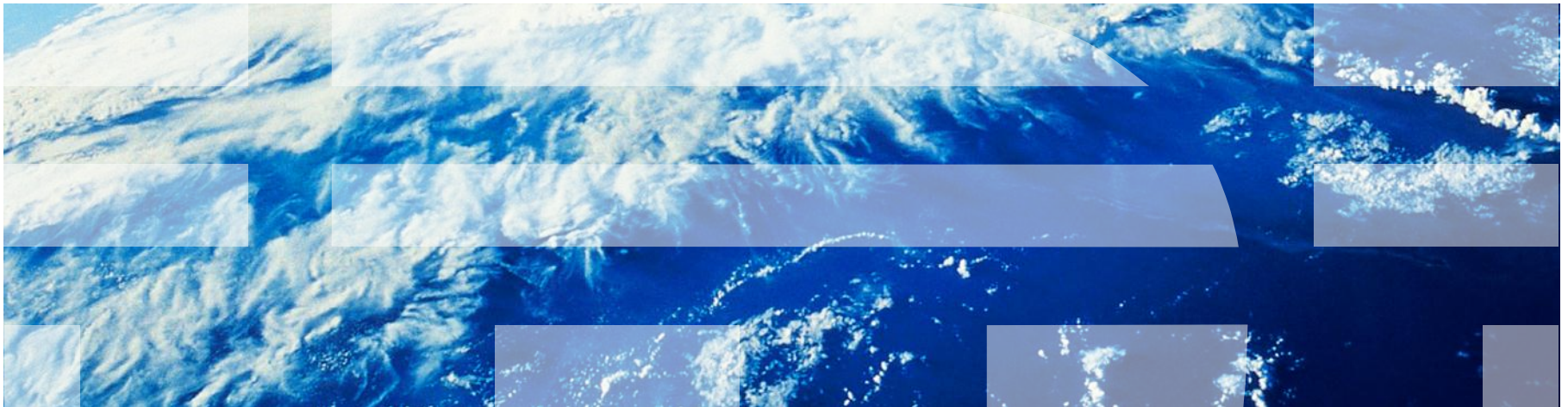IBM

IBM Worklight Foundation V6.2.0
Getting Started

Using the Command Line Interface (CLI) to create, build and manage Worklight Project artifacts

Trademarks

- IBM, the IBM logo, ibm.com, and Worklight are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

- Node.js is an official trademark of Joyent. This documentation is not formally related to or endorsed by the official Joyent Node.js open source or commercial project

- 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.
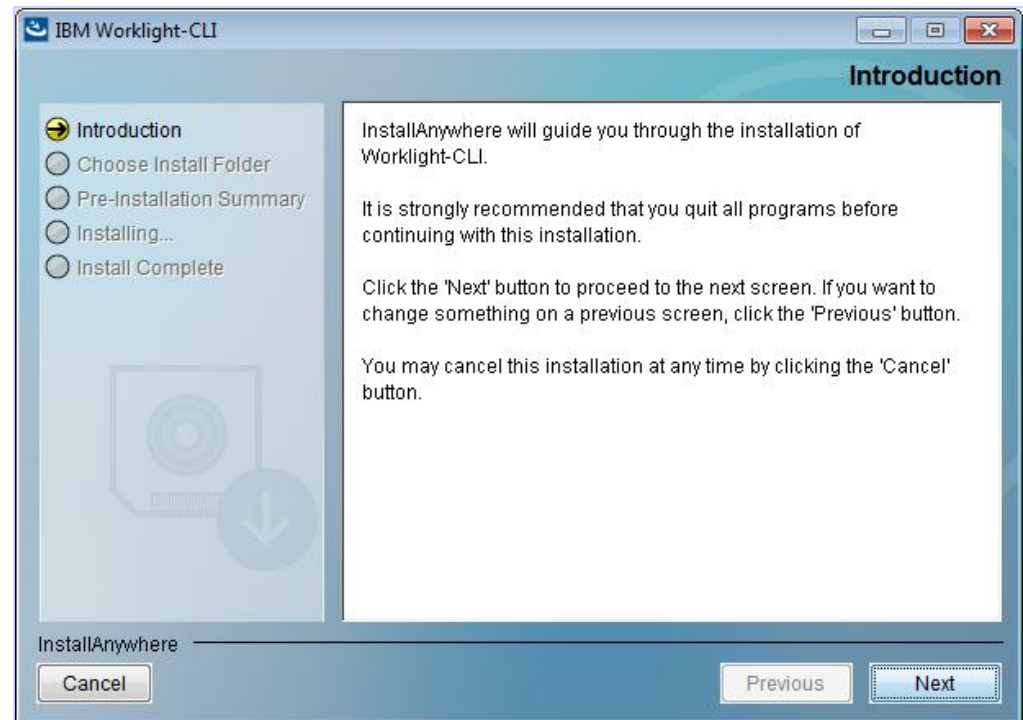
About IBM®

See http://www.ibm.com/ibm/us/en/

Agenda

- CLI installation

- Creating Worklight projects and native API

- Creating hybrid applications

- Creating test servers and the build-and-deploy flow

- Creating and testing adapters

- Using the Export command

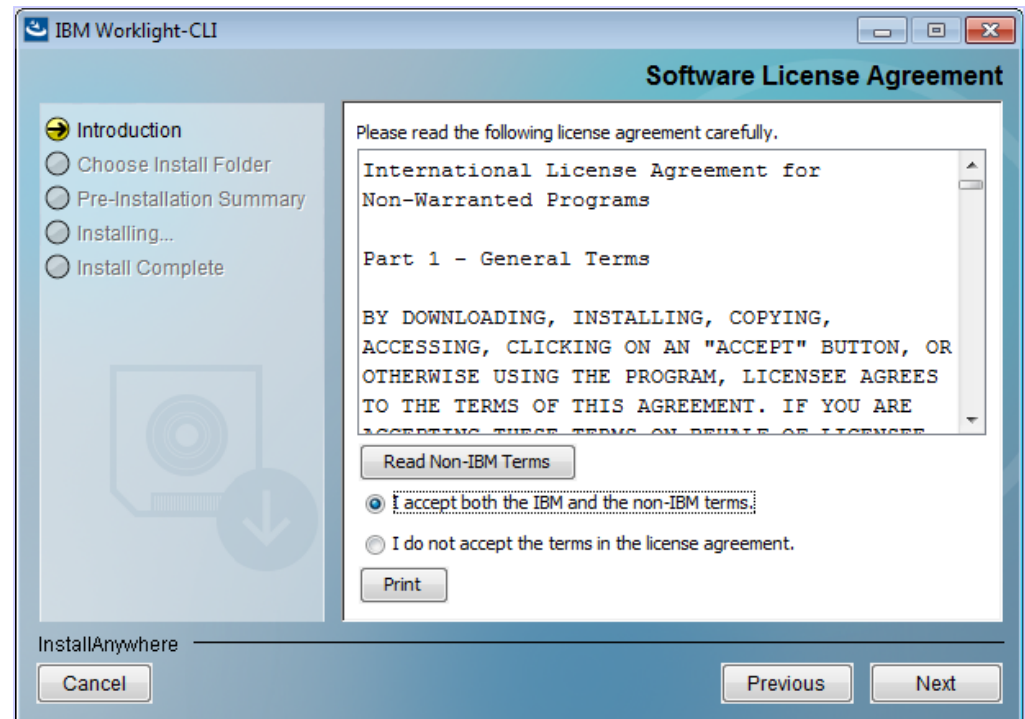- Importing CLI-generated projects into Worklight Studio

CLI Installation (1 of 4)

1. Extract the worklight-cli-installer-6.2.zip archive and navigate to the worklight-cli-installer-6.2 folder.
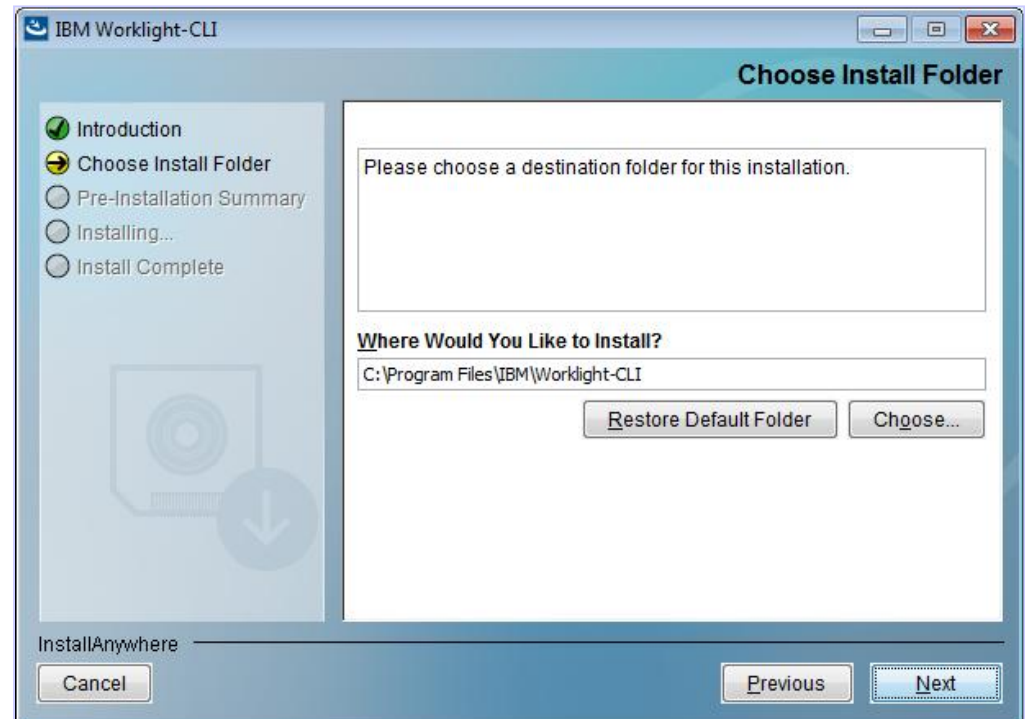
2. Click the **Install** icon to begin the installation.

CLI Installation (2 of 4)
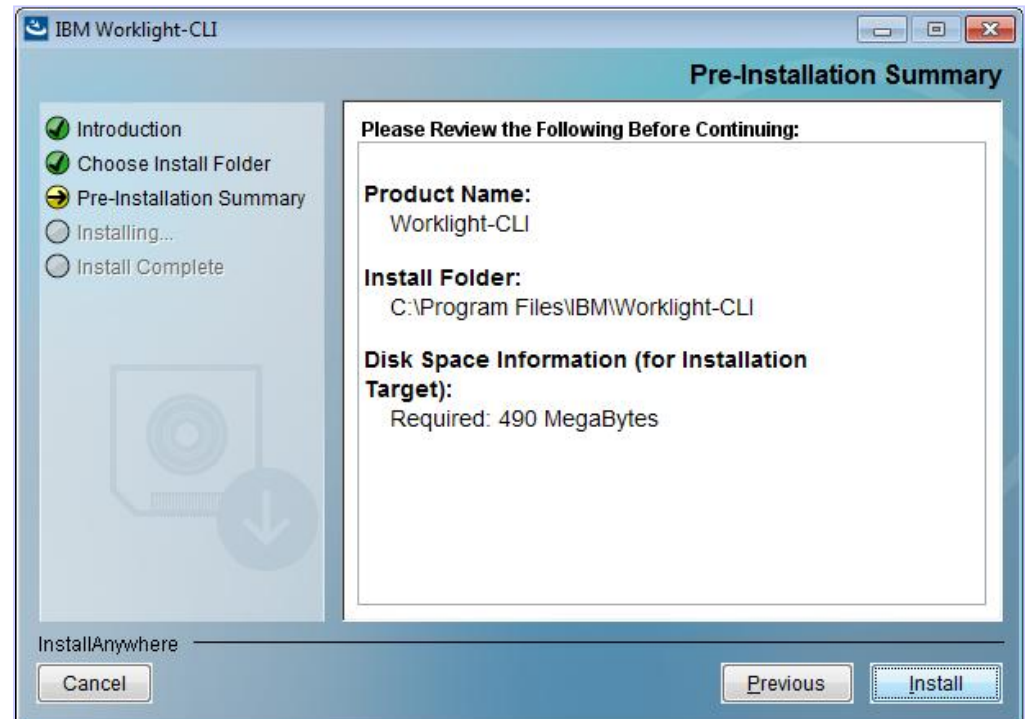
3. Accept the Software License Agreement.

IBM

CLI Installation (3 of 4)

4.   Select the installation folder.

5. Accept the Pre-Installation Summary by clicking **Install** and then select **Done** when installation is complete.

Agenda

- CLI installation

- Creating Worklight projects and native API

- Creating hybrid applications

- Creating test servers and the build-an-deploy flow

- Creating and testing adapters

- Using the Export command

- Importing CLI-generated projects into Worklight Studio

Creating Worklight projects and native APIs (1 of 4)

- The CLI installer adds the installation folder to your `path`, so that Worklight® CLI commands can be run from any directory.

1. Create a Worklight project from the CLI by using either of these commands:

   - `worklight create projectName`, or

   - `wl create projectName`

   - Example:

```
/Users/CLIuser/Desktop/CLIexample$ wl create projectName
A Worklight Project was successfully created at
Users/CLIuser/Desktop/CLIexample/projectName
```

Creating Worklight projects and native APIs (2 of 4)
Helpful Commands

- `wl help` – Shows all command usage:

```
/Users/CLIuser/Desktop/CLIexample/projectName$ wl help

NAME
    worklight -- IBM Worklight Command Line Interface (CLI).


SYNOPSIS
    wl <command> [options]

DESCRIPTION
    Command-line interface to create and manage IBM Worklight's apps.

Global Commands

    create [<name>]
    This command creates a new Worklight project in the current working directory.

    create-server
    This command creates a new Liberty server at the path that is specified by you
    The server is pre-configured to work as a Worklight local test server.


    console
    This command opens the Worklight console in your default browser for the current

    working directory of your project.
```

Creating Worklight projects and native APIs (3 of 4)
Helpful Commands

- `wl info` – Gives OS release, system memory, node.js version, Worklight CLI version

```
/Users/CLIuser/Desktop/CLIexample/projectName$ wl info
OS: darwin ia32
Release: 12.5.0
System Memory: 2946MB free out of 8192MB
Node: v0.10.26
Worklight CLI: 6.2.0.00
```
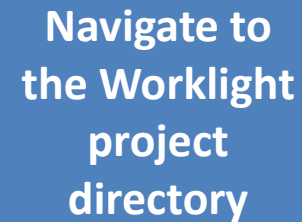
Node.js version

Creating Worklight projects and native APIs (4 of 4)
Helpful Commands

- You can use the `interactive` or `direct` options to add native API.

  - Interactive
    - `wl add api`

  - Direct
    - `wl add api --environment ios apiName`, **or**
    - `-e` for environment

Navigate to the Worklight project directory

```
/Users/CLIuser/Desktop/CLIexample/projectName$ wl add api --environment ios
apiName
A new iOS API was added at
/Users/CLIuser/Desktop/CLIexample/projectname/apps/apiName
```

Agenda

- CLI installation

- Creating Worklight projects and native API

- Creating hybrid applications

- Creating test servers and the build-and-deploy flow

- Creating and testing adapters

- Using the Export command

- Importing CLI-generated projects into Worklight Studio

- Add a hybrid application from the CLI by using these commands:
    - `wl add hybrid [<name>]` – Creates a new hybrid application, which is generated into the apps folder of the current project.
    - `wl add environment` – Adds an environment to your application. The current working directory must be the hybrid app directory.

```
/Users/CLIuser/Desktop/CLIexample/myProject/myHybrid $wl add environment
[?] What environments you want to add to the hybrid app? (Press <space> to select)
>○ iPhone
 ○ iPad
 ○ Android phone and tablets
 ○ BlackBerry 6 and 7
 ○ BlackBerry 10
 ○ Windows Phone 8
 ○ Windows 8 desktop and tablets
(Move up and down to reveal more choices)
```

Application skins

- To add an application skin from the CLI, use these commands:

  - `wl add skin` – Interactive mode prompts for platform to target and name. The current working directory must be under an existing hybrid application, and at least one environment must already be added to this app.

  - `wl add skin [-e] -environment [android|blackberry|blackverry10|iphone|ipad] [name]` – direct mode

  - Ex. `$ wl add skin -e android tablet`

```
/Users/CLIuser/Desktop/CLIexample/projectName/apps/testHybrid$ wl add skin
[?] What platform do you want to target? Android phone and tablets
[?] What do you want to be your skin name suffix? Your skin folder name is
'platform.<suffix>' tablets
A new android Skin was added at
/Users/CLIuser/Desktop/CLIexample/projectName/apps/testHybrid
```

Creating hybrid applications (3 of 5)
Optional features

- To add optional features to a hybrid application from the CLI, use these commands:

  - `wl add feature` – Interactive mode prompts for the features to add.

```
/Users/CLIuser/Desktop/CLIexample/projectName/apps/testHybrid$ wl add feature
[?] What feature you want to install in this application? NOTE: Features that are
already installed are not shown:
(Use arrow keys)
> FIPS 140-2
    IBM Tealeaf® SDK
    JSONStore
```

  - `wl add feature [fips|jsonstore|tealeaf]` – direct mode

Creating hybrid applications (4 of 5)
Optional features

- To remove optional features from a hybrid application by using the CLI, use these commands:

  - `wl remove feature` – Interactive mode prompts for the features to remove.

```
/Users/CLIuser/Desktop/CLIexample/projectName/apps/testHybrid$ wl remove feature
[?] What feature you want to uninstall from this application? NOTE: Features that are
already uninstalled are not shown: (Use arrow keys)
> JSONStore
```

  - `wl remove feature [fips|jsonstore|tealeaf]` – Direct mode

Creating hybrid applications (5 of 5)
Editing files

- BYOE - Bring Your Own Editor

- Use your favorite text editor to develop from CLI

- Example - `vim myAdapterName`

Agenda

- CLI installation

- Creating Worklight projects and native API

- Creating hybrid applications

- Creating test servers and the build-and-deploy flow

- Creating and testing adapters

- Using the Export command

- Importing CLI-generated projects into Worklight Studio

Creating test servers and the the build-and-deploy flow (1 of 2)

- The instance of the Liberty development server is created on the default user directory. For example:

```
/Users/CLIuser/.worklight/6.2.0/server
```

- `wl start` – Starts the server. If the project is not deployed, this command runs the build all and deploy flow.

- `wl stop` – Stops the server.

- `wl run` – Verbose server mode that outputs server events to console or log.

- `wl status` – Gives status of server.

Creating test servers and the the build-and-deploy flow (2 of 2)

- `wl build` – Builds the project. When you are not in the root directory of the project, this command builds the current directory and subdirectories.

- `wl deploy` – Deploys anything that you built in the root directory of the project. When you are in the adapter folder, this command deploys that adapter.

- `wl build && wl deploy` – This command builds and deploys.

- `wl console` - Opens the Worklight console.

Agenda

- CLI installation

- Creating Worklight projects and native API

- Creating hybrid applications

- Creating test servers and the build-an-deploy flow

- Creating and testing adapters

- Using the Export command

- Importing CLI-generated projects into Worklight Studio

Creating and testing adapters (1 of 3)

- Interactive

  - `wl add adapter`

```
/Users/CLIuser/Desktop/CLIexample/projectName/apps
/testHybrid$ wl add adapter
[?] What do you want to name your Worklight
Adapter? myHttpAdapter
[?] What type of adapter would you like? HTTP
[?] Create procedures for offline JSONStore? No
[?] Create procedures for USSD enablement? No
A new http Adapter was added at
/Users/CLIuser/Desktop/CLIexample/projectName/adap
ters/myHttpAdapter
```

Select the adapter type by using the up and down arrow keys. Next, use the prompt to create JSONStore procedures.

- Direct

  - `wl add adapter -type http myAdapterName`, **or**

  - `-t` **for type**

```
/Users/CLIuser/Desktop/CLIexample/projectName/apps/testHybrid$ wl add adapter --type
http testAdapter2
A new http Adapter was added at
/Users/CLIuser/Desktop/CLIexample/projectName/adapters/testAdapter2
```

23

Creating and testing adapters (2 of 3)

- Interactive

    - `wl invoke`

```
/Users/CLIuser/Desktop/CLIexample/projectName/adapters/myHttpAdapter$ wl invoke
[?] Which adapter do you want to use? myHttpAdapter
[?] Which procedure do you want to invoke? (Use arrow keys)
 > getStories
    getStoriesFiltered
```

Select the procedure by using the up and down arrow keys.

Creating and testing adapters (3 of 3)

- Direct
  - `wl invoke adapterName:function [parameter1[,parameter2…]]`

```
/Users/CLIuser/Desktop/CLIexample/projectName/adapters/myHttpAdatpter$
wl invoke myHttpAdapter:getStories \"world\"
{
"statusCode": 200,
"errors": [],
"isSuccessful": true,
"statusReason": "OK",
"rss": {
"feedburner": "http://rssnamespace.org/feedburner/ext/1.0",
"channel": {
"pubDate": "Wed, 07 May 2014 09:58:39 EDT",
"title": "CNN.com - World",
```

Agenda

- CLI installation

- Creating Worklight projects and native API

- Creating hybrid applications

- Creating test servers and the build-and-deploy flow

- Creating and testing adapters

- Using the Export command

- Importing CLI-generated projects into Worklight Studio

Using the Export command (1 of 4)
Interactive Mode Project Export

- By using the `Export` command, you can create a compressed file, which contains the entire Worklight project, or the optimized hybrid assets to use in a native application.

- `wl export` – Running this command in the project root folder prompts you to enter the path and name of the compressed file to export to.

- The resulting compressed file is intended to be shared with other Worklight developers. The compressed file contains source artifacts and everything another developer would need to build the missing artifacts.

```
/Users/CLIuser/Desktop/CLIexample/projectName$ wl export
[?] Where do you want to export the project?
/Users/CLIuser/Desktop
[?] What do you want to name your zip project? projectName
Project is successfully exported to
/Users/CLIuser/Desktop/projectName.zip
```

Using the Export command (2 of 4)
Direct Mode Project Export

- `wl export /Users/user/Desktop/myProject.zip` – If run from the root folder of the project, direct mode takes one additional argument, which is the full path to the compressed file to create.

```
/Users/CLIuser/Desktop/CLIexample/projectName$ wl export
/Users/CLIuser/Desktop/projectName.zip
Project is successfully exported to /Users/CLIuser/Desktop/projectName.zip
```

Using the Export command (3 of 4)
Interactive Mode Hybrid Export

- First make sure that the app is built by using the `wl build` command.

- Run the hybrid asset export within the environment folder of this existing hybrid app.

- `wl export` – Interactive mode asks whether to include the native libraries, the path, and file name of the compressed file to export to.

- The native libraries are built files, which native application IDEs need to make sure that the hybrid assets can be used in a native app.

```
/Users/CLIuser/Desktop/CLIexample/projectName/apps/myHybridApp/iphone$ wl
export
[?] Would you like to include the native libraries? Yes
[?] Where do you want to export the project? /Users/CLIuser/Desktop
[?]  What do you want to name your zip project? projectName.zip
Project is successfully exported to /Users/CLIuser/Desktop/projectName.zip
```

Using the Export command (4 of 4)
Direct Mode Hybrid Export

- `wl export [<path to zip file>] [-i|--includeNativeLibs]]`
  – In direct mode, provide the full path to export hybrid assets.

- If the arguments `-i` or `-includeNativeLibraries` are supplied, the native libraries are included.

```
/Users/CLIuser/Desktop/CLIexample/projectName/apps/myHybridApp/iphone$ wl
export /Users/CLIuser/Desktop/projectName.zip --includeNativelibs
Project is successfully exported to /Users/CLIuser/Desktop/projectName.zip
```

Agenda

- CLI installation

- Creating Worklight projects and native API

- Creating hybrid applications

- Creating test servers and the build-and-deploy flow

- Creating and testing adapters

- Using the Export command

- Importing CLI-generated projects into Worklight Studio

Importing CLI-generated projects into Worklight Studio (1 of 2)

- From Eclipse, import the existing project and point to the project directory that you created with CLI.

Select "Existing Projects into Workspace".

Import

Select

Create new projects from an archive file or directory.

Select an import source:

type filter text

▲ 📂 General
    📇 Archive File
    📂 Existing Projects into Workspace
    📁 File System
    📋 Preferences
▷ 📂 CVS
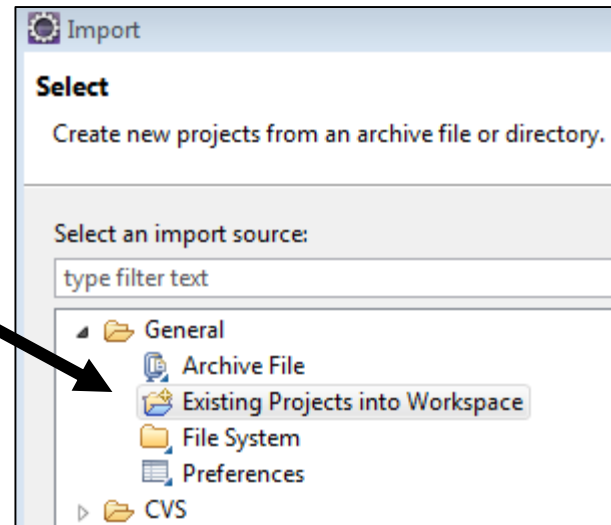
Importing CLI-generated projects into Worklight Studio (1 of 2)
Open Eclipse Worklight Studio Project With CLI

- CLI can also open an existing Eclipse Studio project: Navigate via the command line to a workspace that was created by Eclipse.

Navigate to Eclipse Workspace Project Directory

```
/Users/CLIuser/Desktop/EclipseWorkspace/projectName$ ls
adapters        bin            server
apps        components  services
```

For more information

- For more information about the Command Line Interface, see Command Line Interface in the product user documentation.

Notices

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