IBM® WebSphere® Commerce 7 – Lab exercise

Marketing Customization 1

What this exercise is about

In this tutorial, you will implement a new marketing activity target. Adding a new target is one of many customization options available for the Marketing tool in version 7. This is the first in a series of three labs designed to demonstrate marketing customization. The labs can be completed individually but each will build on concepts and techniques introduced in the previous lab. The recommended starting point for this first lab is a review of the Marketing Customization presentation.

The new target you will create allows marketing managers to target registered customers who have entered their birthday. This target might be used in a promotion designed to encourage more customers to supply a birth date. Once the birth date is on file it can be used to trigger additional marketing initiatives such as special birthday discounts.

What you should be able to do

After completing this exercise, you should be able to:

- Define and register a marketing element implementation template
- Implement a task command to evaluate a target condition
- Add a new target to the Management Center Web activity palette

Introduction

The following naming conventions are used in the exercises:

Reference Variable	Description
<WCDE_INSTALL_DIR>	WebSphere Commerce Developer installation directory
<WCDE_HOST>	Hostname for WebSphere Commerce Developer
<LAB_FILE_DIR>	Location lab files were extracted to

The .zip file file, MarketingCustomizationLabs.zip, provided with this lab contains all the new files necessary to complete this lab. There is also a snippets file provided which contains just the text to be typed in. Unless otherwise stated in the instructions, you should type in the code provided. If you find this is too much typing or need assistance solving a problem, the solution files and snippets file are available.

Requirements

Before beginning this lab, ensure you have:

- Installed WebSphere Commerce Developer 7.0

Part 1: **Define the new target**

In this part of the lab, you will plan your new target and collect all the information you need to proceed with the customization. This information includes parameters specified by the business user when setting up the target and the command that will implement the business logic to evaluate whether the target is met. Once you have all the information, you create an implementation template. This template links all the information together.

____ 1. Identify what parameters the business user needs to specify to create the target. In this example, the target is any customer who has entered their birthday. No additional parameters are needed.

____ 2. Identify the name of the task command that will assess if the target is met. In this example, you will use the task command `com.mycompany.EnterBirthdayTaskCmd`. You will create this task command later in this lab.

____ 3. Define the implementation template for the target using the information gathered in steps 1 and 2. An implementation template is an XML snippet of the form:

```
<FlowElementImplementation type="UNIQUE_NAME">
    <Implementation invocationType="TaskCommand">
        <Class name="TASK_COMMAND_CLASS_NAME">
                <Argument name="ARGUMENT1"
value="VALUE_OF_ARGUMENT1"/>
                <Argument name="ARGUMENT2"
value="VALUE_OF_ARGUMENT2"/>
        </Class>
    </Implementation>
</FlowElementImplementation>
```

__ a. The attribute `type` is a unique name for the target that you create. For this target, set the type to **Enter Birthday**.

__ b. The completed implementation template for this target is shown below**.**

```
<FlowElementImplementation type="Enter Birthday">
    <Implementation invocationType="TaskCommand">
        <Class name="com.mycompany.EnterBirthdayTaskCmd">
        </Class>
    </Implementation>
</FlowElementImplementation>
```

____ 4. Insert the implementation template in the marketing element template table, DMELETEMPLATE.

__ a. Choose a name for the target. This name will be used to identify the target in Management Center. For this lab, use the name **customEnterBirthday.**

__ b. Access the development database using the URL: http://localhost/webapp/wcs/admin/servlet/db.jsp

__ c. Run the following command to insert the implementation template into the DMELETEMPLATE table. The value **1000** was chosen as a unique primary key and the value **2** signifies that this is a target template.

```
insert into dmeletemplate (dmeletemplate_id, dmelementtype_id, name,
implxml) values (1000, 2, 'customEnterBirthday',
'<FlowElementImplementation type="Enter Birthday"><Implementation
invocationType="TaskCommand"><Class
name="com.mycompany.EnterBirthdayTaskCmd"></Class></Implementation></
FlowElementImplementation>');
```

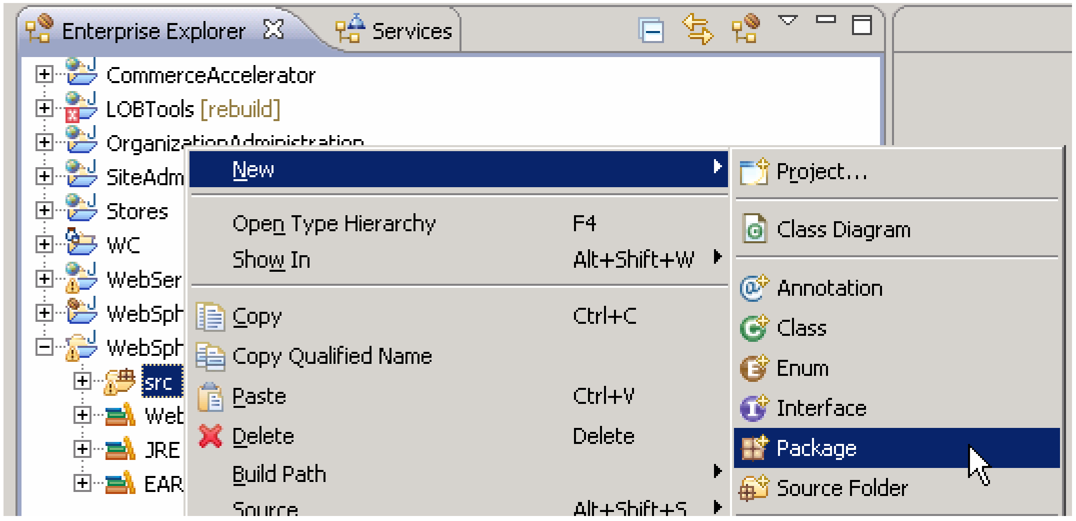
Part 2: Implement the target task command

The target task command provides the business logic that determines whether a target is met or not. A target task command must implement the MarketingCampaignElementTaskCmd interface. In this part of the lab, you will create the new task command specified in the target implementation template.

____ 1. Create the command interface.

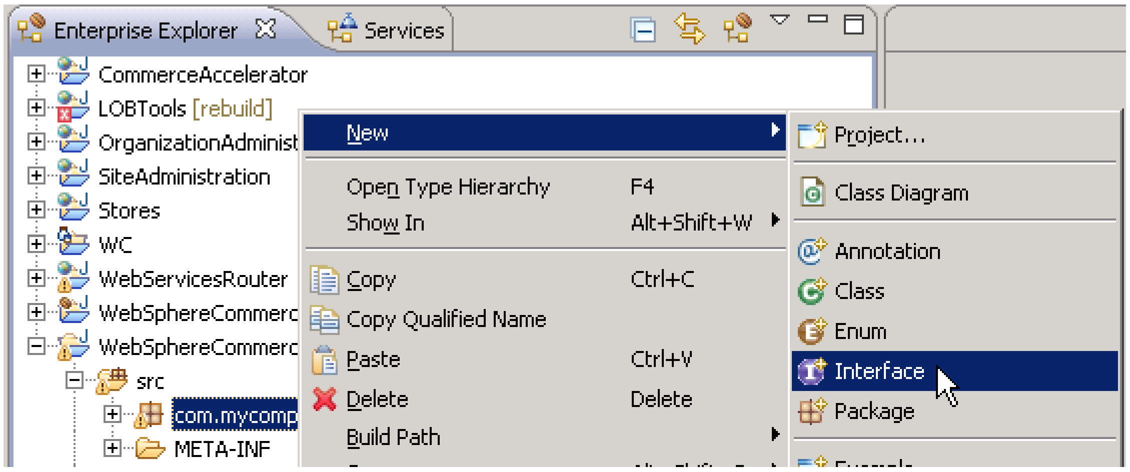
__ a. Open WebSphere Commerce Developer and navigate to the **WebSphereCommerceServerExtensionsLogic** project in the **Enterprise Explorer** tab.

__ b. Expand the project and right click the **src** folder. Select **New > Package**



__ c. Enter the package name **com.mycompany**.

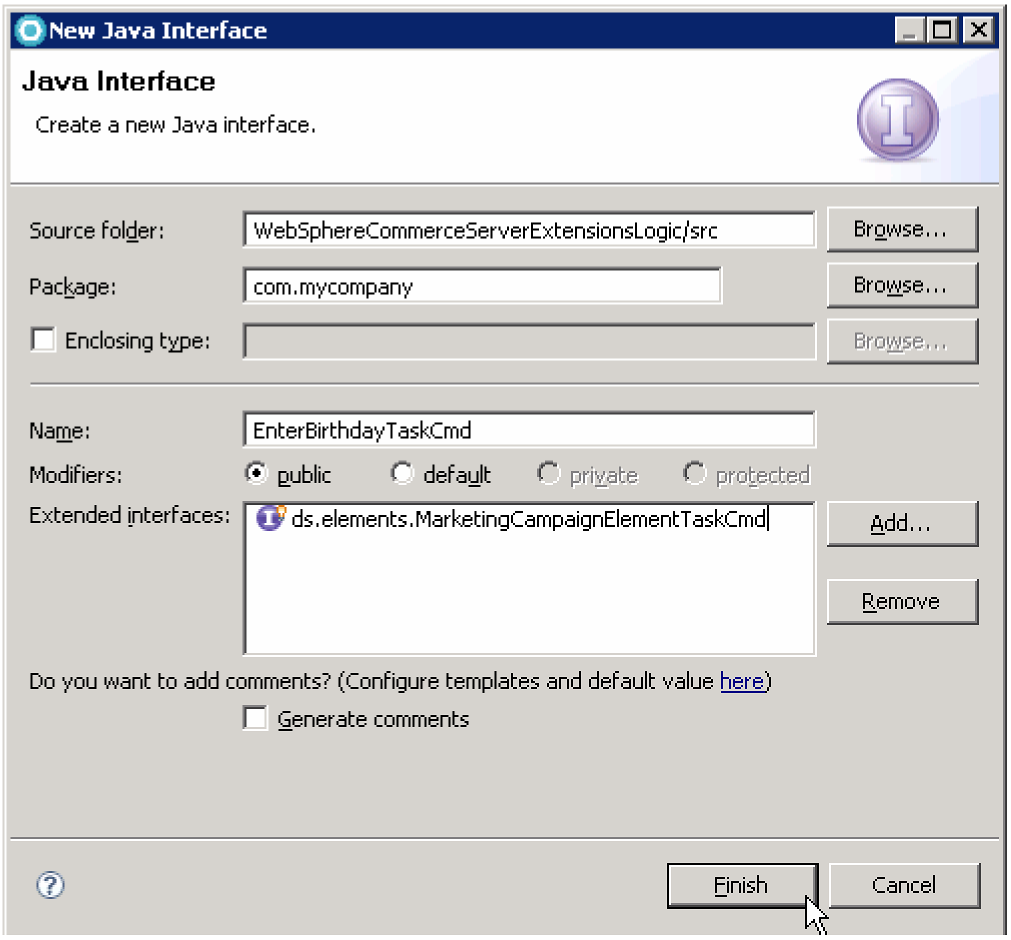
__ d. Right click the **com.mycompany** package and select **New > Interface**.



__ e. Set the interface name to **EnterBirthdayTaskCmd**.

__ f. Click the **Add** button next to the **Extended interfaces** field. Start typing **MarketingCampaingElementTaskCmd** until you see the interface name in the list. Select the interface and click **OK.**

__ g. Click **Finish** to close the dialog.



New Java Interface

Java Interface

Create a new Java interface.

Source folder: WebSphereCommerceServerExtensionsLogic/src

Package: com.mycompany

Enclosing type: [ ]

Name: EnterBirthdayTaskCmd

Modifiers: [✓ public] default private protected

Extended interfaces: ds.elements.MarketingCampaignElementTaskCmd

Do you want to add comments? (Configure templates and default value here)

Generate comments [ ]

____ 2. Update the generated interface.

__ a. Add the following member variable. Do not worry about the error. You will fix that in the next step.

```
public final static String defaultCommandClassName =
EnterBirthdayTaskCmdImpl.class.getName();
```

__ b. When you are done, your interface should look like the one shown below.

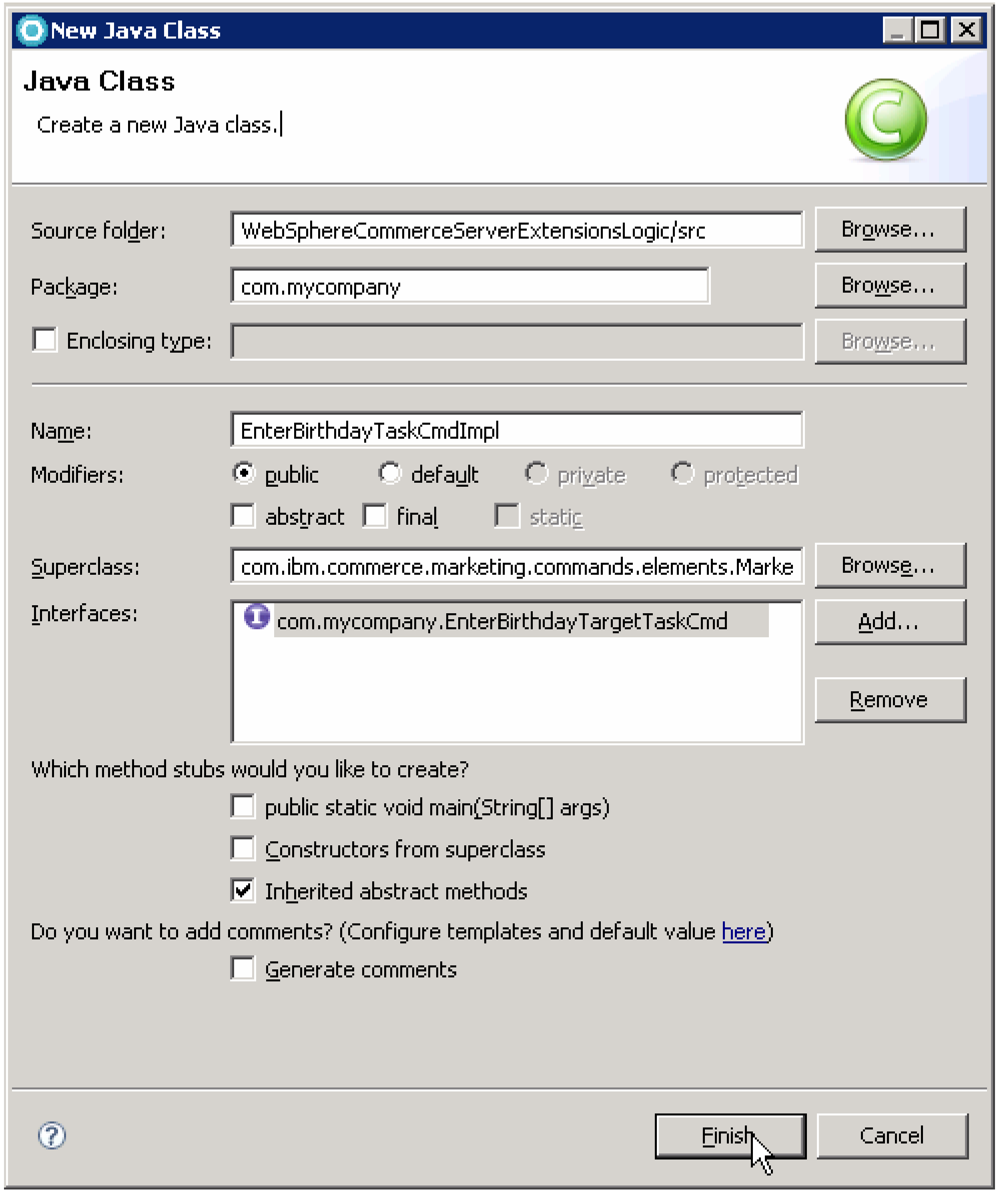
```
package com.mycompany;

import com.ibm.commerce.marketing.commands.elements.MarketingCampaignElementTaskCmd;

public interface EnterBirthdayTaskCmd extends MarketingCampaignElementTaskCmd {
    public final static String defaultCommandClassName = EnterBirthdayTaskCmdImpl.class.getName();
}
```

__ c. Save the file.

____ 3. Create the task command implementation.

__ a. Right click on the **com.mycompany** package and use the pop-up menu to create a new class file**.**

__ b. Set the class name to **EnterBirthdayTaskCmdImpl,** add **MarketingCampaignElementTaskCmdImpl** as the superclass and **EnterBirthdayTaskCmd** as the interface. Click the **Finish** button to create the class.



New Java Class

Java Class

Create a new Java class.

Source folder: WebSphereCommerceServerExtensionsLogic/src

Package: com.mycompany

Enclosing type: [ ]

Name: EnterBirthdayTaskCmdImpl

Modifiers: [✓ public] default private protected

abstract [ ] final [ ] static [ ]

Superclass: com.ibm.commerce.marketing.commands.elements.Marke

Interfaces: com.mycompany.EnterBirthdayTargetTaskCmd

Which method stubs would you like to create?

[ ] public static void main(String[] args)

[ ] Constructors from superclass

[✓] Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

[ ] Generate comments

__ c. Add code to check whether the current user has entered their birthday. If the birthday exists, return true, otherwise return false. To simplify the code for this example, exception handling has been skipped. A screen capture of the completed class is shown below. You can copy the code from the snippets or solution file in <LAB_DIR>/MarketingCustomization1.

```java
package com.mycompany;

import java.util.logging.Logger;

import com.ibm.commerce.foundation.common.util.logging.LoggingHelper;
import com.ibm.commerce.marketing.commands.elements.MarketingCampaignElementTaskCmdImpl;
import com.ibm.commerce.user.objects.DemographicsAccessBean;

public class EnterBirthdayTaskCmdImpl extends
        MarketingCampaignElementTaskCmdImpl implements
        EnterBirthdayTargetTaskCmd {

    private final static Logger LOGGER = LoggingHelper.getLogger(EnterBirthdayTargetTaskCmdImpl.class);
    private final static String CLASSNAME = EnterBirthdayTargetTaskCmdImpl.class.getName();

    public EnterBirthdayTaskCmdImpl() {}

    public void performExecute() {
        final String METHOD_NAME = "performExecute";
        if (LoggingHelper.isEntryExitTraceEnabled(LOGGER)) {
            LOGGER.entering(CLASSNAME, METHOD_NAME);
        }
        boolean rc = false;
        Long memberId = getRegisteredMemberIdForPersonalizationId();
        try {
            DemographicsAccessBean abUserDemo = new DemographicsAccessBean();
            abUserDemo.setInitKey_UserId(memberId.toString());
            abUserDemo.refreshCopyHelper();

            //Return true if the user has entered birthday data
            String birthday = abUserDemo.getDateOfBirth();
            if ((birthday != null) && (!birthday.equals(new String("")))) {
                rc = true;
            }

        }
        catch (Exception e) {/*Exception handling omitted for this example*/}


        //Return true or false for the target
        setReturnValue(rc);

        if (LoggingHelper.isEntryExitTraceEnabled(LOGGER)) {
            LOGGER.exiting(CLASSNAME, METHOD_NAME);
        }
    }
}
```

__ d. Save your work.

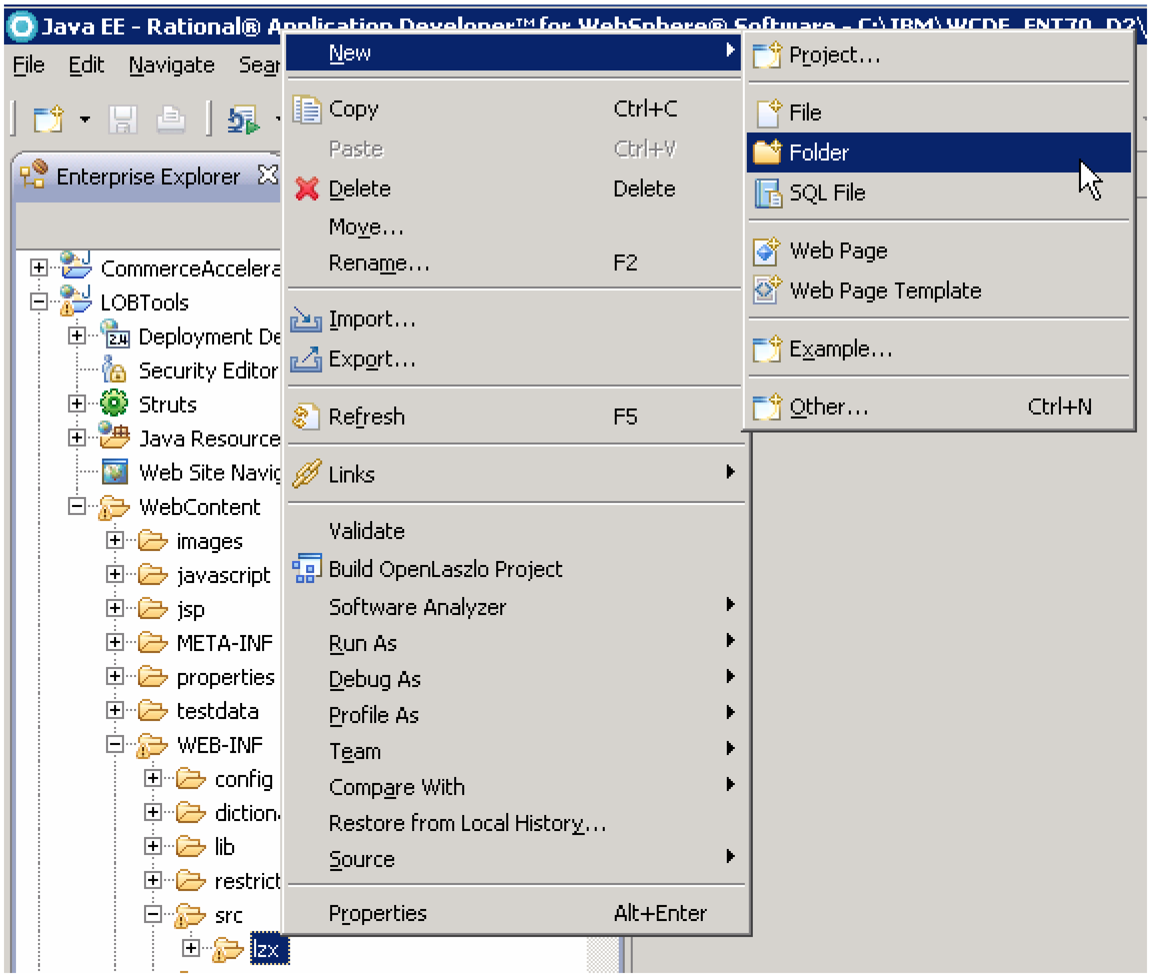
Part 3: Add the new target to Management Center

In this part of the lab, you will update the Management Center activity builder to support the new target.

____ 1. Create a new directory structure for your custom code.

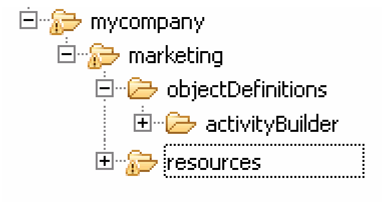
__ a. Open WebSphere Commerce Developer and navigate to the **LOBTools** project.

__ b. Expand the **LOBTools** project directory until you reach the **lzx** directory as shown below. Right click the **lzx** folder and select **New > Folder.**



__ c. Create a folder named **mycompany**. All of your custom OpenLaszlo file will be stored under this directory.

__ d. Repeat steps b. and c. to create the directory structures **mycompany/marketing/objectDefinitions/activityBuilder** and **mycompany/marketing/resources.**



____ 2. Create a new OpenLaszlo file for the target element object definition.

__ a. Right click your new **activityBuilder** directory and select **New > File.**

__ b. Name your new file **EnterBirthdayFlowElementObjectDefinition.lzx**. Click **Finish** to create the file.

____ 3. Create the object definition. Triggers, targets and actions are all activity builder flow elements. When you add a new flow element it must extend the class `mktFlowElementObjectDefinition.`

__ a. Add the following code to the file EnterBirthdayFlowElementObjectDefinition.lzx. Notice the value of the `objectType` attribute and the `elemTemplateName` tag. It is the same as the value entered in the NAME column of DMELETEMPLATE in Part 1. The `displayName` attribute would typically point to a resource string. Text is used here to simplify the exercise. Notice the `elemTemplateType` attribute is set to Target.

```
<library>
      <class name="extEnterBirthdayElementObject"
      extends="mktFlowElementObjectDefinition"
      objectType="customEnterBirthday"
      displayName="Customer has entered their birthday"
      headerIcon="customEnterBirthdayTargetHeaderIcon"
      flowIcon="customEnterBirthdayTargetIcon"
      paletteIcon="customEnterBirthdayTargetPaletteIcon"
      elemTemplateType="Target">

            <mktFlowElementCreateService/>
            <mktFlowElementUpdateService/>

            <dataset name="template">
            <elemTemplateName>customEnterBirthday</elemTemplateName>
            </dataset>
      </class>
</library>
```

__ b. Save the file.

____ 4. Define the display icons. In the object definition above, three icon resources are used. You will define those resources now. The header icon is shown in the properties view, the flow icon represents the target in the activity flow, and the palette icon is the miniature version displayed in the activity builder palette.

__ a. Right click the **resources** directory you created earlier and create a new file called **ExtMarketingResources.lzx.**

__ b. Add the following code to the file. To simplify this example, you will reuse the birthday trigger icons for your target icons.

```
<library>
      <resource name="customEnterBirthdayTargetPaletteIcon"
src="../../../commerce/marketing/resources/pal/birthday.png"/>
      <resource name="customEnterBirthdayTargetIcon"
src="../../../commerce/marketing/resources/dgm/birthday.png"/>
      <resource name="customEnterBirthdayTargetHeaderIcon"
src="../../../commerce/marketing/resources/hdr/birthday.png"/>
```

```
</library>
```

__ c. Save the file.

____ 5. Register your two custom OpenLaszlo files with the Management Center Marketing tool**.**

__ a. Navigate to **…/lzx/commerce/marketing** and open the file **MarketingExtenstionsLibrary.lzx.**

__ b. Add the names of the two files you just created.

```
<include
href="../../mycompany/marketing/objectDefinitions/activityBuilder/Ent
erBirthdayFlowElementObjectDefinition.lzx"/>
<include
href="../../mycompany/marketing/resources/ExtMarketingResources.lzx"/
>
```

__ c. Save the file.

____ 6. Create an instance of the target object**.**

__ a. Navigate to **…/lzx/commerce/marketing/objectDefinitions/activityBuilder** and open the file **FlowPathElementObjectDefinition.lzx**.

__ b. Scroll to the bottom of the file and add an instance of the `extEnterBirthdayElementObject` class to the bottom of the list of element objects. Enter the following line of code.

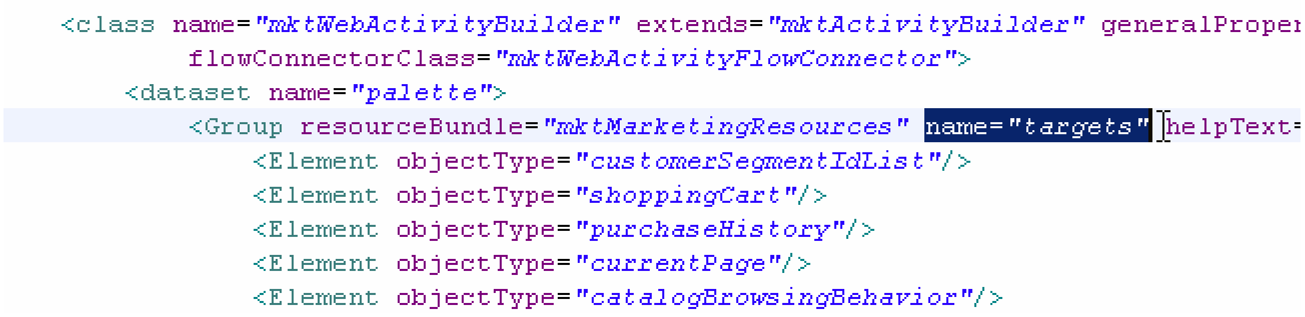
```
<extEnterBirthdayElementObject/>
```

__ c. Save the file.

____ 7. Add the target to the activity builder palette.

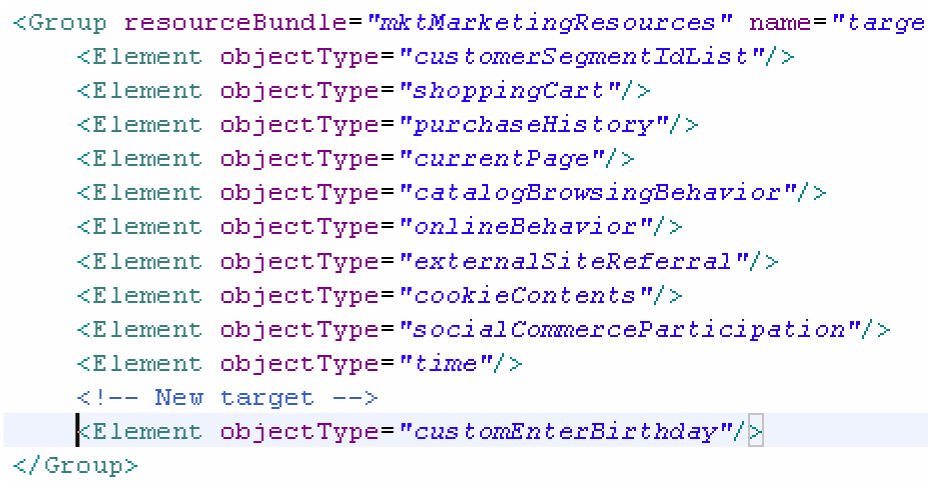
__ a. Navigate to **…/lxz/commerce/marketing/propertiesViews** and open the file **WebActivityBuilder.lzx.**

__ b. Scroll through the file until you locate the targets group.



```
<class name="mktWebActivityBuilder" extends="mktActivityBuilder" generalPropeı
        flowConnectorClass="mktWebActivityFlowConnector">
    <dataset name="palette">
        <Group resourceBundle="mktMarketingResources" name="targets" helpText=
            <Element objectType="customerSegmentIdList"/>
            <Element objectType="shoppingCart"/>
            <Element objectType="purchaseHistory"/>
            <Element objectType="currentPage"/>
            <Element objectType="catalogBrowsingBehavior"/>
```

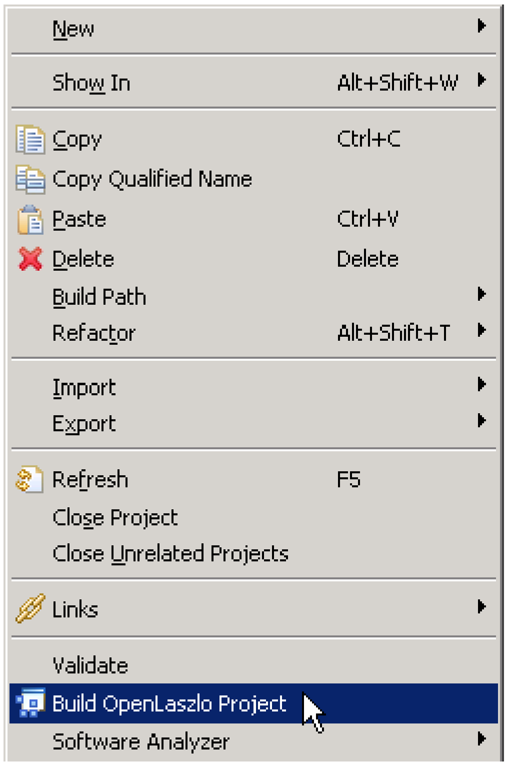
__ c. At the bottom of the targets group, add a new element with `objectType` **customEnterBirthday**. The updated group should look like the following screen capture.



```
<Group resourceBundle="mktMarketingResources" name="targe
    <Element objectType="customerSegmentIdList"/>
    <Element objectType="shoppingCart"/>
    <Element objectType="purchaseHistory"/>
    <Element objectType="currentPage"/>
    <Element objectType="catalogBrowsingBehavior"/>
    <Element objectType="onlineBehavior"/>
    <Element objectType="externalSiteReferral"/>
    <Element objectType="cookieContents"/>
    <Element objectType="socialCommerceParticipation"/>
    <Element objectType="time"/>
    <!-- New target -->
    <Element objectType="customEnterBirthday"/>
</Group>
```

__ d. Save the file. You have now updated the palette for the Web activity builder. You can also add this target to the Web activity template builder, the dialog activity builder and the dialog activity template builder. Each builder definition is in a separate file in this same directory.

____ 8. Compile the Management Center project. Right click the **LOBTools** project and select **Build OpenLaszlo Project.**



____ 9. Update the Struts extension file. In order to read the new target information from the database and correctly display it in Management Center, you need to add a new action to the Struts configuration file.

__ a. Navigate to **LOBTools/WebContent/WEB-INF** and open the file **struts-extension.xml.**

__ b. Click the **Source** tab.

__ c. Within the `<action-mappings>` block, add the following new action.

```
<action path="/SerializeActivityElement-customEnterBirthday"
include="/jsp/commerce/marketing/restricted/SerializeActivitygenericE
lement.jsp" />
```

__ d. Save the file.

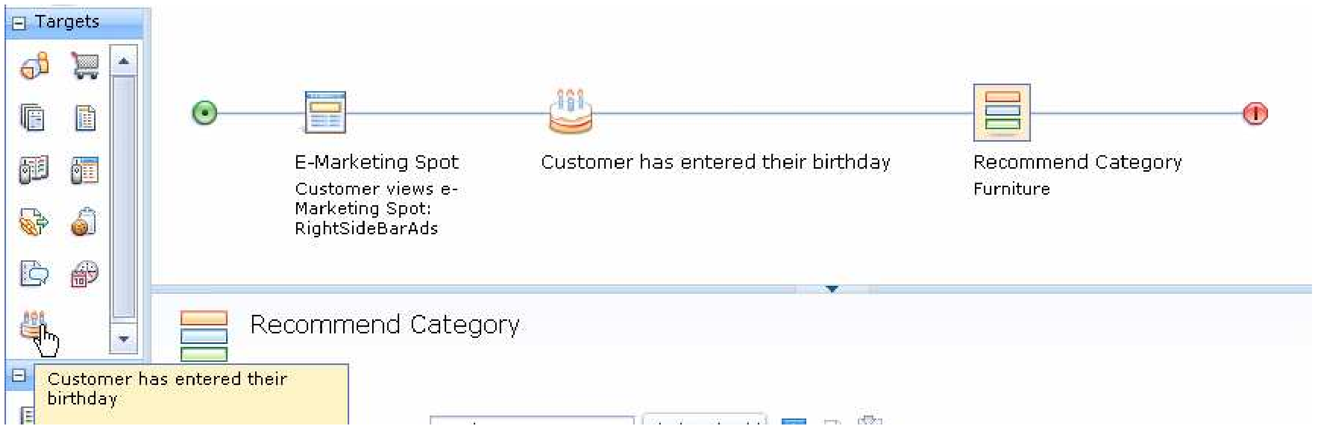
__ e. Start your test server. If it is already running, restart it.

Part 4: Test your new target

In this part of the lab, you will test the changes you have made.

____ 1. Launch Management Center using the URL **https://<WCDE_HOST>:8000/lobtools**.

____ 2. Create a Web activity using your new target. Save and activate the activity.


Targets
E-Marketing Spot
Customer views e-Marketing Spot: RightSideBarAds
Customer has entered their birthday
Recommend Category
Furniture
Recommend Category
Customer has entered their birthday

____ 3. Launch the Madisons starter store using the URL **http://<WCDE_HOST>/webapp/wcs/stores/servlet/Madisons/index.jsp**

____ 4. Register a new shopper but do not supply birthday information.

____ 5. Navigate to the page where your test Web activity is running. Confirm the shopper does not see the marketing activity.

____ 6. Return to the My Account page and enter a birth date.

____ 7. Navigate back to the page with the Web activity and confirm the shopper is now being targeted for the marketing activity.

Part 5: What you did in this exercise

In this tutorial you learned how to create and test a new marketing target element.

You should now understand how to complete the following tasks:

- Define and register a marketing element implementation template
- Implement a task command to evaluate a target condition
- Add a new target to the Management Center Web activity palette