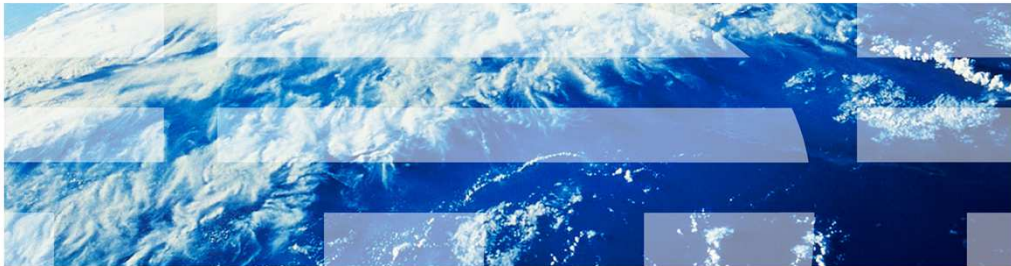


WebSphere Commerce V7 Feature Pack 6

Worklight integration



This presentation provides an overview of the new hybrid application and Worklight integration in Version 7 Feature Pack 6. You should have a general knowledge of the existing WebSphere® Commerce mobile commerce solution before viewing this presentation.

Table of contents

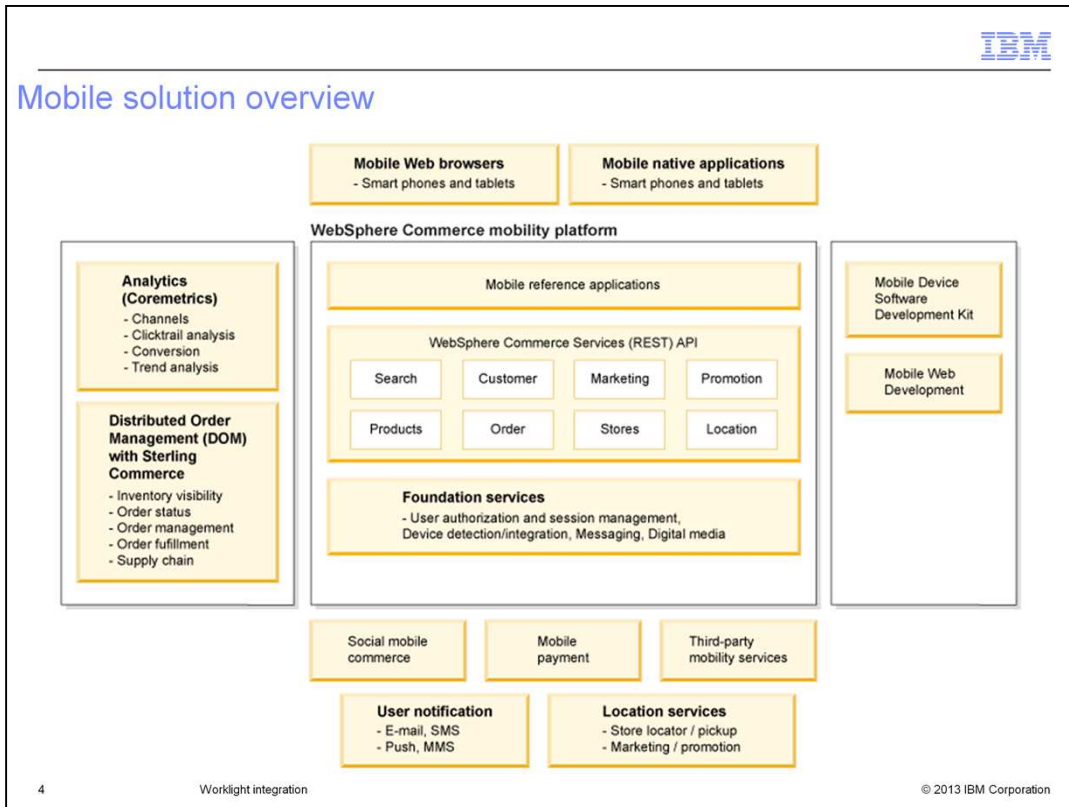
- Overview
- Architecture highlights
- Adoption details
- Troubleshooting

This presentation begins with an overview of the hybrid application and the Worklight development platform. Next a summary of the application architecture is provided. The presentation concludes with some tips for adopting the new hybrid application and troubleshooting.

Overview

This section provides an overview of the Worklight hybrid application.

Mobile solution overview

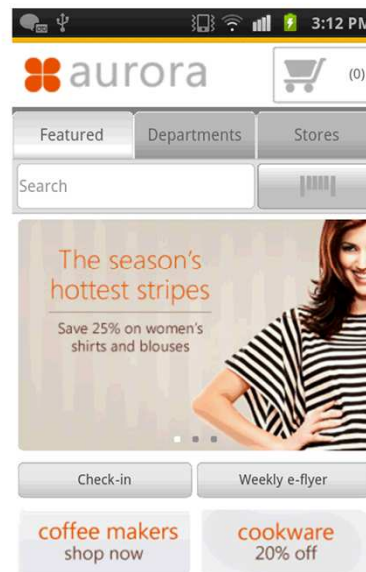


This chart summarizes the mobile solution in Feature Pack 6. The top of the diagram shows the various ways smart phones and tablets can access a WebSphere Commerce store. These include browser-based stores and device-specific applications. Collectively, the mobile web stores and applications are referred to as the mobile reference applications. Similar to the traditional starter stores, the device-specific applications make use of services to access the WebSphere Commerce application.

The left side of the diagram lists WebSphere Commerce capabilities that can be added to the mobile reference applications such as analytics and distributed order management (DOM) integration. For example, cross channel inventory availability and orders submitted through the mobile channel can be processed by your back-end DOM system. The right side of the diagram lists the technologies used in building the mobile reference applications. This presentation will focus on the Worklight Studio IDE. Finally, the bottom of the diagram highlights other integrations that are possible with this solution such as user notification capabilities, social mobile commerce, mobile payment, location services and other vendor mobility services.

Hybrid enhancements

- Updated hybrid application for Android smart phones
 - Based on Worklight platform
 - Delivered as a sample project to deploy cross-platform hybrid applications
 - Server-side content from the Aurora mobile web store
 - Store preview capability
- Same native capabilities as previous hybrid application
 - Shell
 - Barcode scanner
 - Voice search
 - Address book
 - Store locator maps



The new hybrid application for the Aurora store is based on the IBM Worklight platform. This platform is summarized on the next slide. The application is delivered as a sample Worklight project with device-specific code for Android smartphones. It is possible to add device-specific code for other platforms into the sample project if you want to deploy hybrid applications for other platforms. The server-side content for the hybrid store is provided by the Aurora mobile web store. The hybrid application can take advantage of generated store preview links. Preview link generation is covered in the store preview topic in the business user interface section. As in Feature Pack 4, the hybrid application makes use of some native capabilities. These include the application shell, barcode scanner, voice search, address book and maps for displaying store locations.

Worklight summary



Worklight Studio

The most complete, extensible environment with maximum code reuse and per-device optimization



Worklight Server

Unified notifications, runtime skins, version management, security, integration and delivery



Worklight Runtime Components

Extensive libraries and client APIs that expose and interface with native device functionality



Worklight Console

A web-based console for real-time analytics and control of your mobile applications and infrastructure

- Supported versions
 - IBM Worklight Developer Edition V5.0.5 and V5.0.6 – for evaluation purposes only
 - IBM Worklight Consumer Edition V5.0.5 and V5.0.6
 - IBM Worklight Enterprise Edition V5.0.5 and V5.0.6

6

Worklight integration

© 2013 IBM Corporation

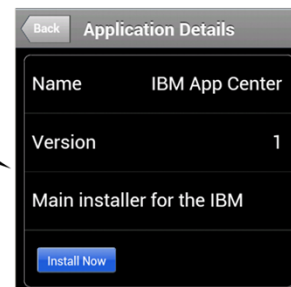
The Aurora hybrid sample application is provided as an IBM Worklight project. IBM Worklight provides an open, comprehensive and advanced mobile application platform for smartphones and tablets. Using standards-based technologies and tools, IBM Worklight provides a single integrated platform that includes a comprehensive development environment, mobile-optimized runtime middleware, an enterprise application store, and an integrated management and analytics console.

Worklight Studio is a stand-alone Eclipse-based IDE for mobile application development. Worklight Server provides application management capabilities for deployed applications. The Worklight Runtime Components include libraries that assist in building the applications. Worklight Console provides a web-based interface to the management capabilities supported by Worklight Server. The supported versions of Worklight for this integration are shown on the slide. Worklight is a separately licensed product. Deploying a hybrid application based on Worklight requires a license. You do not require a license to test the sample hybrid application provided with Feature Pack 6.

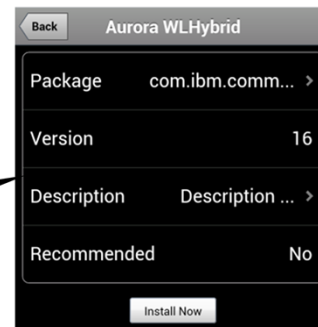
Managing applications under development – IBM Application Center

- Included with Worklight Server
- Acts as an application store during development
 - Upload applications to Worklight Server
 - Manage versions
 - Manage access privileges
- Install the Application Center application on test devices
 - List versions of applications under development that are available to test
 - Install and uninstall applications outside of a vendor marketplace
 - Switch between different versions of the same application

Install IBM Application Center



Use IBM Application Center to install applications under development



7

Worklight integration

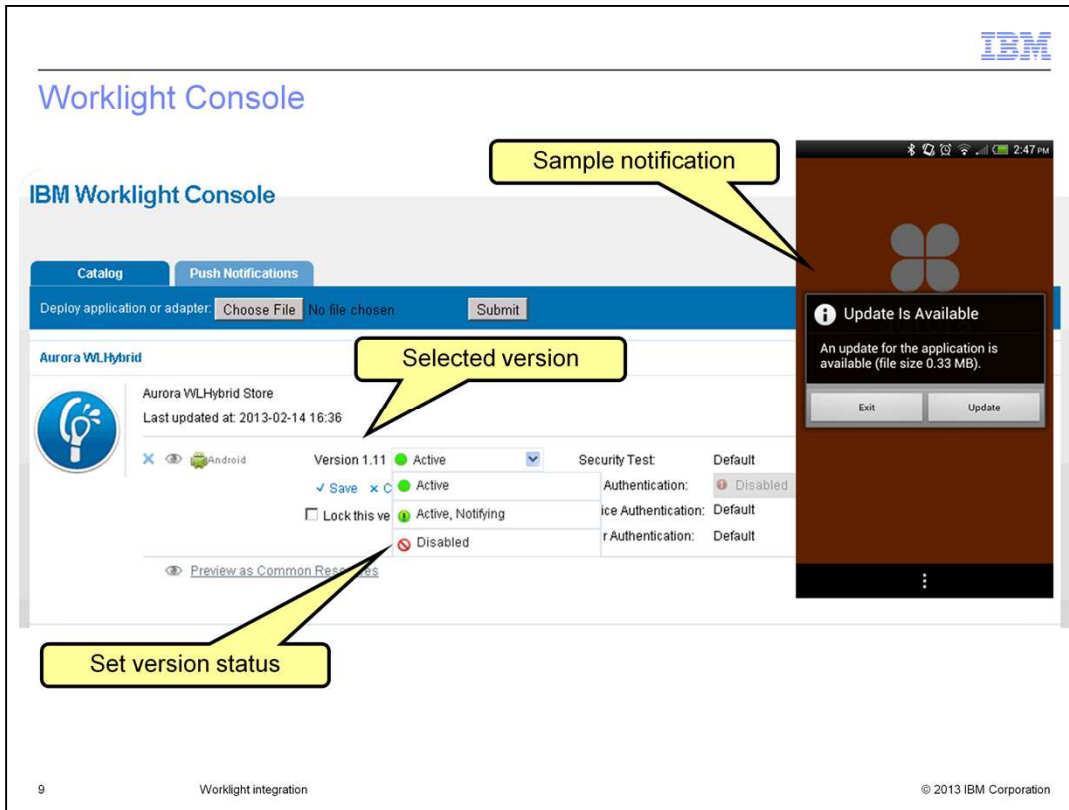
© 2013 IBM Corporation

The Worklight Server provides application management capabilities that were not previously available in the mobile solution. While applications are under development, the IBM Application Center provides a way to easily manage and deploy them to test devices. The Application Center mobile client application needs to be installed on the test device. This application is installed from the Application Center website. This application then provides a list of all store applications under development including version numbers. The store applications can be installed and uninstalled through the Application Center, a task that normally requires a vendor marketplace. Since version numbers are tracked, it is also possible to see when a new version of the application is available to test and to switch between versions.

Managing deployed applications

- Direct update of client-side web assets
 - Latest code uploaded to Worklight Server
 - Installed application prompts customer to install new web assets
- Native code update
 - Requires a platform-specific application store
- Worklight Console
 - GUI frontend for Worklight Server
 - Version control
 - Send notifications to customers when a new version is available to install
 - Disable specific versions to force customers to upgrade

Once an application is deployed to the vendor marketplace, the Worklight Server allows you to retain some control over which version of the application your customers are running. Direct update allows new web assets to be pushed out to customers with a previously installed version and prompts them to install the new web assets. This update method does not work for native code updates. In that case, the application must be reinstalled from the marketplace. In the event that a new version of the native code is released, Worklight Server provides the option to send a message to all users of the current version informing them of the update. It is also possible to disable specific versions of the application to force customers to upgrade to a new version. Version control tasks are accomplished using the Worklight Console which provides a GUI frontend for Worklight Server.



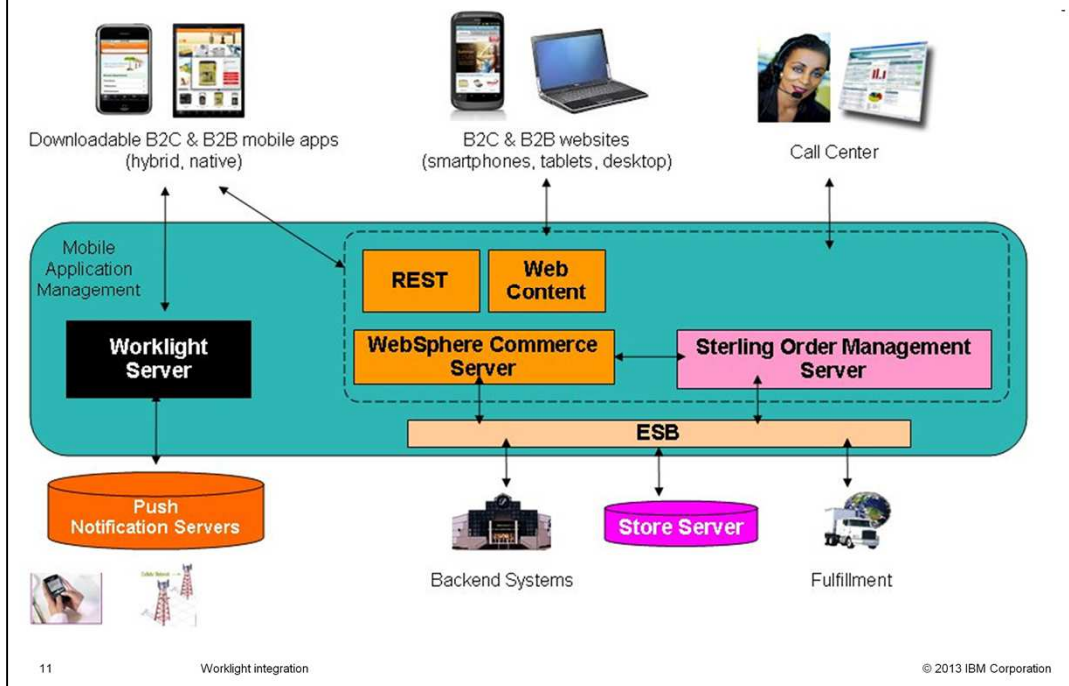
This is a screen capture of Worklight Console which provides access to the Worklight Server capabilities. Using the version management tools, you can send notifications to customers or disable a specific version of the application. A sample notification is shown on the right of the screen.

Worklight Server includes some capabilities not demonstrated in this integration. These are: push notifications, adapters, and application authentication.

Architecture highlights

This section covers the architecture highlights of the new hybrid application.

High level architecture



On this slide, you see a high level architecture of the Worklight integration with WebSphere Commerce. When the hybrid application is launched, a local HTML page initializes the Worklight runtime and connects to the Worklight Server. The connection allows the application to be managed through Worklight Console. If an administrator has disabled this version of the application, the customer is prompted to download the latest copy. If updated client-side web assets are available, they can be downloaded directly from the Worklight Server. Although not used in the Feature Pack 6 sample application, push notifications can also be sent from Worklight Server to connected devices.

The main content for the application is generated server-side by WebSphere Commerce and displayed in an embedded browser in the hybrid application. Some client-side JavaScript libraries such as Cordova and Worklight JS are placed on the WebSphere Commerce server as part of the mobile web store.

IBM

Hybrid application construction

Remote content loaded from WebSphere Commerce (blue) Native search dialog UI integrated using Worklight (green)

Native options menu generated from Worklight (red)

12 Worklight integration © 2013 IBM Corporation

This slide illustrates where different parts of the hybrid application are created. Updates to the native device code still require the customer to download a new version of the application. The area highlighted in blue is remote content loaded from the mobile web store in WebSphere Commerce. The hybrid application uses a custom user-agent to identify itself to WebSphere Commerce. This allows the server-side code to return Worklight-specific web assets. The area highlighted in red is a native Android options menu generated by Worklight using the Worklight JavaScript API . Updates to native elements such as this are made in server-side JavaScript, and do not require the customer to download a new version of the application. The area highlighted in green is a native Android search dialog, an example of integrating native and web pages using the Worklight NativePage API for the remote content. The search dialog is a custom native page and updates to the native elements still require the customer download a new version of the application.

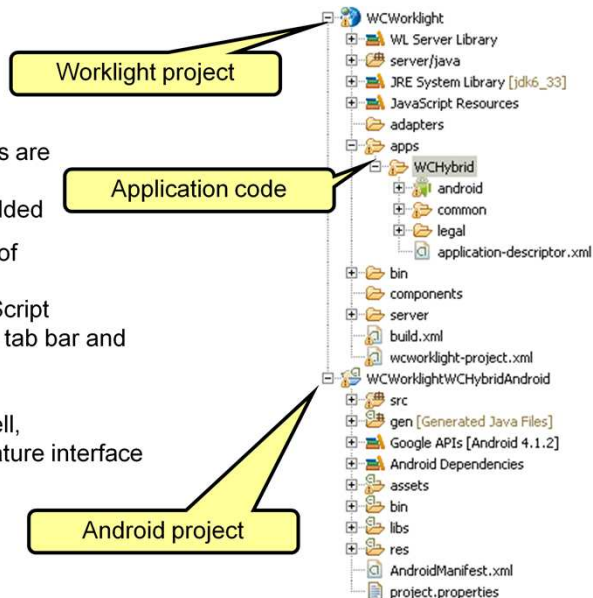
Worklight runtime components

- Worklight runtime API
 - Built on top of Apache Cordova
 - Replaces JavaScript bridge code from Feature Pack 4
 - Used to access native device features
 - Address book
 - Store locator / maps
 - Barcode scanner
 - Voice search
 - Menu options
- Store developers need to import Worklight runtime libraries into their toolkit environment

To support the hybrid application, the mobile web store code includes some hooks to native device capabilities. In Feature Pack 4, these hooks were implemented using custom-built “JavaScript bridge” code. In the updated Feature Pack 6 store, the custom code is replaced with Worklight runtime API calls. The Worklight runtime API is based on Apache Cordova. The native features used by the hybrid application have not changed from Feature Pack 4. They are address book for contacts, Google maps for store location, barcode scanner (providing a vendor application is installed) and voice search. The menu options populated in the device menu bar are also set using the APIs. To make use of these APIs, store developers need to import the Worklight runtime libraries into WebSphere Commerce Developer.

Client-side development - Worklight Studio

- Standalone Eclipse-based IDE
 - Common JavaScript frameworks are included
 - Android SDK plug-ins can be added
- Used to develop client-side portion of smartphone hybrid application
 - Client-side HTML / CSS / JavaScript
 - For example splash screen, tab bar and settings page
 - Android native code
 - For example application shell, initialization code, native feature interface



14

Worklight integration

© 2013 IBM Corporation

Worklight Studio is a stand-alone Eclipse-based IDE for mobile application development. The Worklight development tools are not included in WebSphere Commerce Developer. The Worklight Studio IDE comes with several popular JavaScript frameworks such as Dojo, jQuery Mobile, and Sencha Touch already installed. SDKs for specific mobile platforms, such as Android, need to be added on. In Feature Pack 6, Worklight Studio is used to develop the client-side portion of the Aurora smartphone hybrid application. The files created and updated in Worklight Studio include the client-side HTML, CSS, and JavaScript code for the splash screen, tab bar and developer settings page. Also included is the Android native code for the application shell, initialization and native feature interface.

On the right side of the screen, you see the structure of the sample Worklight project. The WCHybrid folder contains all the common and platform specific code for the application. From the Worklight project, you can generate platform-specific applications such as the Android one shown here. Once the Android project is generated, the application can be signed, exported and published to a application marketplace following the standard process.

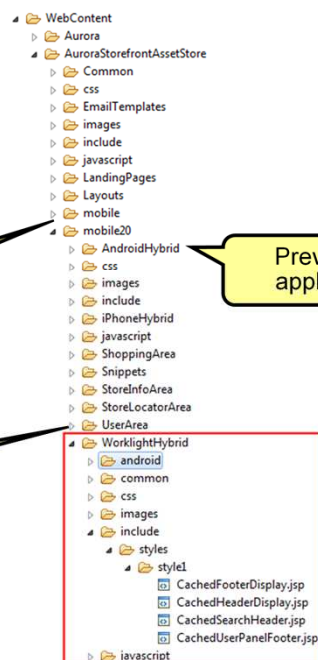
Server-side development - WebSphere Commerce

- Hybrid-specific server-side code is included in the mobile web store
- Some JavaScript files must be copied from your Worklight project

Mobile web store files

Previous hybrid application files

Worklight hybrid application files



15

Worklight integration

© 2013 IBM Corporation

This slide shows the location of the server-side code for the hybrid application. Most of the store JSP files are shared with the smartphone web store. A few files are specific to the hybrid application and these are contained in the WorklightHybrid subdirectory. The application-specific files include CSS, JavaScript, image and JSP files specific to the hybrid application pages. Some are platform-specific and others are common to any device running the hybrid application. The hybrid-specific JavaScript files include the code that makes use of the Worklight runtime API to access the address book, barcode scanner, native map and geolocation integration. The include folder contains a set of Worklight hybrid-specific JSP files such as `CachedHeaderDisplay.jsp`, `CachedFooterDisplay.jsp` to set up the hybrid application client-side environment.

Adoption details

This section summarizes the steps needed to adopt the new hybrid application into an existing environment.

Adoption checklist

- Prerequisites
 - Aurora mobile web store is published
 - IBM Worklight license is available (for hybrid application)
 - IBM Worklight Consumer Edition V5.0.5 or V5.0.6
 - IBM Worklight Enterprise Edition V5.0.5 or V5.0.6
 - IBM Worklight Developer Edition V5.0.5 and V5.0.6 – for evaluation purposes only
- Installation
 - Separate from WebSphere Commerce installation
- Migration
 - None
- Configuration effort
 - Moderate

To make use of the hybrid application, you first need to publish the Aurora mobile web store. To deploy a hybrid application based on the sample, you need to purchase a license for one of the supported versions of IBM Worklight. It is not bundled with WebSphere Commerce. When installing Feature Pack 6, there are no steps specific to the hybrid application. Setting up Worklight is a separate installation process. There is no migration from the previous version of the hybrid application to the new version. The older version continues to be supported but has not been updated for the Aurora store. Configuration for the hybrid application requires a moderate amount of effort since you will need to configure Worklight Studio and import the sample code.

Configuration

- Publish the mobile web store for smartphones (AuroraMobile.sar)

<http://pic.dhe.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.starterstores.doc/tasks/tsmmopub.htm>

- Configure IBM Worklight to build a hybrid application using the sample code

<http://pic.dhe.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.starterstores.doc/concepts/csmmobileappworklight.htm>

The first link provided on this slide is to publish the Aurora mobile web store for smartphones. This store is contained in AuroraMobile.sar.

The second link provides an end-to-end guide to installing and configuring Worklight and importing the sample code. There are also some steps needed to configure WebSphere Commerce.

Customization

- Customize the Worklight sample project for your needs
 - Add a Google map key to allow displaying store location information

<https://developers.google.com/maps/documentation/android/v1/maps-api-signup>

- Adjust native shell settings such as store name and color scheme

http://pic.dhe.ibm.com/infocenter/wrklght/v5r0m5/topic/com.ibm.worklight.help.doc/start/c_gettingstarted.html

http://pic.dhe.ibm.com/infocenter/wrklght/v5r0m6/topic/com.ibm.worklight.help.doc/start/c_gettingstarted.html

Once you have the sample Worklight project loaded, you will likely need to customize the sample code to suit your needs. This slide provides links to help with two common customizations. The first link is for acquiring an Google map key so your application can display maps for store locations. The second link is for getting started with Worklight application development. You will need to become familiar with the project structure and assets to adjust native shell settings such as the store name and color scheme.

Mobile platform support

- Smartphone (mobile web)
 - Android 2.x, 4.0, and 4.1
 - iPhone iOS 4, 5, and 6
 - Blackberry 6 and 7
- Tablet (mobile web)
 - Android 4.0 and 4.1
 - iPad iOS 4, 5 and 6
 - Blackberry 6 and 7
- Hybrid smartphone application
 - Android 2.2, 2.3, 4.0, and 4.1

This slide summarizes the operating systems supported by the various mobile store options. Applications released in previous feature packs and not updated in Feature Pack 6 will continue to be supported as-is.

Log and trace

- Client-side logging
 - Worklight JavaScript logging
 - WL.Logger JavaScript API
 - Writes log output to the native platform console, such as LogCat
 - Controlled by `enableLogger` parameter in `initOptions.js`
 - Native code logging classes
 - Log (Android)
 - NSLog (iOS)
- Server-side logging
 - Standard WebSphere Commerce store trace

There are two options for adding logging to the client-side code in the hybrid application. Worklight has a JavaScript API for logging. The logger outputs the trace to the native platform console for the device being tested. For example LogCat for Android devices. The logger is controlled by the `enableLogger` parameter in `initOptions.js` (client-side) and `initOptionsApp.js` (server-side). In the sample application code, the logger is turned on by default. For debugging problems in native code, there are platform-specific logging classes you can use such as Log for Android and NSLog for iOS. Server-side logging is also available using standard WebSphere Commerce trace statements. This solution does not add any new server-side trace methods.

Troubleshooting

This section summarizes some troubleshooting tips for the hybrid application.

Diagnosing problems – mobile web

- Does the problem exist in the desktop store as well?
 - Yes: Start with the component where the problem is
 - No: Start with the store JSP
- UI elements not working correctly
 - Device does not have WebKit browser support
 - Dojo is not being loaded
- Debugging mobile web stores from a desktop
 - Use Safari or Chrome for WebKit support
 - User agent switching simulates different mobile devices

When debugging problems in the mobile web stores it helps to verify whether the function is working in the desktop store. If not, the component that has the problem is a good place to start. If the desktop store is working correctly and it's only the mobile store that has a problem, start your debugging from the affected store page. If you have problems with the UI not displaying or working correctly there are two likely causes. You should check that you are viewing the store in a browser with WebKit support and that Dojo is being loaded correctly. To debug a mobile web store from a desktop machine, use either Safari or Chrome. Both browsers have WebKit support. You can make use of user agent switching in your desktop browser to simulate different mobile devices.

Diagnosing problems – hybrid applications

- Does the problem exist in the mobile web application as well?
 - Yes: Debug web application
 - No:
 - Check the application configuration
 - Check if the problem scenario invokes any hybrid-specific code
- To debug problems in Worklight code
 - Run the application in the emulator
 - LogCat captures trace from both client-side and server-side Worklight code
- To debug problems in Android code
 - Run the application in the emulator
 - LogCat
 - Java debugger

Since hybrid applications get the majority of their screen content from the same JSP files as the mobile web application, you should start by checking whether the problem also exists in the mobile web store. If the problem is unique to the hybrid application, the next step is to verify the application is configured correctly. The host name, store ID or catalog ID for the application might be incorrect. If the application is configured correctly, the problem might be in one of code paths that invokes hybrid-specific code. If the problem is in the Worklight code, you can use the WL.Logger API to output trace. LogCat can be used to view trace statements generated by either client-side or server-side Worklight code. You can debug problems in the Android code by running the application in the emulator and viewing the log output in the LogCat view. You can also set breakpoints in the Java code and use the Eclipse Java debugger.

Summary

- Overview
- Architecture highlights
- Adoption details
- Troubleshooting

This presentation began with an overview of the hybrid application and the Worklight development platform. Next a summary of the application architecture was provided. The presentation concluded with some tips for adopting the new hybrid application and troubleshooting.

References

- Hybrid applications using IBM Worklight

<http://pic.dhe.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.starterstores.doc/concepts/csmmobileappworklight.htm>

- IBM Worklight 5.0.5 Information Center

– <http://pic.dhe.ibm.com/infocenter/wrklight/v5r0m5/index.jsp>

- Android developer site

– <http://developer.android.com/index.html>

This slide contains some useful references for working with the new hybrid application in Worklight.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WorklightIntegration.ppt

This module is also available in PDF format at: [../WorklightIntegration.pdf](#)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2013. All rights reserved.