

# Lab: Getting Deep Device Integration with Third Party Extensions

JAM 335

Adam Stanley @n\_adam\_stanley

Jorge del Casar Sanchez @jorgecasar

February 6, 2013



## Adam

TL, Developer Relations



## Jorge

Evangelist, Developer Relations




## LED Extension Demo

### Setings

Blink Duration

3

Color 

Magenta

Start

# WebWorks Overview

HTML5, CSS3, JavaScript



WebKit engine



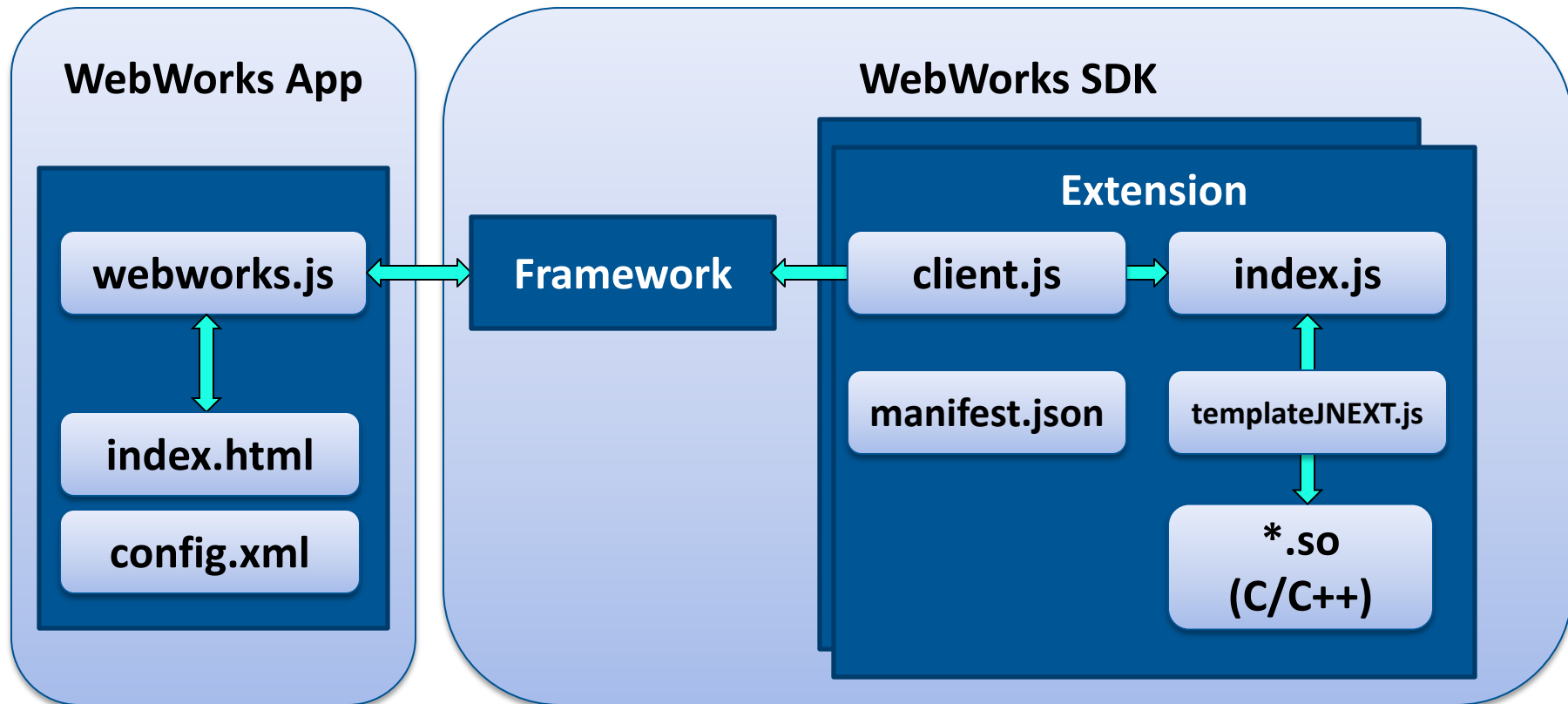
BlackBerry 10 WebWorks



BlackBerry Developer APIs



# Extension Architecture



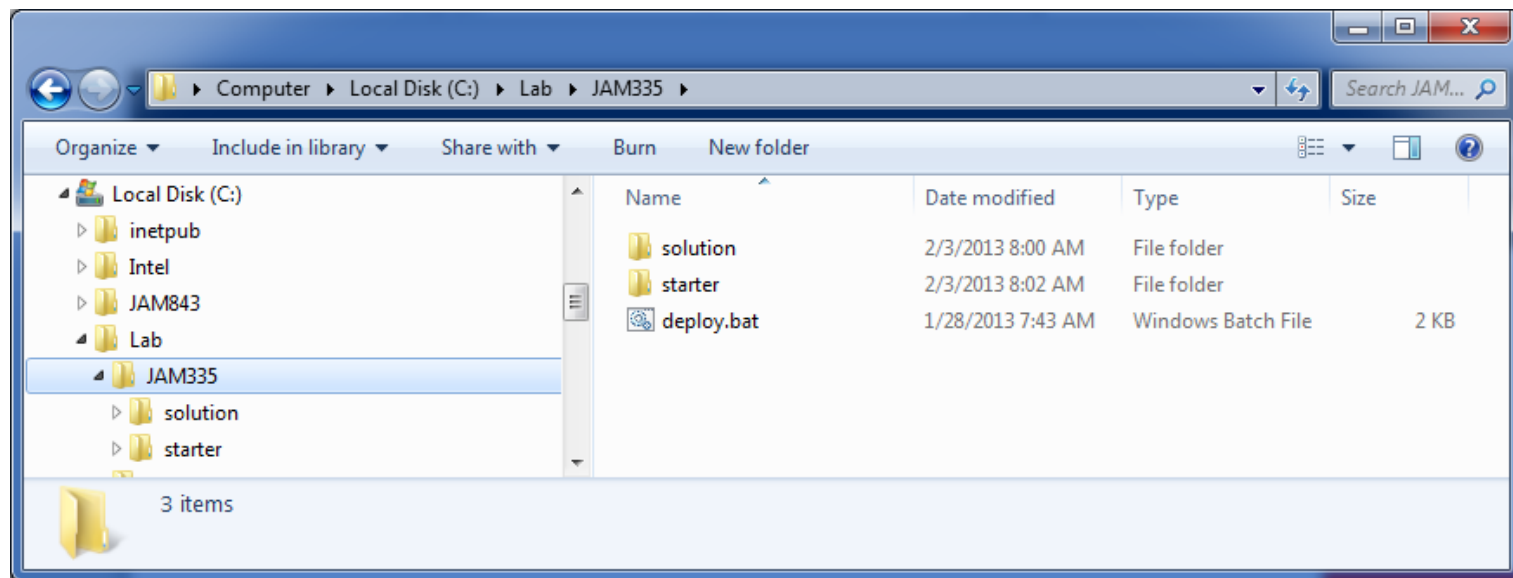
# Lab Requirements

- Package and sign an application
  - ▶ BlackBerry 10 WebWorks SDK
  - ▶ Ripple Mobile emulator
  - ▶ <https://developer.blackberry.com/html5/download>
- Test a native extension
  - ▶ BlackBerry 10 Dev Alpha

- Writing the JavaScript component of the extension
  - ▶ Any editor (e.g. Notepad++)
- Writing the C/C++ component of the extension
  - ▶ BlackBerry 10 Native SDK



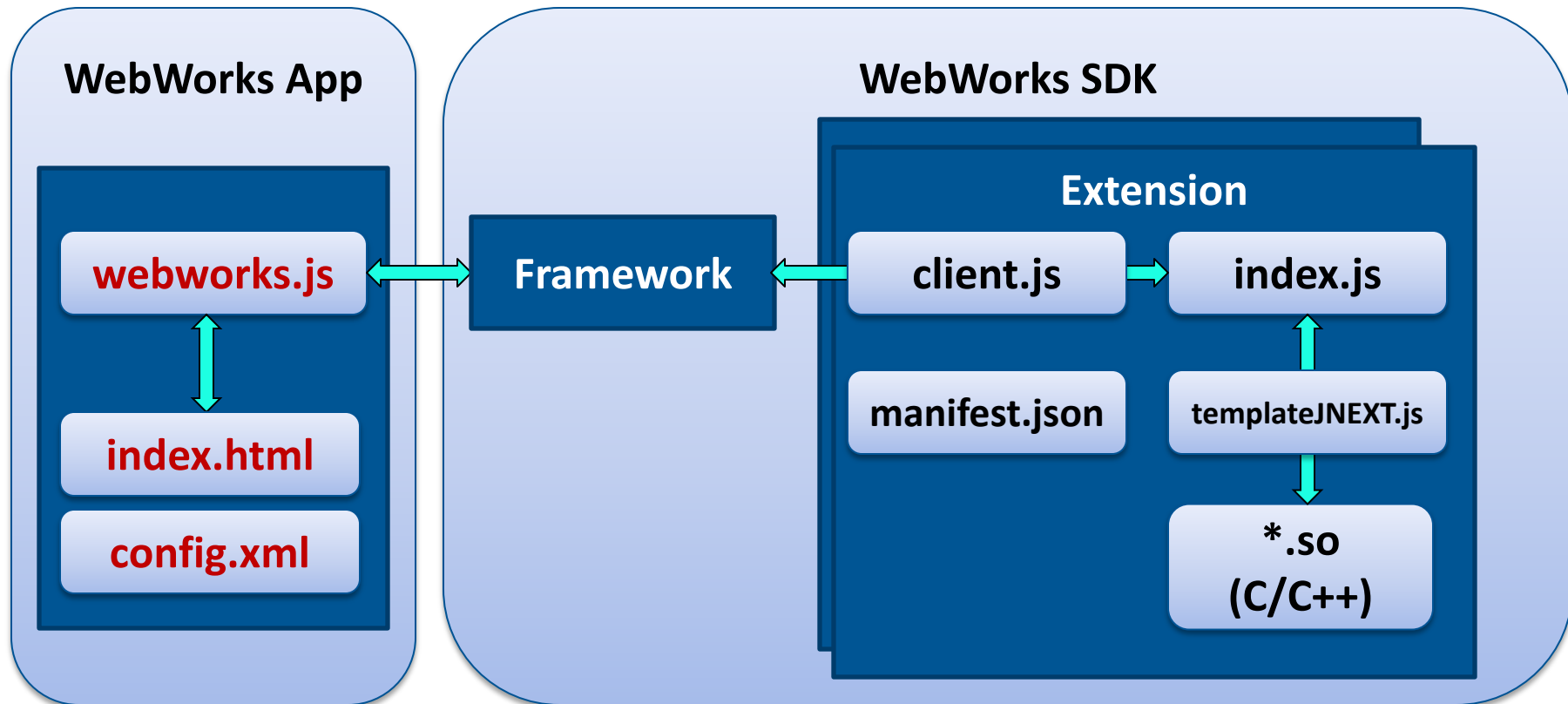
- Sample code
  - ▶ C:\Lab\JAM335



# sample\_application

Today's subject: A WebWorks sample app

# Build sample\_application Goals



- C:\Lab\JAM335\starter\sample\_application
  - ▶ config.xml
  - ▶ index.html
- Open using Notepad++

- **Task:** Modify **config.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets"
        xmlns:rim="http://www.blackberry.com/ns/widgets"
        version="1.0.1"
        id="community.extension.test" xml:lang="en">

    <name>App Name Here</name>
    <description>Test App for the Extension Template</description>
    <author rim:copyright="2013">Your Name Here</author>
    <content src="index.html"/>
    <rim:permissions>
        <rim:permit>access_led_control</rim:permit>
    </rim:permissions>
</widget>
```

- Build and sign app using Ripple mobile emulator
  - ▶ Tip: open any website, then enable Ripple emulator
- Build settings:
  - ▶ Project root: C:\Lab\JAM335\starter\sample\_application
  - ▶ Archive name: sample\_application
  - ▶ Output folder: C:\Temp\sample\_application
  - ▶ Enable remote web inspector

# Build sample\_application

**Settings** ✕

**Package**

Build

SDK Path

C:\Program Files\Research In Motion\BlackBerry 10

Project Root

C:\Lab\JAM335\starter\sample\_application

Archive Name

sample\_application

Output Folder

C:\Temp\sample\_application

Enable Remote Web Inspector

☒

Sign

Signing Password

.....

Bundle Number

1

- Prepare BlackBerry 10 Dev Alpha device
  - ▶ Connect USB (169.254.0.1)
  - ▶ Enable development mode
- Deploy signed app to BlackBerry 10 Dev Alpha
  - ▶ `cd C:\Lab\JAM335\`
  - ▶ `deploy.bat 169.254.0.1 <device password>`



- Test the application
  - ▶ Change Blink Duration and Color selections
  - ▶ Try and start the LED
  - ▶ Where does the message ***community.templateExt is undefined*** originate from in the app source?
  - ▶ Tip: Web Inspector URL <http://169.254.0.1:1337>

# Adding missing features

Let's build an API

# Lets build an API

```
community.templateExt.blinkCount = 5;  
  
var rc = community.templateExt.startLED("FF00FF");
```

JavaScript

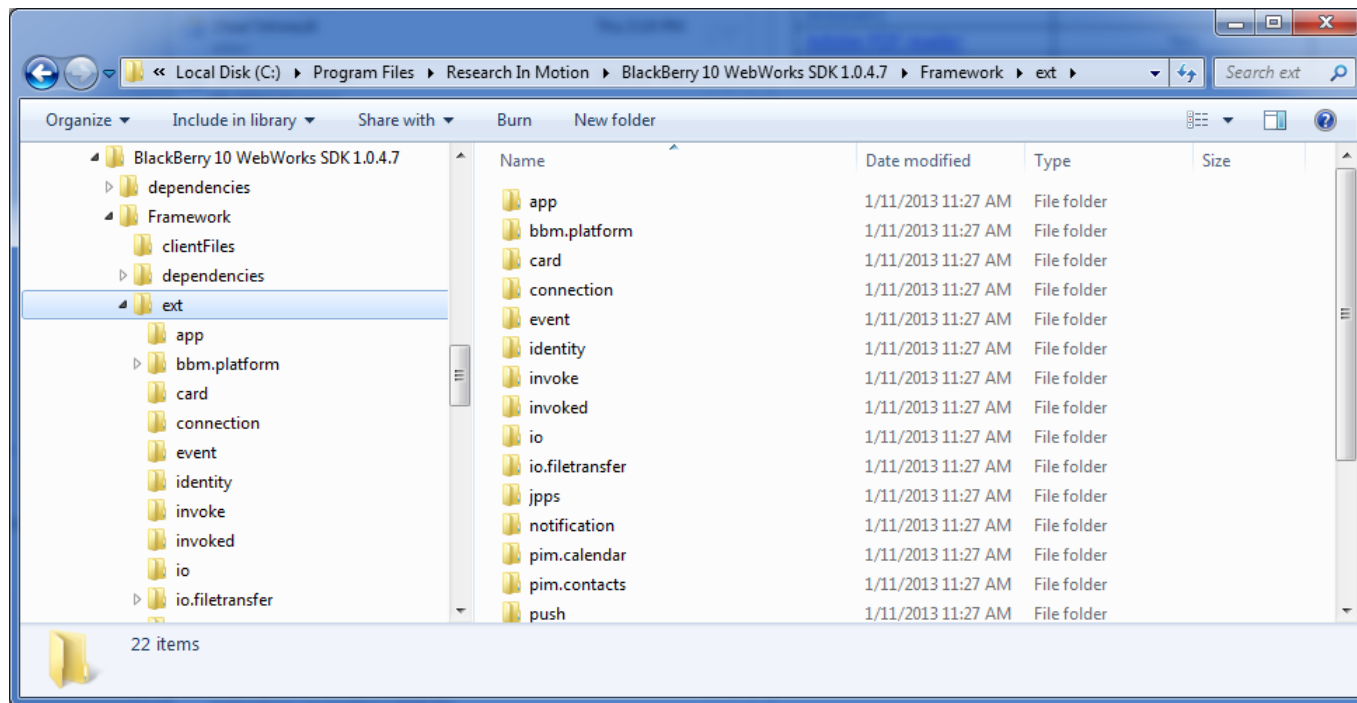
```
std::string TemplateNDK::startLED(int color) {  
    string id = convertLongToString(clock());  
    int status = led_request_rgb(id.c_str(), color, blinkCount);  
}
```

Native

# Getting Started

## Step 1 of 3

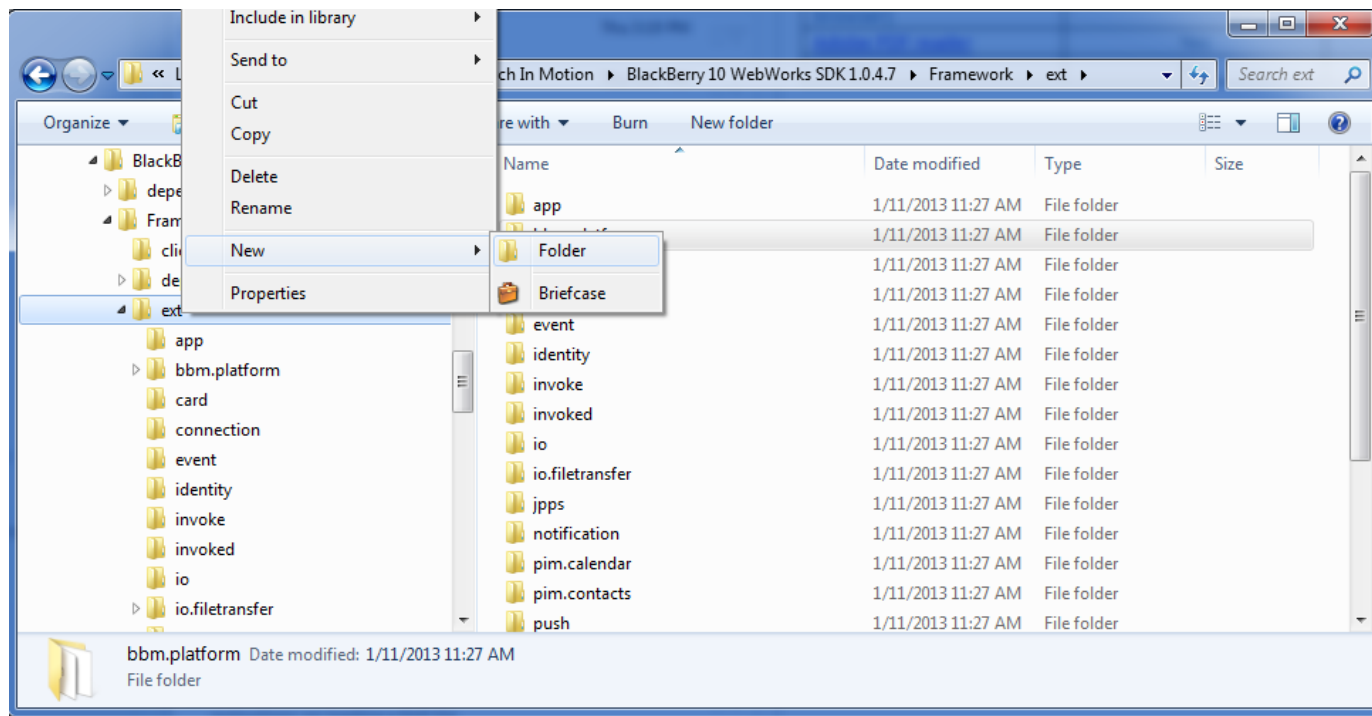
- Find the <WebWorks SDK>/Frameworks/ext folder



# Getting Started

## Step 2 of 3

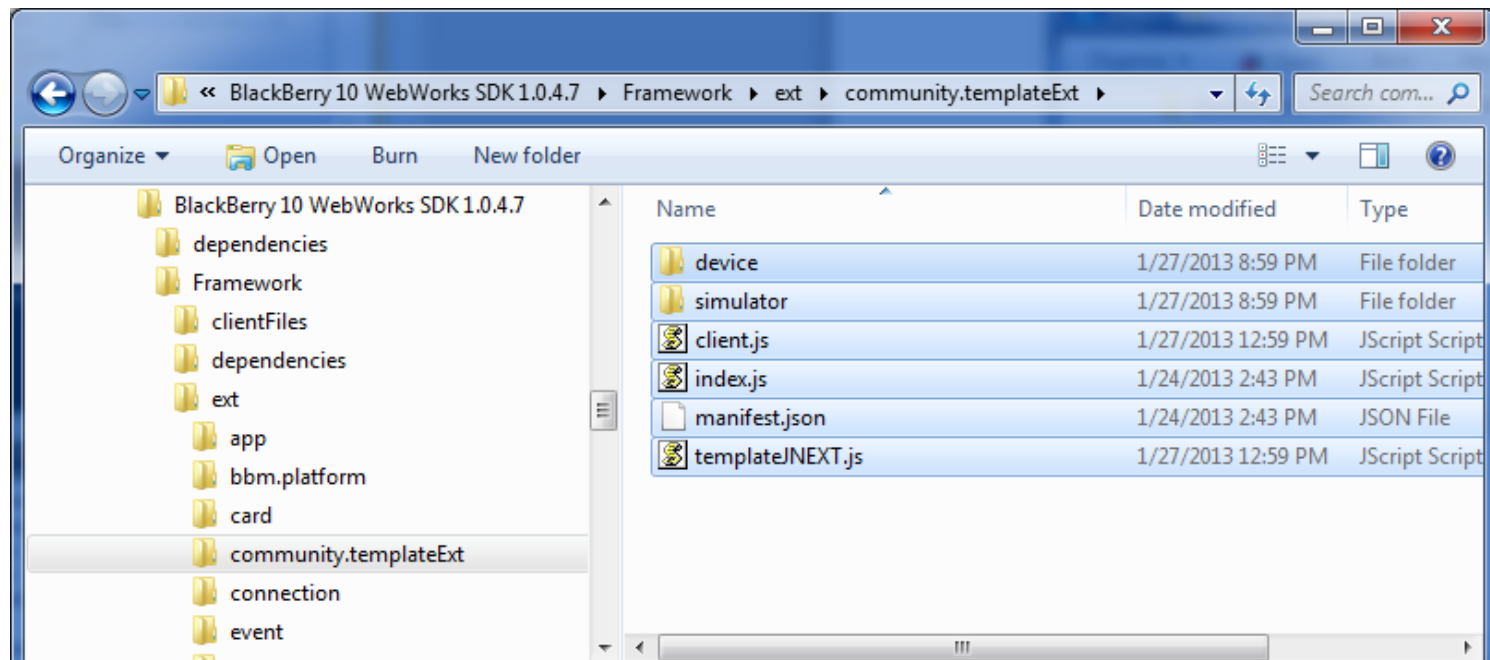
- Create a new folder named **community.templateExt**



# Getting Started

## Step 3 of 3

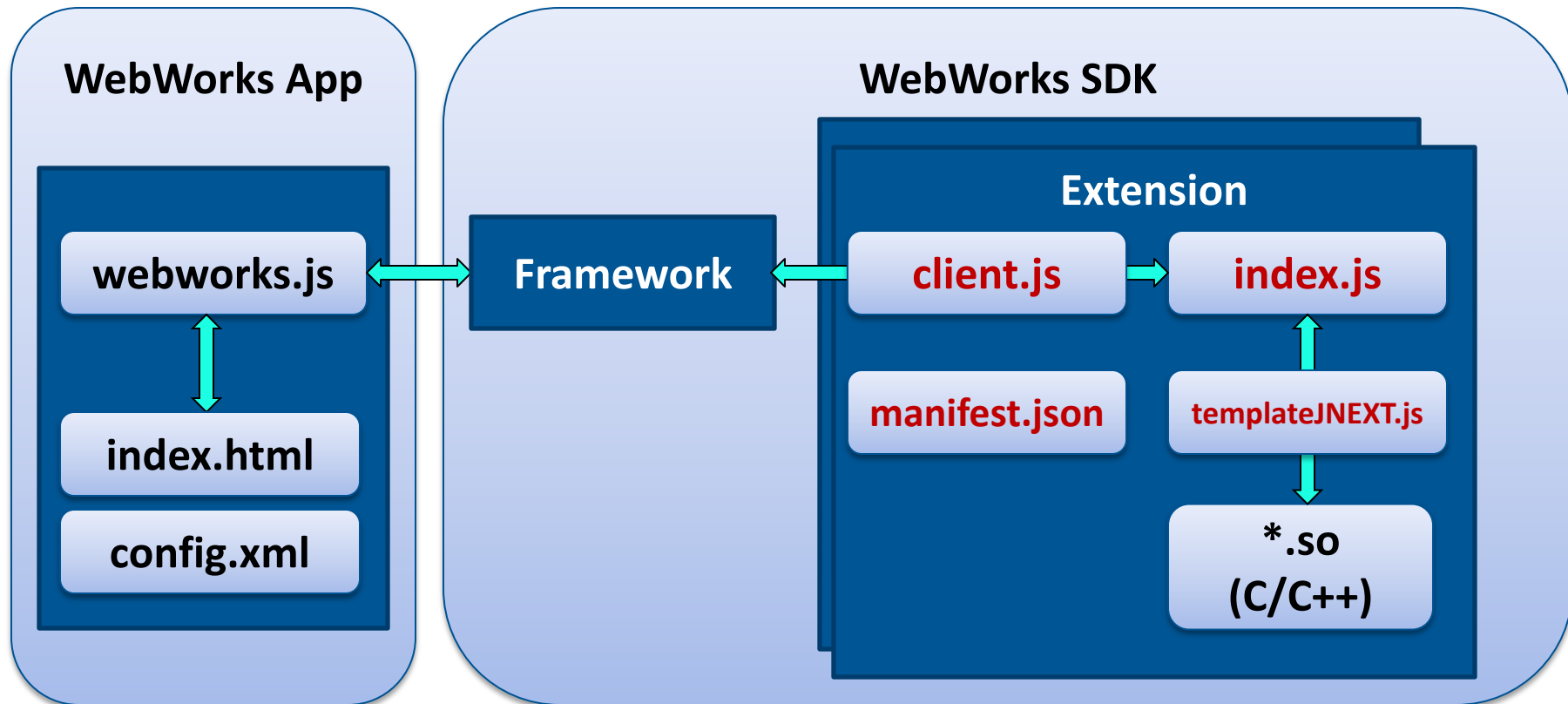
- Copy all template files from
  - ▶ C:\Lab\JAM335\starter\community.templateExt



# Writing Extensions Part 1: JavaScript

JavaScript interface for a native extension

# Writing Extensions: JavaScript Goals





- **Framework / ext / community.templateExt**
  - ▶ manifest.json
  - ▶ client.js
  - ▶ index.js
  - ▶ templateJNEXT.js
- Open using Notepad++

# Writing Extensions: JavaScript

## Step 1 of 4

- Modify **manifest.json**
  - ▶ Defines extension requirements
  - ▶ **Task:** Define the extension

```
{  
    "global"          : false,  
    "namespace"       : "community.templateExt",  
    "dependencies"    : ["utils"]  
}
```

If you require your extension to be whitelisted in the config.xml

Define list of extensions that your extension depend on

e.g. ["connection", "io"]

# Writing Extensions: JavaScript

## Step 2 of 4

- Modify **client.js**
  - ▶ Defines APIs that can be called by a WebWorks application
  - ▶ **Task:** Complete the start & stop client interface for LED function

```
_self.startLED = function (hexcolor) {  
    var args = { hexcolor : hexcolor };  
    return window.webworks.execSync(_ID, "startLED", args);  
};  
_self.stopLED = function (id) {  
    var args = { id : id };  
    return window.webworks.execSync(_ID, "stopLED", args);  
};
```

# Writing Extensions: JavaScript

## Step 2 of 4 (continued)

- Modify **client.js** (continued)
  - ▶ Defines APIs that can be called by a WebWorks application
  - ▶ **Task:** Complete the get & set client interface for blink duration

```
Object.defineProperty(_self, "blinkCount", {  
  get: function () {  
    return window.webworks.execSync(_ID, "blinkCount", null);  
  },  
  set: function (arg) {  
    window.webworks.execSync(_ID, "blinkCount", {"value": arg });  
  }  
});
```

# Writing Extensions: JavaScript

## Step 3 of 4

- Modify **index.js**
  - ▶ The main controller routes calls from client.js to templateJNEXT.js
  - ▶ **Task:** Complete the JNEXT routing of the start & stop LED function

```
startLED: function (success, fail, args) {  
    hexcolor = JSON.parse(decodeURIComponent(args["hexcolor"]));  
    success(template.getInstance().startLED(hexcolor));  
},  
stopLED: function (success, fail, args) {  
    id = JSON.parse(decodeURIComponent(args["id"]));  
    success(template.getInstance().stopLED(id));  
},
```

# Writing Extensions: JavaScript

## Step 3 of 4 (continued)

- Modify **index.js** (continued)
  - ▶ The main controller routes calls from client.js to templateJNEXT.js
  - ▶ **Task:** Complete the JNEXT routing of the get & set blink duration property

```
blinkCount: function (success, fail, args) {  
    var value;  
    if (args && args["value"]) {  
        value = JSON.parse(decodeURIComponent(args["value"]));  
        template.getInstance().blinkCount(value);  
        success();  
    } else {  
        return success(template.getInstance().blinkCount());  
    }  
}
```

- The Bridge between Web and Native
  - ▶ Connects JavaScript and C/C++ code
  - ▶ Cross platform and cross browser framework
  - ▶ Open source

<http://www.jnext.org>



# Writing Extensions: JavaScript

## Step 4 of 4

- Modify **templateJNEXT.js**
  - ▶ Communicates with native code through the JNEXT plugin
  - ▶ **Task:** Complete the JNEXT invocation for the start & stop LED function

```
self.startLED = function (hexcolor) {  
    return JNEXT.invoke(self.m_id, "startLED " + hexcolor);  
};  
self.stopLED = function (id) {  
    return JNEXT.invoke(self.m_id, "stopLED " + id);  
};
```



# Writing Extensions: JavaScript

## Step 4 of 4 (continued)

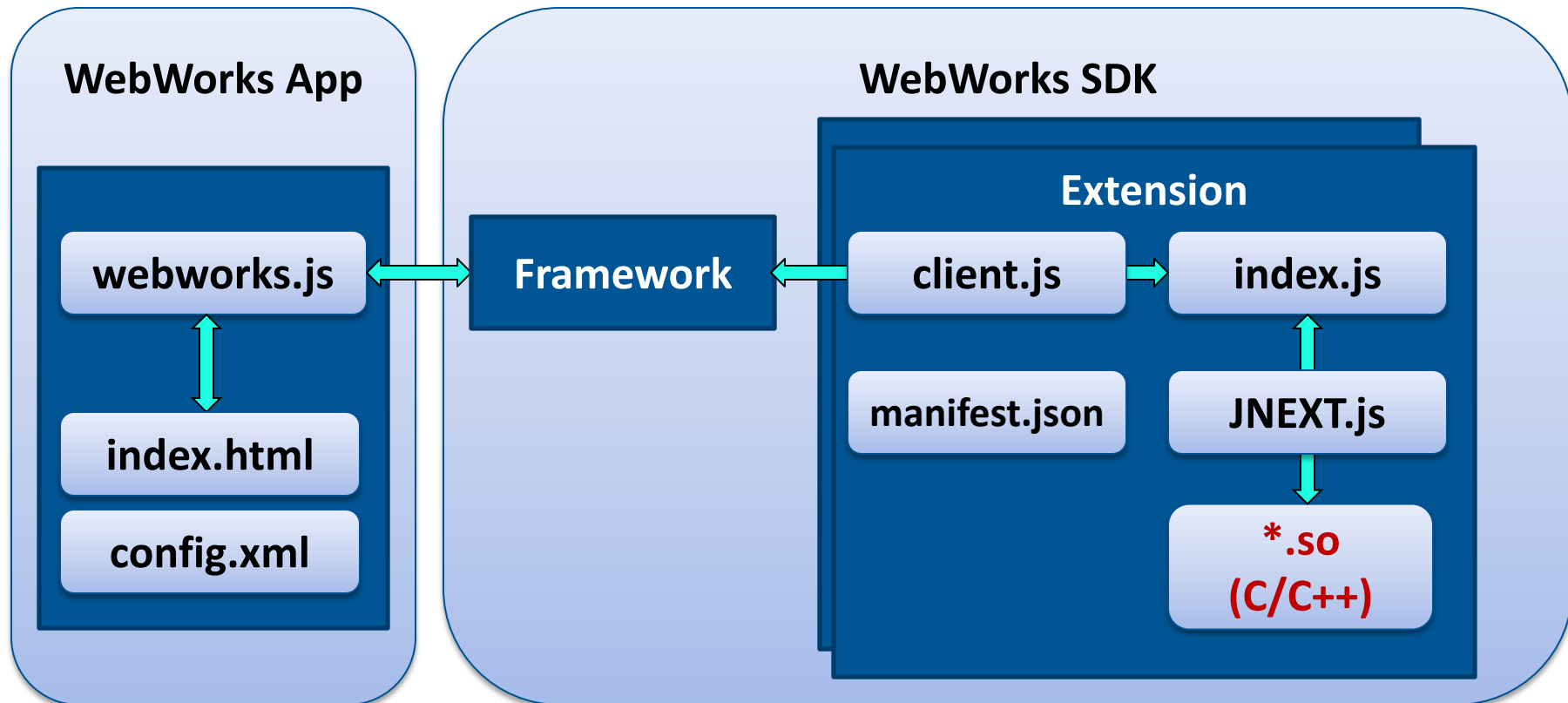
- Modify **templateJNEXT.js** (continued)
  - ▶ Communicates with native code through the JNEXT plugin
  - ▶ **Task:** Complete the JNEXT invocation for the get & set blink duration property

```
self.blinkCount = function (value) {  
    if (value) {  
        return JNEXT.invoke(self.m_id, "blinkCount " + value);  
    } else {  
        return JNEXT.invoke(self.m_id, "blinkCount");  
    }  
};
```

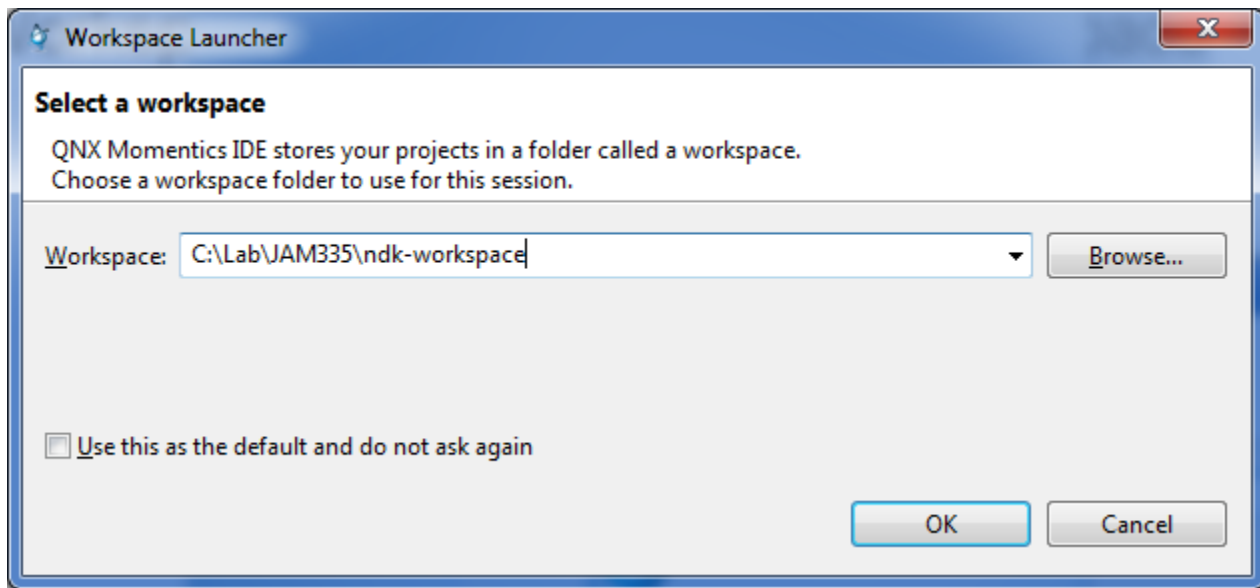
# Writing Extensions Part 2: C/C++

The native component for a WebWorks extension

# Writing Extensions: C/C++ Goals

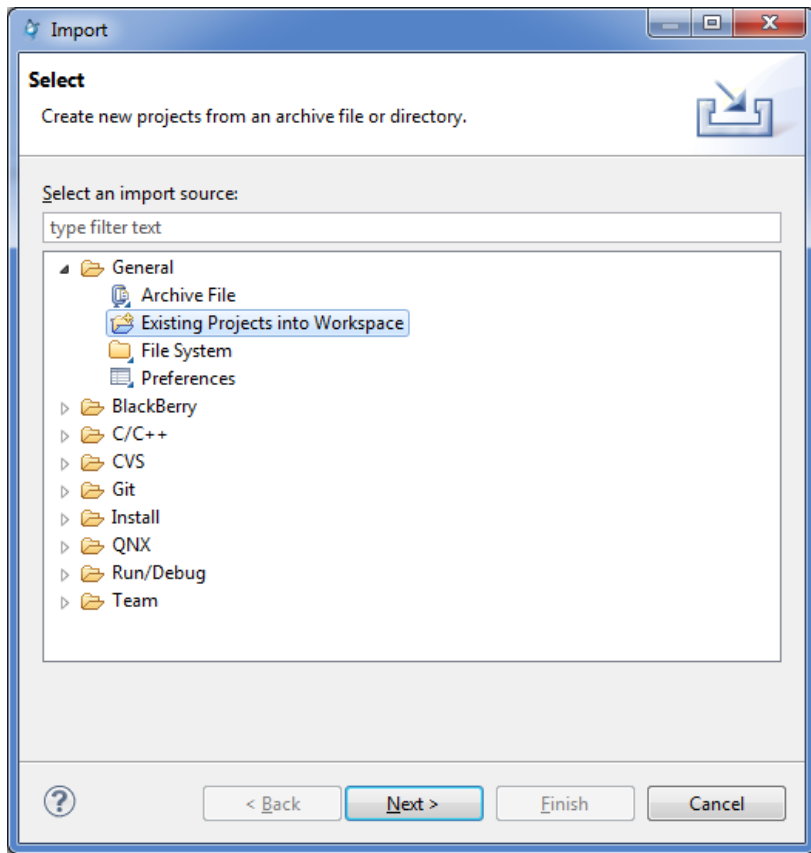


- Start the BlackBerry Native SDK
  - ▶ Workspace = C:\Lab\JAM335\ndk-workspace
  - ▶ Close the Welcome screen

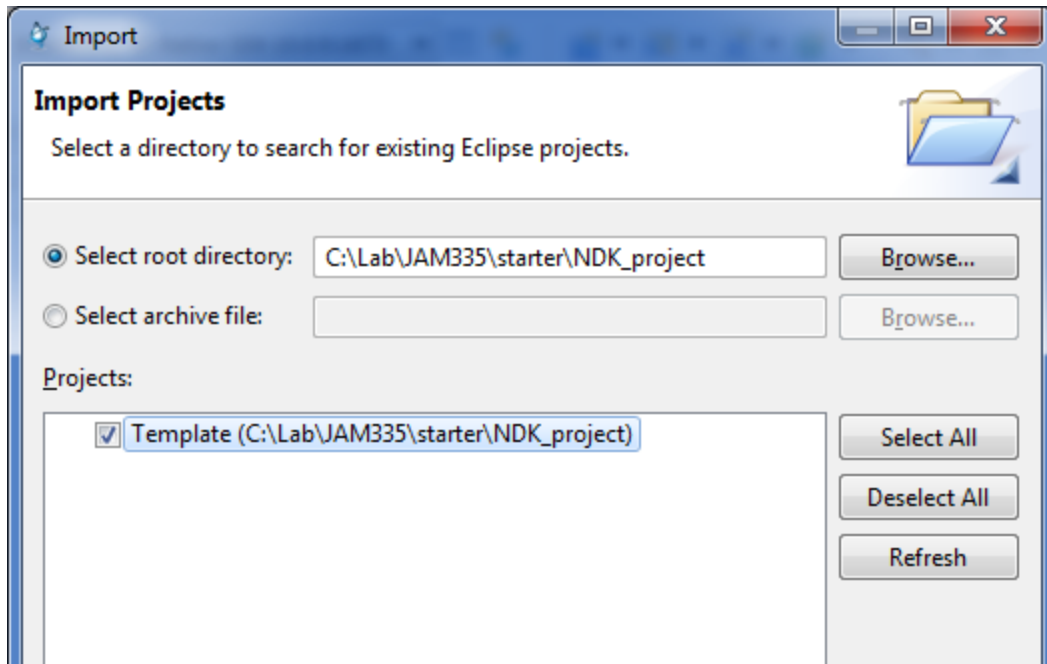


# Writing Extensions: C/C++

- Import template project
  - ▶ File → Import
  - ▶ General →
    - Existing Projects into Workspace

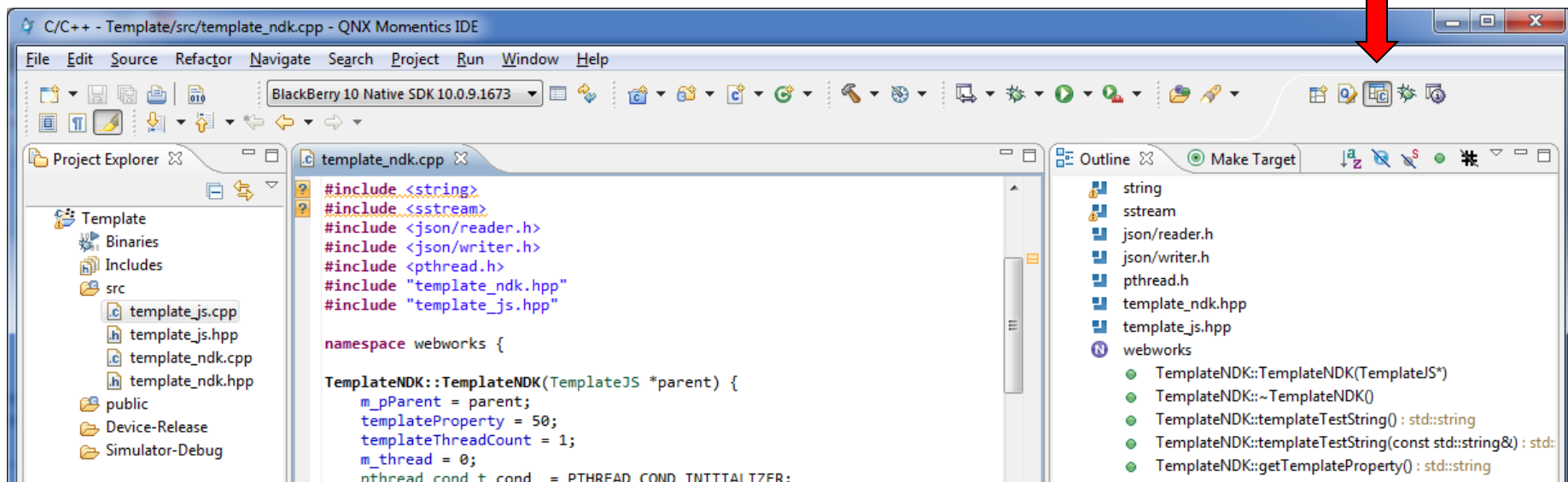


- Import template project
  - ▶ Select root directory → C:\Lab\JAM335\starter\NDK\_project



# Writing Extensions: C/C++

- Import template project
  - ▶ Expand Template/src folder in the Project Explorer window
  - ▶ Confirm the C/C++ perspective is enabled



- Modify src / **template\_js.cpp**
  - ▶ **Task:** Route JNEXT calls into native functions

```
string TemplateJS::InvokeMethod(const string& command) {  
  
    ...  
line85    if (strCommand == "startLED") {  
            return m_pTemplateController->stopLED(arg) ;  
        } else if (strCommand == "stopLED") {  
            return m_pTemplateController->stopLED(arg) ;  
        }  
  
    ...  
}
```



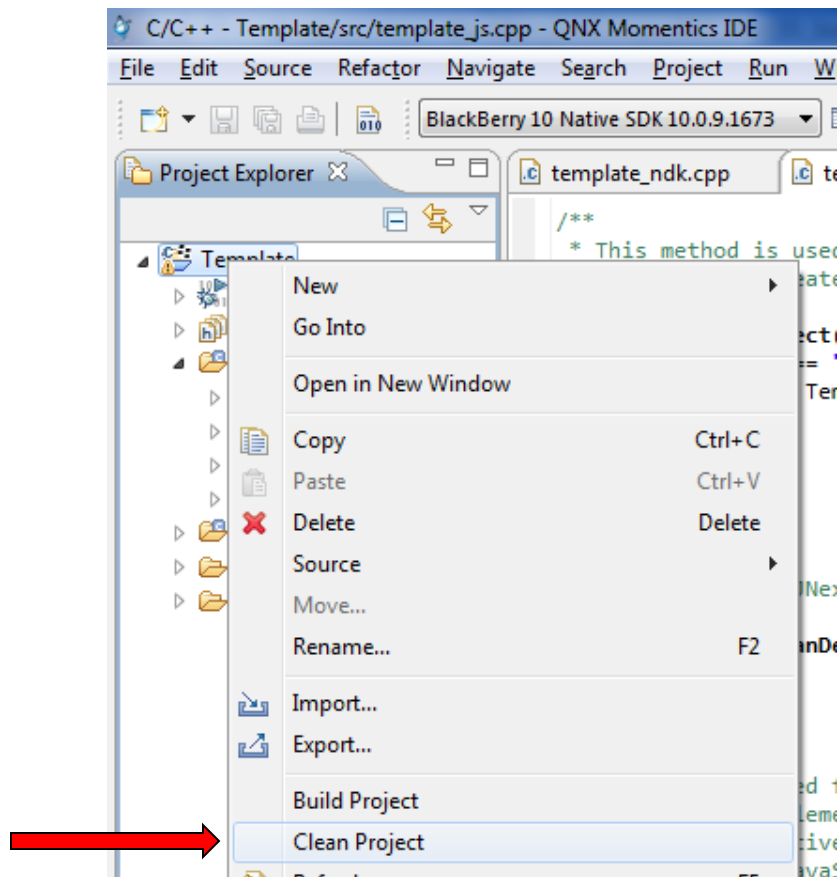
- Modify src / **template\_ndk.cpp**

- ▶ **Task:** Add native code to start and stop the LED

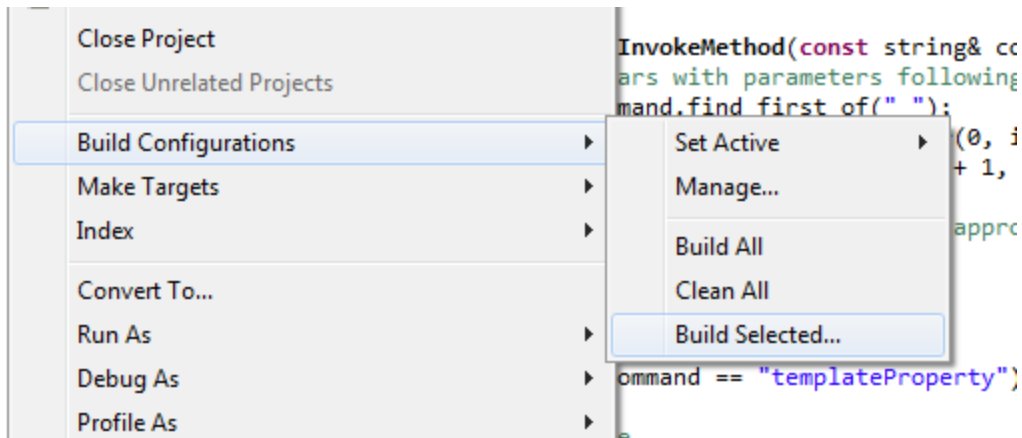
```
std::string TemplateNDK::startLED(std::string arg) {  
    ...  
line80    status = led_request_rgb(id.c_str(), colorCode, blinkCount);  
  
    ...  
}  
std::string TemplateNDK::stopLED(std::string id) {  
line97    status = led_cancel(id.c_str());  
  
    ...  
}
```

# Writing Extensions: C/C++

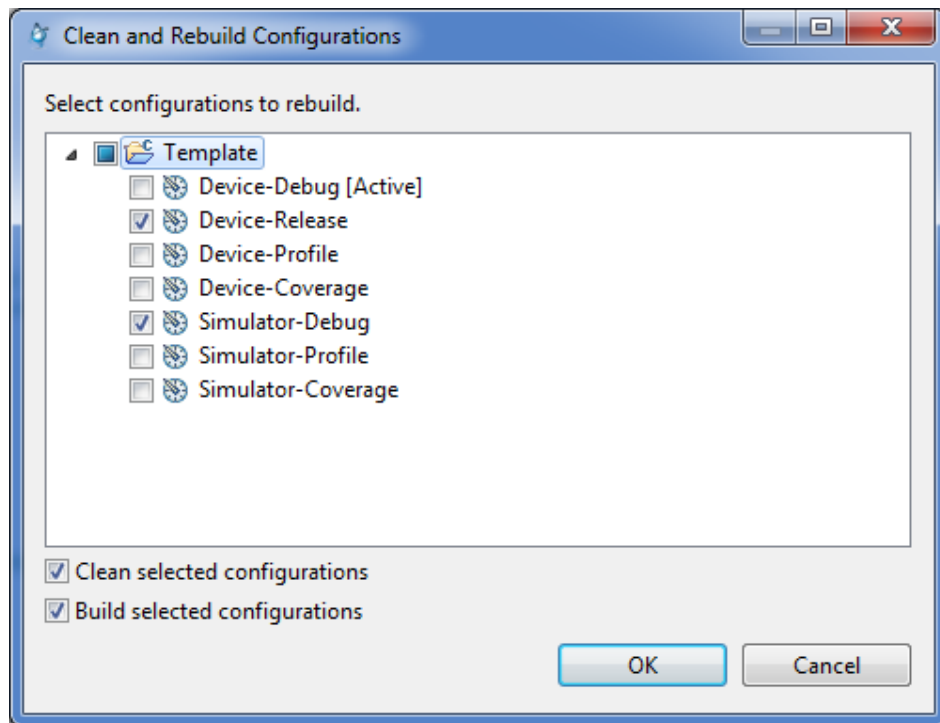
- Build the native code
  - ▶ Project Explorer
  - ▶ Right click on 'Template'
  - ▶ Clean Project



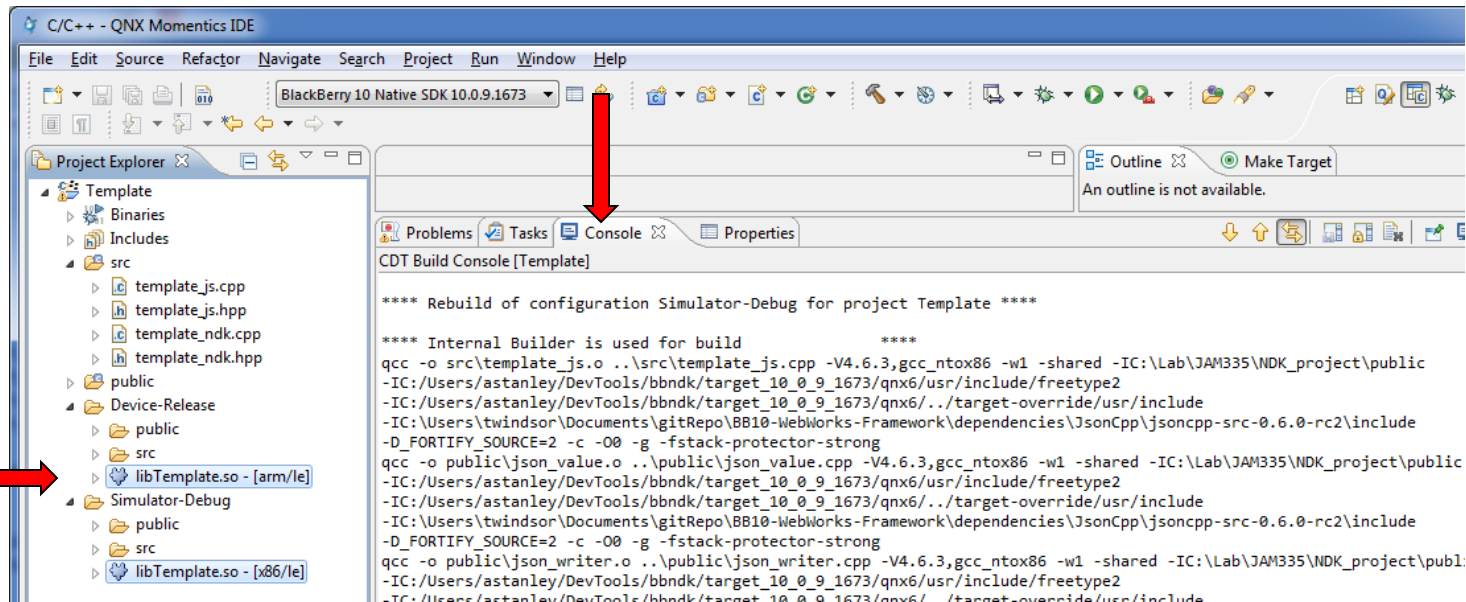
- Build the native code
  - ▶ Project Explorer
  - ▶ Right click on 'Template'
  - ▶ Build Configurations → Build Selected ...



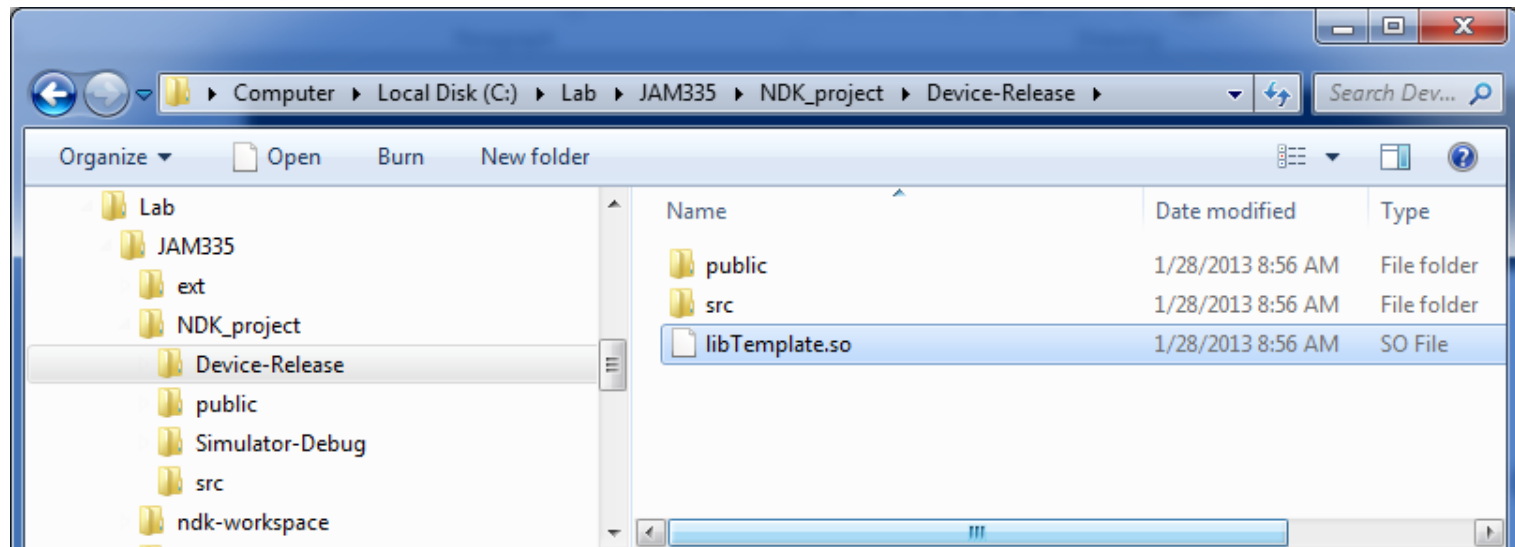
- Build the native code
  - ▶ Select Device-Release
  - ▶ Select Simulator-Debug
  - ▶ OK



- Build the native code
  - ▶ Confirm compiler messages displayed in Console
  - ▶ Confirm compiled **libTemplate.so** files created in Project Explorer



- Confirm compiled native extension \*.so files exist
  - ▶ C:\Lab\JAM335\starter\NDK\_Project\**Device-Release**\libTemplate.so
  - ▶ C:\Lab\JAM335\starter\NDK\_Project\**Simulator-Debug**\libTemplate.so

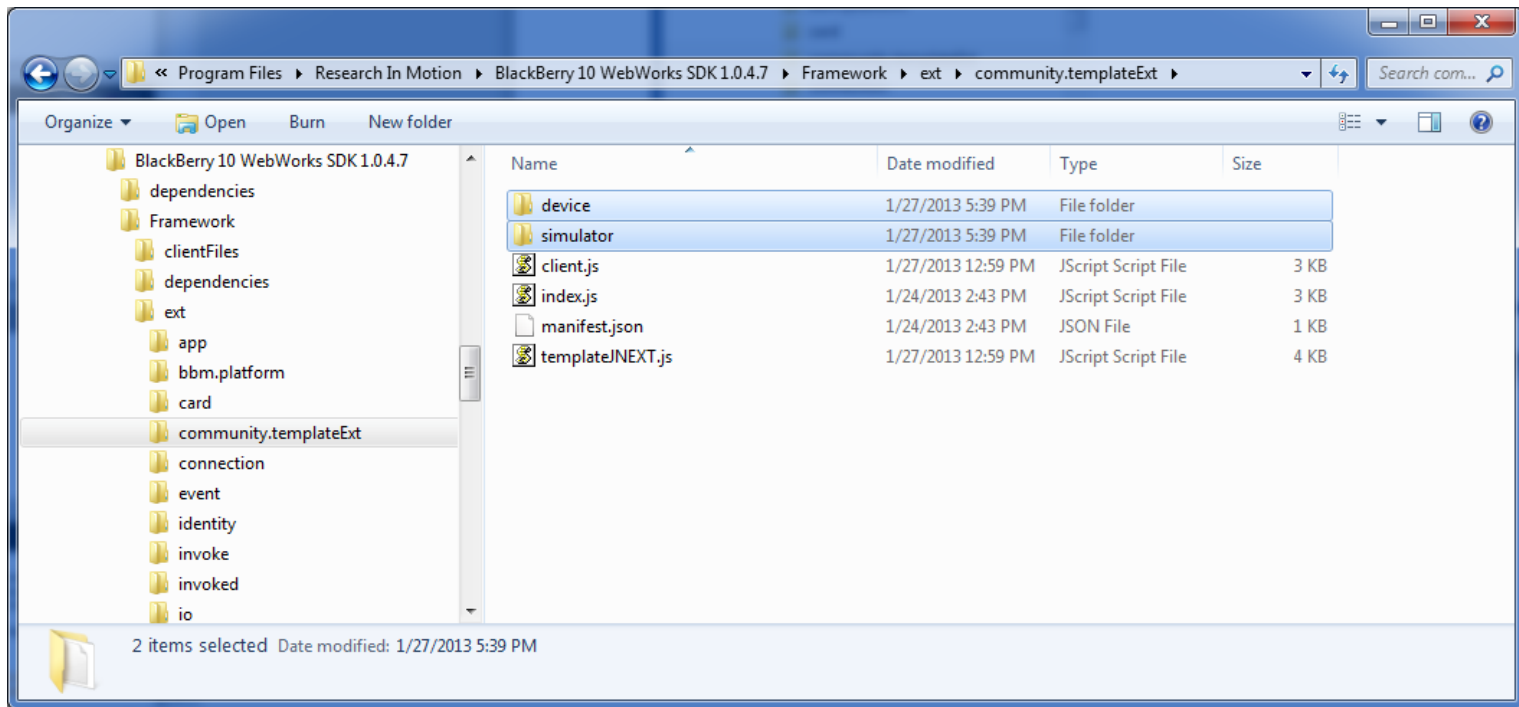


# Putting it all together

Assembling the extension

# Putting it all together

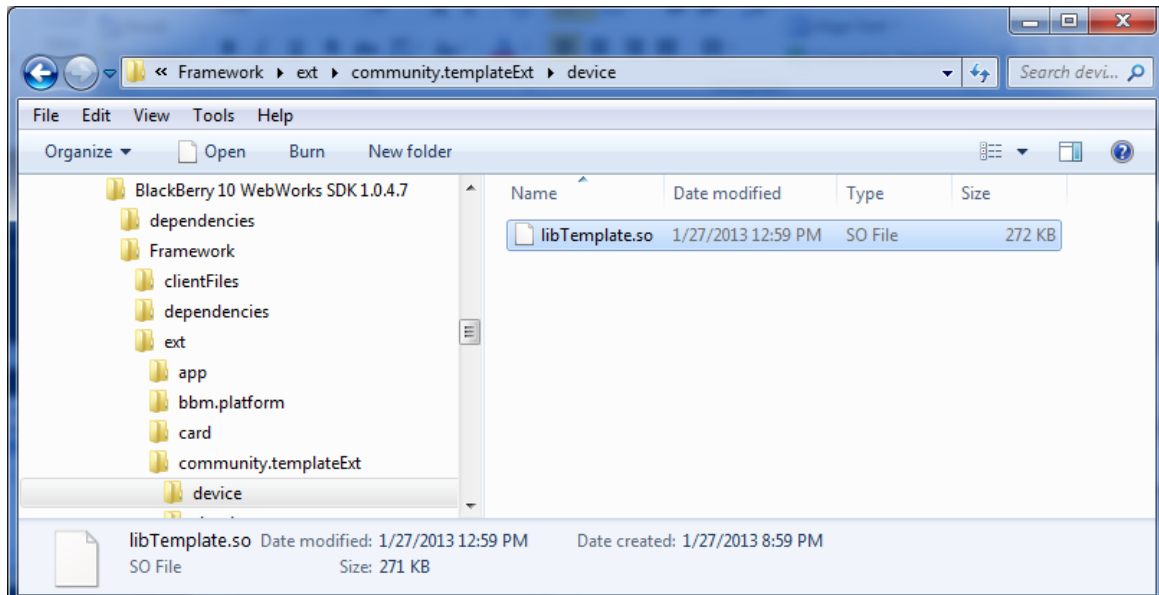
- Browse to SDK / Framework / ext / **community.templateExt**





# Putting it all together

- Copy compiled native \*.so files into WebWorks SDK
  - ▶ **Device-Release/libTemplate.so** → ext/**device**/libTemplate.so
  - ▶ **Simulator-Debug/libTemplate.so** → ext/**simulator**/libTemplate.so



# The finished product

Compiling an app with your completed extension

- C:\Lab\JAM335\starter\sample\_application
  - ▶ **Task:** Add reference to new extension in **config.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets"
        xmlns:rim="http://www.blackberry.com/ns/widgets"
        version="1.0.2"
        id="community.extension.test" xml:lang="en">

    <name>Extension Test</name>
    <author rim:copyright="2013">Your Name Here</author>
    <content src="index.html"/>
    <feature id="community.templateExt" required="true"></feature>
    <rim:permissions>
        <rim:permit>access_led_control</rim:permit>
    </rim:permissions>
</widget>
```

# Re-build sample\_application

## Settings

×

Package

Build

SDK Path

C:\Program Files\Research In Motion\BlackBerry 10

Project Root

C:\Lab\JAM335\starter\sample\_application

Archive Name

sample\_application

Output Folder

C:\Temp\sample\_application

Enable Remote Web Inspector

☒

Sign

Signing Password

.....

Bundle Number

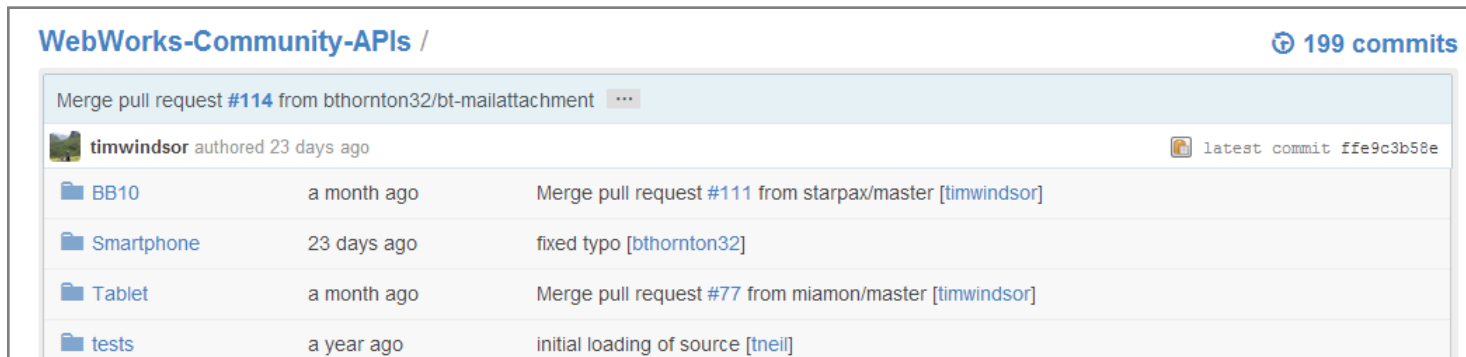
2

- Deploy signed app to BlackBerry 10 Dev Alpha
  - ▶ `cd C:\Lab\JAM335\`
  - ▶ `deploy.bat 169.254.0.1 <device password>`
- Test the application again
  - ▶ Start the LED
  - ▶ The LED should blink using the color and duration selected.

# For more info ...

Building advanced extensions

- Community-developed extensions:
  - ▶ <http://github.com/blackberry/WebWorks-Community-APIs>
  - ▶ Build your own APIs using TEMPLATE example
- We want your help!
  - ▶ Contribute your extensions to open Source





The screenshot shows the GitHub repository page for **WebWorks-Community-APIs**. At the top, it says "WebWorks-Community-APIs /" and "199 commits". Below this, there is a section for "Merge pull request #114 from bthornton32/bt-mailattachment" by **timwindsor**, authored 23 days ago. The latest commit is `ffe9c3b58e`. A table lists recent commits:

Commit	Time	Description
BB10	a month ago	Merge pull request #111 from starpax/master [timwindsor]
Smartphone	23 days ago	fixed typo [bthornton32]
Tablet	a month ago	Merge pull request #77 from miamon/master [timwindsor]
tests	a year ago	initial loading of source [tneil]



- Advanced Extensions Template

- ▶ <http://bit.ly/12dBkqp>
- ▶ Synchronous & asynchronous & callback methods
- ▶ Properties, events






WebWorks-Community-APIs / BB10 / Template / 

 History

giving up on relative links

 timwindsor authored 2 days ago latest commit 12c90fc697 

..

 NDK_project	2 days ago	Initial commit of New Template [timwindsor]
 ext	2 days ago	Initial commit of New Template [timwindsor]
 sample_application	2 days ago	Initial commit of New Template [timwindsor]
 LICENSE	2 days ago	Initial commit of New Template [timwindsor]
 README.md	2 days ago	giving up on relative links [timwindsor]



- Sessions:
  - ▶ JAM16 – Native API Deep Dive: Multimedia
  - ▶ JAM395 – Native SDK: Productivity Tips
  - ▶ JAM345 – Data Storage Web and Native
  - ▶ JAM321 – Native API deep dive: USB, Bluetooth and HDMI
  - ▶ JAM371 – Qt on BlackBerry 10 – why, where and how
- Resources:
  - ▶ Open Source extension samples: <http://bit.ly/10TEiQx>
  - ▶ Development Tools <http://developer.blackberry.com>

# THANK YOU

JAM 335

Adam Stanley @n\_adam\_stanley

Jorge del Casar Sanchez @jorgecasar

February 6, 2013