

Alles neu mit SWT?

Bernd Kolb (b.kolb@kolbware.de, www.kolbware.de)

Martin Lippert (martin.lippert@it-agile.de, www.it-agile.de)

KolbW@re/// *it-agile* 

Inhalt

- Was ist SWT
- Woher kommt SWT
- Wie funktioniert SWT
- Inside SWT
- Ausblick: eRCP / eSWT

Inhalt

- Was ist SWT
- Woher kommt SWT
- Wie funktioniert SWT
- Inside SWT
- Ausblick: eRCP / eSWT

Was ist SWT

The screenshot displays the Eclipse IDE environment. The main editor shows a C source file with code for a text table. The Navigator shows a project structure with folders like 'debian', 'doc', 'man', 'src', 'backend', 'frontend', 'include', and 'lib'. The Outline shows the class hierarchy for 'taj_lib'. The Tasks window is empty. The Lotus Notes interface is visible in the background. The Oxygen XML Editor shows an XSL stylesheet with various formatting rules. A text box in the bottom right corner contains text in four languages:

Arabic: أنا قادر على نكل لرجاج و هذا لا يؤلمني
Hebrew: אני יכול לאכול זכוכית וזה לא מזיק לי
Thai: ฉันสามารถกินแก้วและน้ำร้อนได้โดยไม่เจ็บปวด
Hindi: मैं काँच खा सकता हूँ, मुझे उस से कोई

Was ist SWT

- SWT = Standard Widget Toolkit

The SWT component is designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented.

(www.eclipse.org/swt)

Was ist SWT

- Ein Widget-Toolkit für Java
- Basiert auf den Widgets des Betriebssystems
- Nur eine ganz dünne Abstraktion über die OS-Widgets
- Kein Framework á la Swing
→ JFace

Inhalt

- Was ist SWT
- **Woher kommt SWT**
- Wie funktioniert SWT
- Inside SWT
- Ausblick: eRCP / eSWT

Woher kommt SWT

- AWT (Advanced Widget Toolkit):
 - Bietet Zugriff auf Widgets aus den OS-Fenstersystemen, aber nur auf solche, die in allen Systemen vorhanden sind (keine Bäume, kein Rich Text, ...)
 - → Kleinster gemeinsamer Nenner aller Zielsysteme

Woher kommt SWT

- SWING

- Bekämpft das Problem von AWT durch eigene Implementierungen der Widgets
- → Anderes Look & Feel as native Widgets
 - Durch unterschiedliche Styles wurde eine Anpassung versucht, aber der Unterschied ist immer noch feststellbar
- Man sieht der Anwendung Java an
 - → Vorurteil aus alter Zeit: langsam

Woher kommt SWT

- SWT
 - Vor über 12 Jahren begann OTI, heute Teil von IBM, eine plattformunabhängige, portable und native Oberflächenbibliothek für Smalltalk zu entwickeln.
 - → CW (Common Widgets)
 - IBM brauchte eine Oberflächenbibliothek für J2ME-Geräte. Diese Aufgabe bekam OTI

Woher kommt SWT

- OTI entschied, dass AWT u.a. wegen der Speicherverwaltung nicht passend, und SWING viel zu groß für die Zielplattformen war

Das Ergebnis ist SWT –
allerdings für J2SE

- Derzeit: Portierung von SWT zu eSWT

Inhalt

- Was ist SWT
- Woher kommt SWT
- **Wie funktioniert SWT**
- Inside SWT
- Ausblick: eRCP / eSWT

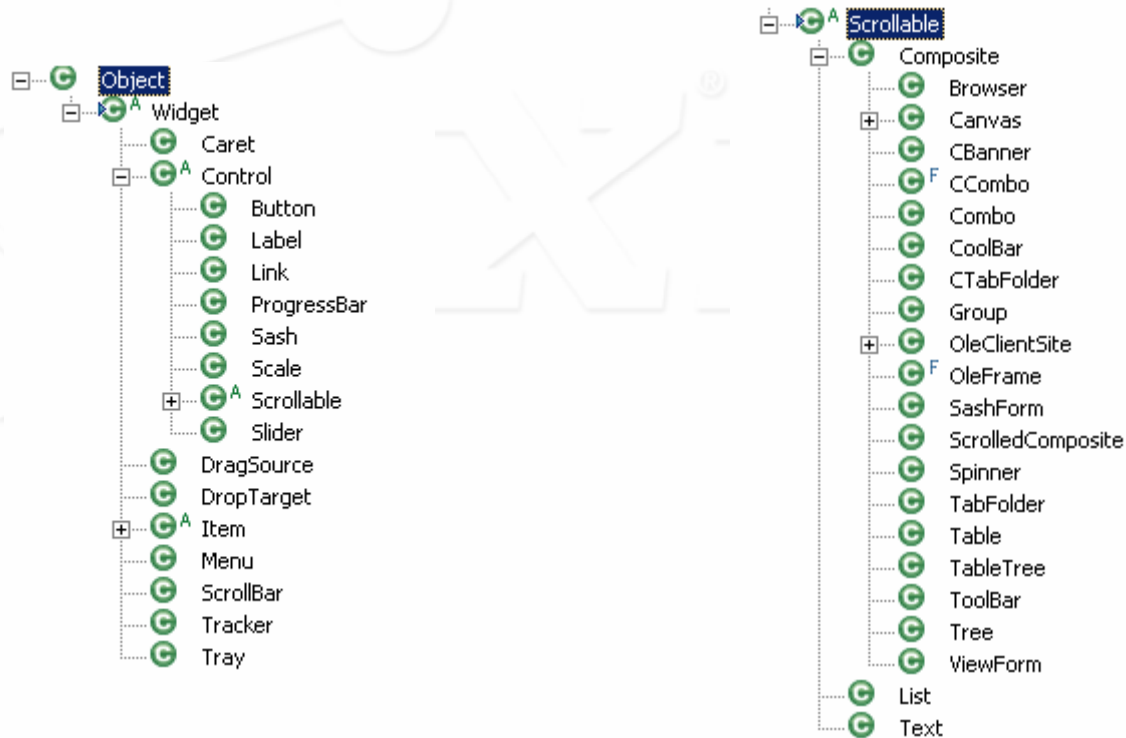
Wie funktioniert SWT

```
public static void main(String[] args) {
    Display display = new Display();
    ImageData data = new ImageData("icon.gif");
    Image icon = new Image(display, data);
    Color blue = new Color(display, 0, 0, 100);
    Font bold = new Font(display, "Arial", 16, SWT.BOLD);
    Shell shell = new Shell(display);
    shell.setText("JAX 2005");
    shell.setImage(icon);
    shell.setBounds(100, 100, 400, 80);
    shell.setLayout(new FillLayout());
    Label label = new Label(shell, SWT.CENTER | SWT.WRAP);
    label.setText("Willkommen in unserem Vortrag zu SWT");
    label.setForeground(blue);
    label.setFont(bold);
    shell.open();
    while (!shell.isDisposed()) {
        if (!display.readAndDispatch()) display.sleep();
    }
    icon.dispose();
    blue.dispose();
    bold.dispose();
    display.dispose();
}
```

DEMO

Wie funktioniert SWT

- Welche Widgets gibt es?



Wie funktioniert SWT

- Was am Code auffällt:

```
new Shell (display) ;
```

Wie funktioniert SWT

- SWT-Widgets halten selbst keinen Zustand
 - → klein und effizient
- Dies führt dazu, dass diese Widgets nicht alleine existieren können.
 - Sobald ein SWT-Widget erzeugt wird, wird der zugehörige Gegenpart im Fenstersystem erzeugt.

Wie funktioniert SWT

- Die meisten Fenstersysteme benötigen zum Erstellen eines Widgets einen Parent
 - Parent muss übergeben werden
- Keine Interfaces
 - → stabile API beruht nur auf Konventionen

Wie funktioniert SWT

- Was am Code auffällt

```
new Label(shell,  
          SWT.CENTER | SWT.WRAP) ;
```

Wie funktioniert SWT

- Widget-Anpassungen geschehen häufig durch das Verändern von Zahlen
- Diese Anpassungen sind meist nicht mehr änderbar wenn ein Widget erzeugt wurde
- Alle Konstanten finden sich in der Klasse **SWT**
- In den Widget-Klassen ist dokumentiert welche Konstanten angenommen werden

Wie funktioniert SWT

- Was am Code auffällt

```
while (!shell.isDisposed()) {  
    if (!display.readAndDispatch())  
        display.sleep();  
}
```

Wie funktioniert SWT

- Ohne diesen Code würde sich das Programm sofort wieder beenden
- Solange eine Shell nicht geschlossen wurde, wird das Device (Grafikkarte) weiter Events verarbeiten

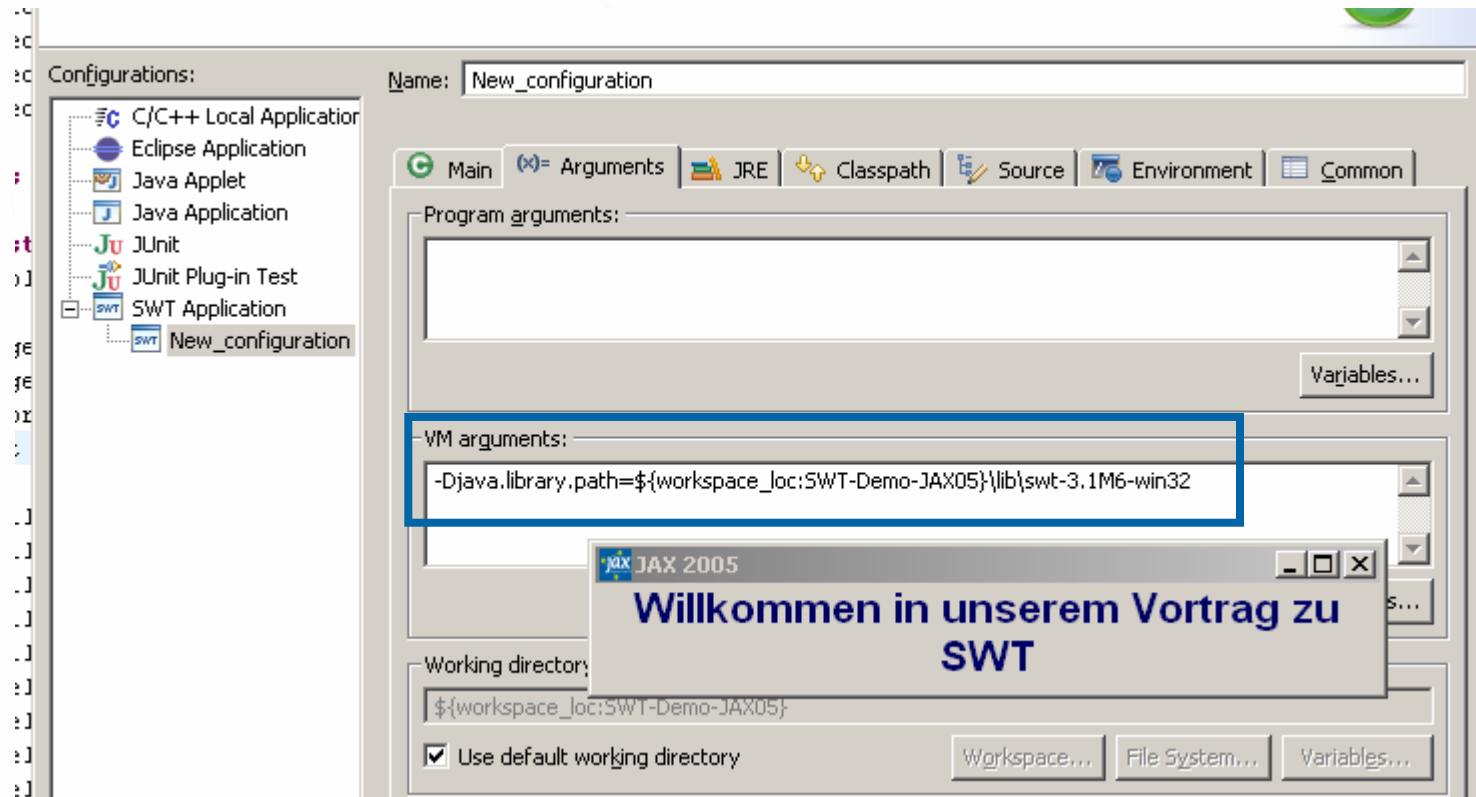
Wie funktioniert SWT

- Was am Code auffällt

```
icon.dispose();  
blue.dispose();  
bold.dispose();  
display.dispose();
```

Wie funktioniert SWT

- Eine SWT-Applikation starten



Wie funktioniert SWT

- Der Programmierer ist für einen Teil der Speicherverwaltung selbst verantwortlich
- Die Ressourcen für Fonts, Images und Colors müssen von Hand an das OS zurückgegeben werden; keine GC
- → Diese Objekte immer über eine Registry erzeugen

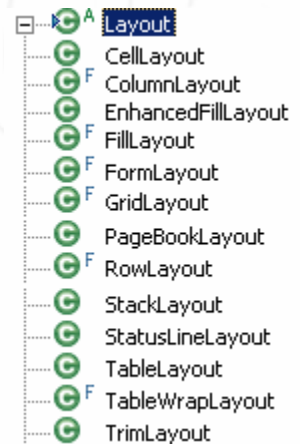
Wie funktioniert SWT

- Registrycode zum Zurückgeben der Ressourcen

```
Display d = new Display();  
// ...  
d.disposeExec(new Runnable() {  
    // dispose Fonts, Colors, Images  
});
```

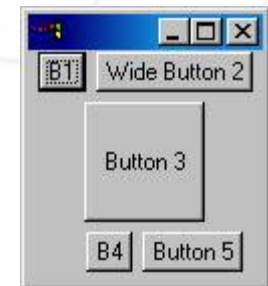
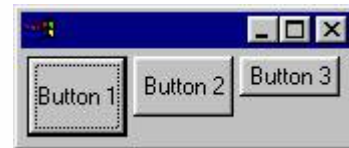
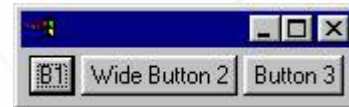
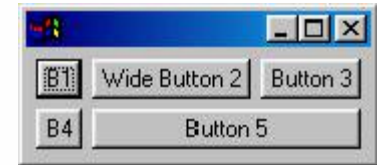
Wie funktioniert SWT

- Welche Layout-Klassen gibt es?
 - Wie in Swing gibt es absolute und relative Layouts
 - Es gibt sehr mächtige Layouts
 - → GridLayout
 - → FormLayout



Wie funktioniert SWT

- Layout-Beispiele



Quelle: <http://www.eclipse.org/articles/>

Wie funktioniert SWT

- Wie funktioniert das Layout?
 - Subklassen von **Composite** können ein Layout annehmen.
 - Das Layout wird, wenn nötig, durch sog. LayoutDatas gesteuert. Alle Subklassen von **Control** können solche LayoutDatas annehmen.

Wie funktioniert SWT

- Eine SWT-Applikation starten

```
-Djava.library.path=
```

```
  ${workspace_loc:
```

```
    SWT-Demo-JAX05}\lib\
```

```
      swt-3.1M6-win32
```

- In der Datei swt-win32-3128.dll (300kb) sind alle Aufrufe an das Betriebssystem gekapselt

Inhalt

- Was ist SWT
- Woher kommt SWT
- Wie funktioniert SWT
- **Inside SWT**
- Ausblick: eRCP / eSWT

Inside SWT

- Wir wollen programmatisch das erste Wort eines Textes markieren

```
text.setText ("Willkommen!");  
text.setSelection (0, 10);
```

Inside SWT

- `setSelection (...)` ist eine Methode aus der Widget-Klasse `Text`

```
public void setSelection (  
    int start, int end) {  
    OS.SendMessage (handle,  
        OS.EM_SETSEL, start, end) ;  
}
```


Inside SWT

- Es folgt sofort der Zugriff auf eine betriebssystem-spezifische Klasse
- Mit Hilfe von **SendMessage** wird das Betriebssystem angehalten etwas zu tun. Dazu wird das Handle sowie die „Aufgabe“ und Parameter übergeben:

```
public static final int EM_SETSEL = 0xb1;  
public static final native int SendMessageW  
(int hWnd, int Msg, int wParam, int lParam);
```

Inside SWT

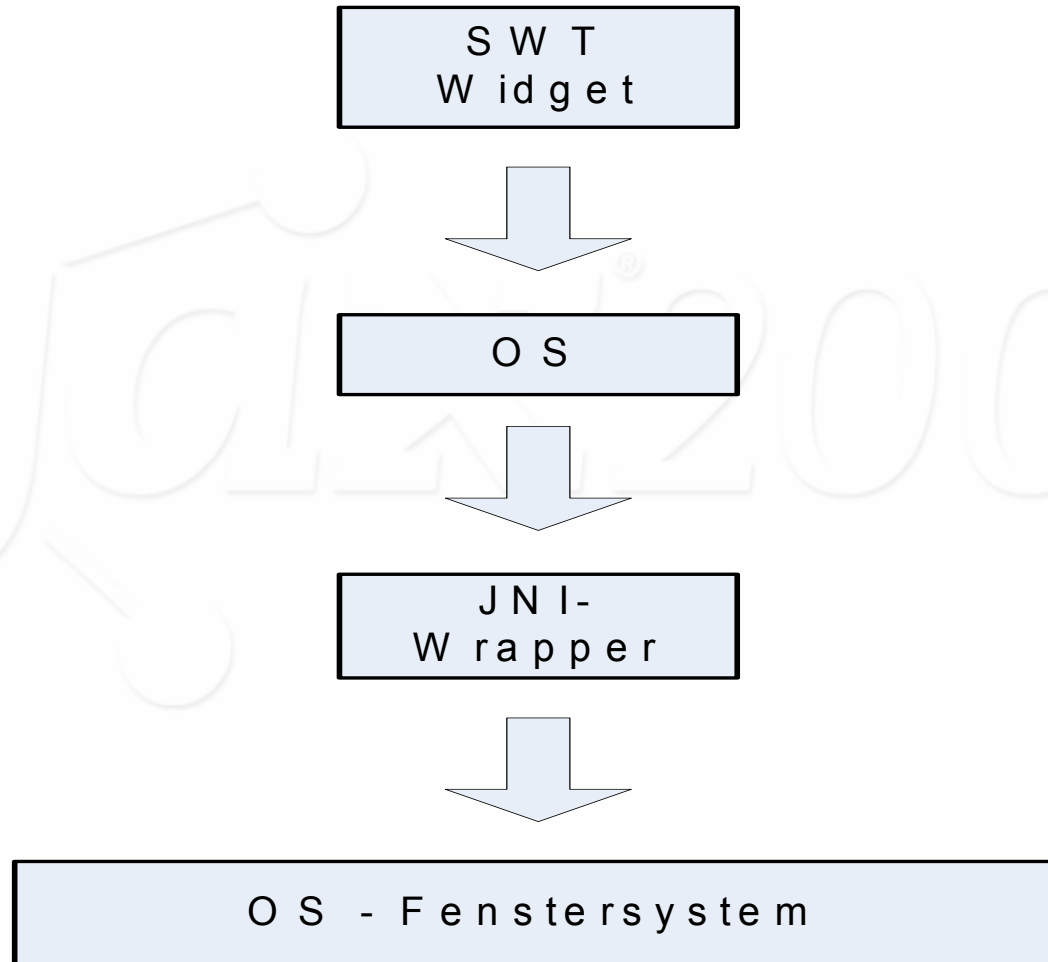
- Der C-Code

```
JNIEXPORT jint JNICALL
    OS_NATIVE(SendMessageW__IIII)
    (JNIEnv *env, jclass that, jint arg0,
     jint arg1, jint arg2, jint arg3)
{
    jint rc = 0;
    rc = (jint) SendMessageW((HWND) arg0, arg1,
        (WPARAM) arg2, (LPARAM) arg3);
    return rc;
}
```

Inside SWT

- Das einzige, was in C programmiert ist, ist die Weiterleitung des Java-Aufrufs an das Betriebssystem
- Die API der Widget-Klasse ist für jedes Betriebssystem die Selbe.
- Die interne Implementierung, sowie **OS** und die in C implementierten JNI-Calls unterscheiden sich.

Inside SWT



Inside SWT

- Vorteile

- **Schnell**

- Der begrenzende Faktor ist das Betriebssystem, nicht die VM

- **Einfach**

- Durch die geringe Menge an Code ist es einfach, SWT zu „durchschauen“

- **Stabil**

- Die Oberflächenbibliotheken der Betriebssysteme sind meist recht stabil und ausführlich getestet.

Inside SWT

- Nachteile

- **Kein Framework**

- Abstraktionsschichten müssen selbst eingebaut werden
 - ABER: JFace bietet diese Möglichkeit

- **Problem bei Fehlern**

- Durch den direkten Aufruf des Fenstersystems sind die Fehlermeldungen nicht immer sprechend. Kommt in der Praxis aber sehr selten vor.
 - Z.B. NoMoreHandles auch evtl. bei falscher Parameterübergabe.

Inside SWT

- Custom Widgets
 - Nicht in allen Betriebssystemen sind alle Widgets vorhanden
 - Beispiel: Erst seit SWT 3.1M5 werden native TableTrees unterstützt.
 - Trotzdem war diese Funktionalität bereits früher vorhanden
 - → Widget wird „emuliert“

Inside SWT

- Was bedeutet emuliert?
 - Mit Hilfe von Standard-Widgets werden die komplexeren Custom-Widgets nachgestellt.
 - Beispiel: siehe CCombo
Im Konstruktor wird ein **Text** und ein **Button** gruppiert sowie Listener für die entsprechenden Events hinzugefügt.

Inhalt

- Was ist SWT
- Woher kommt SWT
- Wie funktioniert SWT
- Inside SWT
- **Ausblick: eRCP / eSWT**

Ausblick: eRCP / eSWT

- eSWT ist ein Subprojekt des eRCP Projekts
- Ziel ist es SWT für *Embedded Devices* anzupassen
- Eine erste Version ist vor Kurzem erschienen

Ausblick: eRCP / eSWT

- Plattformen:

- Intel x86

- Symbian

- PalmOS

- Windows Mobil 2003

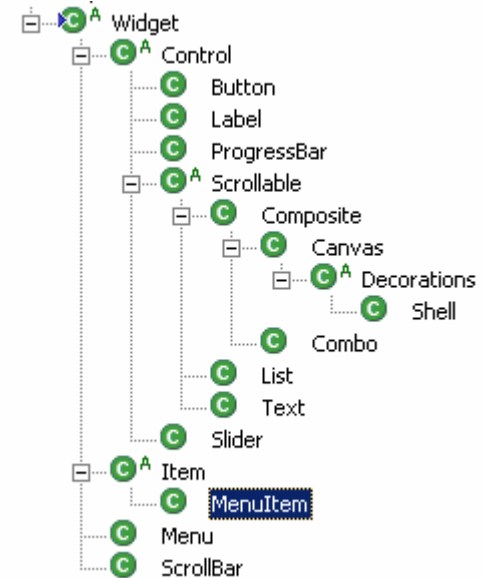
Ausblick: eRCP / eSWT

- Einschränkungen zu SWT:

- Weniger Layouts:
 - Bisher nur Formlayout

- Weniger Widgets

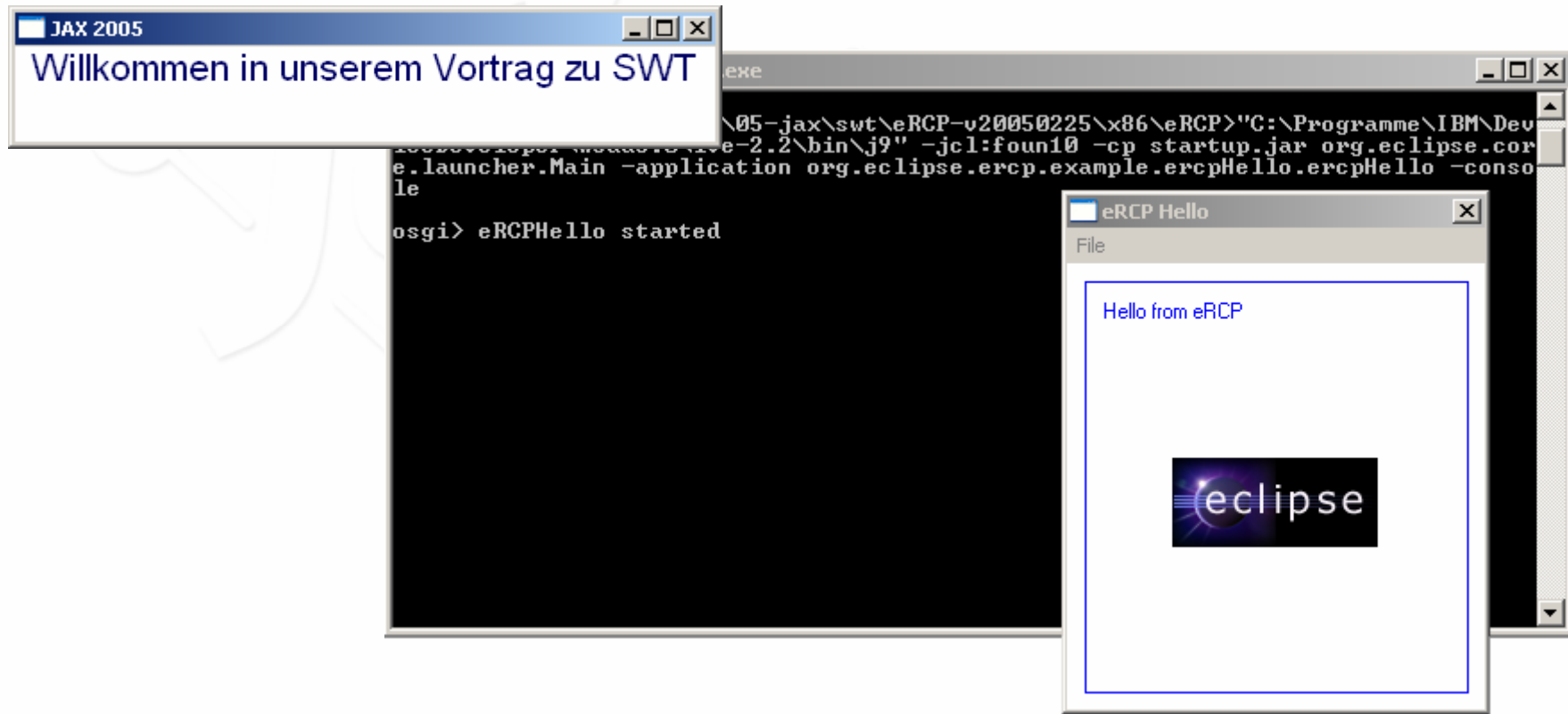
- Erst im Entstehen



Ausblick: eRCP / eSWT

DEMO

- Selbe API wie bei „normalem“ SWT



Referenzen / Links

- Bücher der Eclipse-Series
 - Building Commercial-Quality Plug-ins
 - SWT : The Standard Widget Toolkit, Volume 1
- Artikel auf www.eclipse.org/articles
 - Diverse Artikel zu SWT / JFace
- www.eclipse.org/swt
- www.eclipse.org/ercp

Ende

Fragen?, Kritik?, Anregungen?

Vielen Dank für Ihre Aufmerksamkeit

Bei Fragen: einfach melden ☺

Martin Lippert martin.lippert@it-agile.de

Bernd Kolb b.kolb@kolbware.de