

## **LAB5b: Brand Mobiliser State Development**

Implement a New Brand Mobiliser State

## TABLE OF CONTENTS

AGENDA .....	3
BEFORE YOU START... ..	3
EXTEND THE PROJECT SERVICES INTERFACE.....	3
IMPLEMENT THE INTERFACE FOR SOAP AND OSGi.....	4
IMPLEMENT A CHECK DEPARTMENT STATE .....	4
UPDATE THE Brand BUNDLE .....	8
USE THE NEW STATE IN YOUR Brand Mobiliser APPLICATION.....	8
ACTIVATE THE APPLICATION .....	13
SIMULATE/TEST THE APPLICATION .....	13

## AGENDA

1. Extend the Project Services Interface
2. Implement the Interface for SOAP and OSGi
3. Implement a Check Department State
4. Update the Spring Configuration
5. Use the new State in your BM Application

## BEFORE YOU START...

Please start the Mobiliser 5.1 Lab Virtual machine.

The Login with user is "mobiliser" and password "sybase".

Start the JMS Broker:

```
$ cd ~workspace/custom/dist/target/com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT
$ cd jms
$ bin/activemq
```

## EXTEND THE PROJECT SERVICES INTERFACE

Navigate to product bundle brand in the eclipse and add a new interface method to the **com.sybase365.mobiliser.custom.project.brand.client.IProjectServices.java** interface class.

```
(...)  
  
import com.sybase365.mobiliser.custom.project.services.contract.api.IBlacklistEndpoint;  
import com.sybase365.mobiliser.custom.project.services.contract.api.IDepartmentEndpoint;  
  
(...)  
  
IDepartmentEndpoint getDepartmentEndpoint();  
  
(...)
```

## IMPLEMENT THE INTERFACE FOR SOAP AND OSGi

Implement the new method in the existing SOAP and OSGi implementation beans in the *brand.client.impl* package.

**ProjectServicesDirectImpl.java/ProjectServiceSOAPImpl.java:**

```
(...)  
  
import com.sybase365.mobiliser.custom.project.brand.client.IProjectServices;  
  
private IBlacklistEndpoint blacklistEndpoint;  
private IDepartmentEndpoint departmentEndpoint;  
  
public void setBlacklistEndpoint(IBlacklistEndpoint blacklistEndpoint) {  
    this.blacklistEndpoint = blacklistEndpoint;  
}  
  
@Override  
public IBlacklistEndpoint getBlacklistEndpoint() {  
    return this.blacklistEndpoint;  
}  
  
public void setDepartmentEndpoint(IDepartmentEndpoint departmentEndpoint) {  
    this.departmentEndpoint = departmentEndpoint;  
}  
  
@Override  
public IDepartmentEndpoint getDepartmentEndpoint() {  
    return this.departmentEndpoint;  
}  
  
(...)
```

- Format the code: “Ctrl+A” then “Ctrl+Shift+F” to format the contents of the files.
- Organize the imports: “Ctrl+Shift+O” then verify the imports.
- Save the **ProjectServicesDirectImpl.java/ProjectServiceSOAPImpl.java**

## IMPLEMENT A CHECK DEPARTMENT STATE

Now you can implement a new state using the SOAP or OSGi services to check for the department name.

**CheckDepartment.java (NEW):**

**Bundle = com.sybase365.mobiliser.custom.project.brand-1.2.0-SNAPSHOT.jar**

**Package = com.sybase365.mobiliser.custom.project.brand.states.impl**

```
package com.sybase365.mobiliser.custom.project.brand.states.impl;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.sybase365.mobiliser.brand.dao.DBException;
import com.sybase365.mobiliser.brand.jpa.SmappState;
import com.sybase365.mobiliser.brand.plugins.api.smapp.SmappStateProcessingAction;
import com.sybase365.mobiliser.brand.plugins.api.smapp.SmappStateProcessingContext;
import com.sybase365.mobiliser.brand.plugins.smapp.controls.Attribute;
import com.sybase365.mobiliser.brand.plugins.smapp.controls.OutputAttribute;
import com.sybase365.mobiliser.brand.plugins.smapp.controls.TextBoxAttribute;
import com.sybase365.mobiliser.brand.plugins.smapp.state.SmappStatePlugin;
import com.sybase365.mobiliser.brand.processing.exceptions.MwizProcessingException;
import com.sybase365.mobiliser.custom.project.brand.client.IProjectServices;
import com.sybase365.mobiliser.custom.project.services.contract.v1_0.GetDepartmentRequest;
import com.sybase365.mobiliser.custom.project.services.contract.v1_0.GetDepartmentResponse;

public class CheckDepartment extends SmappStatePlugin {

    private static final Logger LOG = LoggerFactory
        .getLogger(CheckDepartment.class);

    protected IProjectServices services;

    private static final TextBoxAttribute inDepartmentId = new TextBoxAttribute(
        "DEPARTMENT_ID", "Department ID [Format: 0123456789]", false);

    private static final OutputAttribute outDepartmentName = new OutputAttribute(
        "DEPARTMENT_NAME", "Department Name");

    private static Attribute[] stateAttr;
    static {
        stateAttr = new Attribute[] { inDepartmentId, outDepartmentName };
    }

    @Override
    protected Attribute[] getStateAttributes() {
        return stateAttr;
    }

    public IProjectServices getServices() {
        return this.services;
    }

    public void setServices(IProjectServices services) {
        this.services = services;
    }
}
```

```
@Override
public long getStateId() {
    return 900500L;
}

@Override
public String getStateName() {
    return "Check Department Name";
}

@Override
public boolean supportsFailTransition() {
    return true;
}

@Override
public String getRevisionString() {
    return "$Revision: 262450 $";
}

@Override
protected SmappState processStateLogic(SmappStateProcessingContext context,
    SmappStateProcessingAction action) throws MwizProcessingException,
    DBException {
    LOG.info("Entering state #{}; {}", getStateId(), getStateName());
    GetDepartmentRequest request;
    try {
        LOG.info("Trying to get Department with ID #{}", inDepartmentId
            .getInputValue().getLong());
        request = new GetDepartmentRequest();
        request.setOrigin("brand");
        request.setId(inDepartmentId.getInputValue().getLong());
    } catch (Exception e) {
        LOG.info("Exception Occurred ({}!)", e.getMessage());
        return continueFail();
    }
    GetDepartmentResponse response = services.getDepartmentEndpoint()
        .getDepartment(request);
    if (response.getStatus().getCode() != 0) {
        LOG.info("Department not found!");
        return continueFail();
    }
    LOG.info("Department found with Name '{}'", response.getDepartment()
        .getName());
    outDepartmentName.setValue(response.getDepartment().getName());

    return continueOk();
}
```

1. Format the code: "Ctrl+A" then "Ctrl+Shift+F" to format the contents of the file.
2. Organize the imports: "Ctrl+Shift+O" then verify the imports.
3. Save the CheckDepartment.java

## UPDATE THE SPRING CONFIGURATION

Update the Spring configuration to setup the new SOAP and OSGi services. Make sure to expose the new State to the OSGi service registry using the `com.sybase365.mobiliser.brand.plugins.api.smapp.StatePlugin` interface in your service-context.

### beans-context.xml

```
(...)  
  
<bean id="checkDepartment"  
    class="com.sybase365.mobiliser.custom.project.brand.states.impl.CheckDepartment">  
    <property name="services" ref="soapServices"/>  
</bean>  
  
(...)  
  
<bean id="directServices"  
class="com.sybase365.mobiliser.custom.project.brand.client.impl.ProjectServicesDirectImpl">  
    <property name="blacklistEndpoint" ref="blacklistEndpointService" />  
    <property name="departmentEndpoint" ref="departmentEndpointService" />  
</bean>  
  
<bean id="soapServices"  
class="com.sybase365.mobiliser.custom.project.brand.client.impl.ProjectServiceSOAPImpl">  
    <property name="blacklistEndpoint" ref="blacklistSOAPClient" />  
    <property name="departmentEndpoint" ref="departmentSOAPClient" />  
</bean>  
  
(...)  
  
<bean id="departmentSOAPClientSource"  
    class="com.sybase365.mobiliser.util.tools.clientutils.api.RefreshableClientTargetSource">  
    <property name="clientFactory" ref="soapClientFactory" />  
    <property name="configuration" ref="serviceConfiguration" />  
    <property name="clientInterfaces">  
    <list>  
        <value>com.sybase365.mobiliser.custom.project.services.contract.api.IDepartmentEndpoint  
        </value>  
    </list>  
    </property>  
    <property name="endpointSuffix" value="/custom" />  
</bean>  
  
<bean id="departmentSOAPClient" class="org.springframework.aop.framework.ProxyFactoryBean">  
    <property name="targetSource" ref="departmentSOAPClientSource" />  
    <property name="interfaces">  
    <list>  
        <value>com.sybase365.mobiliser.custom.project.services.contract.api.IDepartmentEndpoint  
        </value>  
    </list>  
    </property>  
</bean>  
  
(...)
```

**service-context.xml**

```
(...)  
  
<osgi:reference id="departmentEndpointService"  
  interface="com.sybase365.mobiliser.custom.project.services.contract.api.IDepartmentEndpoint"  
  availability="optional" />  
  
(...)  
  
<osgi:service ref="checkDepartment"  
  interface="com.sybase365.mobiliser.brand.plugins.api.smapp.StatePlugin" />  
  
(...)
```

**UPDATE THE Brand BUNDLE**

1. Change directory to ~/workspace/custom/brand
2. Run "mvn clean install".
3. Once the maven shows "BUILD SUCCESS", go to the following directory:
  - a. cd ~/workspace/custom/brand/target
  - b. You should see following two files (among others):
    - i. com.sybase365.mobiliser.custom.project.brand-1.2.0-SNAPSHOT.jar
    - ii. com.sybase365.mobiliser.custom.project.brand-1.2.0-SNAPSHOT-sources.jar
  - c. Copy the updated bundle (com.sybase365.mobiliser.custom.project.brand-1.2.0-SNAPSHOT.jar) to  
"/workspace/custom/dist/target/com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT/brand/bundle/application".

**USE THE NEW STATE IN YOUR Brand Mobiliser APPLICATION**

Accessing the Application Composer (The following steps are also found in Lab 5a)

1. Login to the Brand Mobiliser application. (admin/Brand!23)
2. Select the "Workspace Administration", "Manage Workspaces" and "default" options.
3. Click the "SHORTCODE" tab.
4. Add a shortcode, check the "Use For Reply" checkbox and click the "Add" button.
5. Click the "DASHBOARD" option in the upper left of the screen.
6. Click the "Create Interactive Application" link located at the top of the "My Applications" section. (Under "Schedule".)
7. Enter the "Name" of the application being created such as "MyLab5b".
8. Enter the "From" and "To" dates and click the "Save" button.
9. Click the "KEYWORDS" tab.
10. Enter a Keyword and the "Active To/From" dates.
11. Click the save icon in the "Actions" column.
12. Click the "APPLICATIONS COMPOSER" tab. Double-Click on the state node displaying the name you selected for the application. Select the "Send SMS [New]" option in the dropdown box.
13. Click "Save". Your screen should look similar to the following:



SYBASE 365

BRAND MOBILISER WEB

Workspace: default | Logged-in as admin (Logout) | My Account | Help

DASHBOARD

EVENTS

SUBSCRIBERS

ASSETS

REPORTS

WORKSPACE ADMINISTRATION

Actions

Assets > MyLab5b

MyLab5b | Type: Interactive Application

APPLICATION DETAILS

KEYWORDS

APPLICATION COMPOSER

Application Actions

SAVE LAYOUT

REVERT LAYOUT

Gridlines (All)

Zoom (100%)

MYLAB5B

Start Application

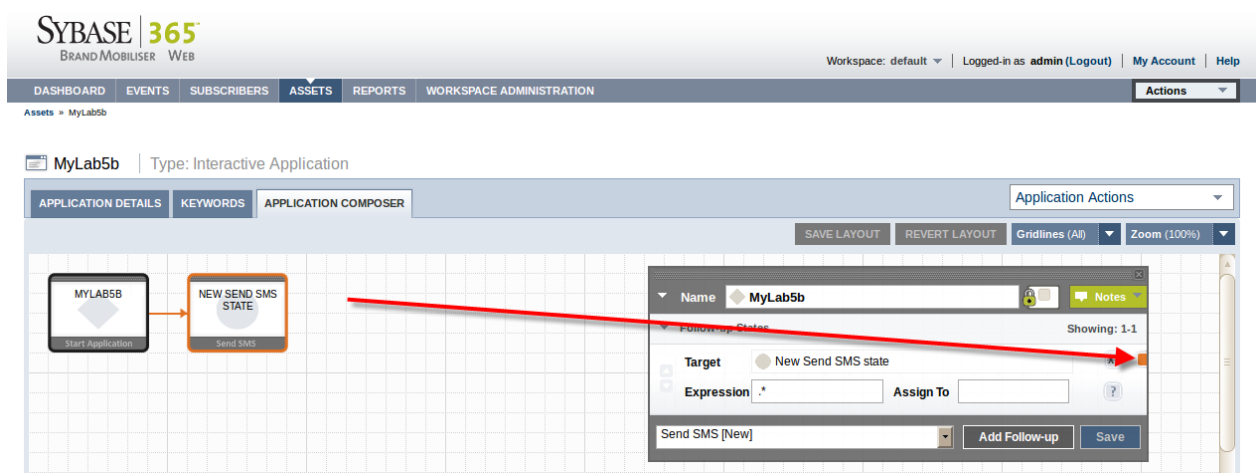
Name MyLab5b

Send SMS [New]

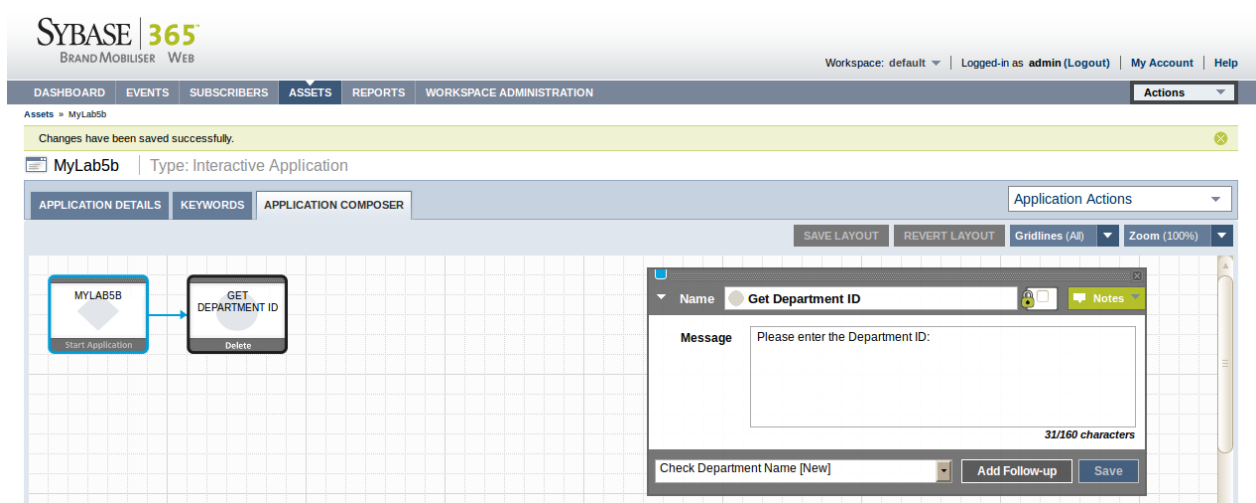
Add Follow-up

Save

14. Click the “Add Follow-up Button to create another state node.
15. Click the “Go to target State” button at the far right of the node properties box as indicated by the arrow.

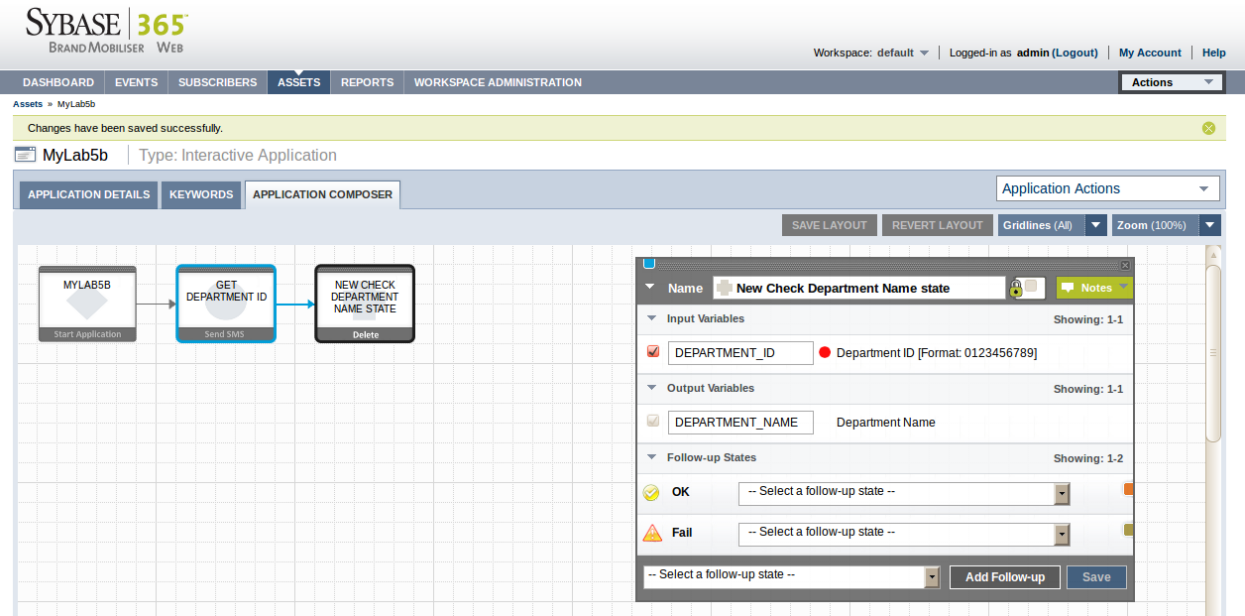


16. Add the node Name, the Message prompting for the Department ID and select the “Check Department Name [New]” option from the dropdown box.
17. Click the “Save” button.

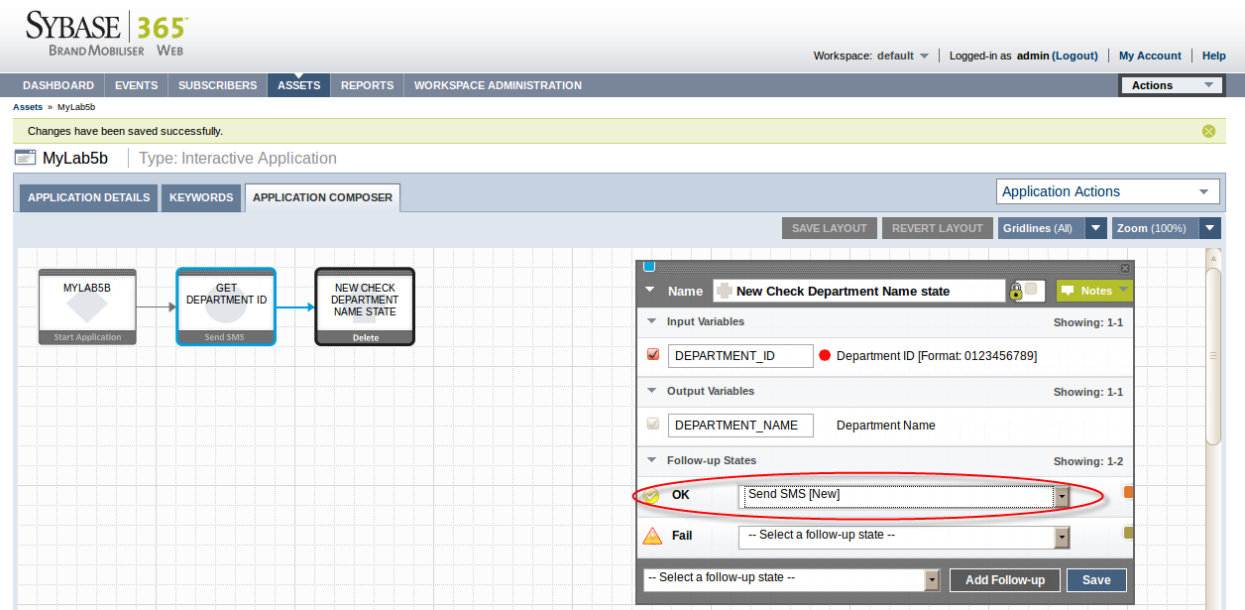


18. Click the “Add Follow-up” button.
19. Enter the characters “(.\*)” in the “Expression” textbox.
20. Enter “DEPARTMENT\_ID” in the “Assign To” textbox.
21. Click “Save”.
22. Click the “Go to target State” button at the far right of the node properties box.

23. Under “Output Variables” enter the session variable name. This example uses “DEPARTMENT\_NAME”. Click the “Save” button. Your screen should look similar to the one below:

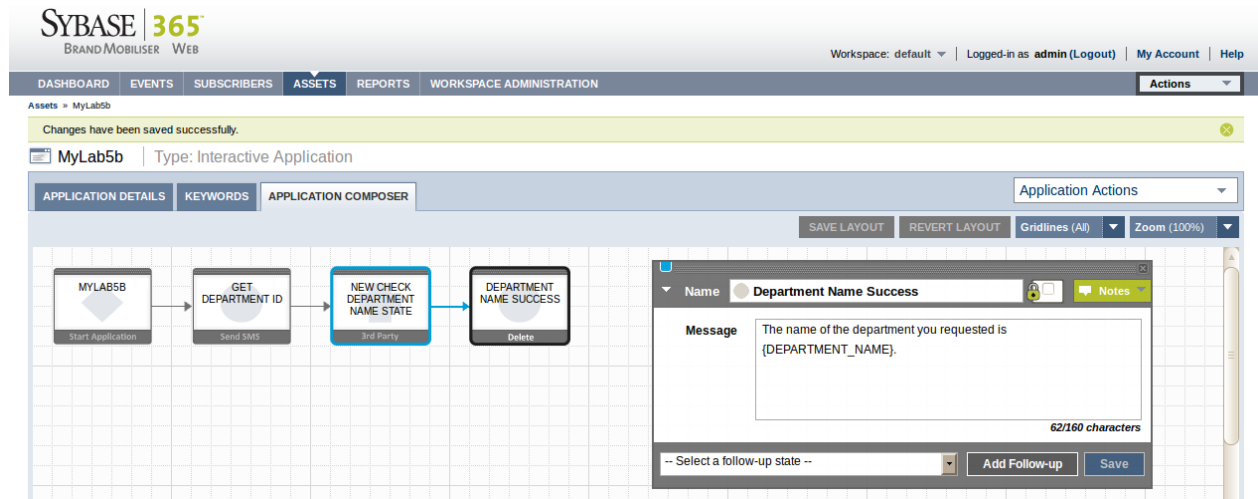


24. Select the “Send SMS [New]” state from the “OK” dropdown box.

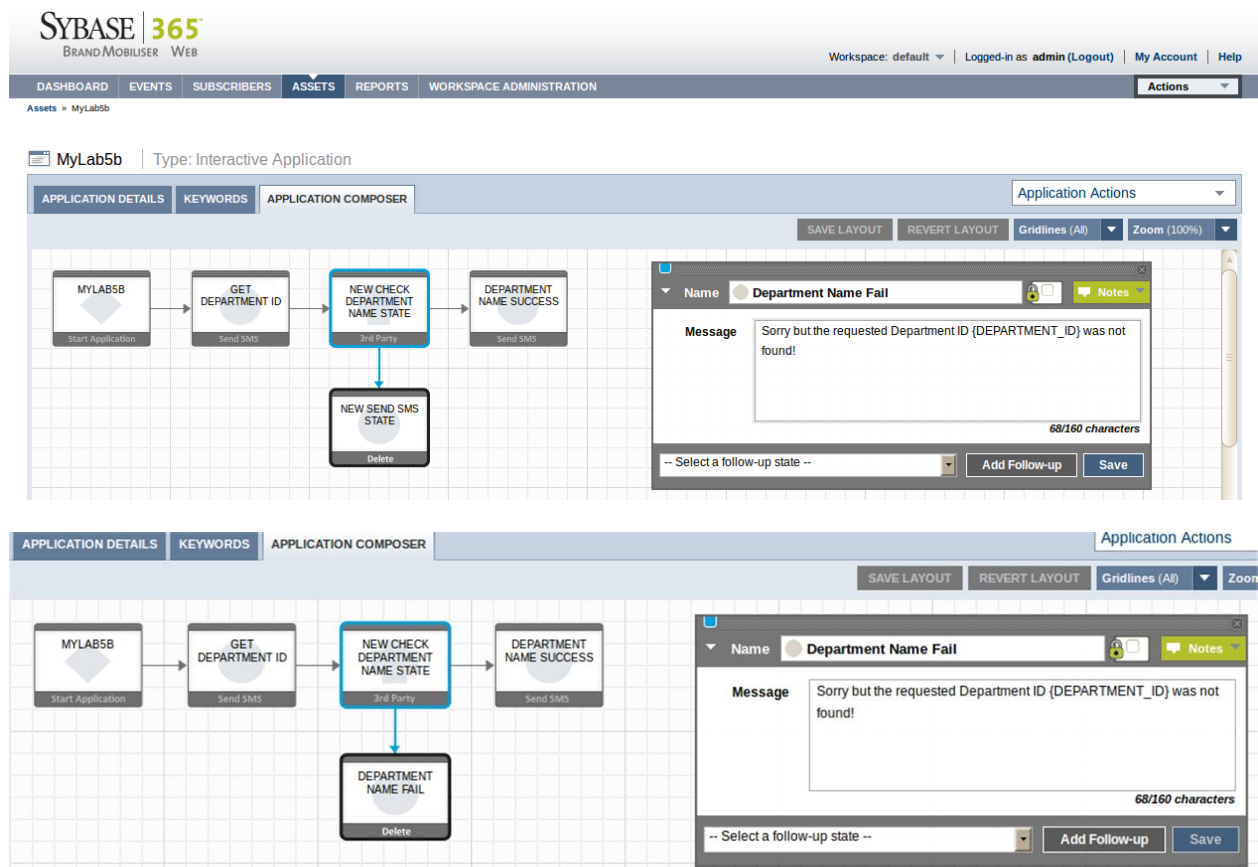


25. Click “Save”.
26. Click the “OK” “Go to target State” button at the far right of the node properties box.

27. Update the node state properties box by changing the “Name” and adding a Message similar to the one used below. Click the “Save” button.
28. Your screen should look similar to the one below:



29. Click on the “New Check Department Name State” node.
30. Update the properties by selecting the “Send SMS [New]” option from the “Fail” dropdown box.
31. Click the “Save” button.
32. Click the “Fail” “Go to target State” button at the far right of the node properties box.
33. Update the new node similar to the following:



34. Click the “Save” button.

## ACTIVATE THE APPLICATION

To test the application it must be activated first. This is to support the maker/checker concept and to avoid changes being visible to the end user instantly.

1. Select the “Activate Application” option from the “Application Actions” dropdown box locate at the upper-right of the page.
2. Click the “Load Applications for Activation” button.
3. Locate your application or the application for activation and click the dropdown for it in the “Actions” column.
4. Select the “Activate” option.
5. When prompted, click “OK” to activate.

## SIMULATE/TEST THE APPLICATION

To test your application, select the “Simulate Application” option from the “Actions” drop down box located at the upper right of your screen.

1. Enter the MSISDN
2. Select the Shortcode
3. Enter the Keyword as the “Message Text” box
4. Click the “Send to Brand Mobiliser” button
5. Replace the Keyword with the department number in the “Message Text” box
6. Click the “Send to Brand Mobiliser” button

Using the above steps your screen should look similar to the following indicating application access to the new state(s).

<b>Customer MSISDN</b>	8052221212
<b>Workspace Short   Long Code</b>	11111
<b>Message Encoding</b>	normal
<b>Message Text</b>	fun

Enter the Department\_ID created in Lab 4a when prompted. You should receive messages similar to the following:

Direction	Sender	Application	Receiver	Message
OUT	11111	MyLab5b	8052221212	The name of the department you requested is test_department104.
IN	8052221212	MyLab5b	11111	500006900
OUT	11111	MyLab5b	8052221212	Please enter the Department ID:
IN	8052221212	MyLab5b	11111	fun

© 2013 SAP AG. All rights reserved.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

