



JavaOne

# *Visual Development and Deployment of Advanced Mobile Applications*

**Robert Virkus**

CEO

Enough Software

[www.j2mepolish.org](http://www.j2mepolish.org)

**David Kaspar**

Software Engineer

Sun Microsystems, Inc.

[www.netbeans.org](http://www.netbeans.org)

TS-9452

# Developing Visually— Only for Beginners?

Rapid development for professionals and beginners

**Learn how to develop mobile applications visually while keeping full control over your application to fight device fragmentation!**

# Agenda

Introduction to Visual Development  
Device Fragmentation and Solution  
Next Generation Visual Development  
Content Delivery and Content Sales  
Demo: Mobile RSS Reader  
Summary and Questions

# Agenda

## **Introduction to Visual Development**

Device Fragmentation and Solution

Next Generation Visual Development

Content Delivery and Content Sales

Demo: Mobile RSS Reader

Summary and Questions

# Visual Development

- Advantages
  - Natural way of development
  - Easier navigation and editing of an application
  - Various views on data
- Disadvantages
  - Visualization of complex application
  - Source code update issue
  - Handling diversity of a target environment

# Visual Development

Current generation: NetBeans™ IDE 5.5

The screenshot displays the NetBeans IDE 5.5 interface for visual development. The main window, titled 'Flow Designer: MIDP-2.0', shows a flow diagram with a 'Mobile Device' icon, an 'Exit Point', a 'Start Point', and an 'alert [Alert]' component. A 'Screen Designer: helloForm [Form]' window is overlaid, showing a device screen with text 'Hello Hello, World!' and an 'Input' field labeled 'Volume:'. A 'Properties' window for 'gauge1 [Gauge]' is also visible, showing settings like 'Is Interactive', 'Maximum Value', and 'Value'.

gauge1 [Gauge] - Properties	
Properties	
Label	Volume: [...]
Is Interactive	<input checked="" type="checkbox"/>
Maximum Value	100
Value	50
Layout	LAYOUT_DEFAULT [...]
Preferred Size	[Unlocked, Unlocked] [...]
Default Command	[None] [v]
Code Properties	
Instance Name	gauge1 [...]
Lazy Initialized	<input checked="" type="checkbox"/>
Is Interactive	

# Visual Development

Current generation: NetBeans™ IDE 5.5

- Drag-and-drop development of mobile applications
- Application flow designer
- Screen designer
- User code editing in the source code
- Supports MIDP–1.0 and MIDP–2.0
- Additional custom components

# Agenda

Introduction to Visual Development

**Device Fragmentation and Solution**

Next Generation Visual Development

Content Delivery and Content Sales

Demo: Mobile RSS Reader

Summary and Questions



# What Fragmentation?

The real world can be cruel

- Different hardware
- Support of different APIs
- Different interpretations of standards
- Device bugs
- More than a billion handsets sold in 2006



Source: Gartner

# Overcome Fragmentation—But How?

## Available options

- Using the lowest common denominator
- Using dynamic code and defensive programming
- Using different source folders
- Using preprocessing
- Using automatic tools

# Overcome Fragmentation

## Lowest common denominator

- Using the lowest common denominator
  - Modern variant: MIDP 2.0, CLDC 1.1 and Java™ technology for the wireless industry (JTWI) or MSA
- Advantages
  - Easy deployment
  - Simple programming
- Disadvantages
  - One size fits all?
  - Testing burden is not really reduced
  - Unable to use advanced features of newer devices

# Overcome Fragmentation

## Dynamic code and defensive programming

- Using dynamic code and defensive programming
  - Dynamically analyze device capabilities and react accordingly
- Advantages
  - Easy deployment
  - With dynamic class loading quite powerful
- Disadvantages
  - Code bloat
  - Different resource sets

# Overcome Fragmentation

## Dynamic code and defensive programming

- Load classes dynamically

```
Canvas splash = null;  
try {  
    Class.forName(  
        "javax.microedition.pki.Certificate" );  
    splash = (Canvas) Class.forName  
        ( "com.company.Midp2Splash" ).newInstance();  
}
```

- Only use imports in the device specific classes
- Use a dispatcher class for
  - Querying the capabilities and
  - Loading the correct implementation class

# Overcome Fragmentation

## Preprocessing

- Why preprocessing?
  - More powerful than dynamic class-loading
  - Reduced code size
  - Easy to understand
- Preprocessing directives and variables
  - Directives use variables to process
  - User defined and device database variables
- Many predefined preprocessing directives and property functions
  - `if` `foreach` `debug` `condition` `define` etc.

# Overcome Fragmentation

## Preprocessing

- Changing the source code before compilation:

```
//#if polish.midp2
//#elif polish.api.nokia-ui
//#else
//#endif
```

- Use properties from the device database:

```
//#= int screenHeight =
//#=      ${polish.FullCanvasHeight};
```

- Use property functions to transform values:

```
//#= player = Player.createPlayer(in,
//#=      "${ contenttype(audio/mp3) }");
```

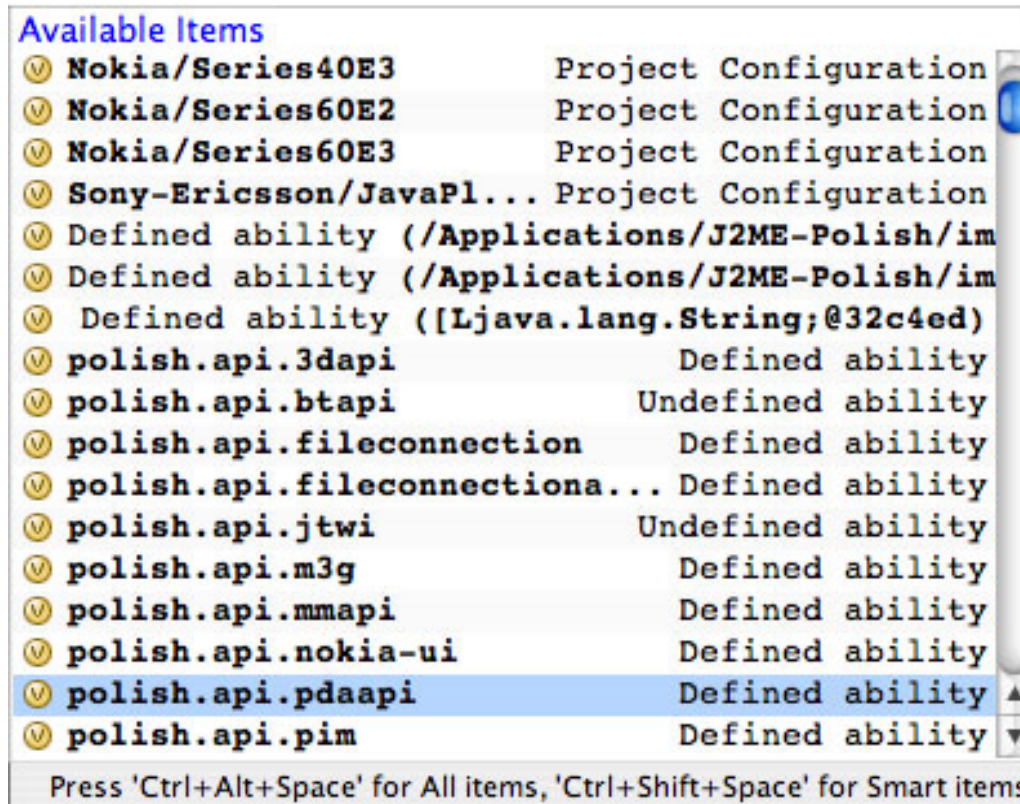
- Check for known issues:

```
//#if polish.Bugs.drawRgbOrigin
//#else
//#endif
```

# Overcome Fragmentation

## Preprocessing

- Use auto-completion in NetBeans IDE





# Overcome Fragmentation

## Automatic tools

- Post-compilers and finalizers can be used to adjust the application code automatically
- Examples
  - Floater: Allows floating point calculations on CLDC 1.0 devices
  - MasterCanvas: on Series 60 devices the screen flickers when a new screen is set, the MasterCanvasPostCompiler redirects all Canvases to a single “MasterCanvas” when the `polish.Bugs.displaySetCurrentFlickers` is present

# Agenda

Introduction to Visual Development

Device Fragmentation and Solution

**Next Generation Visual Development**

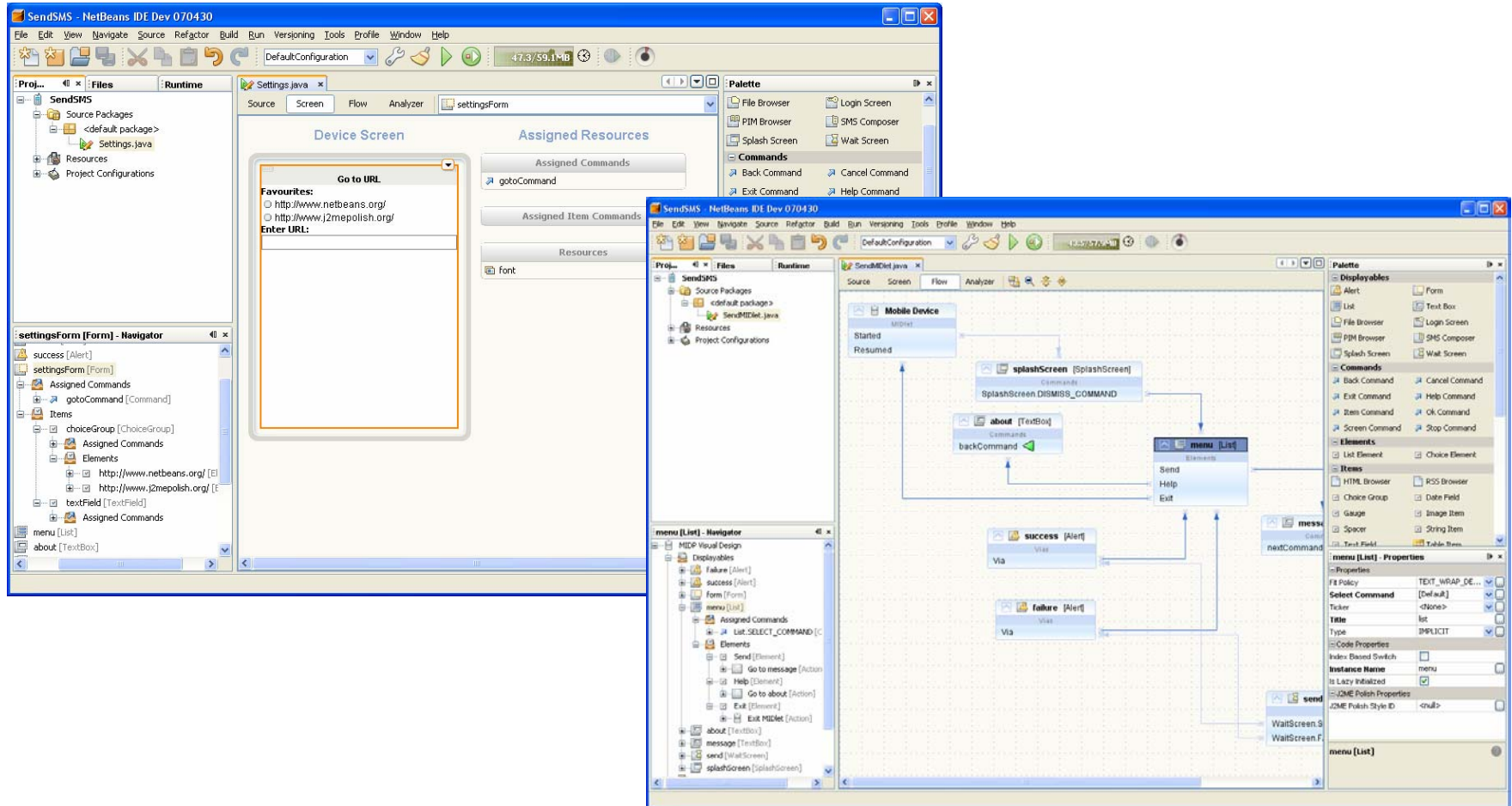
Content Delivery and Content Sales

Demo: Mobile RSS Reader

Summary and Questions

# Visual Development Solution

## Next generation: NetBeans™ IDE 6.0



# Visual Development Solution

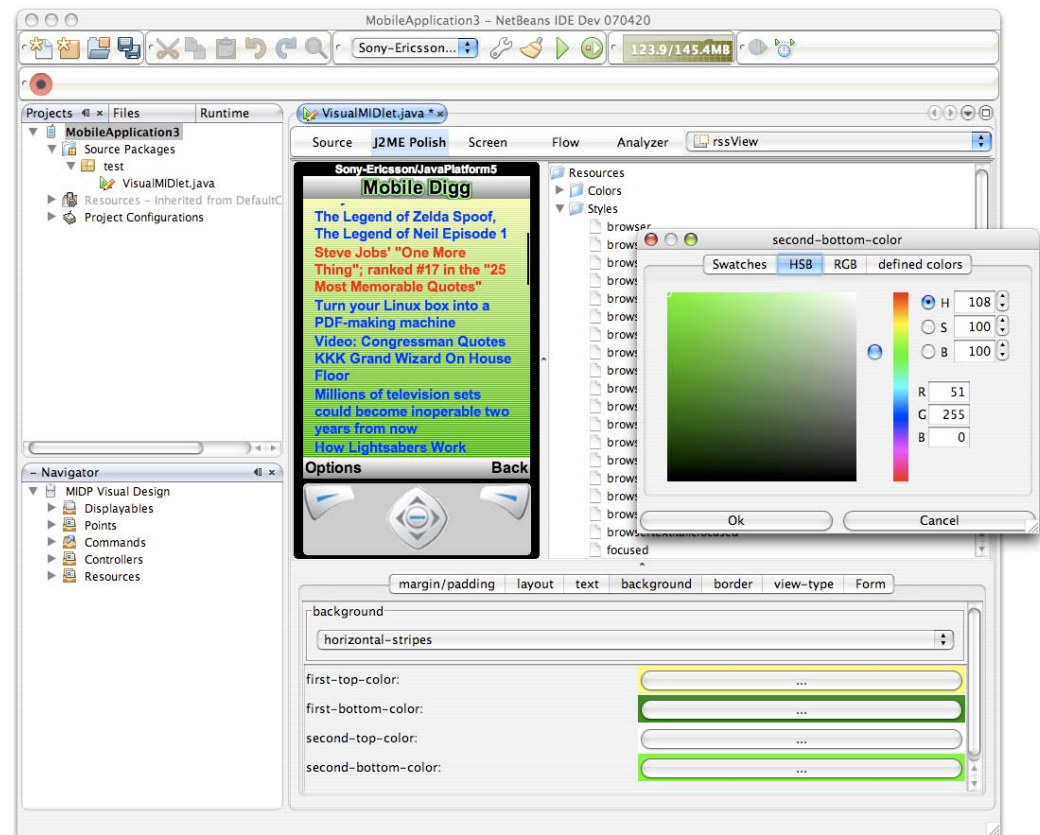
Next generation: NetBeans™ IDE 6.0

- Still drag-and-drop development of mobile applications
- Cleaner flow designer organization
- Extended flow control
- Finer code control
- Design analyzes
- Game support—scenes, layers, sprites builder
- Basic/advanced properties

# Visual Development Solution

Next generation: NetBeans™ IDE 6.0 and J2ME Polish

- Integrated device database
- Control animations and effects



# Agenda

Introduction to Visual Development

Device Fragmentation and Solution

Next Generation Visual Development

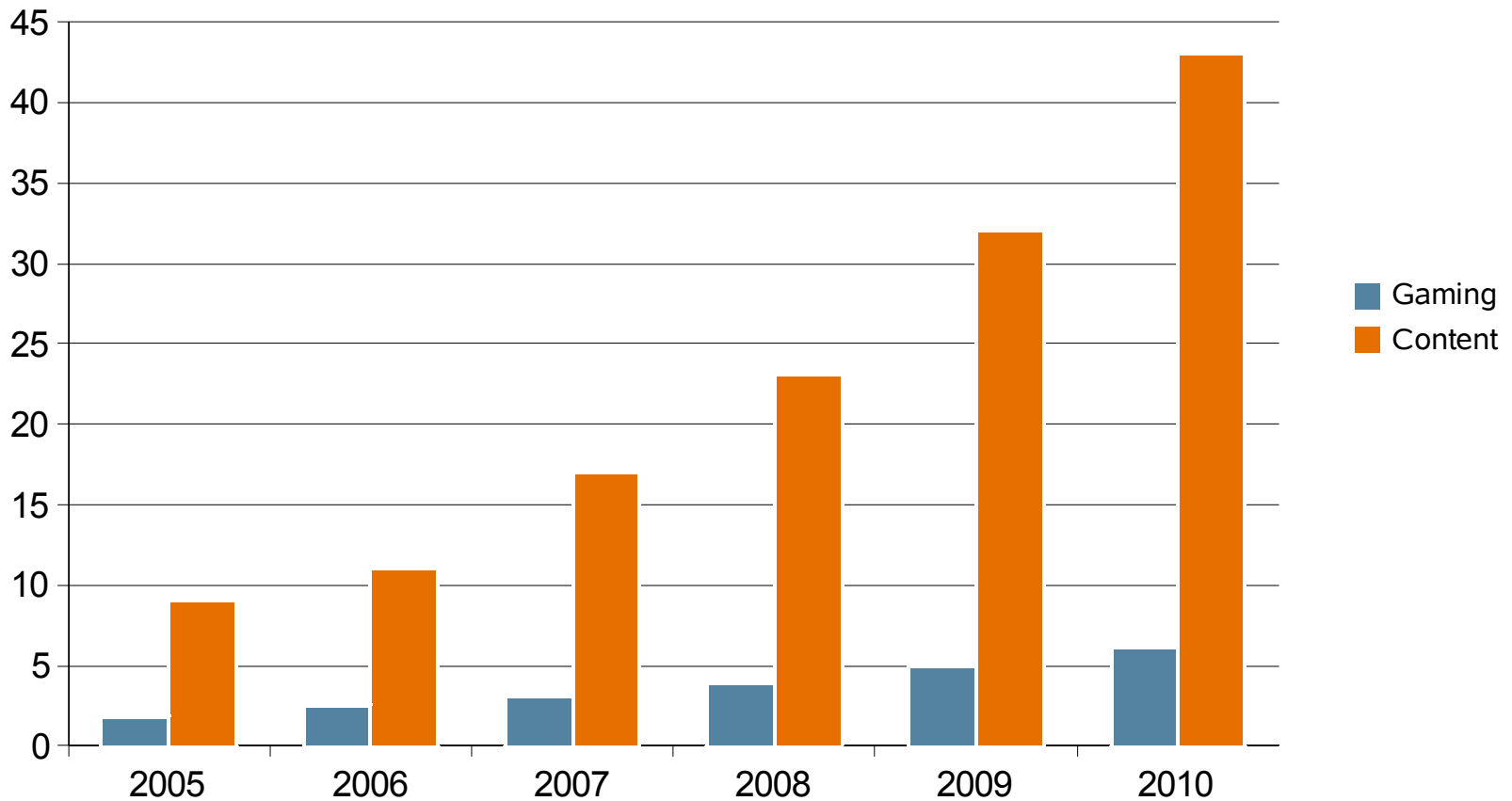
**Content Delivery and Content Sales**

Demo: Mobile RSS Reader

Summary and Questions

# Mobile Content Sales 2005–2010

In million US \$



Source: iSuppli Corp

# Selling Content

## Real world strategies

- Channels
  - Carriers
  - Handset manufacturers
  - Aggregators
  - Direct sales
  - Business-to-business applications
- Business models
  - Revenue share
  - One-off deals
  - Open Source



# Selling Content

## Real world strategies

- Localization
  - Act globally, think locally!
  - Translations
  - Culture
- Collaborate
  - Use networking platforms
  - Attend conventions ;-)
- Focus
  - Idea → presentation → implementation

# Agenda

Introduction to Visual Development

Device Fragmentation and Solution

Next Generation Visual Development

Content Delivery and Content Sales

**Demo: Mobile RSS Reader**

Summary and Questions

# Demo Phases

## 1. Phase

- Focus on functionality
- Visually develop working application

## 2. Phase

- Focus on appearance
- Polishing the application



# DEMO

## Creating a Mobile RSS Reader





# DEMO

## Polishing Our Mobile RSS Reader



# Agenda

**Introduction to Visual Development**

Device Fragmentation and Solution

Next Generation Visual Development

Content Delivery and Content Sales

Demo: Mobile RSS Reader

**Summary and Questions**

# Future Plans

- Extending device database
- Extending set of built-in components
- Data-binding support

# Summary

- You can create your mobile application visually
- You can make it look visually stunning
- You can use high-level components
- You can use device database
- Still you have full control over it
  - Tackle device fragmentation
  - Control design and customization



# For More Information

- NetBeans IDE and J2ME Polish
  - [www.j2mepolish.org](http://www.j2mepolish.org)
  - [www.netbeans.org](http://www.netbeans.org)
  - “Pro J2ME Polish” book
- Related sessions
  - TS-5626—data binding and Java™ platform, Micro Edition (Java™ ME)
- Download RSS reader
  - <http://www.j2mepolish.org/dl>



# Q&A

**Robert Virkus**

CEO

Enough Software

[www.j2mepolish.org](http://www.j2mepolish.org)

**David Kaspar**

Software Engineer

Sun Microsystems, Inc.

[www.netbeans.org](http://www.netbeans.org)



JavaOne

# *Visual Development and Deployment of Advanced Mobile Applications*

**Robert Virkus**

CEO

Enough Software

[www.j2mepolish.org](http://www.j2mepolish.org)

**David Kaspar**

Software Engineer

Sun Microsystems, Inc.

[www.netbeans.org](http://www.netbeans.org)

TS-9452