

# AT&T Developer Program –Webcast Series

## New Development Tools for BlackBerry smartphones

Presented by:

**Hsuan-Hua Chang**  
Sr. Product Development Manager  
The AT&T Developer Program

**Brent Thornton**  
Technical Channel Manager – Applications  
RIM



# Introduction - devCentral – AT&T Developer Program

## Commitment to Data Developers – support for each step of the wireless application development process

- **Access to the latest information on building solutions for the AT&T wireless network**
  - Detailed Device & Modem Specifications
  - APIs, Toolkits, and Sample Code
  - Developer Gateways
- **Technical support and educational opportunities**
  - Webcasts
  - Developer Forums
  - Documentation and FAQs
- **Subsidized wireless devices & discounted data service plans\***
- **Device Specific Information**
  - [developer.att.com/RIM](http://developer.att.com/RIM)
- **Alignment with AT&T's enterprise business powerhouse**
  - Certified solutions program ([developer.att.com/certifiedsolutionscatalog](http://developer.att.com/certifiedsolutionscatalog))
  - Co-marketing and co-selling opportunities

<http://developer.att.com>

\* Available for  
deluxe members

# Devices That Will Support 5.0

- BlackBerry® Bold™ 9700 smartphone
- BlackBerry® Bold™ 9000 smartphone
- BlackBerry® Curve™ 8900 smartphone
- BlackBerry® Curve™ 8520 smartphone



# Win an AT&T Smartphone!



BlackBerry® Bold™ 9700 smartphone



- High Speed email and web
- Built in GPS and Wi-Fi
- Sharp 3.2 megapixel camera
- Visual Voicemail

Answer the question at the end of the webcast and win an AT&T Smartphone!

**Brent Thornton**

**Technical Channel Manager, Applications**

**Research In Motion**

# Agenda

- 5.0 Browser
- BlackBerry Widgets
- Java 5.0 APIs & Tools
- 5.0 BlackBerry User Interface Elements
- Other Announcements From The BlackBerry Developers Conference

# 5.0 Browser

# 5.0 Browser Features

- SVG
  - PME, SVG Tiny 1.1 & 1.2
- HTML 5.0
  - Parsing, Forms, datalist
- Web APIs
  - AJAX – XMLHttpRequest Object
  - Gears 0.5 (Except LocalServer)
- Web Feeds
  - Atom, RSS 0.9, 1.0, 2.0
- DOM Level 2
  - Core, Events, HTML, Range, Style, Traversal
- CSS 3
  - Selectors, Color (except opacity), Marquee, Media Queries, Namespaces
- Speed improvements (especially in JavaScript)
- JavaScript is enabled by default
- HttpOnly cookies



# BlackBerry Widgets

# What are widgets?

- A “small”, very focused web application that typically uses real-time data to provide relevant context to the end user
- Widgets are developed using web technologies like HTML, CSS, JavaScript and AJAX
- Entire user interface (UI) is created with HTML and CSS and there is no browser UI used or present
- Browser engine is rendering the UI

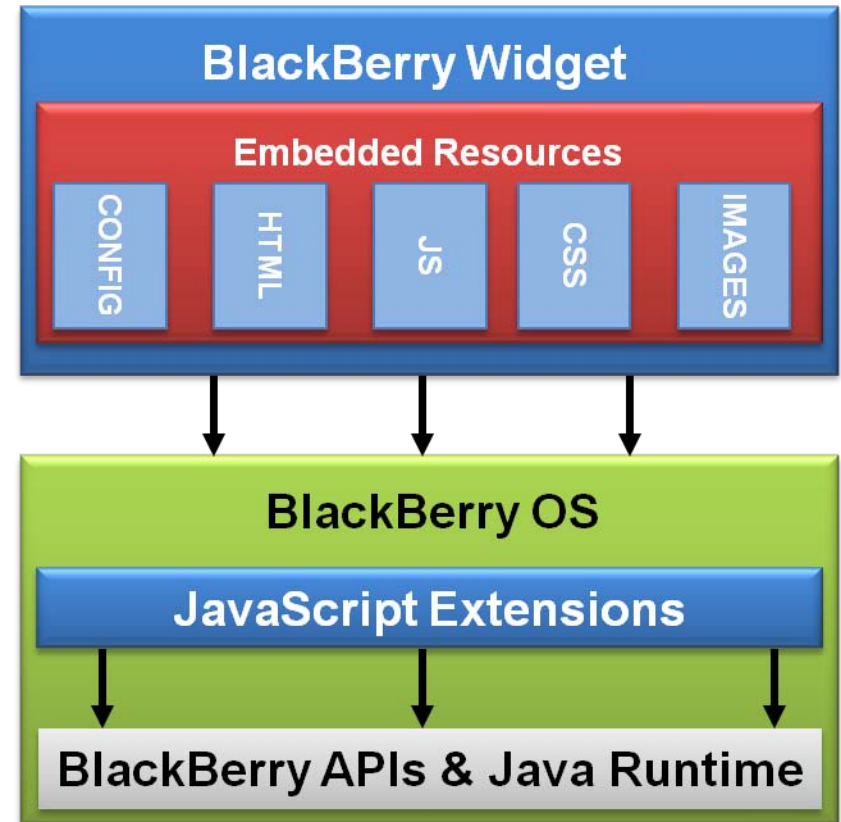
# What are BlackBerry Widgets?

- A BlackBerry Widget looks, behaves and has the same security mechanisms as a native BlackBerry application.
- BlackBerry Widgets can be installed on a BlackBerry smartphone like any native application and can be extended to use device-specific information and data using the BlackBerry Widget APIs.
- BlackBerry Widgets can be distributed via App World, BES or via web servers



# BlackBerry Widget Architecture

- Create native applications using web technologies
  - User interface is entirely authored in HTML/CSS
- Application logic is written in JavaScript
- Access to BlackBerry device functionality through JavaScript Extension APIs (known as Widget APIs)



# What are BlackBerry Widget APIs?

- The “glue” between Widgets and the BlackBerry native platform
- RIM provides JavaScript interfaces for device resource access – Calendar, PIM, GPS, File System, and more
- RIM will continue to add new JavaScript Extension interfaces for BlackBerry Widgets. Most will not require new device OS versions to be used
- As a Java developer you can create your own custom JavaScript Extensions (known as Widget Extensions) to package with your widget
- Wrap any of the native BlackBerry API’s by implementing the provided JavaScript interfaces
- Package your Widget Extension in the widget archive to be distributed with your application

# Device integration with rich JavaScript APIs

PIM	Create, retrieve and modify Calendar, Appointments, and Contacts
Phone	Invoke/Accept call
File system	Access device file system
Native applications integration	Invoke Maps, Calendar, Phone, Messaging, Camera...,
Gears APIs	For multi-threading, local storage via SQLite and Geolocation
SD Card	Accessing information from SD Card
Data push	The push.Data Class provides access to the data that has arrived as the result of a push.

## How is Widget Security maintained?

- BlackBerry Widgets must be signed with a Research In Motion (RIM) code signing key
- The same access control policies can be applied to widgets as can be applied to native applications
- Web resources outside of the widget can be pulled in from external sources as long as those sources match the white list provided with your widget
- Widget API's are only provided in a widget, and not in the BlackBerry Browser

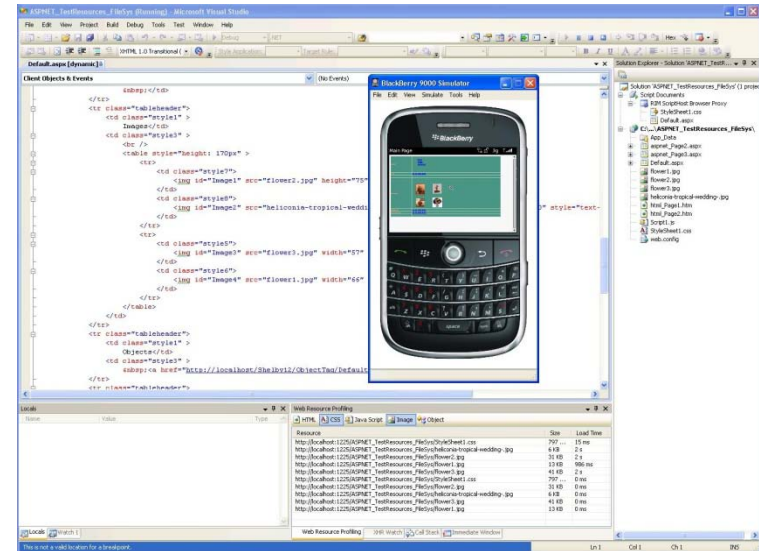
## Tooling - what is the BlackBerry Widget Packager?

- Widget Packager is a standalone command line utility for BlackBerry Widget packaging:
  - Input: Widget bundle
    - zip file of HTML, CSS and JavaScript + config.xml file
  - Output: Packaged BlackBerry Widget ready to be deployed
    - Cod, jad, alx
- Integrated into BlackBerry Web Plug-in developer tools
  - Widget creation and debugging in upcoming tooling releases



# Tooling - Visual Studio and Eclipse Support

- BlackBerry Widget Packager
  - Utility to package BlackBerry Widgets for deployment
- Beta: Web Plug-ins for Visual Studio 2.0 and Eclipse 2.0
  - Everything in previous version of the tooling plus support for BlackBerry Widget Debugging, Profiling, and Packaging



## How do you distribute a BlackBerry Widget?

- Same distribution/management model as native applications and can be distributed through BlackBerry App World™
- When the Widget has been signed and packaged with the Widget Packager tool it can be submitted to App World
- Alternatively, it can be distributed through BlackBerry Enterprise Server or websites.

# Web standards supported by BlackBerry Widgets

- Support for latest Web standards

HTML 4.01 (including Forms, Maps, Tables, iFrames, Objects)	Partial HTML5
CSS 2.1	Partial CSS 3.0
XHTML 1.1	DHTML
JavaScript® 1.5/1.6	DOM L2 (including Core, HTML, Style, Events)
XMLHttpRequest	SVG Tiny 1.1 (with scripting)
Support for inline video streaming	Support for Gears®

- BlackBerry Widgets based on W3C Recommendations (WIP)
  - <http://dev.w3.org/2006/waf/widgets-reqs/#widgets-1.0-family-of-specifications>

## Resources

- BlackBerry Widget SDK –
  - Contains tooling, product documentation and sample code
  - <http://www.blackberry.com/developers/widget>
- BlackBerry Web Plug-in tooling –
  - <http://www.blackberry.com/developers>

# 5.0 APIs

# 5.0 APIs

- Video Capture
- Phone Screen Integration
- SQLite
- Media Keys
- Other New APIs
- Tools- Java Plug-in for Eclipse

# Video Capture

## Use the camera to record video in your application

JSR 135 - MMAPI

- RecordControl

### Supported encodings

```
System.getProperty("video.encodings");
```

```
encoding=video/3gpp&mode=standard
```

```
encoding=video/3gpp&mode=mms
```

```
encoding=video/3gpp&width=480&height=320&video_codec=MPEG-4&audio_codec=AMR
```

```
encoding=video/3gpp&width=176&height=144&video_codec=MPEG-4&audio_codec=AMR
```

```
encoding=video/3gpp&width=480&height=320&video_codec=MPEG-4&audio_codec=PCM
```

```
encoding=video/3gpp&width=176&height=144&video_codec=MPEG-4&audio_codec=PCM
```

```
encoding=video/3gpp&width=480&height=320&video_codec=H263&audio_codec=AMR
```

```
encoding=video/3gpp&width=176&height=144&video_codec=H263&audio_codec=AMR
```

```
encoding=video/3gpp&width=480&height=320&video_codec=H263&audio_codec=PCM
```

```
encoding=video/3gpp&width=176&height=144&video_codec=H263&audio_codec=PCM
```

# Video Capture - Example

```
String encoding = "encoding=video/3gpp&mode=standard";

Player player = Manager.createPlayer("capture://video?" + encoding);
player.realize();

videoControl = (VideoControl) _player.getControl("VideoControl");
recordControl = (RecordControl) _player.getControl("RecordControl");

// Initialize the video display – MIDlet
videoControl.initDisplayMode(VideoControl.USE_DIRECT_VIDEO, _myCanvas);

/** Initialize the video display – BlackBerry Java Application
Field videoField = (Field) videoControl.initDisplayMode(VideoControl.USE_GUI_PRIMITIVE,
    "net.rim.device.api.ui.Field");
add(videoField);
*/

// Create a ByteArrayOutputStream to capture the audio stream.
ByteArrayOutputStream output = new ByteArrayOutputStream();
recordControl.setRecordStream(output);

//start recording
recordControl.startRecord();
player.start();
```

# Phone Screen Integration

Ability to add a Field to:

- Incoming call screen
- Active call screen

To use:

- Implement PhoneListener or extend AbstractPhoneListener
- Implement or override callIncoming() and callWaiting()
- Create a new PhoneScreenField
- Add text, images, or other Fields to the PhoneScreenField

Example:

```
public void callIncoming(int callid) {  
    PhoneScreenField f = new PhoneScreenField(callid);  
    f.add(new LabelField("Demo 2"));  
    f.addImage(_image);  
    //note: _image was loaded previously, don't load in callIncoming()  
}
```

Remember:

- Do not perform any long or blocking operations
- Data should be simple and already available for use



# SQLite

Package net.rim.device.api.database

- Provides relational database functionality

Example for a SELECT operation over a table:

```
try {
    Database db = DatabaseFactory.create("x"); // Defaults to /SDCard/x.db
    Statement s = db.createStatement("SELECT * FROM mytable;");
    s.prepare();
    Cursor c = s.execute();
    while (cursor.hasMoreElements()) {
        Row r = (Row)cursor.nextElement();
        // Read row. For example:
        int idx1 = r.getColumnIndex("col1");
        int idx2 = r.getColumnIndex("col2");
        int val1 = r.getInteger(idx1);
        int val2 = r.getInteger(idx2);
    }
} catch (Throwable e) {
    throw e;
}
```

# Media Keys

Package `net.rim.device.api.media`

- API to handle dedicated media actions
  - Volume changes
  - forwards / backwards requests
  - speakerphone / mute / play / pause requests

MIDP application:

- `Controllable.getControl("net.rim.device.api.lcd.ui.control.MediaBehaviourControl")`
- `MediaBehaviourControl.setMediaPlayerModeEnabled(true)`
- Media keys will now be sent to
  - `Canvas.keyPressed()`, `keyReleased()`, `keyRepeated()`

BlackBerry Java application:

- Option A: Implement `MediaActionHandler`
  - `Application.addBackgroundMediaHandler()`
- Option B: Extend `MediaKeyListener`
  - Abstract class: implements `MediaActionHandler` and `KeyListener`
  - `Application.addKeyListener()`

# Other New APIs

- Push API:
  - store push service registration information
  - auto-start a push application when a message arrives and bring it up to foreground
  - intelligently detect SIM swap
  - auto-unregister when application is removed.
- Network API:
  - simplify the establishment of connections over the different transports available on the BlackBerry device
  - means for querying the availability of transports and for selecting the most appropriate one to establish a connection



# 5.0 BlackBerry User Interface Elements

# UI Development: History

- Evolution of the BlackBerry® application platform and BlackBerry smartphones
- UI API growth of over 1000% in the last 5 years
- Exponential growth in display size & resolution



# New UI Elements

- Components
- Containers
- Extensions
- Utilities

# Components

- Auto Complete
- Spin Boxes
- Pickers
  - Date and Time
  - Location
  - File

# Auto Complete Field

- Helps users complete text based input
- Numerous sources for completion
- Designed for speed
- Customizable



# Auto Complete Field: Sample Code

```
import net.rim.device.api.ui.component.*;

// Create a list of suggestions using default comparison method
String[] suggestions = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};
BasicFilteredList list = new BasicFilteredList("Days Of Week", suggestions);
AutoCompleteField autoCompleteField = new AutoCompleteField(list);

// Retrieve an e-mail address from BlackBerry Contacts
BasicFilteredList list = new BasicFilteredList("Contact",
    BasicFilteredList.DATA_SOURCE_CONTACTS);
AutoCompleteField autoCompleteField = new AutoCompleteField(list);

// Retrieve a phone number from BlackBerry Contacts
BasicFilteredList list = new BasicFilteredList();
list.addDataSource("Contact", BasicFilteredList.DATA_SOURCE_CONTACTS);
list.addDataFields("Contact", BasicFilteredList.DATA_SOURCE_CONTACTS_NAME,
    BasicFilteredList.COMPARISON_STARTS_WITH);
list.addDataFields("Contact", BasicFilteredList.DATA_SOURCE_CONTACTS_PHONE_ALL,
    BasicFilteredList.COMPARISON_CONTAINS);
AutoCompleteField autoCompleteField = new AutoCompleteField(list);
```

# Spin Boxes

- Select items from a sequential list
- Can have multiple spin boxes
- Managed by a “spin box manager”
- Developers can implement own spin box



# Spin Boxes: Sample Code

```
import net.rim.device.api.ui.component.*;

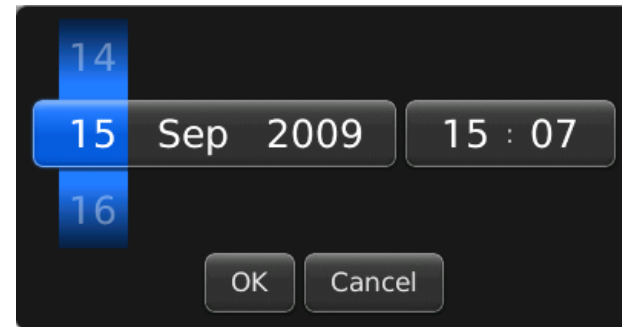
// Create some options
final String[] TWO_OPTIONS = {"one", "two"};
final String[] THREE_OPTIONS = {"one", "two", "three"};

// Setup the manager
SpinBoxFieldManager spinBoxManager = new SpinBoxFieldManager();
spinBoxManager.setVisibleRows(3);

// Add spin boxes to the manager
spinBoxManager.add(new TextSpinBoxField(TWO_OPTIONS));
spinBoxManager.add(new TextSpinBoxField(THREE_OPTIONS));
```

# Date and Time Picker

- Select date and/or time using spin boxes
- Developer specifies date format



# Date and Time Picker: Source Code

```
import net.rim.device.api.ui.picker.*;

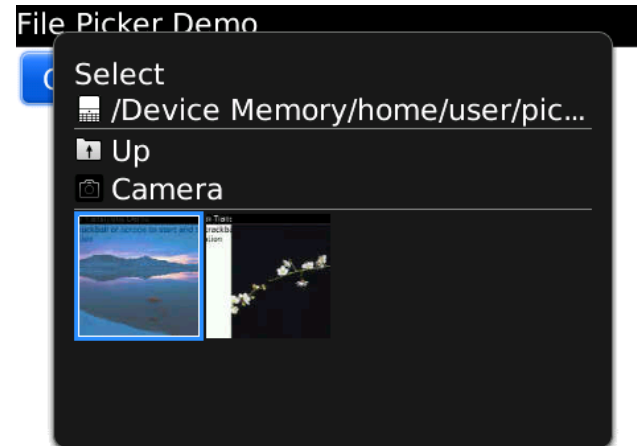
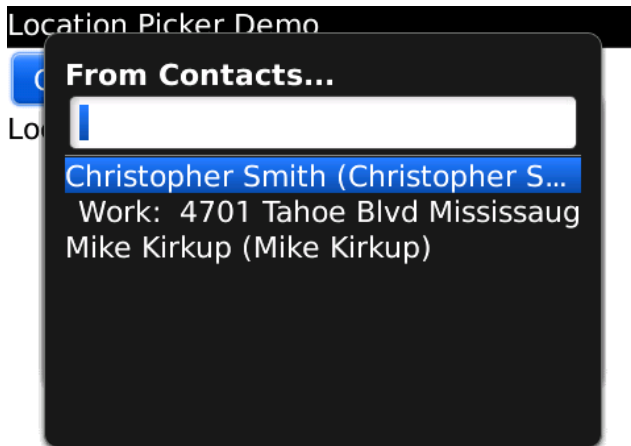
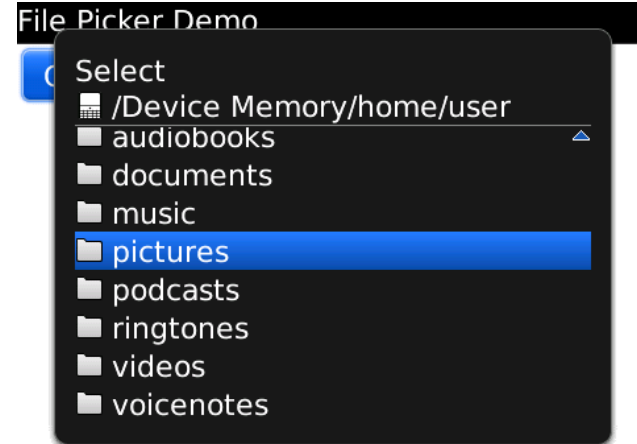
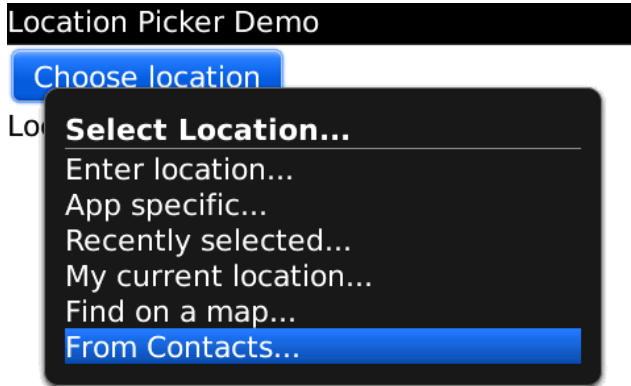
// Create a formatted date picker
final DateTimePicker datePicker =
    DateTimePicker.createInstance( null, "yyyy-MM-dd", null);

datePicker.doModal();

// Create a date picker for expiry date
final DateTimePicker expiryDatePicker =
    DateTimePicker.createInstance( null, "MM/yyyy", null);

expiryDatePicker.doModal();
```

# Location Picker & File Picker



# Location Picker: Source Code

```
import net.rim.device.api.lbs.picker.*;

// Create a prompt for a location using GPS
LocationPicker picker = LocationPicker.getInstance();
LocationListener listener = new LocationListener();
picker.setListener(listener);
picker.addLocationPicker(GPSLocationPicker.getInstance());
picker.show();

// Handle the selected location
class LocationListener implements LocationPicker.Listener {
    public void locationPicked (LocationPicker.Picker picker, Landmark
location) {
        // Process selected location
    }
}
```

# File Picker: Source Code

```
import net.rim.device.api.ui.picker.*;

// Create a prompt for a file
FilePicker picker = FilePicker.getInstance();
FileListener listener = new FileListener();
picker.setListener(listener);
// Select the start directory
picker.setPath("file://store/home/user/documents");
// Specify file filter
picker.setFilter("jpg");
picker.show();

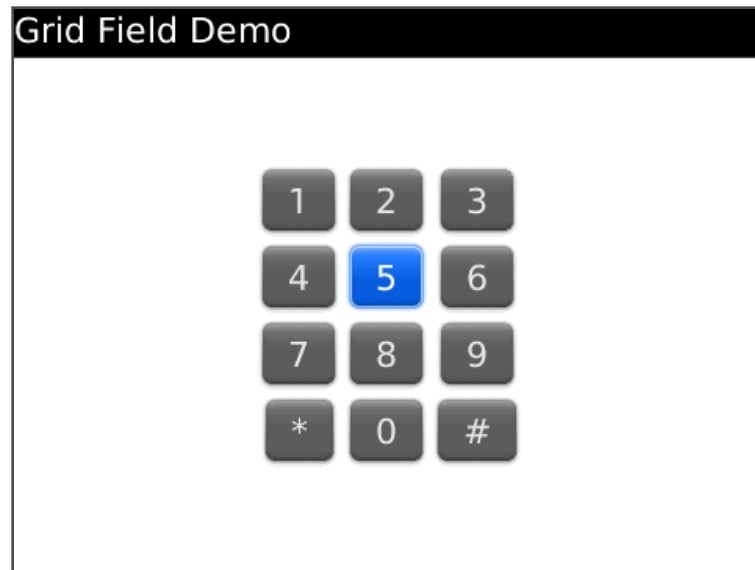
// Handle the selected file
class FilePicker implements FilePicker.Listener {
    public void fileSelected(String fullPathAndFilename) {
        // Process the selected filename
    }
}
```

# Containers

- Grid Field Manager
- Screen Transitions
- Menu Customization

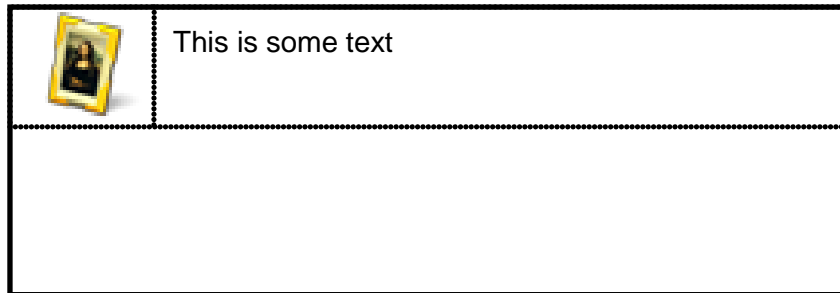
# Grid Field Manager

- Organize fields in rows and columns
- Useful for screens requiring components to be laid out in a table



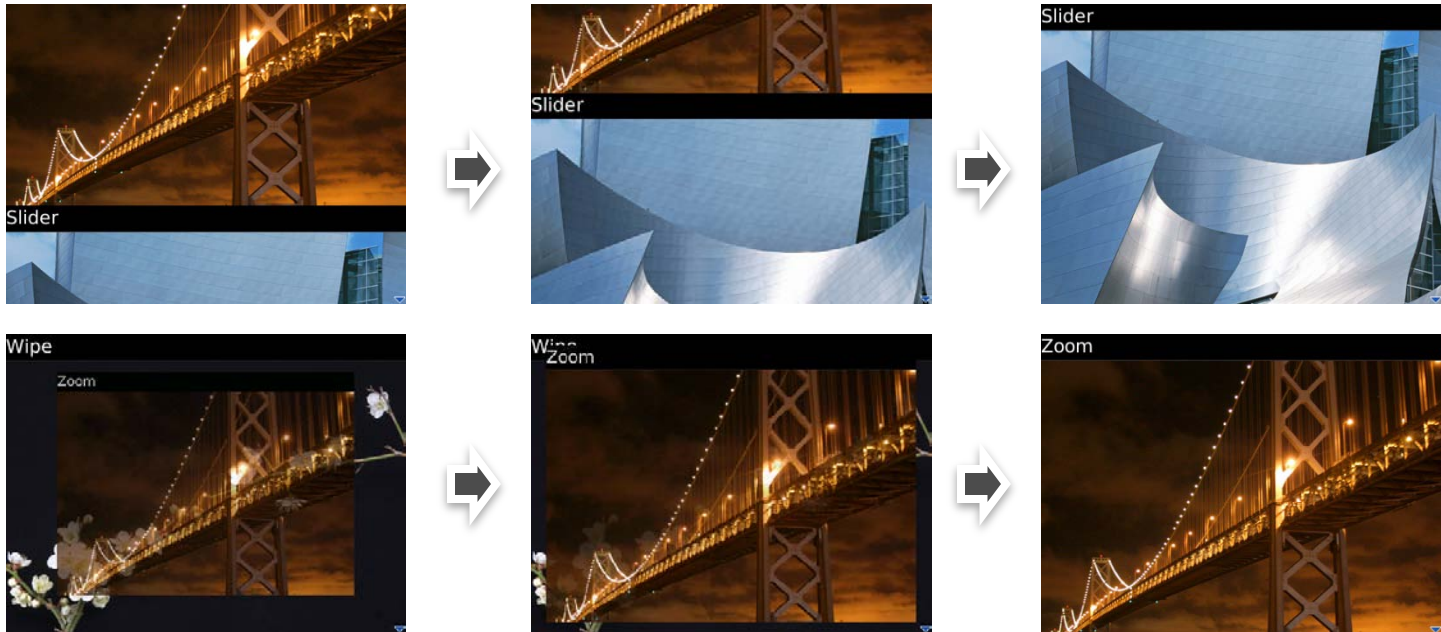
# Grid Field Manager: Sample Code

```
import net.rim.device.api.ui.container.*;  
  
GridFieldManager grid = new GridFieldManager(2, 2, Field.USE_ALL_WIDTH);  
BitmapField bitmap = new BitmapField(Bitmap.getBitmapResource("img.gif"));  
LabelField label = new LabelField("This is some text");  
grid.add(bitmap);  
grid.add(label);
```



# Screen Transitions

- Animated pushing and popping of screens
- Specify a “transition context” to the UI Engine



# Screen Transitions: Sample Code

```
import net.rim.device.api.ui.*;

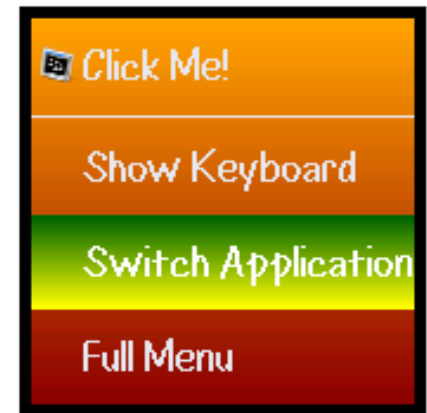
// Create a slide up transition when screen is pushed
TransitionContext context = new
    TransitionContext(TransitionContext.TRANSITION_SLIDE);

// Add transition type parameters
context.setIntAttribute(TransitionContext.ATTR_DURATION, 1000);
context.setIntAttribute(TransitionContext.ATTR_DIRECTION,
    TransitionContext.DIRECTION_UP);

// Set when the transition occurs
UiEngineInstance engine = Ui.getUiEngineInstance();
engine.setTransition(null, screen, UiEngineInstance.TRIGGER_PUSH, context);
```

# Menu Customization

- Adding icons to menu items
  - Uses the new Image API
- Customizing menu item backgrounds
  - Uses the Background API
- Customizing menu (popup) screen backgrounds and borders
  - Uses the Background and Border APIs
- Customizing menu (popup) screen font
  - Can use the new Font Loading APIs



# Menus: Sample Code 1/2

```
import net.rim.device.api.ui.*;
import net.rim.device.api.ui.component.*;
import net.rim.device.api.ui.decor.*;
import net.rim.device.api.ui.image.*;

protected void makeMenu(Menu menu, int context) {

    // Add menu item with icon
    MenuItem item = new MenuItem("Click Me!", 0, 0) {
        public void run() {
            Dialog.alert("Clicked!");
        }
    };
    item.setIcon(ImageFactory.createImage(
        EncodedImage.getEncodedImageResource("menuicon.png")));
    menu.add(item);
}
```

## Menus: Sample Code 2/2

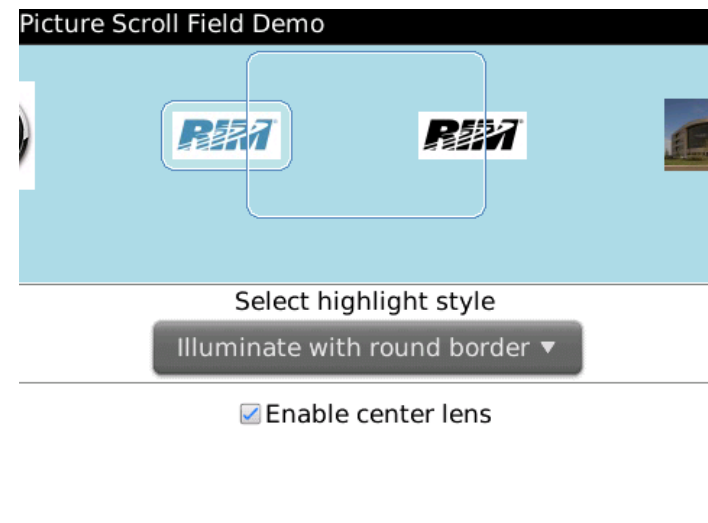
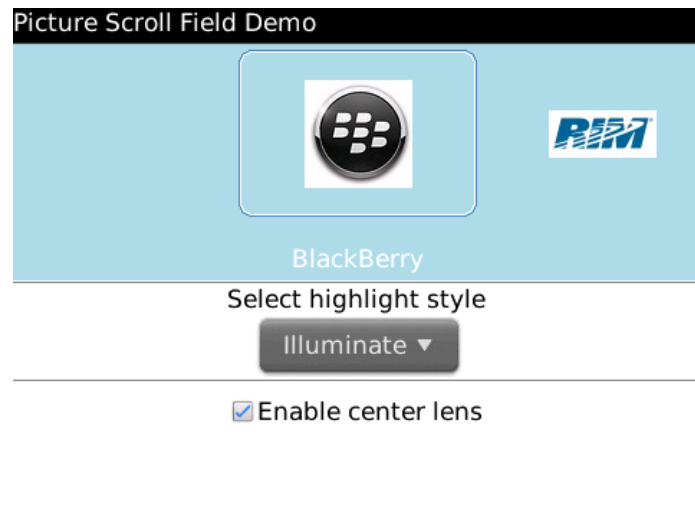
```
// Customize look and feel
try {
    FontFamily typeface = FontFamily.forName("BBCasual");
    menu.setFont(typeface.getFont(Font.BOLD, 20));
} catch(ClassNotFoundException e) {
    System.err.println("Could not load system font!");
}
menu.setBorder(BorderFactory.createSimpleBorder(
    new XYEdges(5, 5, 5, 5),
    new XYEdges(Color.BLACK, Color.BLACK, Color.BLACK, Color.BLACK),
    Border.STYLE_SOLID));
menu.setBackground(BackgroundFactory.createLinearGradientBackground(
    Color.ORANGE, Color.ORANGE, Color.DARKRED, Color.DARKRED));
menu.setCaretBackground(BackgroundFactory.createLinearGradientBackground(
    Color.DARKGREEN, Color.DARKGREEN, Color.YELLOW, Color.YELLOW));
super.makeMenu(menu, context);
}
```

# Extensions

- Picture Scroll
- Zoom Screen
- Eyelids

# Picture Scroll Field

- A horizontally scrolling field for pictures
- Supports various input methods
- Configurable display options



# Picture Scroll Field: Sample Code

```
import net.rim.device.api.ui.extension.component.*;

// Initialize an array of scroll entries
ScrollEntry[] entries = new ScrollEntry[ 2 ];
entries[0] = new ScrollEntry(Bitmap.getBitmapResource("berry.jpg"),
    "BlackBerry", "bb");
entries[1] = new ScrollEntry(Bitmap.getBitmapResource("logo_blue.jpg"),
    "Blue logo", "blue");

// Initialize the picture scroll field
PictureScrollField pictureScrollField = new PictureScrollField(150, 100);
pictureScrollField.setData(entries, 0);
pictureScrollField.setHighlightStyle(HighlightStyle.ILLUMINATE);
pictureScrollField.setLabelsVisible(true);
```

## Zoom Screen

- A screen for displaying images that allows for zooming and panning



# Zoom Screen: Sample Code

```
import net.rim.device.api.ui.extension.container.*;

EncodedImage image =
    EncodedImage.getEncodedImageResource("img/building.jpg");

// Somewhere on the event thread
UiApplication.getUiApplication().pushScreen(
    new ZoomScreen(image));
```

# Eyelids

- A manager that overlays fields on demand



# Eyelids: Sample Code

```
import net.rim.device.api.ui.extension.container.*;

// Step 1: Create the eyelid and configure it
EyelidFieldManager eyelidFieldManager = new EyelidFieldManager();
// Change the display time from the default 1.2s
eyelidFieldManager.setEyelidDisplayTime(2000);
// Show the eyelid on any user input
eyelidFieldManager.showOnInput(true);

// Step 2: add fields to the top eyelid
eyelidFieldManager.addTop(new LabelField("Eyelid Field Demo"));

// Step 3: add fields to the bottom eyelid
eyelidFieldManager.addBottom(new ButtonField("Eyelid button"));

// Step 4: add fields to the "eye"
eyelidFieldManager.add(new LabelField("Retina"));
eyelidFieldManager.add(new LabelField("Pupil"));
```

# Utilities

- Bitmap Scaling
- Bitmap Differences
- Font Loading

# Bitmap Scaling

- Create a scaled copy of a Bitmap image
- Three interpolation filters:
  - Lanczos
  - Bilinear
  - Box (Mean)
- Three aspect ratio parameters:
  - Stretch
  - Scale-to-Fit
  - Scale-to-Fill

# Bitmap Scaling: Code Sample

```
import net.rim.device.api.system.*;

// Load a 200x80 Bitmap
Bitmap bitmapOriginal = Bitmap.getBitmapResource("RIM.png");
// Create a 100x100 Bitmap
Bitmap bitmapScaled1 = new Bitmap(100, 100);
// Create a 200x50 Bitmap
Bitmap bitmapScaled2 = new Bitmap(200, 50);
// Scale without preserving aspect ratio
bitmapOriginal.scaleInto(bitmapScaled1, Bitmap.FILTER_LANCZOS);
// Scale preserving aspect ratio and fit vertically
bitmapOriginal.scaleInto(bitmapScaled2, Bitmap.FILTER_BILINEAR, Bitmap.SCALE_TO_FIT);
```



# Bitmap Differences

- Compares two Bitmap images and localizes where differences are found
- Intended for image rendering optimizations, i.e. minimizing redrawing region



# Bitmap Differences: Code Sample

```
import net.rim.device.api.system.*;

// Load two Bitmaps for comparison
Bitmap bitmap1 = Bitmap.getBitmapResource("bitmap1.png");
Bitmap bitmap2 = Bitmap.getBitmapResource("bitmap2.png");

// Compare the two Bitmaps
XYRect diffRect = new XYRect();
bitmap1.locateDifference(diffRect, bitmap2, 0, 0);

// Create difference Bitmap
Bitmap diffBitmap = new Bitmap(diffRect.width, diffRect.height);
bitmap2.scaleInto(diffRect.x, diffRect.y, diffRect.width, diffRect.height,
    diffBitmap, diffRect.x, diffRect.y, diffRect.width, diffRect.height,
    Bitmap.FILTER_BOX);
```

# Font Loading

- Allows applications to use custom fonts
- System-level fonts:
  - Appear in Screen/Keyboard Options
  - Available to all applications
  - Persistent
- Application-level fonts:
  - Only available to the loading application
  - Automatically unloads upon exit
- Currently supports TrueType Unicode fonts up to 90K in size

# Font Loading: Code Sample

```
import net.rim.device.api.ui.*;

String filename = "MyApplicationFont.ttf";
String family = "ApplicationFont"; // typeface name
LabelField sampleFont = new LabelField("The quick brown fox jumped over the lazy dog");

// 1. Get instance of the FontManager and load application font
int rc = FontManager.getInstance().load(filename, family,
    FontManager.APPLICATION_FONT);

// 2. Get FontFamily object for the loaded font
if(rc == FontManager.SUCCESS) {
    FontFamily typeface = FontFamily.forName(family);

// 3. Create Font object (with style and size) and set on component
    Font myFont = typeface.getFont(Font.PLAIN, 50);
    sampleFont.setFont(myFont);
}
```

# Other Announcements From The BlackBerry Developer Conference

# BlackBerry Theme Studio 5.0

- Announcing availability of Theme Studio 5.0
- Enhanced UI and workflow
- Support for new devices (Bold 9700) and 5.0 'Today' themes
- Support for ringtones
- Screen transitions
- Ability to import Photoshop files
- Accepting themes in AppWorld today

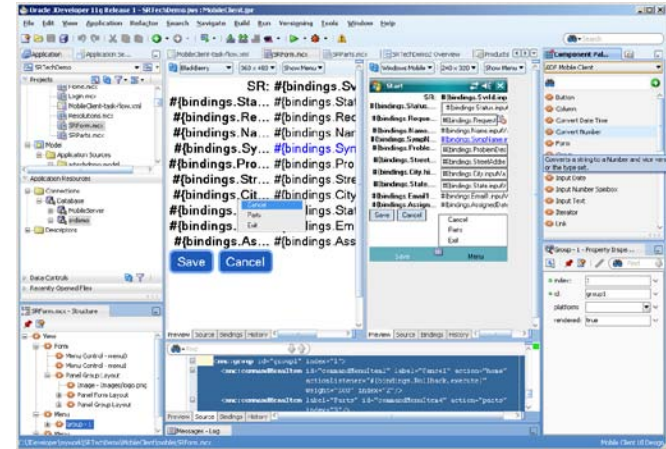


# Platform Services

- Commercial Services
  - Advertising Service
    - Provides one-stop access to multiple networks, centralized management console, reporting, and Omniture analytics
  - Payment Service
    - Support taking payments within an application
    - Provides end users a trusted, streamlined experience
- Location Services
  - Reverse Geocoding
    - Given a lat/long, provides an approximate address
  - Locate
    - Estimation of a user's location based on cell tower positioning
  - ETA App and Service
    - Uses anonymous crowd-sources data + traffic history data
- Connectivity Services
  - Push Service
    - Free, unlimited frequency, no push acknowledgements

# Partnerships

- Oracle
  - JDeveloper 11g available today for mobile web development
  - ADF Mobile will allow building java applications
- Adobe
  - Help content designers and developers work better together
  - Optimize graphics for BlackBerry
  - Dreamweaver – create mobile web content optimized for BlackBerry



# More Announcements

- Support for OpenGL in future devices
- WebKit Browser Support
- BlackBerry Certified Application Developer
  - Developing Applications for the BlackBerry Solution
  - Developing Java Applications for the BlackBerry Platform
  - Developing Web Applications for the BlackBerry Platform
  - <http://www.blackberry.com/certification>



Win an AT&T Smartphone!

# CONTEST

Question: What new development platform allows you to create an application using standard web technologies, but package them as a Java application?

- a) BlackBerry Java Plug-in for Eclipse
- b) BlackBerry Widgets
- c) Java Web Toolkit
- d) BlackBerry Theme Studio 5.0

Send answers to:

[devCentral-biz@awsmail.att.com](mailto:devCentral-biz@awsmail.att.com)

Winners will be selected randomly from correct responses

# New Development Tools for BlackBerry smartphones

## Questions and Answers

Webcast slides will be posted shortly on  
<http://developer.att.com/webcasts>