl.ums
 - org.kplab.smat.domainModel
 - org.kplab.smat.form
 - org.kplab.smat.servlet
 - org.kplab.smat.userProfilingTO
 - org.kplab.smat.ws.annotationActivity
 - org.kplab.smat.ws.configureAnnotationSession

	Run Main Project	F6
	Debug Main Project	Ctrl+F5
	Test "MultimediaAnnotationServices"	Alt+F6
	Run File	
	Attach Debugger...	
	Finish Debugger Session	Maj+F5
	Pause	
	Continue	F5
	Step Over	F8
	Step Over Expression	Maj+F8
	Step Into	F7
	Step Out	Ctrl+F7
	Run to Cursor	F4
	Run Into Method	Maj+F7
	Apply Code Changes	
Stack		
	Toggle Line Breakpoint	Ctrl+F8
	New Breakpoint...	Ctrl+Maj+F8
	New Watch...	Ctrl+Maj+F7
	Evaluate Expression...	Ctrl+F9

ContentSegmentCom

ge this templat
a the template

g.kplab.smat.co

a.util.Comparat

kplab.smat.dor

Morpheus

ss ContentSegme

```
int compare(Co  
return o1.getOrde
```

Présentation de l'interface de Netbeans

The screenshot displays the NetBeans IDE 6.0.1 interface. The 'Profile' menu is open, showing options for configuring and managing profiling sessions. The background shows a code editor with a Java class named 'ContentSegmentComparedByOrder'.

NetBeans IDE 6.0.1

Factor Build Run Profile Versioning Tools Window Help

Profile Main Project...
Attach Profiler...
Profile Other

Rerun Profiling Session (Ctrl+Maj+F2)
Modify Profiling Session... (Alt+Maj+F2)
Stop Profiling Session

Reset Collected Results
Run GC

Take Snapshot of Collected Results (Ctrl+F2)
Take Heap Dump...
Load Snapshot...
Load Heap Dump...
Compare Memory Snapshots...

Advanced Commands

ContentSegmentComparedByOrder.java

```

import java.util.Comparator;
import org.netbeans.smat.domainModel.ContentSegment;

public class ContentSegmentComparedByOrder implements Comparator<ContentSegment> {

    public int compare(ContentSegment o1, ContentSegment o2) {
        return o1.getOrder() - o2.getOrder();
    }
}

```

Présentation de l'interface de Netbeans

NetBeans IDE 6.0.1

Factor Build Run Profile Versioning Tools Window Help

Profile Main Project...
Attach Profiler...
Profile Other

Rerun Profiling Session Ctrl+Maj+F2
Modify Profiling Session... Alt+Maj+F2
Stop Profiling Session

Reset Collected Results
Run GC

Take Snapshot of Collected Results Ctrl+F2
Take Heap Dump...
Load Snapshot...
Load Heap Dump...
Compare Memory Snapshots...

Advanced Commands

ContentSegmentComparedByOrder.java x

his template, choose Tools | Templates
e template in the editor.

```
lab.smat.comparator;

il.Comparator;
lab.smat.domainModel.ContentSegment;

pheus

public class ContentSegmentComparedByOrder implements

public int compare(ContentSegment o1, ContentSegm
return o1.getOrder() - o2.getOrder();
}
```

Présentation de l'interface de Netbeans

The screenshot displays the NetBeans IDE 6.0.1 interface. The 'Versioning' menu is open, showing options such as 'Checkout...', 'Import into Repository...', 'Relocate...', 'Update Project with Dependencies', 'Show Changes', 'Diff', 'Update', 'Commit...', 'Export Diff Patch...', 'Apply Diff Patch...', 'Copy to...', 'Switch to...', 'Merge to...', 'Show Annotations', 'Search History...', 'Revert Modifications...', 'Resolve Conflicts...', 'Ignore', 'Svn Properties', 'CVS', 'Subversion', and 'Local History'. The code editor shows a file named 'ContentSegmentComparedByOrder.java' with the following code:

```

change this template, choose Tools | Templates
open the template in the editor.

org.kplab.smat.comparator;

java.util.Comparator;
org.kplab.smat.domainModel.ContentSegment;

hor Morpheus

class ContentSegmentComparedByOrder implements Comparat

lic int compare(ContentSegment o1, ContentSegment o2) {
return o1.getOrder() - o2.getOrder();
    
```

COMPIL

Présentation de l'interface de Netbeans

The screenshot displays the NetBeans IDE 6.0.1 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, and Help. The Tools menu is open, showing options like Java DB Database, Add to Favorites, Create JUnit Tests (Ctrl+Maj+U), Add to Palette..., Apply Diff Patch..., Internationalization, Java Platforms, NetBeans Platforms, Libraries, Ruby Gems, Component Libraries, Servers, Mobility Deployment, Keystores, Templates, DTDs and XML Schemas, Palette (highlighted), Plugins, and Options. The Palette sub-menu is also open, listing Swing/AWT Components, HTML/JSP Code Clips, BPEL Activities, Visual Web JSF 1.1 Components, Visual Web JSF 1.2 Components, MIDP Visual Mobile Designer, and SVG Visual Composer.

The main editor window shows a Java file named ContentSegmentComparedByOrder.java. The code includes a package declaration, imports, and a class definition implementing the Comparator interface for ContentSegment objects.

```
package org.kplab.smat;

import java.util.Comparator;

public class ContentSegmentComparedByOrder implements Comparator<ContentSegment> {

    @Override
    public int compare(ContentSegment o1, ContentSegment o2) {
        // ...
    }
}
```

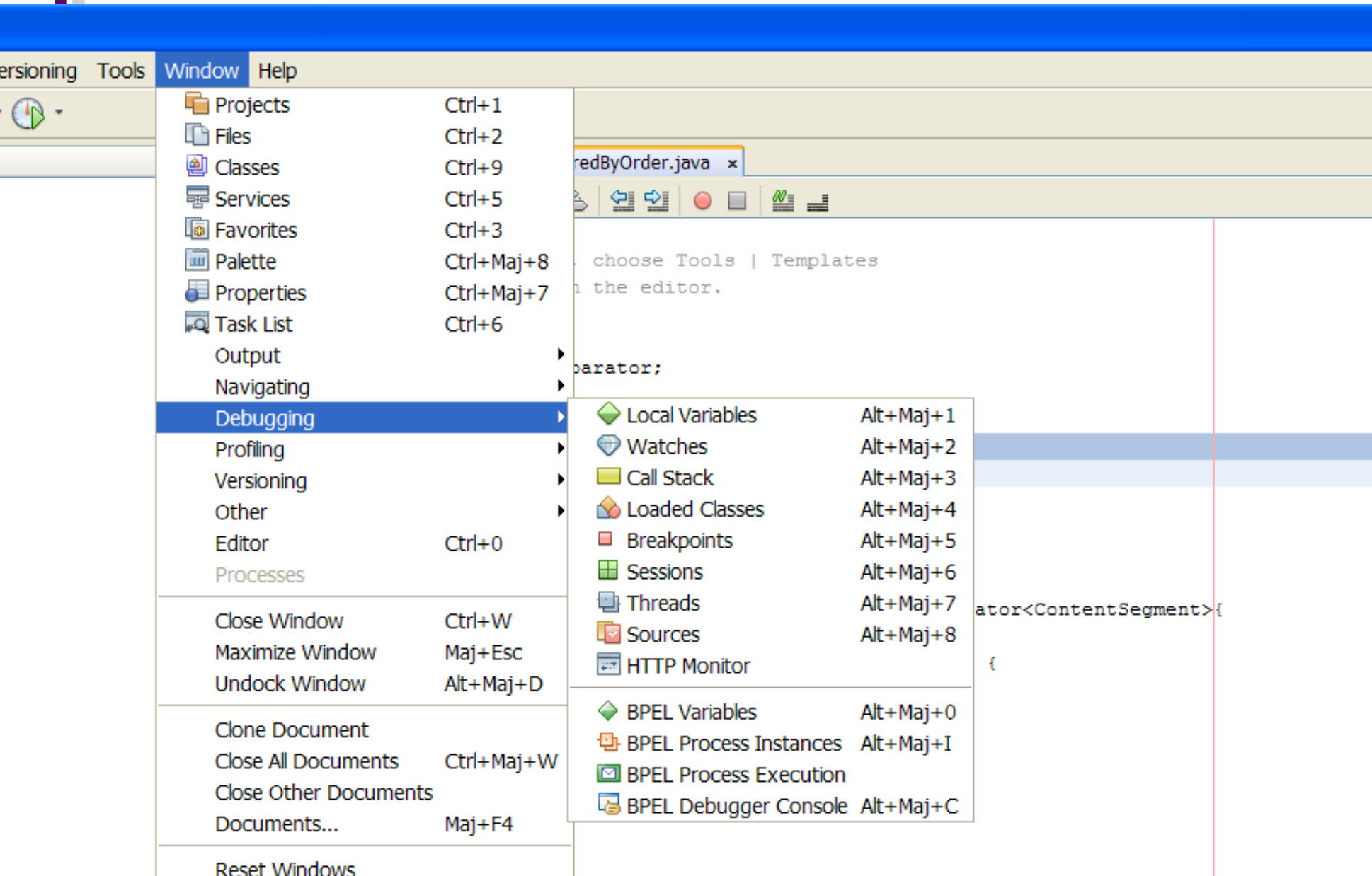
The Palette Manager dialog box is open, showing a list of components available for the current project. The components are organized into categories like Swing Containers, Swing Controls, Swing Menus, Swing Windows, AWT, Borders, Beans, Java Persistence, and Look and Feels. The dialog includes buttons for adding components from JAR files, libraries, or the project, as well as buttons for moving, removing, and resetting the palette.

Palette Content:

- Swing Containers
 - Panel
 - Tabbed Pane
 - Split Pane
 - Scroll Pane
 - Tool Bar
 - Desktop Pane
 - Internal Frame
 - Layered Pane
- Swing Controls
- Swing Menus
- Swing Windows
- AWT
- Borders
- Beans
- Java Persistence
- Look and Feels

Buttons: Add from JAR..., Add from Library..., Add from Project..., Move Up, Move Down, Remove..., New Category..., Reset Palette, Close

Présentation de l'interface de Netbeans





Présentation de l'interface de Netbeans

The screenshot displays the NetBeans IDE interface. At the top, there is a menu bar with 'Tools', 'Window', and 'Help'. The 'Help' menu is open, showing options like 'Help Contents', 'Online Docs and Support', and 'Javadoc References'. A sub-menu for 'Javadoc References' is also open, listing 'Java EE 5 SDK', 'JAX-WS 2.1', and 'JUnit API'. Below the menu, a code editor window titled 'yOrder.java' is visible, containing Java code. The code includes imports for 'java.util.Comparator' and 'org.kplab.smat.domainModel.ContentSegment', followed by a class definition 'ContentSegmentComparedByOrder' that implements the 'Comparator' interface. The code snippet is as follows:

```

package org.kplab.smat.domainModel;

import java.util.Comparator;
import org.kplab.smat.domainModel.ContentSegment;

/**
 *
 * @author Morpheus
 */
public class ContentSegmentComparedByOrder implements Comparator<ContentSegment>{

    public int compare(ContentSegment o1, ContentSegment o2) {
        return o1.getOrder() - o2.getOrder();
    }

}

```

Projects :

- Choix du projet actif
- Visualisation des éléments Java du projet

Styles d'affichage :

- Source : codage Java
- Screen Design : conception d'un seul écran de l'IHM en utilisant la palette de composants
- Flow Design : conception de l'enchaînement des écrans de l'IHM en utilisant la palette de composants

Palette :

Composants graphiques disponibles



The screenshot shows the NetBeans IDE 5.0 interface for the project 'pjtIHMNetbeansAuto'. The main window is titled 'HelloMiclet *' and is in 'Flow Design' mode. The central canvas, titled 'Flow Designer: MIDP-2.0', displays a flow diagram with a 'Mobile Device' icon and a 'helloForm [Form]' component. The 'Palette' window on the right is open, showing a list of components under 'Screens' (Alert, List, SplashScreen, Form, TextBox, WaitScreen), 'Commands' (Back Command, Exit Command, Item Command, Screen Command, Cancel Command, Help Command, Ok Command, Stop Command), and 'Form Items'. The 'Inspector' window at the bottom left shows the hierarchy of the 'helloForm [Form]' component, including 'Assigned Commands' and 'Items'. The 'Output' window at the bottom shows debug messages from the build process.

Inspector :

Visualisation de l'organisation des composants graphiques

Output :

Affichage des messages de construction du projet et des sorties console

Properties :

Visualisation et modification des propriétés du composant graphique

2. L'utilisation de base de NetBeans

a) Présentation de l'interface de Netbeans

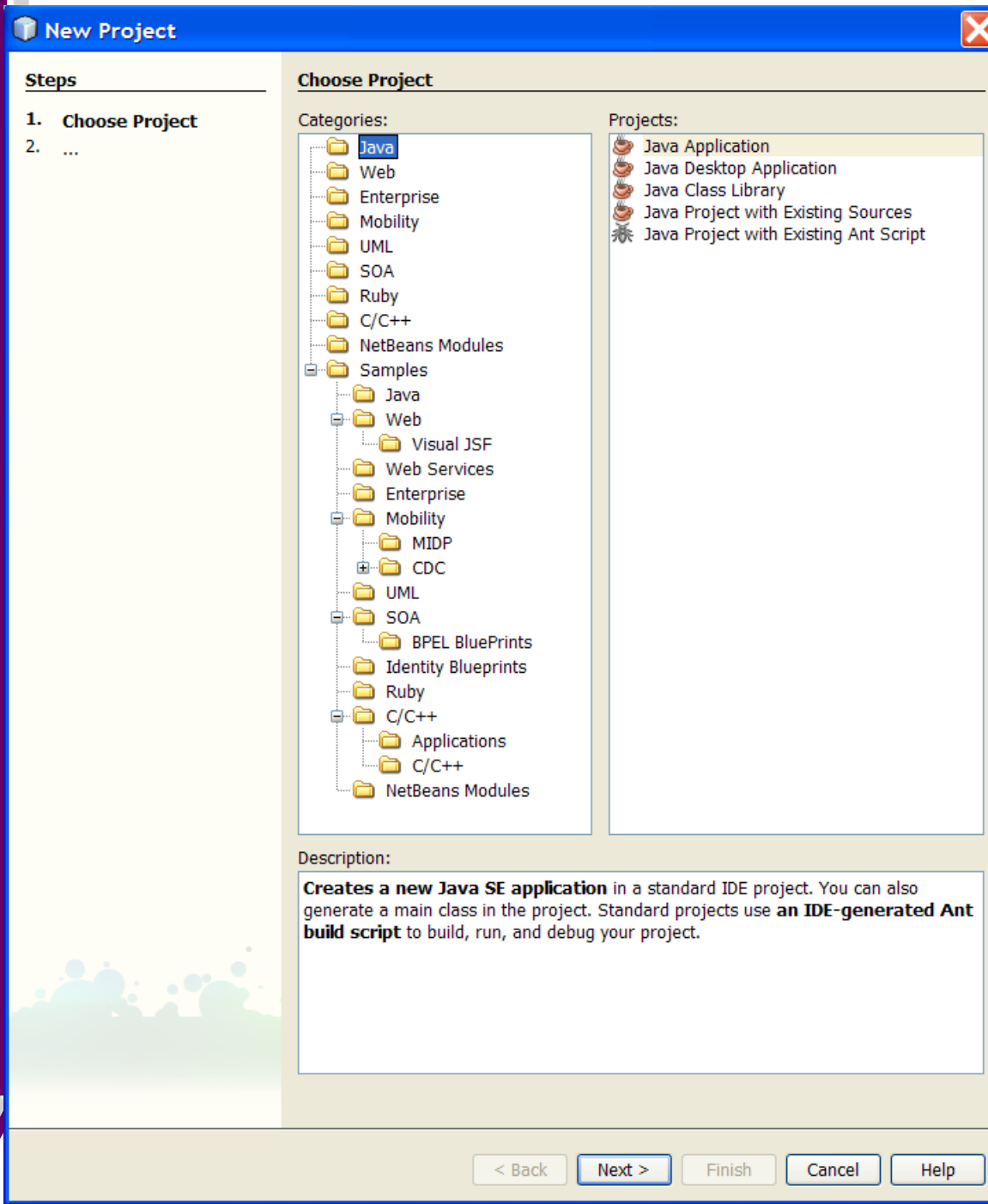
b) Création d'un projet

c) Edition d'un projet

d) Exécution d'un projet

e) Débogage d'un projet

Création d'un projet



- Typologie des projets
- Templates de projet
- Projet orienté J2EE, applications WEB
- NetBeans => les beans du net

Création d'un projet

How to set up the SoyLatte JDK 6 and JavaFX on Intel Macs with Mac OS X 10.4 and 32-bit CPUs, and configure NetBeans IDE as a tool

New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Frameworks

Name and Location

Project Name:

Project Location:

Project Folder:

Add to Enterprise Application:

Server:

Java EE Version:

Context Path:

Set as Main Project

Add Server Instance

Steps

1. **Choose Server**
2. ...

Choose Server

Server:

Name:

Product Awards! 08/10

Product Awards. Vote to have your favorite developer and open-source tools
 of the Last year, community participation was...

06/10

application's header information, and links for navigating between two

03/10

ed web documents using the jQuery JavaScript library in the NetBeans

[ALL ARTICLES](#)

Blo

08/10

missions (fro

08/10

of them. So coc

08/10

n JRuby and th

08/10

s having some

08/10

ion for millions

[ALL BLOGS](#)

Création d'un projet

New Web Application

Steps

1. Choose Project
2. Name and Location
3. **Frameworks**

Frameworks

Select the frameworks you want to use in your web application.

- Visual Web JavaServer Faces
- JavaServer Faces
- Struts 1.2.9**

Struts 1.2.9 Configuration

Action Servlet Name:

Action URL Pattern: ▼

Application Resource:

Add Struts TLDs

< Back Next > Finish Cancel Help

2. L'utilisation de base de NetBeans

- a) Présentation de l'interface de Netbeans
- b) Création d'un projet
- c) Edition d'un projet**
- d) Exécution d'un projet
- e) Débogage d'un projet

Création de paquetage, de classe

The screenshot shows the NetBeans IDE interface. The 'Projects' window on the left displays a project structure with a package named 'org.kplab.ms.server.file'. A context menu is open over this package, with the 'New' option selected. The 'New' submenu is also open, showing options like 'Java Package...', 'Web Service Client...', 'Session Bean...', 'Message-Driven Bean...', 'Java Class...' (which is highlighted), 'Entity Class...', 'Entity Classes from Database...', 'Web Service...', and 'Other...'. The 'Files' window shows a list of Java files including 'RectanglePb.java', 'RegionPb.java', 'SegmentPb.java', 'StreamPb.java', 'TaskPb.java', and 'VideoPb.java'. The 'Services' window is empty. The main editor window shows the code for 'DaoFileClient.java', which includes package declarations, imports, and an interface definition.

```

package org.kplab.ms.server.file;

import java.io.File;
import java.net.URI;

/**
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

public interface DaoFileClient {

    public String CreatePath(URI uri, String pathDest) ;
    public URI createContent(String sUrlFile) ;
    public String getMIMETYPE(File file);
    public void getContent();
    public void createSymbolicLink(String srcFile, String destPath, String
}
    
```

COMPIL

Navigation dans le code par onglets

The screenshot displays the NetBeans IDE interface with the following components:

- Projects View:** Shows a tree structure of the project 'MultimediaAnnotationServices'. The 'org.kplab.smat.controller' package is expanded, showing various action classes like 'AgentACTION.java', 'AnnotationPhaseACTION.java', etc.
- Code Editor:** Displays the source code of 'FragmentAnnotationACTION.java'. The code includes package declarations, imports (e.g., 'import java.net.URI;', 'import java.util.ArrayList;'), and class definitions. A red box highlights the 'Unused Import' for 'java.util.ArrayList'. The class 'FragmentAnnotationACTION' contains several private methods and fields, such as 'private ContentSequenceDAO cSqdDAO' and 'private Agent verifyAgentRegistered'.
- FragmentAnnotationACTION.java - Navigator:** Located at the bottom, it provides a list of methods and fields from the current class, including 'createFragmentAnnotation', 'deleteFragmentAnnotation', 'findFragmentAnnotations', etc.
- Toolbar:** Contains standard IDE navigation icons like 'Go to Source', 'Go to Declaration', and 'Go to Implementation'.
- Status Bar:** Shows the current cursor position as '31:31 INS'.

COMPIL

Navigation dans le code

```
import org.kplab.smat.dao.impl.kr.KRDAOFactory;
import org.kplab.smat.dao.impl.kr.converter;
import org.kplab.smat.domainModel.Agent;
import org.kplab.smat.domainModel.FragmentA

/**
 *
 * @author dayre
 */
public class CRUDFragmentAnnotationDAOTest

    DAOFactory daoFactory;

    public CRUDFragmentAnnotationDAOTest()
        daoFactory = new KRDAOFactory();
    }

    // Create Read Update Delete FragmentAn
    public boolean CRUDFragmentAnnotationte
```

- Navigate
- Show Javadoc Alt+F1
- Find Usages Alt+F7
- Refactor
- Format Alt+Maj+F
- Fix Imports Ctrl+Maj+I
- Insert Code... Alt+Insérer
- Reverse Engineer...
- Run File Maj+F6
- Debug File Ctrl+Maj+F5
- Run Into Method Maj+F7
- New Watch... Ctrl+Maj+F7
- Toggle Line Breakpoint Ctrl+F8
- Profiling
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Code Folds
- Select in
- EJB Methods
- Enterprise Resources
- Web Service Client Resources

- Go to Source Ctrl+Maj+B
- Go to Declaration Ctrl+B
- Go to Super Implementation Ctrl+Maj+P
- Inspect Members Ctrl+Maj+F12
- Inspect Hierarchy Alt+Maj+F12

:47 INS

Type

Recherche de l'usage des classes

The screenshot illustrates the process of finding class usages in an IDE. A context menu is open over the `AnnotationScenarioACTION` class in the project tree. The `Find Usages` option is selected, showing the keyboard shortcut `Alt+F7`. The code editor displays the `FragmentAnnotationACTION` class, which uses `ContentSequenceDAO`, `ContentItemDAO`, `AnnotationPhaseDAO`, `AnnotationScenarioDAO`, and `FragmentAnnotationDAO`. The `Usages` window at the bottom shows the following usages:

```

import org.kplab.smat.controller.AnnotationScenarioACTION
f.contentItem = new AnnotationScenarioACTION().findWorkingContentItem(userURI)
f.contentItem = new AnnotationScenarioACTION().createAnnotationScenario(userURI, contentSequenceURI)
f.contentItem = new AnnotationScenarioACTION().deleteAnnotationScenario(userURI, contentSequenceURI, annotationScenarioURI)
f.contentItem = new AnnotationScenarioACTION().updateAnnotationScenario(userURI, contentSequenceURI, annotationScenarioURI, annotationScenarioName)
f.annotationScenario = new AnnotationScenarioACTION().getAnnotationScenarioDetail(userURI, contentSequenceURI, annotationScenarioURI)
f.annotationScenario = new AnnotationScenarioACTION().setAnnotationScenarioDetail(userURI, contentSequenceURI, annotationScenario)
    
```

COMPIL

Affichage de la documentation d'une classe

The screenshot displays an IDE interface with three main panels:

- Navigator:** Shows the project structure with `Clip.java` selected. Below, it lists the class's methods and fields, including `getDc_title() : String`, `getHasContentItem() : ContentItem`, and `date : Date`.
- Code Editor:** Shows the source code for `Clip`, which extends `ObjectOfActivity`. It includes fields like `mSegments`, `date`, `hasContentItem`, and `mAnnotationSequence`, along with a constructor and a `setDate` method.
- Documentation Panel:** Displays the Javadoc for `org.kplab.smat.domainModel.Clip`. It shows the `setDc_alternative` method signature and an "Overrides" section indicating that `setDc_alternative` is overridden from the `ObjectOfActivity` class.

At the bottom of the IDE, there is a status bar with the text "Press Ctrl+F1 to enlarge" and a "Usages" tab.

Refactoring du code

The screenshot shows an IDE interface with a project tree on the left, a code editor on the right, and a refactoring menu open over the file `AnnotationScenarioACTION.java`. The menu options are:

- Open
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Compile File (F9)
- Run File (Maj+F6)
- Debug File (Ctrl+Maj+F5)
- Profile File
- Add
- Delete (Supprimer)
- Save As Template...
- Find Usages (Alt+F7)
- Refactor** (highlighted)
 - Rename... (Ctrl+R)
 - Move...
 - Copy...
 - Safe Delete...
 - Change Method Parameters...
 - Encapsulate Fields...
 - Pull Up...
 - Push Down...
 - Extract Interface...
 - Extract Superclass...
 - Use Supertype Where Possible...
 - Move Inner to Outer Level...
 - Introduce Variable...
 - Introduce Constant...
 - Introduce Field...
- Reverse Engineer...
- File Members (Ctrl+F12)
- File Hierarchy (Alt+F12)
- Subversion
- Local History
- Tools
- Properties

The code editor shows the following imports and the start of a class:

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import org.kplab.smat.dao.AnnotationPhaseDAO;
import org.kplab.smat.dao.AnnotationScenarioDAO;
import org.kplab.smat.dao.ContentItemDAO;
import org.kplab.smat.dao.ContentSequenceDAO;
import org.kplab.smat.dao.DAOFactory;
import org.kplab.smat.dao.FragmentAnnotationDAO;
import org.kplab.smat.domainModel.AnnotationPhase;
import org.kplab.smat.domainModel.AnnotationScenario;
import org.kplab.smat.domainModel.ContentItem;
import org.kplab.smat.domainModel.ContentSequence;
import org.kplab.smat.form.AgentRuleForm;
import org.kplab.smat.ws.SMATException;

/**
 *
 * @author Morpheus
 */
public class FragmentAnnotationACT
```

2. L'utilisation de base de NetBeans

- a) Présentation de l'interface de Netbeans
- b) Création d'un projet
- c) Edition d'un projet
- d) Exécution d'un projet**
- e) Débogage d'un projet

COMPIL

Compilation, exécution d'un projet

The screenshot displays the NetBeans IDE 6.0.1 interface. The title bar reads "SMAT_Test - NetBeans IDE 6.0.1". The menu bar includes "File", "Edit", "View", "Navigate", "Source", "Refactor", "Build", "Run", "Profile", "Versioning", "Tools", "Window", and "Help". The toolbar contains various icons for file operations and execution, with a dropdown menu set to "<default config>".

The left sidebar shows a project tree with "SMAT" selected. A context menu is open over the "SMAT" project, listing actions such as "New", "Build", "Clean and Build", "Clean", "Generate Javadoc", "Run", "Debug", "Profile", "Test", "Set Configuration", "Set as Main Project", "Open Required Projects", "Close", "Rename...", "Move...", "Copy...", "Delete", "Find...", "Reverse Engineer...", "Versioning", "Local History", and "Properties". The "Run" option is highlighted in blue.

The right pane shows a code editor with the following Java code:

```
...va DaoFileClient.java x AgentDB.java x SMAT.java x CRUDSegr
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package Main;

import DAOtest.CRUDVideoDAOtest;

/**
 *
 * @author dayre
 */
public class Main {

    public static void main(String[] args) {

        // TODO code application logic here

        CRUDVideoDAOtest vt = new CRUDVideoDAOtest();

        vt.CRUDVideotest1();

    }
}
```

Exécution d'une classe avec Main

The screenshot shows an IDE interface with the following components:

- Project Explorer (Left):** Shows a project named 'SMAT_Test' with a 'Main' package containing 'CRUDVideoDAOTest.java'. A context menu is open over this file, listing actions like 'Open', 'Run File' (Maj+F6), and 'Debug File'.
- Code Editor (Right):** Displays the source code for 'CRUDVideoDAOTest.java'. The code includes a 'scenarioTest()' method and a 'main()' method that creates a 'CRUDVideoDAOTest' instance and calls its 'CRUDVideotest1()' method.
- Output Console (Bottom):** Shows a log of messages from 'WSIX-COMMON-2007', indicating successful execution of web service methods like 'setAuthorizedUserRoom' and 'deleteRoom'.

```

}

public boolean scenarioTest () {

    // get an agnet from the agent list

    // get video list

    // create a clip

    // add a clip to a video (video update)

    return true;
}

public static void main(String[] args) {
    // TODO code application logic here

    CRUDVideoDAOTest vt = new CRUDVideoDAOTest();

    vt.CRUDVideotest1();
}
}
    
```

Output Console Log:

```

WSIX-COMMON-2007: Map CMT EJB web method 'org.kplab.ms.server.ws.mswebservice.'openroom' with effective transaction attribute of 'REQUIRED'
WSIX-COMMON-2007: Map CMT EJB web method 'org.kplab.ms.server.ws.mswebservice.'setAuthorizedUserRoom' with effective transaction attribute of 'REQUIRED'
WSIX-COMMON-2007: Map CMT EJB web method 'org.kplab.ms.server.ws.mswebservice.'getAuthorizedUserRoom' with effective transaction attribute of 'REQUIRED'
WSIX-COMMON-2007: Map CMT EJB web method 'org.kplab.ms.server.ws.mswebservice.'deleteRoom' with effective transaction attribute of 'REQUIRED'
    
```

2. L'utilisation de base de NetBeans

- a) Présentation de l'interface de Netbeans
- b) Création d'un projet
- c) Edition d'un projet
- d) Exécution d'un projet
- e) Débogage d'un projet**

COMPIL

Sélection d'un débogueur

The screenshot shows the NetBeans IDE interface with the 'Run' menu open. The menu items are as follows:

- Run Main Project (F6)
- Debug Main Project (Ctrl+F5)
- Test "SMAT_Test" (Alt+F6)
- Run File
- Attach Debugger...
- Finish Debugger Session (Maj+F5)
- Pause
- Continue (F5)
- Step Over (F8)
- Step Over Expression (Maj+F8)
- Step Into (F7)
- Step Out (Ctrl+F7)
- Run to Cursor (F4)
- Run Into Method (Maj+F7)
- Apply Code Changes
- Stack
- Toggle Line Breakpoint (Ctrl+F8)
- New Breakpoint... (Ctrl+Maj+F8)
- New Watch... (Ctrl+Maj+F7)
- Evaluate Expression... (Ctrl+F9)

The main editor window displays the following Java code:

```
package org.kplab.smat.domainModel;

import java.net.URI;
import java.util.Date;
import java.util.List;

/**
 * @author David
 */
public class Circle extends Region {

    private int x;
    private int y;
    private int radius;

    public Circle() {
    }

    public int getRadius() {
        return radius;
    }

    public void setRadius(int radius) {
        this.radius = radius;
    }

    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }
}
```

COMPIL

Insertion d'un point d'arrêt ligne

The screenshot shows the NetBeans IDE interface with the following components:

- Projects View:** Shows a project named 'SMAT_Test' with a package structure including 'DAO_Test' and 'Main'.
- Editor:** Displays the source code of 'CRUDVideoDAOTest.java'. A red vertical line indicates a breakpoint set on the line: `List<Video> lvs = cd.findVideos();`. A context menu is open over this line, listing various actions such as 'Navigate', 'Show Javadoc', 'Find Usages', 'Refactor', 'Format', 'Fix Imports', 'Insert Code...', 'Reverse Engineer...', 'Run File', 'Debug File', 'Run Into Method', 'New Watch...', 'Toggle Line Breakpoint', 'Profiling', 'Cut', 'Copy', 'Paste', 'Code Folds', 'Select in', 'EJB Methods', 'Enterprise Resources', and 'Web Service Client Resources'.
- Output View:** Shows the execution log of the program, including messages like 'Listening on 2676', 'Not able to submit breakpoint LineBreakpoint UserManagement.java : 9, reason: The breakpoint is set outside of any class.', and 'Invalid LineBreakpoint UserManagement.java : 9'.

COMPIL

Insertion d'un point d'arrêt méthode

The screenshot displays the NetBeans IDE interface. The main editor window shows the file `CRUDVideoDAOTest.java` with the following code:

```
DaoFileClient.java x AgentDB.java x SMAT.java x CRUDSegmentDAOTest11.java x CRUDVideoDAOTest.java x CRUDClipDAOTest.java x Main.java x  
v.setDc_creator(new Agent()); // que ce passe-t-il si l'agent n'existe pas ?  
v.setDc_created(new Date());  
v.setDc_description("It is a video contentItem for DAO testing");  
v.setDc_title("DAOVideoTest1");  
v.setContentURI("ftp://kplab/kplab/childbirth.flv"); //new URI ("ftp://kplab/kplab/childbirth.flv");  
  
// Create  
v.setUri(cd.createVideoByObject(v));  
System.out.println("--- Video Created: " + v.getUri());  
  
// Read  
List<Video> lvs = cd.findVideos();  
System.out.println("--- Number of Videos read: " + lvs.size());  
  
Iterator it = lvs.iterator();  
  
while (it.hasNext()) {  
    v = (Video) it.next();  
    // TODO: verifier de la semantique de l'egalite  
    //  
    System.out.println("- Video: " + v.getDc_title() + " - Description : " + v.getDc_description() + " - URI : " + v.getUri() + " - Date : " + v.ge  
}  
  
// Update  
System.out.println("--- Video updated: Start : " + v.getUri());  
String uri = new String("http://server/file");  
v.setContentURI(uri);  
cd.updateVideoByObject(v);  
System.out.println("--- Video updated: End : " + v.getUri());  
  
// Retrieve of the updated video  
Video v2 = cd.findOneVideo(v.getUri(), null);  
System.out.println("--- Video updated retrieved by URI: " + v2.getUri());
```

The `while` loop is highlighted in blue, and a red line indicates a breakpoint is set on the line `v = (Video) it.next();`. The `Run` menu is open, showing options like `Run Main Project`, `Attach Debugger...`, `Pause`, `Continue`, `Step Over`, `Step Into`, `Toggle Line Breakpoint`, `New Breakpoint...`, `New Watch...`, and `Evaluate Expression...`. The `Output` window at the bottom shows the following log:

```
http://trac.kp-lab.org/svn x Java DB Database Process x GlassFish V2 x Debugger Console x  
Listening on 2676  
Not able to submit breakpoint LineBreakpoint UserManagement.java : 9, reason: The breakpoint is set outside of any class.  
Invalid LineBreakpoint UserManagement.java : 9  
User program running  
LineBreakpoint CRUDVideoDAOTest.java : 59 successfully submitted.  
LineBreakpoint CRUDVideoDAOTest.java : 56 successfully submitted.  
User program finished
```

COMPIL

Insertion d'un point d'arrêt méthode

The screenshot displays the NetBeans IDE interface with the following components:

- Debugger Console:** Shows the execution log with messages such as "Listening on 2676", "Not able to submit breakpoint LineBreakpoint UserManagement.java : 9, reason: The breakpoint is set outside of any class.", and "LineBreakpoint CRUDVideoDAOTest.java : 59 successfully submitted."
- Project Explorer:** Shows the project structure for SMAT_Test, including source packages like DAOVideoDAOTest and Main.
- Code Editor:** Displays the source code of DAOVideoDAOTest.java with several lines highlighted in red, indicating the location of the breakpoint.
- New Breakpoint Dialog:** A modal dialog box is open, configured with the following settings:
 - Debugger: Java
 - Breakpoint Type: Method
 - Class Name: DAOVideoDAOTest
 - Method Name: CRUDVideoDAOTest1 ()
 - Stop On: Method Entry
 - Suspend: All threads
 - Print Text: Method breakpoint hit in {className}.{methodName} at line {

COMPIL

Insertion d'un point d'arrêt méthode

The screenshot displays the NetBeans IDE interface for SMAT_Test - NetBeans IDE 6.0.1. The main editor window shows the source code of the `CRUDVideoDAOTest` class. A **Method Breakpoint** is set on the `CRUDVideotest1()` method. The code includes a constructor, a `CRUDVideotest1()` method, and a `daoFactory` field. The `CRUDVideotest1()` method contains logic for creating and reading video objects. The `daoFactory` field is of type `DAOFactory`.

```
public class CRUDVideoDAOTest {  
  
    DAOFactory daoFactory;  
  
    public CRUDVideoDAOTest() {  
        daoFactory = new KRDAOFactory();  
    }  
  
    Method Breakpoint  
    public boolean CRUDVideotest1() {  
        //try {  
  
        VideoDAO cd = daoFactory.getVideoDAO();  
  
        //  
        System.out.println("--- Video CRUD Start Test: " + new Date());  
  
        // Create a video  
        Video v = new Video();  
        v.setDc_creator(new Agent()); // que ce passe-t-il si l'agent n'existe pas ?  
        v.setDc_created(new Date());  
        v.setDc_description("It is a video contentItem for DAO testing ");  
        v.setDc_title("DAOVideoTest1");  
        v.setContentURI("ftp://kplab/kplab/childbirth.flv"); //new URI ("ftp://kplab/kplab/childbirth.flv");  
  
        // Create  
        v.setUri(cd.createVideoByObject(v));  
        System.out.println("--- Video Created: " + v.getUri());  
  
        // Read  
        List<Video> lvs = cd.findVideos();  
        System.out.println("--- Number of Videos read: " + lvs.size());  
  
        Iterator it = lvs.iterator();  
  
    }  
}
```

The **Output** window shows the following messages:

```
http://trac.kp-lab.org/svn x Java DB Database Process x GlassFish V2 x Debugger Console x  
Listening on 2676  
Not able to submit breakpoint LineBreakpoint UserManagement.java : 9, reason: The breakpoint is set outside of any class.  
Invalid LineBreakpoint UserManagement.java : 9  
User program running  
LineBreakpoint CRUDVideoDAOTest.java : 59 successfully submitted.  
LineBreakpoint CRUDVideoDAOTest.java : 56 successfully submitted.  
User program finished
```

COMPIL

Exécution pas à pas

MultimediaAnnotationServices - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Projects: Files

- Test Packages
- Libraries
- Test Libraries
- msClient
 - Source Packages
 - msclient
 - Main.java
- Test Packages
- Libraries
- Test Libraries
- MultimediaAnnotationServices
 - Web Pages
 - Web Services
 - Configuration Files
 - Server Resources
 - Source Packages
 - <default package>
 - org.kplab.smat.comparator
 - ContentSegmentComparedByOrder.java
 - org.kplab.smat.controller
 - AgentACTION.java
 - AnnotationPhaseACTION.java
 - AnnotationScenarioACTION.java
 - ContentItemACTION.java
 - ContentSegmentACTION.java
 - ContentSequenceACTION.java
 - DomainOntologyACTION.java
 - FragmentAnnotationACTION.java
 - PdfACTION.java
 - SmatAgentSessionManagement.java
 - org.kplab.smat.dao
 - org.kplab.smat.dao.impl.kr
 - org.kplab.smat.dao.impl.mysql
 - org.kplab.smat.dao.impl.ums
 - org.kplab.smat.domainModel
 - org.kplab.smat.form
 - org.kplab.smat.servlet
 - org.kplab.smat.userProfilingTO
 - org.kplab.smat.ws.annotationActivity
 - org.kplab.smat.ws.configureAnnotationSession
 - org.kplab.smat.ws.launchAnnotationTool

FragmentAnnotationACTION.java - Navigator

- createFragmentAnnotation(URI userURI, URI segmentURI, URI phaseURI, Fragme...
- deleteFragmentAnnotation(URI userURI, URI phaseURI, URI annotationURI) : Li...
- findFragmentAnnotations(URI userURI, URI phaseURI) : List<FragmentAnnotation...
- findLeaderFragmentAnnotations(URI userURI, URI phaseURI) : List<FragmentAnn...
- moveFragmentAnnotation(URI userURI, URI phaseScrURI, URI phaseDestURI, UR...
- verifyAccessOnSequence(URI userURI, URI contentItemURI, URI sequenceURI) :
- verifyAgentRegistered(URI userURI) : Agent
- verifyContentItemChoice(URI userURI) : URI
- annotateDAO : FragmentAnnotationDAO
- cSqdDAO : ContentSequenceDAO
- ciDAO : ContentItemDAO

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package org.kplab.smat.controller;

import java.net.URI;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import org.kplab.smat.dao.AnnotationPhaseDAO;
import org.kplab.smat.dao.AnnotationScenarioDAO;
import org.kplab.smat.dao.ContentItemDAO;
import org.kplab.smat.dao.ContentSequenceDAO;
import org.kplab.smat.dao.DAOFactory;
import org.kplab.smat.dao.FragmentAnnotationDAO;
import org.kplab.smat.domainModel.Agent;
import org.kplab.smat.domainModel.AnnotationPhase;
import org.kplab.smat.domainModel.ContentItem;
import org.kplab.smat.domainModel.FragmentAnnotation;
import org.kplab.smat.form.AgentRules;
import org.kplab.smat.ws.smatException.SmatException;

/**
 *
 * @author Morpheus
 */
public class FragmentAnnotationACTION {

    private ContentSequenceDAO cSqdDAO = DAOFactory.getDAOFactory().getContentSequenceDAO();
    private ContentItemDAO ciDAO = DAOFactory.getDAOFactory().getContentItemDAO(null);
    private AnnotationPhaseDAO phaseDAO = DAOFactory.getDAOFactory().getAnnotationPhaseDAO();
    private AnnotationScenarioDAO scenarioDAO = DAOFactory.getDAOFactory().getAnnotationScenarioDAO();
    private FragmentAnnotationDAO annotatedDAO = DAOFactory.getDAOFactory().getFragmentAnnotationDAO(null);

    private SmatAgentSessionManagement sasm = SmatAgentSessionManagement.getInstance();

    private Agent verifyAgentRegistered(URI userURI) throws SmatException{
        Agent agent = (Agent) sasm.getAgentMap().get(userURI);
        if(agent == null){
            throw new SmatException("Agent not registered", null);
        }
        return agent;
    }

    private URI verifyContentItemChoice(URI userURI) throws SmatException{
        // verify if user have choice a ContentItem
        URI contentItemURI = (URI) sasm.getContentItemURIMap().get(userURI);
        if(contentItemURI == null){
            throw new SmatException("You don't choice ContentItem", null);
        }
        return contentItemURI;
    }
}
```

COMPIL

Navigation dans la pile d'appel (step ...)

SMAT_Test - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

205,6 / 367,2MB

CRUDVideoDAOTest.java x CRUDClipDAOTest.java x Main.java x KRClipDAO.java x CRUDFragmentAnnotationDAOTest.java x KRDAOFactory.java x PersistenceM...

Test Packages
Libraries
Test Libraries
MS
msClient
MultimediaAnnotationServices
SMAT
SMAT_Test
Source Packages
DAOTest
@CRUDAgentDAOTest1.java
@CRUDAnnotationPhaseDAOTest11.java
@CRUDAnnotationSequenceDAOTest11.j
@CRUDClipDAOTest.java
@CRUDFragmentAnnotationDAOTest.java
@CRUDFragmentDAOTest11.java
@CRUDSegmentDAOTest11.java
CRUDVideoDAOTest.java
Main
WSTest
WS_Test
@LaunchAnnotationWSTest.java
smat_test.AnnotationActivityWSClient
smat_test.LaunchWSClient
Test Packages
DAOTest
@CRUDVideoDAOTestTest.java
Web Service References
Libraries
Test Libraries
JUnit 4.1 - junit-4.1.jar
SMAT-Model
SMATServer

Main.java - Navigator
Main
main(String[] args)

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package Main;  
  
import DAOTest.CRUDVideoDAOTest;  
  
/**  
 *  
 * @author dayre  
 */  
public class Main {  
  
    public static void main(String[] args) {  
  
        // TODO code application logic here  
  
        CRUDVideoDAOTest vt = new CRUDVideoDAOTest();  
  
        vt.CRUDVideotest1();  
  
    }  
  
}
```

Stack

Toggle Line Breakpoint Ctrl+F8
New Breakpoint... Ctrl+Maj+F8
New Watch... Ctrl+Maj+F7
Evaluate Expression... Ctrl+F9

Output

JUnit Test Results Usages Task List VM Telemetry Overview Watches Local Variables Call Stack Breakpoints

http://trac.kp-lab.org/svn Java DB Database Process
GlassFish V2 Reverse Engineering Log SMAT_Test (debug-stepinto) Debugger Console

Listening on 2457
Not able to submit breakpoint LineBreakpoint UserManager.java : 9, reason: The breakpoint is set outside of any class.
Invalid LineBreakpoint UserManager.java : 9
User: program running

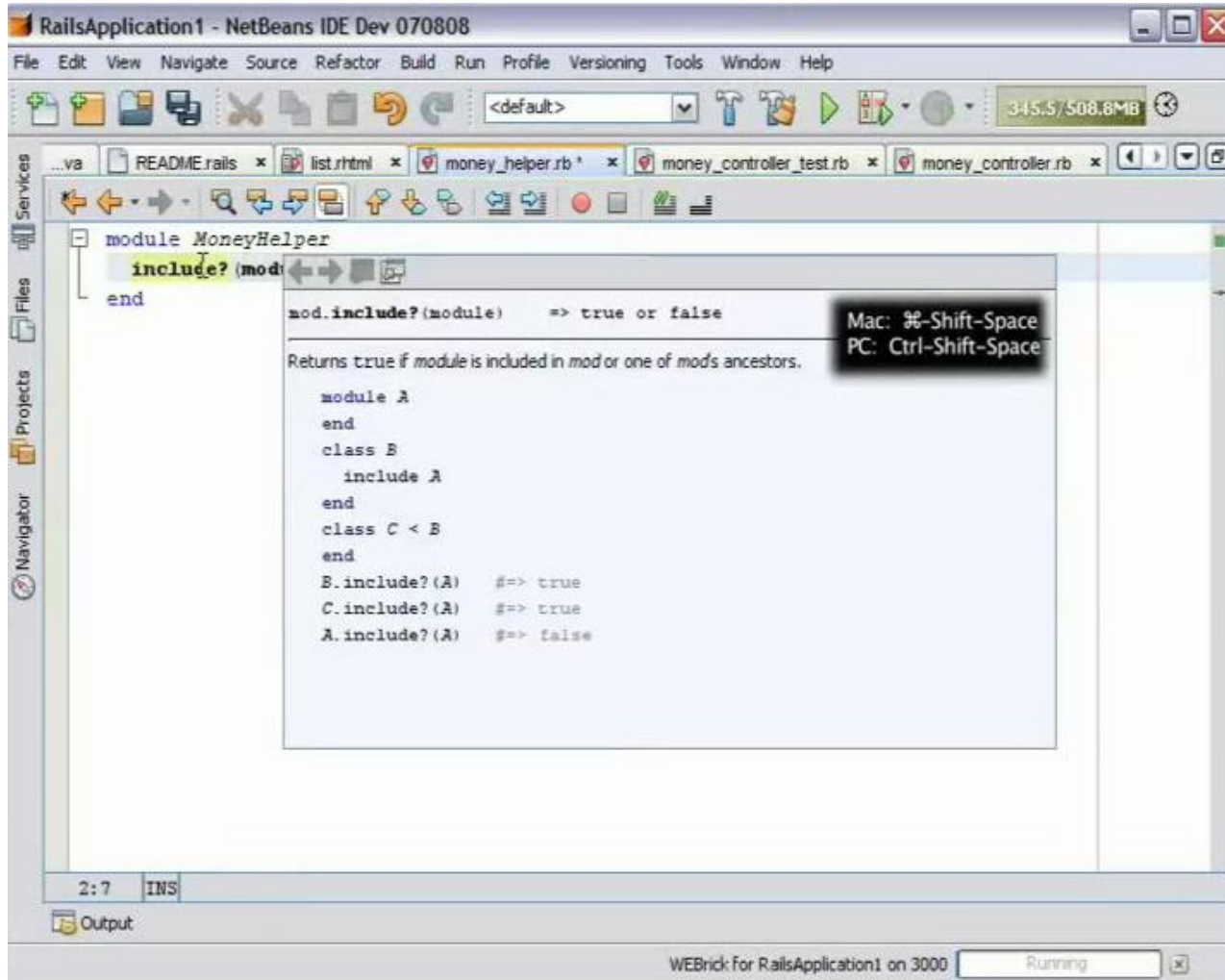
SMAT_Test (debug-stepinto) running...

démarrer Gestonn... 9 Mozil... SMAT_Te... 6 Open... ZA ZoneAlarm 6 Oper... Invite de ... backup0... 2 Adob... 2 Expl... Microsoft... Microsoft... Intel(R) P... FR 15:45

3. Utilisation avancée de Netbeans

- a) Gestion de la documentation
- b) Listes des tâches à effectuer
- c) Personnalisation de son IDE : Installation d'un plugin
- d) Importation de bibliothèques et de projets
- e) Tests unitaires avec Junits
- f) Profiling
- g) Gestion de version
- h) Reverse engineering

Accéder à la documentation d'une classe



COMPIL

Générer la documentation d'un projet

The screenshot shows the NetBeans IDE interface. The 'Projects' window on the left displays a project structure for 'MultimediaAnnotationServices'. A context menu is open over the project, with 'Generate Javadoc' selected. The main editor window shows the source code for 'FragmentAnnotationACTION.java'. The 'Output' window at the bottom displays the Javadoc generation process, including file paths and warnings.

```
package org.kplab.smat.controller;

import java.net.URI;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import org.kplab.smat.dao.AnnotationPhaseDAO;
import org.kplab.smat.dao.AnnotationScenarioDAO;
import org.kplab.smat.dao.ContentItemDAO;
import org.kplab.smat.dao.ContentSequenceDAO;
import org.kplab.smat.dao.DAOFactory;
import org.kplab.smat.dao.FragmentAnnotationDAO;
import org.kplab.smat.domainModel.Agent;
import org.kplab.smat.domainModel.AnnotationPhase;
import org.kplab.smat.domainModel.ContentItem;
import org.kplab.smat.domainModel.FragmentAnnotation;
import org.kplab.smat.form.AgentRules;
import org.kplab.smat.ws.smatException.SmatException;

/**
 *
 * @author Morpheus
 */
public class FragmentAnnotationACTION {

    private ContentSequenceDAO cSeqDAO = DAOFactory.getDAOFactory().getContentSequenceDAO();
    private ContentItemDAO ciDAO = DAOFactory.getDAOFactory().getContentItemDAO(null);
    private AnnotationPhaseDAO phaseDAO = DAOFactory.getDAOFactory().getAnnotationPhaseDAO();
    private AnnotationScenarioDAO scenarioDAO = DAOFactory.getDAOFactory().getAnnotationScenarioDAO();
    private FragmentAnnotationDAO annotateDAO = DAOFactory.getDAOFactory().getFragmentAnnotationDAO(null);
}
```

Output window content:

```
C:\dev\netbean\M36\SMAT2\server\MultimediaAnnotationServices\trunk\src\java\org\kplab\smat\controller\ContentSequenceACTION.java:147: warning - @return tag has no arguments.
C:\dev\netbean\M36\SMAT2\server\MultimediaAnnotationServices\trunk\src\java\org\kplab\smat\controller\DomainOntologyACTION.java:42: warning - @param argument "ontologiesURI" is not a parameter name.
C:\dev\netbean\M36\SMAT2\server\MultimediaAnnotationServices\trunk\src\java\org\kplab\smat\dao\ContentSegmentDAO.java:27: warning - @return tag has no arguments.
C:\dev\netbean\M36\SMAT2\server\MultimediaAnnotationServices\trunk\src\java\org\kplab\smat\dao\ContentSegmentDAO.java:41: warning - @return tag has no arguments.
C:\dev\netbean\M36\SMAT2\server\MultimediaAnnotationServices\trunk\src\java\org\kplab\smat\dao\ContentSegmentDAO.java:48: warning - @param argument "segment" is not a parameter name.
C:\dev\netbean\M36\SMAT2\server\MultimediaAnnotationServices\trunk\src\java\org\kplab\smat\dao\impl\mysql\MysqlContentSegmentDAO.java:35: warning - @return tag has no arguments.
C:\dev\netbean\M36\SMAT2\server\MultimediaAnnotationServices\trunk\src\java\org\kplab\smat\dao\impl\mysql\MysqlContentSegmentDAO.java:124: warning - @return tag has no arguments.
C:\dev\netbean\M36\SMAT2\server\MultimediaAnnotationServices\trunk\src\java\org\kplab\smat\dao\impl\mysql\MysqlContentSegmentDAO.java:274: warning - @param argument "segment" is not a parameter name.
Building index for all the packages and classes...
Building index for all classes...
Generating C:\dev\netbean\M36\SMAT2\server\MultimediaAnnotationServices\trunk\dist\javadoc\stylesheet.css...
47 warnings
javadoc-browse:
Browsing: file:/C:/dev/netbean/M36/SMAT2/server/MultimediaAnnotationServices/trunk/dist/javadoc/index.html
javadoc:
BUILD SUCCESSFUL (total time: 26 seconds)
```

COMPIL

Liste des tâches à effectuer

The screenshot displays the NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Build, Run, Profile, Versioning, Tools, Window, and Help. The 'Window' menu is open, showing options like Projects, Files, Classes, Services, Favorites, Palette, Properties, Task List (highlighted), Output, Navigating, Debugging, Profiling, Versioning, Other, Editor, and Processes. The 'Task List' window is open at the bottom, displaying a table of tasks.

Description	File	Line	Location
TODO process result here	LaunchAnnotationWSTest.java	146	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO handle custom exceptions here	LaunchAnnotationWSTest.java	150	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO initialize WS operation arguments here	LaunchAnnotationWSTest.java	163	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO process result here	LaunchAnnotationWSTest.java	165	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO handle custom exceptions here	LaunchAnnotationWSTest.java	169	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO initialize WS operation arguments here	LaunchAnnotationWSTest.java	183	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO process result here	LaunchAnnotationWSTest.java	187	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
port.getClipList(userSID, SMATSession); // TODO sessionID in place	LaunchAnnotationWSTest.java	189	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO handle custom exceptions here	LaunchAnnotationWSTest.java	192	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO initialize WS operation arguments here	LaunchAnnotationWSTest.java	213	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO: add concepts to the annotation TODO TODO TODO TODO	LaunchAnnotationWSTest.java	231	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO process result here	LaunchAnnotationWSTest.java	233	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO Stupide de retourner la liste des annotations	LaunchAnnotationWSTest.java	234	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO handle custom exceptions here	LaunchAnnotationWSTest.java	240	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO code application logic here	LaunchAnnotationWSTest.java	267	C:/dev/netbean/M24/SMAT_Test/src/WS_Test

COMPIL

Installation d'un plugin

SMAT - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Projects: M24_SVN, MS, Enterprise Beans, Configuration Files, Server Resources, Source Packages, Test Packages, Libraries, Test Libraries, MS, msClient, MultimediaAnnotationServices, SMAT, Enterprise Beans, Web Services, Web Service References, Configuration Files, Server Resources, Source Packages, org.kplab.smat.dao, org.kplab.smat.dao.ejb, PersistenceManagerBean.java, PersistenceManagerLocal.java, org.kplab.smat.dao.impl.kr, org.kplab.smat.dao.impl.kr.converter, org.kplab.smat.dao.impl.kr.converter.error, org.kplab.smat.dao.impl.kr.persistenceBeans, org.kplab.smat.domainModel, org.kplab.smat.domainModel.localController, org.kplab.smat.domainModel.to, org.kplab.smat.server.controller, org.kplab.smat.server.ejb, org.kplab.smat.server.ejb.user

Circle.java - Navigator

Output

Description	Location
TODO process result here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO handle custom exceptions here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO initialize WS operation arguments here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO process result here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO handle custom exceptions here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO initialize WS operation arguments here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO process result here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO handle custom exceptions here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO initialize WS operation arguments here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO process result here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO handle custom exceptions here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO initialize WS operation arguments here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO: add concepts to the annotation TODO TODO TODO TODO	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO process result here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO Stupide de retourner la liste des annotations	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO handle custom exceptions here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO code application logic here	C:/dev/netbean/M24/SMAT_Test/src/WS_Test
TODO: 44	

Task scanning

Importation de bibliothèques et de projets

The screenshot displays the NetBeans IDE interface with several key components:

- Project Properties - SMAT_Test:** Shows the 'Compile' tab with a list of compile-time libraries. The 'Java Platform' is set to 'JDK 1.6 (Default)'. The list includes various JAR files such as 'activation.jar', 'commons-collections-3.1.1.jar', and 'commons-httpclient-3.0.jar'.
- Add Library:** A dialog box for adding external libraries. The 'Libraries' list includes 'Absolute Layout', 'ActiveSync Java Wrapper Library', 'ANT Contrib', 'Beans Binding', 'CDC Build System Ant Utilities', 'CopyLibs Task', 'Default Theme', 'Exception Handler', 'Gray Theme', and 'Green Theme'.
- Library Manager:** A dialog for managing installed libraries. The 'Library Name' is 'Swing Layout Extensions'. The 'Library Classpath' is set to 'C:\Program Files\NetBeans 6.0.1\platform'.
- Output:** Shows the results of a deployment process, including messages like 'Undeployment of application MS completed successfully' and 'BUILD SUCCESSFUL (total time: 40 seconds)'.
- Background IDE:** Shows the 'Projects' and 'Files' panes on the left, and the 'Main.java - Navigator' pane at the bottom left. The main editor area is partially visible on the right.

COMPIL

Tests unitaire avec Junits (création)

SMAT_Test - NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Projects: org.kplab.smat.session.ejb, Test Packages, Libraries, Test Libraries, SMAT_Test, Source Packages, DAOtest, CRUDAgentDAOtest1.java, CRUDAnnotationPhaseDAOtest11.java, CRUDAnnotationSequenceDAOtest11.java, CRUDClipDAOtest.java, CRUDFragmentAnnotationDAOtest.java, CRUDFragmentDAOtest11.java, CRUDSegmentDAOtest11.java, CRUDVideoDAOtest.java, Main, WSTest, WS_Test, smat_test.AnnotationActivityWSClient, smat_test.LaunchWSClient, Test Packages, Web Service References, Libraries, Test Libraries, JUnit 4.1 - junit-4.1.jar, SMAT-Model, SMATServer

CRUDVideoDAOtest.java - Navigator: CRUDVideoDAOtest, CRUDVideoDAOtest(), CRUDVideotest1(): boolean, main(String[] args), scenarioTest(): boolean, daoFactory: DAOFactory

Output: No tests executed.

JUnit Test Results: Usages

Create Tests Dialog Box:

Class to Test: DAOtest.CRUDVideoDAOtest

Class Name: DAOtest.CRUDVideoDAOtestTest

Location: Test Packages

Code Generation:

Method Access Levels	Generated Code
<input checked="" type="checkbox"/> Public	<input checked="" type="checkbox"/> Test Initializer
<input checked="" type="checkbox"/> Protected	<input checked="" type="checkbox"/> Test Finalizer
<input checked="" type="checkbox"/> Package Private	<input checked="" type="checkbox"/> Default Method Bodies

Generated Comments:

- Javadoc Comments
- Source Code Hints

Help: Create Tests Dialog Box

This dialog box lets you specify the options for generating JUnit test classes and suites for existing Java classes. You open the dialog box by right-clicking any Java file, package, or the Source Packages node and choosing Tools > Create JUnit Tests (Ctrl-Shift-U).

In addition to specifying the location of the generated test classes, you can specify the following options:

- Method Access Levels.** Specifies which methods to generate test methods for.
- Class Types.** Specifies which class types to generate test classes for. This option is only available when generating tests for multiple classes.
- Optional Classes.** Specifies whether to generate a test suite or whether to just generate the individual test classes. This option is only available when generating tests for multiple classes.
- Generated Code.** Specifies whether to generate the following:
 - Test Initializer.** Method for tasks like initializing fields, enabling on logging, and resetting environment variables.
 - Test Finalizer.** Method for cleaning up after tests have finished running.
 - Default Method Bodies.** Instructions for printing out information about test status to the command output.
- Generated Comments.** Specifies whether to create Javadoc comments for the test methods and comments that suggest how to implement the test methods.

See Also: [Creating a JUnit Test](#), [Running a JUnit Test](#), [Configuring JUnit](#), [About JUnit](#)

Legal Notices

Choix et paramétrage d'un profiler

The screenshot displays the NetBeans IDE interface. On the left, the 'Projects' and 'Files' panels show a project named 'SMAT_Test' with a package structure including 'DAOTest' and 'Main'. The 'CRUDVideoDAOTestTest.java - Navigator' panel shows the class structure with methods like 'main()', 'scenarioTest()', 'setUp()', 'setUpClass()', 'tearDown()', and 'tearDownClass()'. The main editor window shows the source code of 'CRUDVideoDAOTestTest.java', featuring a test method 'CRUDVideotest1()' that uses 'System.out.println()', 'new CRUDVideoDAOTest()', 'boolean' variables, and 'assertEquals()'. An 'Attach Profiler' dialog box is open, allowing configuration of profiling options. The 'Attach to' field is set to '<External Application>'. The 'Monitor' section includes 'CPU' and 'Memory' (selected). Under 'Memory', 'Record object creation only' is selected, and 'Track every 10 object allocations' is set. The 'Attach Mode' is 'No attach settings defined, define...'. The 'Output' panel at the bottom shows 'No tests executed.' and the 'JUnit Test Results' panel is empty.

Gestion des versions

SubVersion (SVN) : dépôt centralisé de sources
[http://fr.wikipedia.org/wiki/Subversion_\(logiciel\)](http://fr.wikipedia.org/wiki/Subversion_(logiciel))

Plugin SVN de Collabnet

<http://www.collab.net/downloads/subversion/>

- Chargement d'un projet sur un dépôt SVN
- Récupération d'un projet d'un dépôt SVN
- Mettre à jour le SVN
- Comparer sa version locale avec celle du dépôt
- Mettre à jour sa copie locale du projet
- Synchronisation d'une classe
- Historique des versions locales d'une classe

COMPIL

SVN: Récupération d'un projet sur un serveur SVN

The screenshot shows the NetBeans IDE 6.0.1 interface. The main editor displays a Java file with the following code:

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package Main;  
  
import DAOTest.CRUDVideoDAOTest;
```

The 'Checkout' dialog is open, showing the following steps and options:

- Steps:**
 - Subversion Repository
 - Folders to Checkout
- Folders to Checkout:**
 - Repository Folder(s): [Browse...]
 - Repository Revision: [Search...]
(empty means repository HEAD)
 - Skip "<selected-folder>" and checkout only its content
 - Specify the local folder to checkout folders into.
Local Folder: C:\dev\netbean\M36 [Browse...]
(local Subversion working copy)
 - Scan for Netbeans Projects after Checkout

The 'Browse Repository Folders' dialog is also open, showing a table of repository folders:

http://trac.kp-lab.org/svn	Revision	Date	Author	History
http://trac.kp-lab.org/svn	START			
.project	3647	25 avr. 2008 09:36:17	visapa	
wp2	4905	15 sept. 2008 15:45:04	cgeorges	
wp4	4462	8 juil. 2008 23:39:53	sfaggi	
wp5	24	18 sept. 2007 12:19:27	Patrick Aus...	
wp6	5479	13 oct. 2008 16:48:20	alir	

COMPIL

SVN: Mettre à jour le SVN

The screenshot shows the NetBeans IDE 6.0.1 interface. The 'SVN' menu is open, displaying options such as 'Checkout...', 'Import into Repository...', 'Update Project with Dependencies', 'Show Changes', 'Diff', 'Update', 'Commit...', 'Export Diff Patch...', 'Apply Diff Patch...', 'Copy to...', 'Switch to...', 'Merge to...', 'Show Annotations', 'Search History...', 'Revert Modifications...', 'Resolve Conflicts...', 'Ignore', and 'Svn Properties'. The 'Main.java' file is open in the editor, showing a `main` method. The 'Output' window at the bottom displays the following text:

```
http://trac.kp-lab.org/svn x MS (run) x Java DB Database Process x GlassFish V2 x
Undeployment of application MS completed successfully
All operations completed successfully
In-place deployment at C:\dev\netbean\M24\MS\MS\build\jar
Start registering the project's server resources
Finished registering server resources
moduleID=MS
deployment started : 0%
deployment finished : 100%
Deploying application in domain completed successfully
Trying to create reference for application in target server completed successfully
Trying to start application in target server completed successfully
Deployment of application MS completed successfully
Enable of MS in target server completed successfully
Enable of application in all targets completed successfully
All operations completed successfully
post-run-deploy:
run-deploy:
run:
BUILD SUCCESSFUL (total time: 40 seconds)
```

COMPIL

SVN: Comparer sa version locale avec celle du dépôt

The screenshot displays the NetBeans IDE interface with the 'Subversion' menu open. The menu options include: Checkout..., Relocate..., Update Project with Dependencies, Show Changes, Diff, Update, Commit..., Export Diff Patch..., Apply Diff Patch..., Copy to..., Switch to..., Merge to..., Show Annotations, Search History..., Revert Modifications..., Resolve Conflicts..., Ignore, and Svn Properties. The 'Output' window at the bottom shows the following log:

```
http://trac.kp-lab.org/svn x MS (run) x Java DB Database Process x GlassFish V2 x
Undeployment of application MS completed successfully
All operations completed successfully
In-place deployment at C:\dev\netbean\M24\MS\MS\build\jar
Start registering the project's server resources
Finished registering server resources
moduleID=MS
deployment started : 0%
deployment finished : 100%
Deploying application in domain completed successfully
Trying to create reference for application in target server completed successfully
Trying to start application in target server completed successfully
Deployment of application MS completed successfully
Enable of MS in target server completed successfully
Enable of application in all targets completed successfully
All operations completed successfully
post-run-deploy:
run-deploy:
run:
BUILD SUCCESSFUL (total time: 40 seconds)
```

COMPIL

SVN: historique des versions locales

The screenshot displays the NetBeans IDE 6.0.1 interface. The main window shows the local history of the file `KRDAOFactory.java`. The history table lists a single entry: "6 mai 2008 10:50:38". The editor displays the code for this version, with a diff view on the right side. The diff shows changes in the `public KRDAOFactory()` method, specifically the initialization of the `properties` map and the `manager` field. The `closeManager()` and `getClipDAO()` methods are also visible. The `Output` window at the bottom shows the compilation process, including warnings and an exception related to WSDL access. The `Watches` window is currently empty.

NetBeans IDE 6.0.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Projects: Files Services

...va @ CRUDDaoTest.java x Main.java x KRClipDAO.java x CRUDDaoTest.java x KRDAOFactory.java x KRDAOFactory.java [Local History] x

Version	Label
Older than 6 days ago	
6 mai 2008 10:50:38	

KRDAOFactory.java 6 mai 2008 10:50:38 1/1 Current File

```
private FragmentAnnotationDAO fragmentAnnotationDAO = null;
private FragmentDAO fragmentDAO = null;
private AnnotationPhaseDAO annotationPhaseDAO = null;
private AnnotationSequenceDAO annotationSequenceDAO = null;

public KRDAOFactory() {

    Map<String, String> properties = new HashMap<String, String>();
    properties.put("serviceURL", "http://139.91.183.100:3027/Swkm");
    manager = EntityManagerFactory.createEntityManager(properties);

}

public void closeManager() {
    manager.close();
}

@Override
public ClipDAO getClipDAO() {
    if (clipDAO == null) {
        clipDAO = new KRClipDAO(manager);
    }
    return clipDAO;
}
```

public KRDAOFactory() {

 Map<String, String> properties = new HashMap<String, String>();
 //properties.put("serviceURL", "http://139.91.183.100:3027/Swkm");
 // Production: http://athena.ics.forth.gr:3027/SwkmMiddle
 // for the development server (mielikki.mobile.evtek.fi:3027)
 properties.put("serviceURL", "http://kr.mobile.evtek.fi:3027/SwkmMiddle");
 manager = EntityManagerFactory.createEntityManager(properties);

}

public void closeManager() {
 manager.close();
}

@Override
public ClipDAO getClipDAO() {
 if (clipDAO == null) {
 clipDAO = new KRClipDAO(manager);
 }
 return clipDAO;
}

Output: Usages

http://trac.kp-lab.org/svn x Java DB Database Process x Glassfish V2 x Debugger Console x SMAT_Test (run-single) x

The class DAOTest.CRUDAnnotationPhaseDAOTest11 in file C:\dev\netbean\M24\SMAT_Test\build\classes\DAOTest\CRUDAnnotationPhaseDAOTest11.java

The class DAOTest.CRUDFragmentAnnotationDAOTest11 in file C:\dev\netbean\M24\SMAT_Test\build\classes\DAOTest\CRUDFragmentAnnotationDAOTest11.java

The class DAOTest.CRUDAnnotationSequenceDAOTest11 in file C:\dev\netbean\M24\SMAT_Test\build\classes\DAOTest\CRUDAnnotationSequenceDAOTest11.java

Deleted 9 out of date files in 10 seconds

Compiling 1 source file to C:\dev\netbean\M24\SMAT_Test\build\classes

compile-single:

run-single:

log4j:WARN No appenders could be found for logger (org.kplab.swkm.client.RdfSuiteProxyImpl).

log4j:WARN Please initialize the log4j system properly.

Exception in thread "main" javax.xml.ws.WebServiceException: Failed to access the WSDL at: http://kr.mobile.evtek.fi:3027/SwkmMiddle

http://kr.mobile.evtek.fi:3027/SwkmMiddle?wsdl

Watches

Name	Type	Value
v		

démarrer | Gestionnaire d... | 5 Explorateur... | 4 Opera Inte... | ZoneAlarm | Invite de com... | NetBeans IDE ... | backup04 - Blo... | 8 Mozilla Thu... | 2 Adobe Rea... | ENVOL_EnvDe... | FR | 02:22

COMPIL

Reverse engineering

The screenshot shows the NetBeans IDE 6.0.1 interface. The main editor displays the source code for the `Circle` class in the `org.kplab.smat.domainModel` package. A **Reverse Engineer** dialog box is open, showing the following configuration:

- Selected Nodes: SMAT
- Java Project Source Package Folders:

Reverse Engineer	Package Folder	Package Folder Label
<input checked="" type="checkbox"/>	java	Source Packages
<input checked="" type="checkbox"/>	test	Test Packages
- Use Existing UML Project:
- Target Project: SMAT-Model
- Create New UML Project:
- Project Name: SMAT-Model1
- Project Location: C:\dev\netbean\M24\SMAT\SMAT
- Project Folder: C:\dev\netbean\M24\SMAT\SMAT\SMAT-Model1

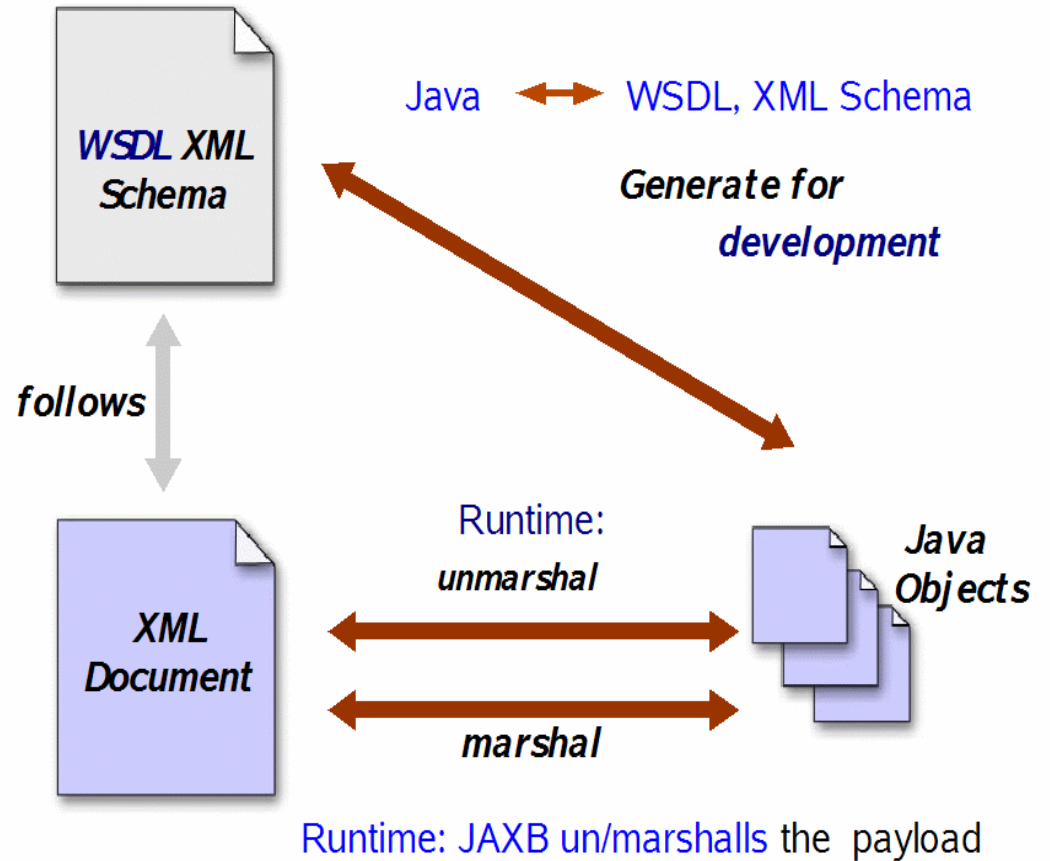
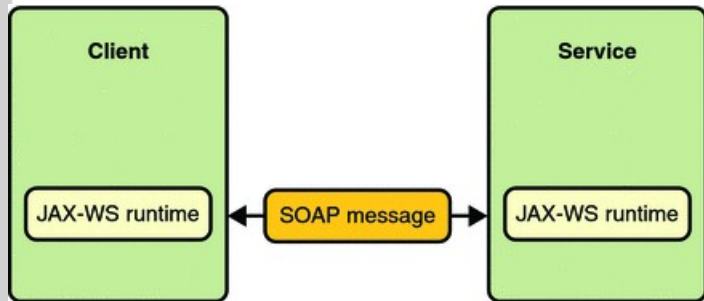
The **Output** window at the bottom shows the progress of the reverse engineering task:

```
http://trac.kp-lab.org/svn x Java DB Database Process x Glassfish V2 x Debugger Console x SMAT (javadoc) x Reverse Engineering Log x
Parsing 176 elements
Analyzing attributes & operations for 175 symbols
Resolving 102 attribute types
Integrating 175 elements
Building the query cache.
=====
Task Successful (total time: 25 seconds)
```

4. Cas d'étude d'un projet avec WS

- ✓ Génération d'un client à partir d'une URL de WS
- ✓ Implémentation de WS avec des EJBs
- ✓ Déploiement (WS, J2EE) sur Glassfish
- ✓ Environnement de test, environnement d'exploitation

JAX-WS, EJB avec Glassfish



Undeploy and deploy

The screenshot shows an IDE interface. On the left, a context menu is open over a project, with 'Undeploy and Deploy' highlighted. The menu includes options like 'Generate Javadoc', 'Run', 'Debug', 'Profile', 'Set as Main Project', 'Open Required Projects', 'Close', 'Rename...', 'Move...', 'Copy...', 'Delete', 'Find...', 'Reverse Engineer...', 'Subversion', 'Local History', 'Properties', and 'verifyContentItemChoice'. Below the menu, a snippet of Java code is visible, including a method signature: `verifyContentItemChoice(URI userURI) : URI`. The main editor displays a Java class with several imports, including `ContentSequenceDAO`, and a class definition for `FragmentAnnotationACTION`. The console window at the bottom shows the output of the deployment process.

Output

<http://trac.kp-lab.org/svn> × MS (run) × MS (run) #2 × Java DB Database Process × GlassFish V2 ×
 In-place deployment at C:\dev\netbean\M24\MS\MS\build\jar
 Start registering the project's server resources
 Finished registering server resources
 moduleID=MS
 While redeploying, trying to stop the application in target server completed successfully
 While redeploying, trying to remove reference for application in target server completed successfully
 deployment started : 0%
 deployment finished : 100%
 Deploying application in domain completed successfully
 Trying to create reference for application in target server completed successfully
 Trying to start application in target server completed successfully
 Deployment of application MS completed successfully
 Enable of MS in target server completed successfully
 Enable of application in all targets completed successfully
 All operations completed successfully
 post-run-deploy:

References

- Le site officiel de netbeans <http://www.netbeans.org>
- Editor code completion API
<http://platform.netbeans.org/tutorials/nbm-code-completion.html>
- netbeans.tv
- Glassfish <https://glassfish.dev.java.net/>
- EJB and WS
http://weblogs.java.net/blog/caroljmcDonald/archive/2007/09/sample_applicat_2.html