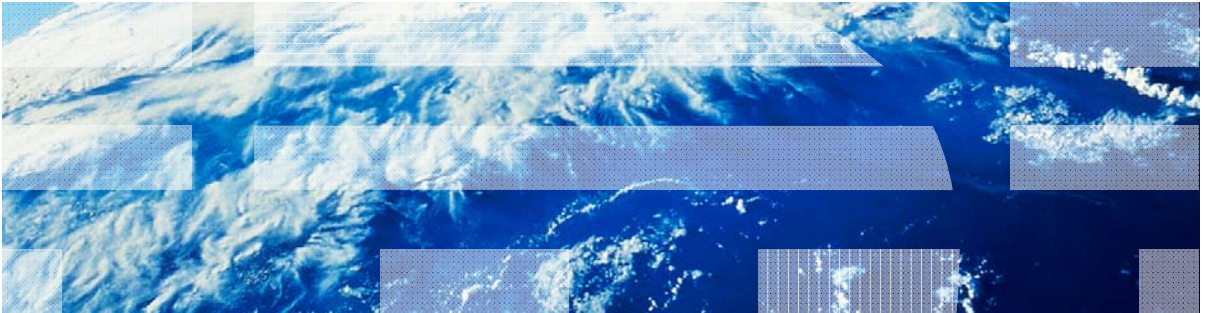


IBM Worklight V6.0.0 Getting Started

Shell development concepts



Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Worklight is a trademark or registered trademark of Worklight, an IBM Company. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)” at www.ibm.com/legal/copytrade.shtml.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Other company products or service names may be trademarks or service marks of others.
- This document may not be reproduced in whole or in part without the prior written permission of IBM.

About IBM®

- See <http://www.ibm.com/ibm/us/en/>

Agenda

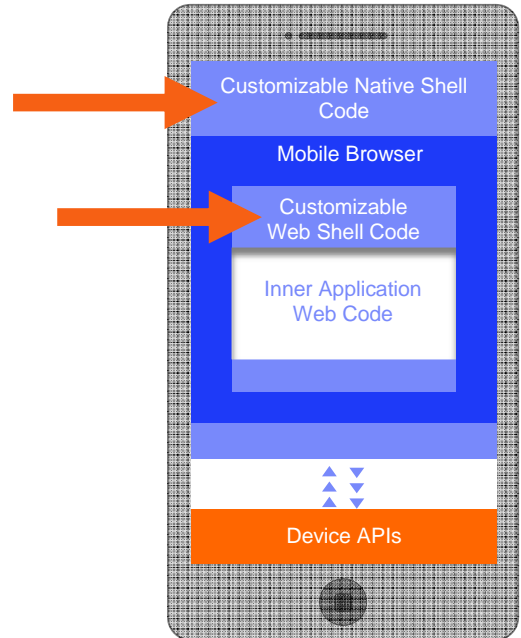
- Overview
- Creating a shell component
- Using a shell component in a test application
- Creating and using a shell bundle in an Inner application

Overview

- The main idea behind shell component methodology is to create two levels of development inside the organization:
 - Developers who are skilled in native development implement native and web code-bases that can be used as a starting point for one or more applications. For example:
 - Native functionality to be invoked from JavaScript (Cordova plug-ins)
 - Authentication framework
 - Security configuration
 - Web resources that are shared between applications, such as logotypes and themes
 - Developers who have less native development skills but more web expertise receive a ready-to-use shell component and use it as a wrapper to create the organization applications. For example:
 - Business logic
 - UI development
 - Data integration

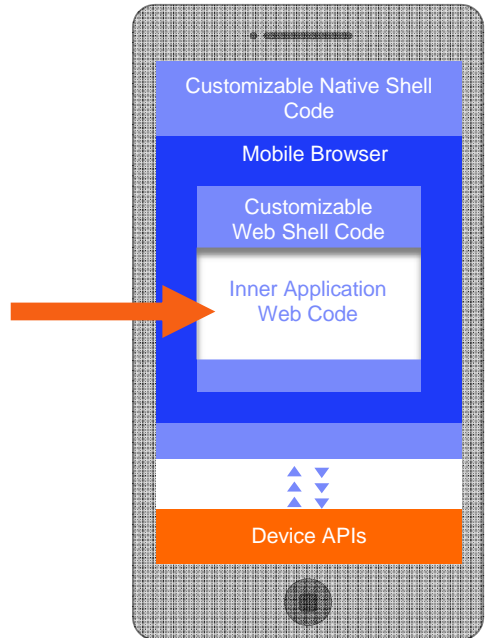
Overview – Architecture of the Shell-based Application

- **Shell component** - a component to be used by inner applications as a code base wrapper. It usually consists of native classes and shell-specific web resources that are going to be used in inner applications. The shell component is implemented by shell developers and sent to inner application developers to use.



Overview – Architecture of the Shell-based Application

- **Inner application** – web resources (HTML / JavaScript / CSS) that are run inside the shell component.
- **Test application** – The shell component is not executable by itself. After it is created, an inner application is automatically added to the project by Worklight® Studio. This application is used by the shell developer to test the shell component functionality.



Agenda

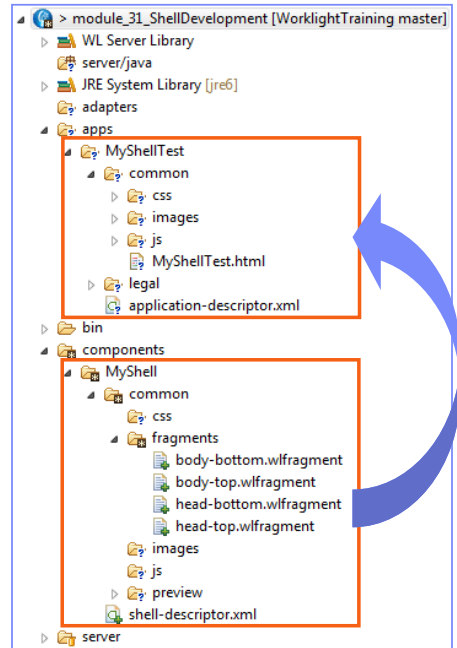
- Overview
- **Creating a shell component**
- Using a shell component in a test application
- Creating and using a shell bundle in anInner application

Creating a shell component

- This training module covers the basics of creating and using a shell component and an inner application.
- A prerequisite for successful completion of this and subsequent training modules is a solid knowledge of Worklight hybrid application development concepts.
- Make sure that you have a solid understanding of Worklight development principles, specifically iOS and Android development.
- This module covers shell development concepts, using a common folder.
- The following modules cover Android and iOS shell development.

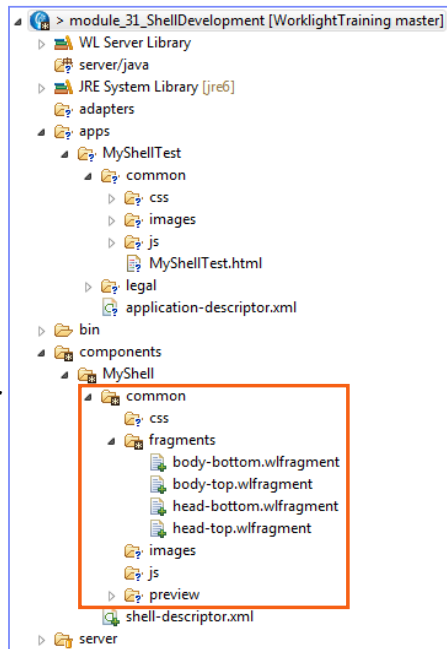
Creating a shell component

- A *shell component* is a building block that is used to create inner applications.
- Add a shell component to your project and name it **MyShell**.
- Note that the **MyShellTest** application was automatically created for you; this is a test application as described in the *Overview* section. You can use it to test and debug the shell component.



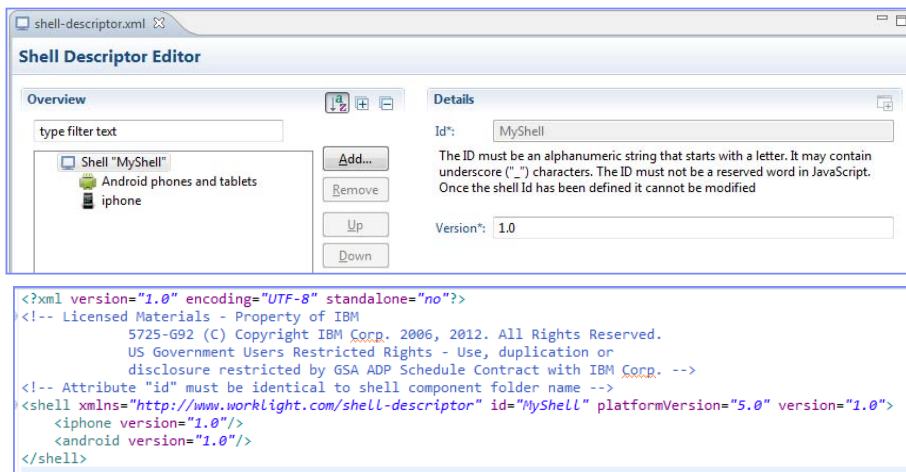
Creating a shell component

- The `common` folder of the shell component contains the following folders:
 - `css`, `images`, `js` – these folders contain web resources that are added automatically to inner applications at build time,
 - `fragments` – this folder contains HTML fragments that will be added to predefined locations in the main HTML file of the inner application.
 - `preview` – this folder can be used to implement stubs for simulating native functionality in the Worklight Console preview instead of receiving exceptions.



Creating a shell component

- The `shell-descriptor.xml` file contains shell component metadata and application-specific properties
- Application-specific properties that are set in the shell descriptor are used in all inner applications
- `shell-descriptor.xml` can be edited in either Design or Source mode



The screenshot shows the 'Shell Descriptor Editor' window. The 'Overview' pane displays a tree view with 'Shell "MyShell"' selected, containing 'Android phones and tablets' and 'iphone'. The 'Details' pane shows 'Id': 'MyShell' and 'Version': '1.0'. Below the panes is a code editor showing the XML content of the descriptor file.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Licensed Materials - Property of IBM
      5725-G92 (C) Copyright IBM Corp. 2006, 2012. All Rights Reserved.
      US Government Users Restricted Rights - Use, duplication or
      disclosure restricted by GSA ADP Schedule Contract with IBM Corp. -->
<!-- Attribute "id" must be identical to shell component folder name -->
<shell xmlns="http://www.worklight.com/shell-descriptor" id="MyShell" platformVersion="5.0" version="1.0">
  <iphone version="1.0"/>
  <android version="1.0"/>
</shell>
```

Agenda

- Overview
- Creating a shell component
- Using a shell component in a test application
- Creating and using a shell bundle in an Inner application

Using a shell component in a test application

- Follow these instructions to develop a functioning shell component.
- Create a `myshell.js` file in the `MyShell\common\js` folder.
- Add the following function to it:

```
function sayHelloFromShell(){  
    alert("Hello from Shell");  
}
```

- Modify the `body-top.wlfragment` file and add the following lines to it.

```
<h1>This is a header that will be visible in all inner applications that use this Shell</h1>  
<script src="js/myshell.js"></script>
```

Using a shell component in a test application

- Modify the `MyShellTest.js` file in the `apps/MyShellTest/common/js` folder.
- Invoke the function that you previously added in the shell component.

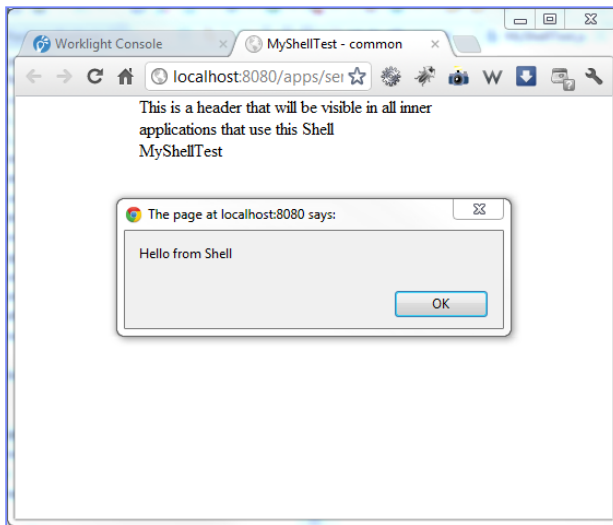
```
function wlCommonInit(){  
    sayHelloFromShell();  
}
```

- Note that the **sayHelloFromShell()** function is not a part of the inner application, but is from the shell component.
- Build and deploy the `MyShellTest` application.
- When you have built and deployed your application, you will find it in Worklight Console as a regular hybrid application.



Using a shell component in a test application

- Preview your MyShellTest application. Note that it contains web resources from both the shell component and the inner application.



Agenda

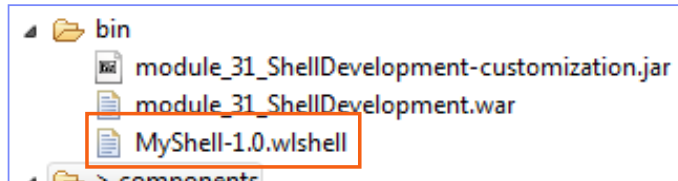
- Overview
- Creating a shell component
- Using a shell component in a test application
- **Creating and using a shell bundle in an inner application**

Creating and using a shell bundle

- When the shell developer builds a shell component a `.wshell` file is created in the project `bin\` folder. This file is called a *shell bundle* and can be sent to inner application developers to use.
- A shell developer working with a test application is not required to explicitly create a shell bundle. The test application references the shell component source code directly from the location that is specified in its `application-descriptor.xml` file.
- However, when the shell developer wants to send the shell component to the inner application developer the need to create a shell bundle arises.

Creating and using a shell bundle

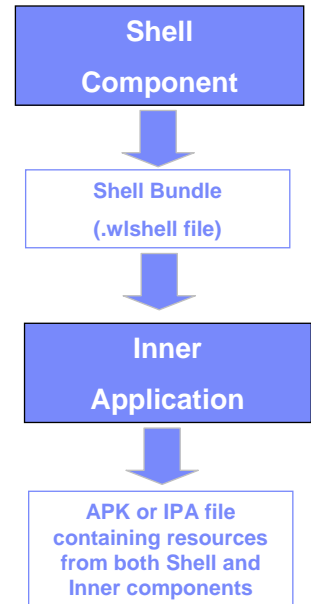
- To create a shell bundle, right-click a shell component folder and select **Run > Build Shell Component**.
- The `.wshell` file is created in the `bin\` folder of your project, as described previously.



- This file can be sent to inner application developers.

Creating and using a shell bundle

- The inner application developer must copy the shell bundle file to a Worklight project.
- When inner application developers create a new inner application, they must specify the location of a shell bundle file.
- If a new shell bundle file is received from shell component developers, inner application developers must replace the existing shell bundle file and rebuild their applications.



Next training modules

- Shell development for iOS and Android environments is covered in subsequent training modules.
- Ensure that you have a solid understanding of this training module before you move to the next ones.

Check yourself questions

- In which case should the shell development approach be taken?
 - The company has only two developers
 - The company wants to create a number of mobile applications sharing the common native code base and authentication mechanism
 - The company wants to create a native iOS and Android application, but its resources are limited
 - The company has distributed development teams that are situated on different continents but requiring to share the source code
- Which of the following should not be a part of the shell component?
 - Authentication module
 - Native functionality JavaScript wrapper
 - Application UI components
 - Company logotype that should be shared between several applications
- When a shell developer completes developing the shell components, what is the correct way to distribute it to inner application developers?
 - Compressing the Worklight project and emailing it to inner application developers
 - Committing the Worklight project to a source control management system and telling inner application developers to use source code from it
 - The shell developer should not distribute the shell component to inner application developers. They should send their inner applications to the shell developer in order to build them
 - Sending the `worklight` shell bundle file to inner application developers

Check yourself questions

- In which case should the shell development approach be taken?
 - The company has only two developers
 - The company wants to create a number of mobile applications sharing the common native code base and authentication mechanism
 - The company wants to create a native iOS and Android application, but its resources are limited
 - The company has distributed development teams situated on different continents but requiring to share the source code
- Which of the following should not be a part of the shell component?
 - Authentication module
 - Native functionality JavaScript wrapper
 - Application UI components
 - Company logotype that should be shared between several applications
- When a shell developer completes developing the shell components, what is the correct way to distribute it to inner application developers?
 - Compressing the Worklight project and emailing it to inner application developers
 - Committing the Worklight project to a source control management system and telling inner application developers to use source code from it
 - The shell developer should not distribute the shell component to inner application developers. They should send their inner applications to the shell developer in order to build them
 - Sending the `wlshell` shell bundle file to inner application developers

Notices

- Permission for the use of these publications is granted subject to these terms and conditions.
- This information was developed for products and services offered in the U.S.A.
- IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
- IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
 - IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.
- For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:
 - Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan
- **The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.**
- This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.
- Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
- IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
- Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:
 - IBM Corporation
Dept F6, Bldg 1
294 Route 100
Somers NY 10589-3216
USA

- Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.
- The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.
- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

- This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
- Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:
 - © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp., enter the year or years. All rights reserved.

Privacy Policy Considerations

- IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.
- Depending upon the configurations deployed, this Software Offering may use session cookies that collect session information (generated by the application server). These cookies contain no personally identifiable information and are required for session management. Additionally, persistent cookies may be randomly generated to recognize and manage anonymous users. These cookies also contain no personally identifiable information and are required.
- If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent. For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy>; and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details>; the sections entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Support and comments

- For the entire IBM Worklight documentation set, training material and online forums where you can post questions, see the IBM website at:
 - <http://www.ibm.com/mobile-docs>
- **Support**
 - Software Subscription and Support (also referred to as Software Maintenance) is included with licenses purchased through Passport Advantage and Passport Advantage Express. For additional information about the International Passport Advantage Agreement and the IBM International Passport Advantage Express Agreement, visit the Passport Advantage website at:
 - <http://www.ibm.com/software/passportadvantage>
 - If you have a Software Subscription and Support in effect, IBM provides you assistance for your routine, short duration installation and usage (how-to) questions, and code-related questions. For additional details, consult your IBM Software Support Handbook at:
 - <http://www.ibm.com/support/handbook>
- **Comments**
 - We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.
 - For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.
 - When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state.
 - Thank you for your support.
 - Submit your comments in the IBM Worklight Developer Edition support community at:
 - <https://www.ibm.com/developerworks/mobile/worklight/connect.html>
 - If you would like a response from IBM, please provide the following information:
 - Name
 - Address
 - Company or Organization
 - Phone No.
 - Email address

Thank You

