

Lotusphere 2011 Lotus Solutions Development Lab

Lab 2

Creating and deploying a Java plugin for Notes
and Symphony

Part 3: Deploying the Java plugin using a
Composite Application



Introduction

There are many options to deploy a plugin. The Composite Applications framework includes a deployment model where required plugins are installed or upgraded.

Objective

This lab will explain the following tasks:

- How to develop an Eclipse feature and update site.
- How to create a Composite Application using a view from feature

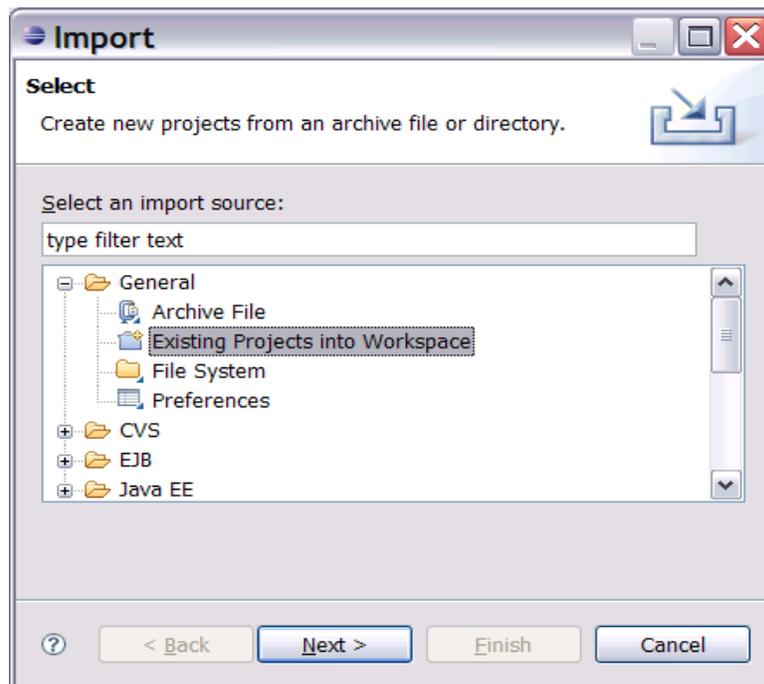
Procedure

Step 1: Launch Eclipse and import the plugin

Note: You can skip this step if you completed it in part 2 of this lab.

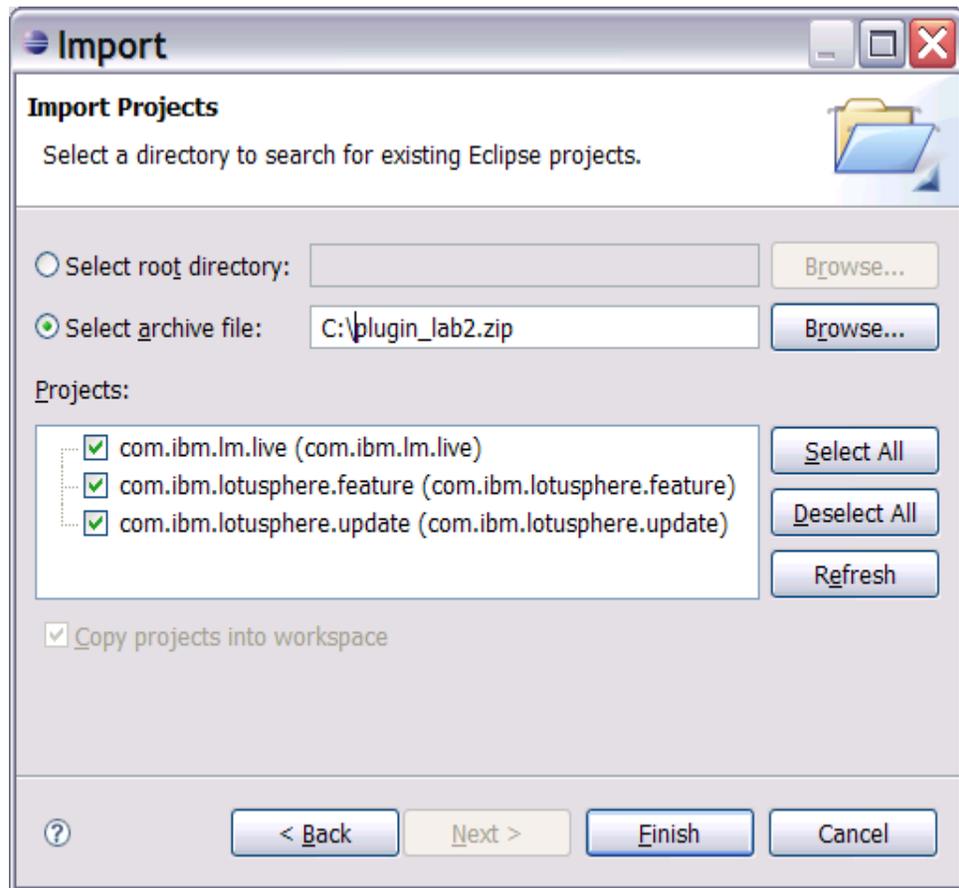
a) Launch Eclipse

b) Import the plugin, feature and update site using File->Import. Select Existing Projects into Workspace and press Next





- c) Select the archive file in [c:\plugin_lab2.zip](#)
 Select all the project and press Finish



Step 2: Plugins are included in a feature

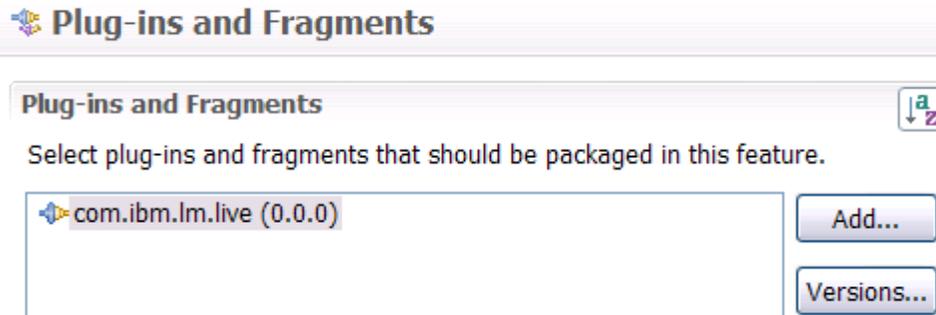
- a) Open feature.xml in the feature



On the Overview tab you can change the version after making a change. The CA can define which version to install, providing an upgrade scenario.
 Note: When a feature is installed, all plugins in the feature will be installed.

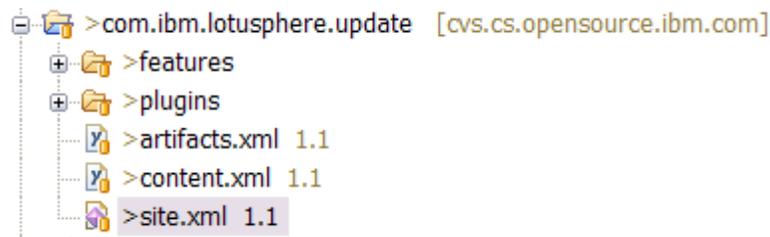


b) The plugins tab lists the included plugins



Step 3: Features are packaged in an update site

a) Open site.xml in the update site



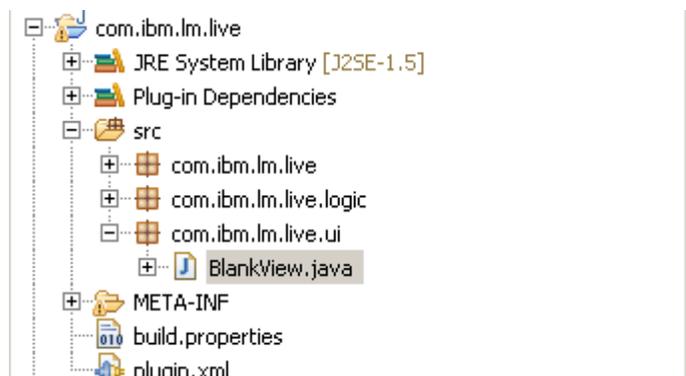
b) Features must be included in a category. In this lab, we use the lotusphere category.



The *Build All* button will refresh the update site. Press the *Build All* button and then explore C:\Documents and Settings\Administrator\workspace\com.ibm.lotusphere.update

Step 4: Build the component

- a) Composite Application can only include components which are Eclipse views
- b) Open BlankView.java, this view will be added to the CA.

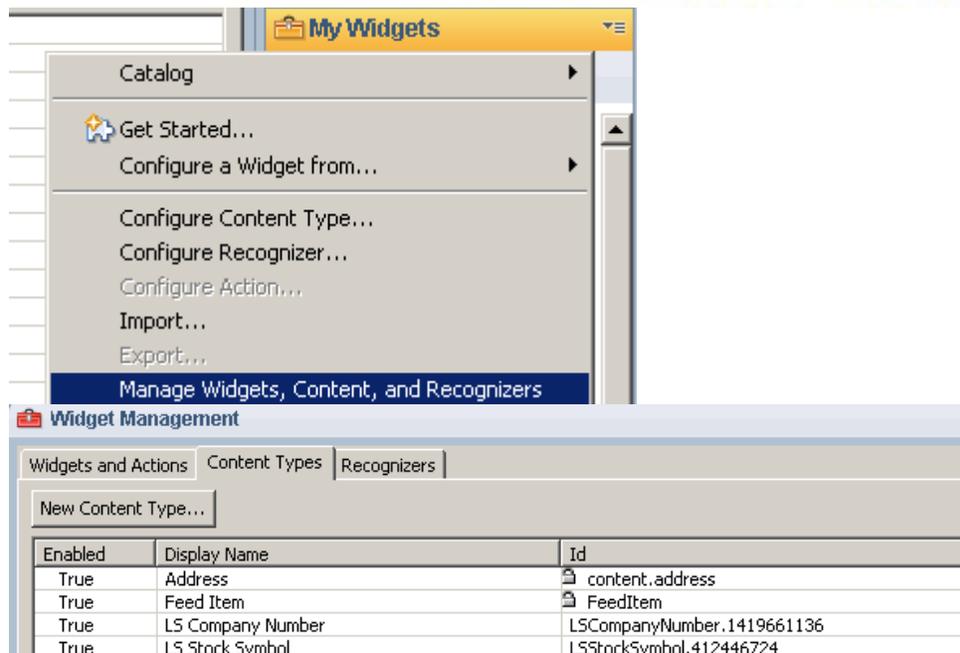


The Text control contains a message for the user. This view will be displayed when the CA is opened. It can provide feedback; informing the user that the plugin has been installed and additional instructions. For example, it could contain a link to the feature's help page.

Note: If you change BlankView.java, you will need to rebuild the update site again (Step 3).

- c) Before creating the CA, confirm the content type value in the plugin.xml matches the content type ID for the Live Text. This was covered in the *Create a Java Plugin to look up Customer ID* Lab (The steps have been included below as a reference).
- d) Launch Notes to view the required Ids in the Widget Management page.





Note: This value is different per instance. In the example above, the content.type value is DCCT.LSStockSymbol.412446724.

e) Associate the plugin with the content type. Open plugin.xml and update content.type for the Display Company Extension

Extension Element Details

Set the properties of "objectState". Required fields are denoted by "*".

name*:

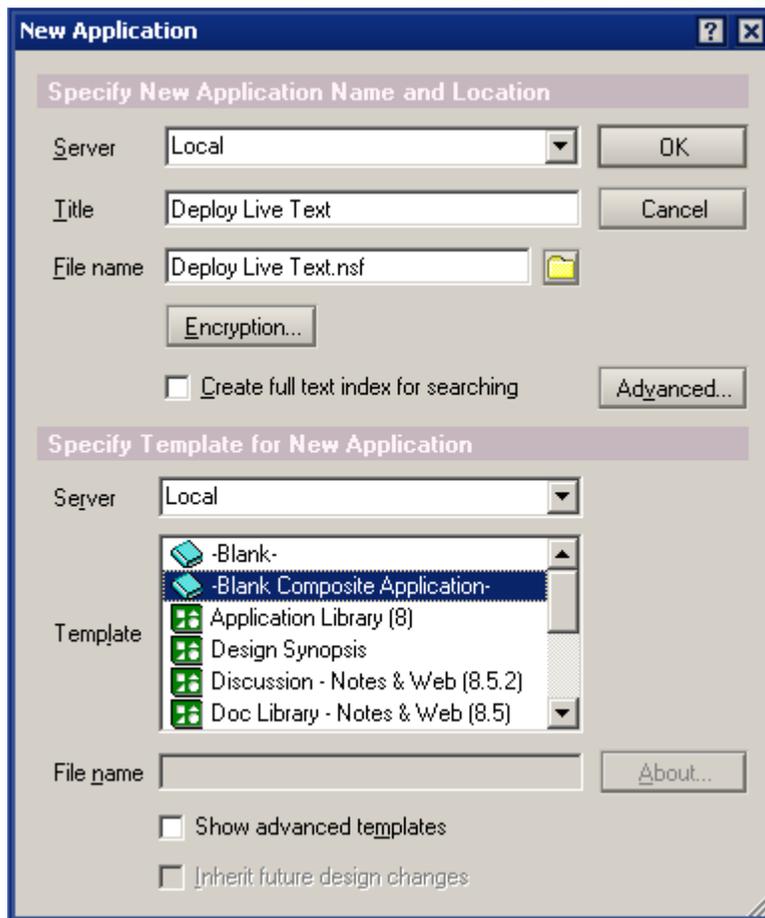
value*:

f) If you update the plugin, rebuild the update site using the *Build All* button.

Step 5: Build the CA

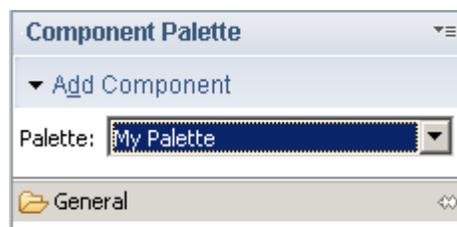
a) From within Notes, create the CA using File → Application → New and fill in the following values:



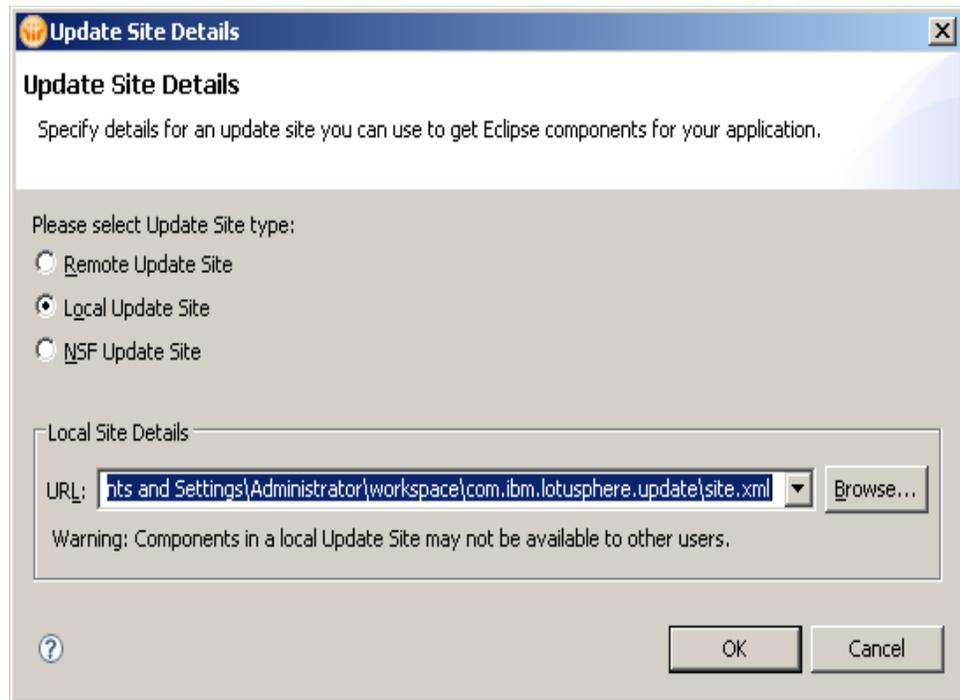


b) Launch the Composite Application Editor because the CA is empty. This is done with the Actions menu (Action → Edit Application)

c) Select My Palette from the the Component Palette on the right.

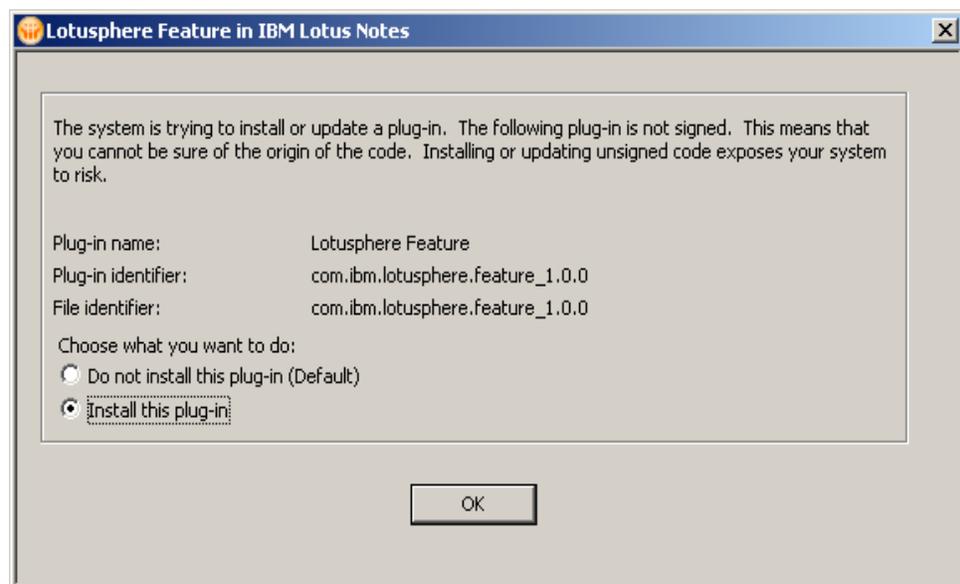


d) Press the Add Component dropdown and select *Add Components from Update Site*. The URL is C:\Documents and Settings\Administrator\workspace\com.ibm.lotusphere.update\site.xml



Select the *Welcome* component, keep the Category setting as *General* and press OK

e) Drag the *Welcome* component from the palette on the right to the main CA area. When prompted, install the plugin,



f) Restart when prompted. If asked to Save the changes, respond Yes.

g) The Lead Manager composite application will restart

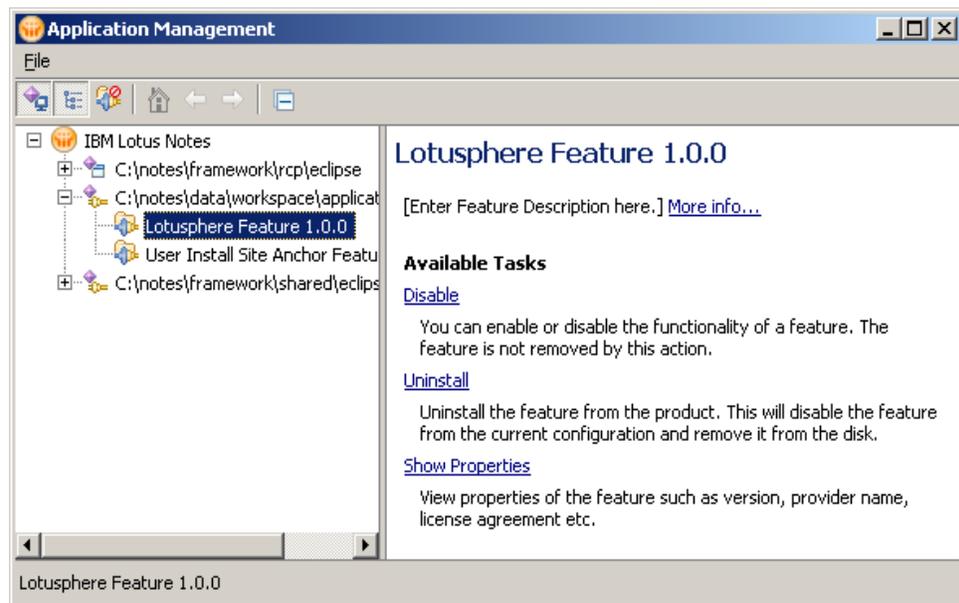
Step 6: Test

a) As a side-effect of building the CA, the plugin is installed. To test, you must first uninstall the plugin. This is accomplished using the File menu → Application → Application Management...

Note: The Application Management menu will not be available in default Notes installations. You will need to enable it by shutting down Notes, and modifying the c:\notes\framework\rcp\plugin_customization.ini file to include the line:

```
com.ibm.notes.branding/enable.update.ui=true
```

Save the file, restart Notes, and then access the Application Management menu.



b) Select the Lotusphere Feature 1.0.0 on the left, then click [Uninstall](#) on right. Confirm uninstall, then restart when prompted.

c) Open the Deploy Live Text CA



Lotus. software



You will be prompted to install the plugin and to restart

d) Open the *Sample account numbers* document in Lab2 Discussion NSF and validate the Live Text is recognized.



Summary:

In this lab you learned how to create a composite application and add an Eclipse view. When the CA is opened, all the plugins in the feature containing the Eclipse view will be installed. The code for Live Text is packaged with the view and will be installed. Displaying information or a link in the Eclipse view for user feedback is an extra benefit of using a CA.

References:

[How to Write an Eclipse Component in 40 minutes or less](#)

