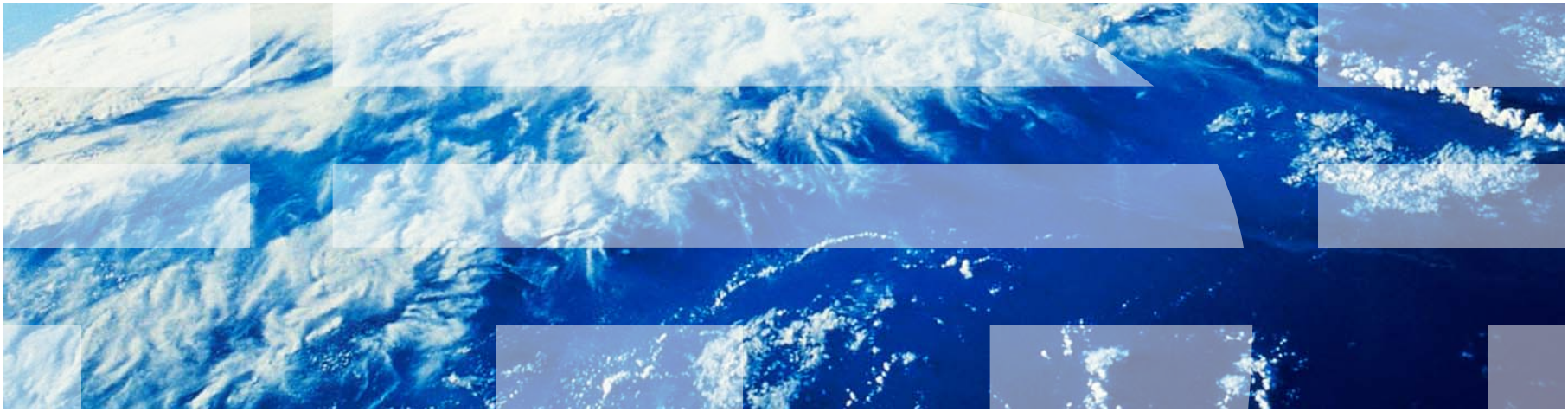
IBM Worklight V6.1.0
Getting Started

Using IBM Worklight API in native Java ME applications



17 March 2013

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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" 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/

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Agenda

- Creating an IBM Worklight native API
- Creating and configuring a Java ME native application
- Initializing WLCLient
- Invoking an IBM Worklight procedure
- Receiving a procedure response
- For BlackBerry

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

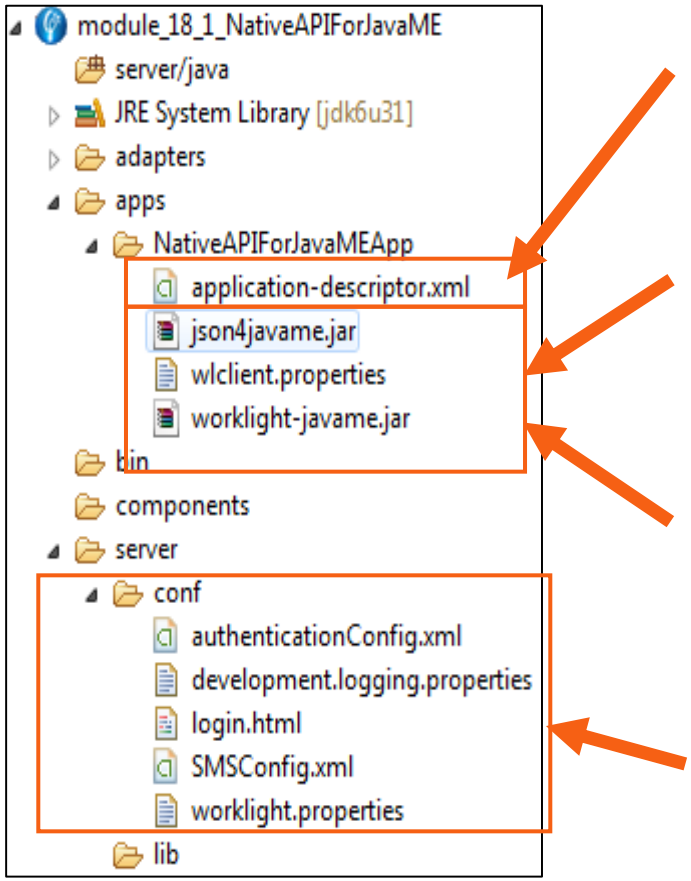
Creating an IBM Worklight native API (1 of 3)

- IBM Worklight® provides the ability for Java™ Platform, Micro Edition (Java ME) applications to communicate with a Worklight Server by using an IBM Worklight native API library.
- To serve a Java ME application, the Worklight Server must be aware of it.
- The IBM Worklight native API is located in the **`apps`** folder of your IBM Worklight project.
- The IBM Worklight native API folder serves two purposes:
  - It contains a native API library and configuration file that you must copy to your Java ME project.
  - It contains the **`application-descriptor.xml`** file, which you can deploy to a Worklight Server to serve as an entry point.
- In this module, you learn how to create an IBM Worklight native API and use its components in your Java ME application.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Creating an IBM Worklight native API (2 of 3)

- An IBM Worklight native API contains several components:



You use the **`application-descriptor.xml`** file to define the application metadata and to configure the security settings that the Worklight Server enforces.

The **`wlclient.properties`** file contains the connectivity settings that a native Java ME application uses. You must copy this file to your native Java ME project.

The **`worklight-javame.jar`** and **`json4javame.jar`** files define the IBM Worklight API library that you must copy to your native Java ME project.

Like any other Worklight project, you define the server configuration by modifying the files in the **`server\conf`** folder.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Creating an IBM Worklight native API (3 of 3)

1. In Worklight Studio, create a Worklight project, and add an IBM Worklight native API.
2. In the **New Worklight Native API** dialog, enter your native API name, and select **JavaME** for the **Environment** field.
3. Right-click the **IBM Worklight native API** folder and select **Run As > Deploy Native API.**

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Agenda

- Creating an IBM Worklight native API
- Creating and configuring a Java ME native application
- Initializing WLCLient
- Invoking an IBM Worklight procedure
- Receiving a procedure response
- For BlackBerry

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Creating and configuring a Java ME native application

1. Create a Java ME native application.
2. Copy the **`worklight-javame.jar`** and **`json4javame.jar`** files from the IBM Worklight native API folder to the Java ME native application, in the **`/lib`** directory.
3. Copy the **`wlclient.properties`** file from the IBM Worklight native API folder to the new Java ME native application under the **`/res`** directory.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Agenda

- Creating an IBM Worklight native application
- Creating and configuring a Java ME native application
- Initializing WLCLient
- Invoking an IBM Worklight procedure
- Receiving a procedure response
- For BlackBerry

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Initializing WLClient

- Create an instance of **`WLClient`**.

```
private WLClient client;


public JavaMENativeApp() {
    client = WLClient.createInstance(this);
```

- To establish the connection to a Worklight Server, use the **`connect`** method, and specify a **`MyConnectListener`** class instance as the parameter.

```
public void commandAction(Command command, Item item) {
    StringItem itemName = (StringItem)item;
    if(itemName.getText().equals("1.Connect")) {
        updateTextView("\nConnecting...");
        client.connect(new MyConnectListener());
```

  - See next slides to learn how to create it.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

MyConnectListener

- The **`WLClient`** instance first connects to the Worklight Server, following the properties of the **`wlclient.properties`** file.
- After the connection is done, it calls one of the methods of the **`MyConnectListener`** class.
  - The **`MyConnectListener`** class implements the **`WLResponseListener`** interface.

```
public class MyConnectListener implements WLResponseListener  {
```

- The **`WLResponseListener`** interface specifies the following methods:
  - public void **`onSuccess`** `(WLResponse response) { }`
  - public void **`onFailure`** `(WLFailResponse response) { }`
- Use these methods to process connection success or connection failure.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Agenda

- Creating an IBM Worklight native application
- Creating and configuring a Java ME native application
- Initializing WLCLient
- Invoking an IBM Worklight procedure
- Receiving a procedure response
- For BlackBerry

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Invoking an IBM Worklight Procedure (1 of 3)

- After the connection is established with a Worklight Server, you can use the **`WLClient`** instance to call the adapter procedures:

```
}else if(itemName.getText().equals("2.Invoke Procedure")) {
    updateTextView("\nInvoking procedure...");
    String adapterName = "RSSReader";
    String procedureName = "getStoriesFiltered";

    WLProcedureInvocationData invocationData = new WLProcedureInvocationData(adapterName, procedureName);

    Object[] parameters = new Object[] {};
    invocationData.setParameters(parameters);

    WLRequestOptions options = new WLRequestOptions();
    client.invokeProcedure(invocationData, new MyInvokeListener(), options);

  }
```

- Create a **`WLProcedureInvocationData`** object with the adapter and procedure names.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Invoking an IBM Worklight Procedure (2 of 3)

- After the connection is established with a Worklight Server, you can use the **`WLClient`** instance to call the adapter procedures:

```
}else if(itemName.getText().equals("2.Invoke Procedure")) {
    updateTextView("\nInvoking procedure...");
    String adapterName = "RSSReader";
    String procedureName = "getStoriesFiltered";

    WLProcedureInvocationData invocationData = new WLProcedureInvocationData(adapterName, procedureName);

    Object[] parameters = new Object[] {};
    invocationData.setParameters(parameters);

    WLRequestOptions options = new WLRequestOptions();
    client.invokeProcedure(invocationData, new MyInvokeListener(), options);

 }
```

- Add the required parameters as an object array, and set the request options.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Invoking an IBM Worklight Procedure (3 of 3)

- After the connection is established with a Worklight Server, you can use the **`WLClient`** instance to call the adapter procedures:

```
}else if(itemName.getText().equals("2.Invoke Procedure")) {
    updateTextView("\nInvoking procedure...");
    String adapterName = "RSSReader";
    String procedureName = "getStoriesFiltered";

    WLProcedureInvocationData invocationData = new WLProcedureInvocationData(adapterName, procedureName);

    Object[] parameters = new Object[] {};
    invocationData.setParameters(parameters);

    WLRequestOptions options = new WLRequestOptions();
    client.invokeProcedure(invocationData, new MyInvokeListener(), options);

  }
```

- Specify a **`MyInvokeListener`** class instance as a parameter.
  - See next slides to learn how to define such an instance.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Agenda

- Creating an IBM Worklight native application
- Create and configure a Java ME native application
- Initializing WLCLient
- Invoking an IBM Worklight procedure
- Receiving a procedure response
- For BlackBerry

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Receiving a procedure response (1 of 3)

- After the procedure call completes, the **`WLClient`** instance calls one of the methods of the **`MyInvokeListener`** class.
- The **`MyInvokeListener`** class implements the **`WLResponseListener`** interface.

```
import com.worklight.wlclient.api.WLFailResponse;
import com.worklight.wlclient.api.WLResponse;
import com.worklight.wlclient.api.WLResponseListener;

public class MyInvokeListener implements WLResponseListener {
```

- The **`WLClient`** calls its **`onSuccess`** or **`onFailure`** methods (see the next slide).

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Receiving a procedure response (2 of 3)

- If the procedure call is successful, the **`onSuccess`** method of the **`MyInvokeListener`** instance is called.
- Use it to get the data that is retrieved from the adapter.

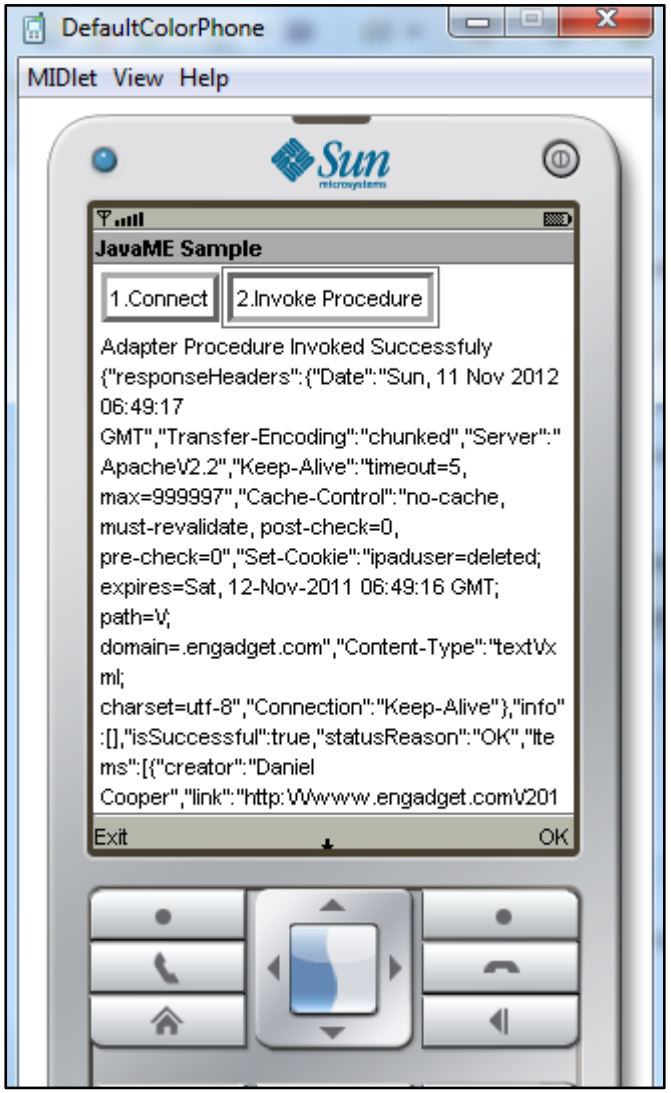
```
public class MyConnectListener implements WLResponseListener  {
    public void onSuccess(WLResponse response) {
        String responseText = response.getResponseText();
        JavaMENativeApp.updateTextView("Connected Successfuly\n" + responseText);
    }

    public void onFailure(WLFailResponse response) {
        String responseText = response.getResponseText();
        JavaMENativeApp.updateTextView("Connection Failure\n" + responseText);
    }

}
```

- The **`response`** object contains the response data.
  - You can use its methods and properties to retrieve the required information.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Receiving a procedure response (3 of 3)

- You can find the sample for this training module in the Getting Started page of the IBM Worklight documentation website at http://www.ibm.com/mobile-docs
- The sample contains two projects:
  - The **`NativeAPIForJavaME.zip`** file contains an IBM Worklight native API that you can deploy to your Worklight Server.
  - The **`JavaMENativeApp.zip`** file contains a native Java ME application that uses the IBM Worklight native API library to communicate with the Worklight Server.
- **Important**: Make sure to update the **`wlclient.properties`** file in **`JavaMENativeApp`** with the relevant server settings.



© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Agenda

- Creating an IBM Worklight native application
- Creating and configuring a Java ME native application
- Initializing WLCLient
- Invoking an IBM Worklight procedure
- Receiving a procedure response
- For BlackBerry

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

For BlackBerry

- Use the **`createInstance(String connectionString, MIDlet midlet)`** method for BlackBerry to create the **`WLClient`** instance.
- BlackBerry has different ways to make network (HTTP or Socket) connection.
  - To identify the type of network connection that you use to connect to the Worklight Server, pass the appropriate string argument to the **`createInstance(String connectionString, MIDlet midlet)`** method.
    - For example: `deviceside=true`
  - For more information, see the BlackBerry Developers Knowledge Base at http://supportforums.blackberry.com/t5/Java-Development/Different-ways-to-make-an-HTTP-or-socket-connection/ta-p/445879.

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

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.

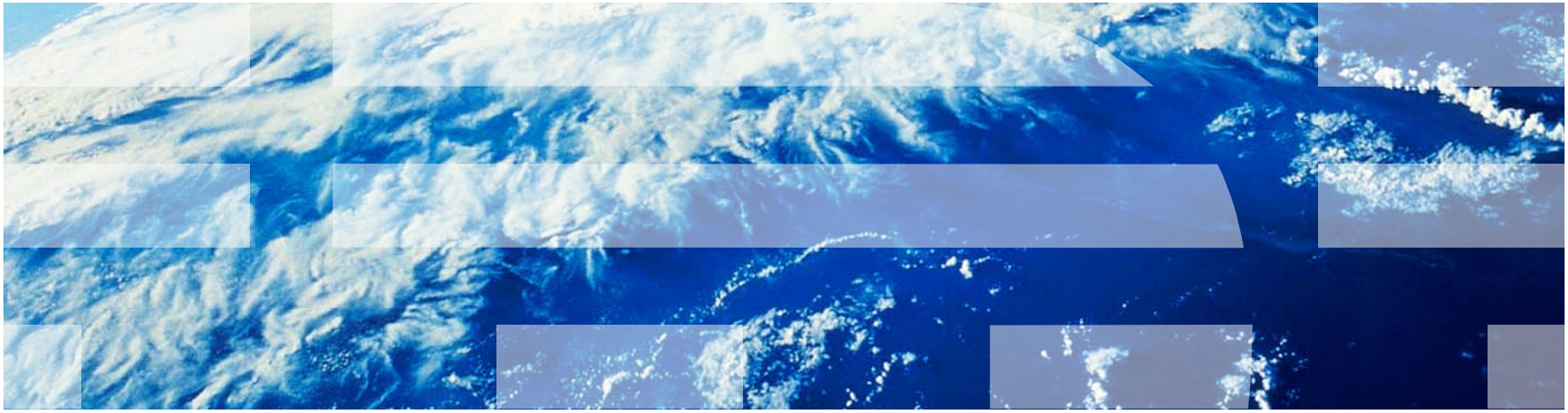
© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

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

© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.

Thank You



© Copyright International Business Machines Corporation 2012, 2013. All rights reserved.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.