**PSG-DEV**

## Installed Software

JRE 1.4.2

LMe 6.1a

LMe ‘Right Click’ functionality

JDK 7.0 (Java Development Kit)

Eclipse Helios (For demonstrating Plants in Subversion)

Subclipse Plugin

## Demo Application Instructions

### DataSearch Application

DataSearch is a small employee records application written in VB, which connects to an SQL database. It is built using MSBuild, initiated from Jenkins, then deployed and installed onto the Test and Prod servers. The instructions to demo this application are given below.

Open up a Terminal Services session to the following servers;

PSG DEV

PSG BUILD

PSG DBSVR

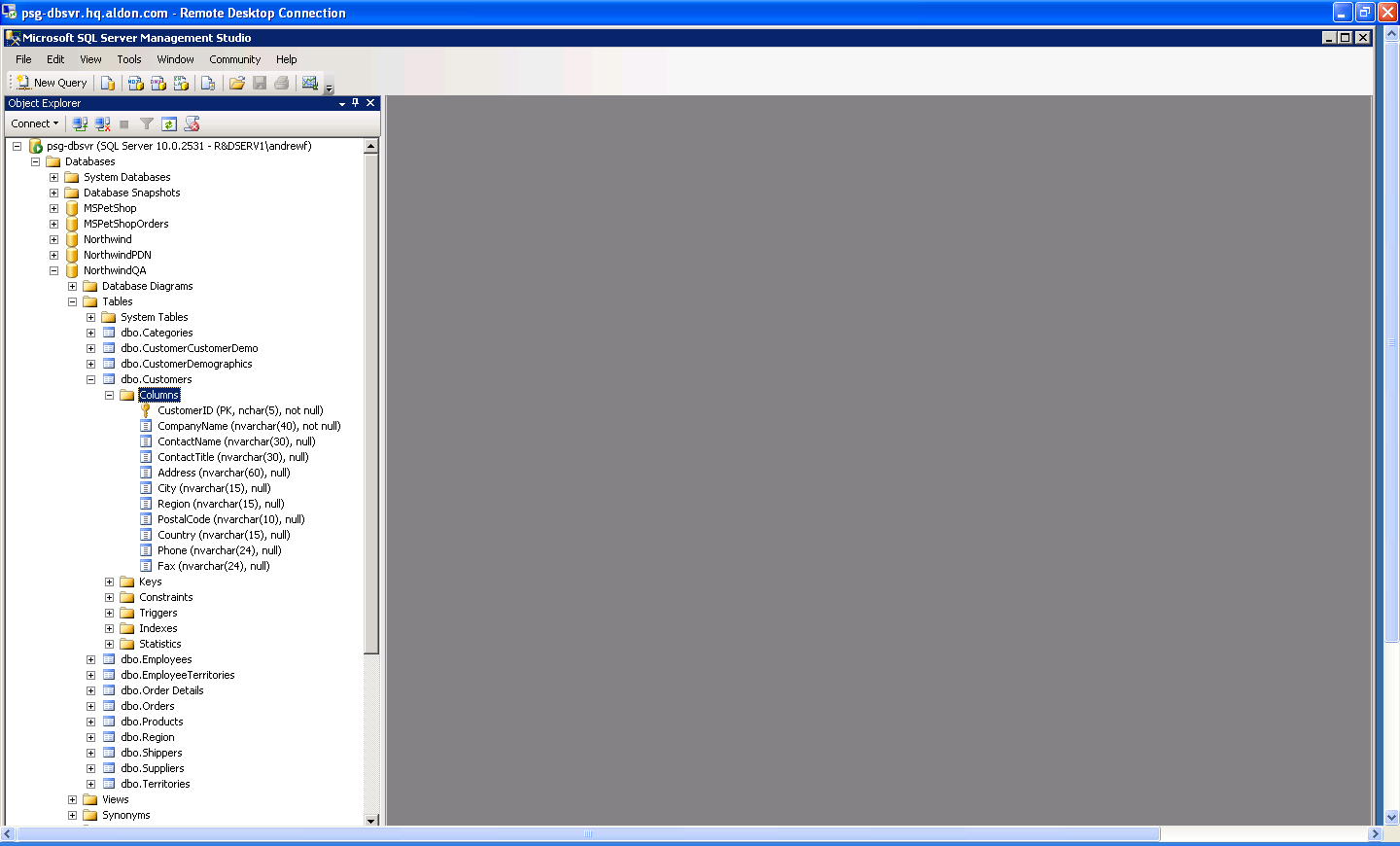
PSG TEST

PSG APPSVR

Open the following Applications on the servers;

PSG DEV : LMe GUI, Visual Studio 2010 (Project - DataSearch)

PSG DBSVR : Microsoft SQL Server Management Studio. Go into Databases, and expand the ‘NorthwindQA’ and ‘NorthwindPDN’ Databases, as below;



PSG TEST : Just leave the server displaying the Desktop

PSG APPSVR : Just leave the server displaying the Desktop

Open up a browser session to the JENKINS Build Server;

<http://psg-build:8080/>

Open up a browser session to the Deployment Web Portal;

<http://psg-lm.hq.aldon.com/aldonlmw/login.php?return_uri=%2Faldonlmw%2F>

1. Create an Issue in CM (\_DEVELOPMENT MANAGEMENT PROJECT) – Pass through workflow so that it is assigned to a Developer, in a status of ‘Development’ . ***Make sure the XD box is ticked in the ALM tab.***
2. Create a Task in LMe, in the Personal Banking\Customer Master\DataSearch Application using the CM Issue number as the name of the Task.
3. On PSG DESKTOP, in Visual Studio 2010, go into the DataSearch Project.

Checkout the following files;

ControlFile.txt

DataSearchForm.vb

NorthwindDataSet.xsd

Before you know which database script to Checkout, first check the Database, to see if it requires a DROP or an INSERT. Go onto PSG-DBSVR, and look at the Databases, and see if the ‘EMAIL’ Column is there. If it is, Checkout the **northwind\_customers\_DROP\_script.sql**. If the ‘EMAIL’ Column is not there, checkout the **northwind\_customers\_ADD\_script.sql**.

1. (**VB Development**)

Update the ControlFile.txt as follows;

On the first line, type the name of the Task created in LMe above.

On the second line, type the CM Issue Number created in CM above.

Right Click on NorthwindDataSet.xsd and choose ‘View Designer’. This will open up on the main screen. If you are dropping the ‘Email’ column, right click it and choose ‘Delete’. If you are adding the ‘Email’ column, right click anywhere in the top box and choose ‘Add – Column’ and give it the name ‘Email’. Close and save the changes.

Double click the DataSearchForm.vb, and it will open up in the main window. Click on the ‘City’ Field in the centre of the form, and it will select the complete grid. Use the keyboard to delete this grid.

On the right hand side of the screen, choose the ‘Data Sources’ Tab.

Drag the ‘Customers’ data set onto the form on the right. Drag to expand the grid to use all of the form. Close and save the form. On the right hand side select the ‘Solutions Explorer’ Tab.

Checkin the files into LMe.

This will move everything from Development into STAGING Environment, which can be seen using the LMe GUI.

1. Select the Task in LMe, and do a Promote. The Promote will deploy all of the files to the PSG BUILD Server.
2. Show the files being deployed using the Web Portal. Please note, it also deploys the control file over to the PSG-CM-NEW server, which is used later on in this process.
3. This deployment then starts the Jenkins Build job, so show the build being started using the Jenkins browser window.
4. The Jenkins build calls the build script, which basically does the following;

Get latest of the project

Overwrite the project files with the newly deployed files.

Obtains the correct APP.CONFIG file which is used to point the Application to the correct database.

Runs the SQLCMD.exe utility to update the TEST database on PSG-DBSVR.

Uses MSBUILD to build the application.

Imports the build result (.EXE), associated files back into LMe.

1. Once the Jenkins build completes, it will start the IMPORT into LMe, which can be shown from the Web Portal window. This deployment will do the following;

* Import into Build Release ***PersonalBanking\CustomerMaster\BuildQUA***
* Deploy the files to PSG TEST, into **C:\ProgramFiles\Aldon\TESTEnvironment**
* Initiate the post deployment script which will read through the ControlFile.txt file which has just been deployed, and send an email using BLAT into CM, which will update the existing Issue number with a description that the testing has completed, attach a TestCase document (which is located in C:\ProgramFiles\Aldon\TESTEnvironment\Scripts) and change the status to **PASSED QA.**
* This then invokes an Escalation in CM, and changes the status to ‘**PRODUCTION READY’**.
* This then invokes an Approval in CM.
* You can show the deployed application on PSG-TEST by double clicking the ‘DataSearch’ icon on the desktop. You can prove that the application has been built pointing to the TEST database by clicking on ‘**Show all companies’** and showing that the data in the **‘Company Name’** field has **TEST** in it.

1. To move to Production, either;
2. Approve the Email sent by CM, and it does the same as above, but sends the application to the production server, PSG-APPSVR.
3. Promote the Task as previously done above.

# ‘PlantsByWebsphere’ Application in SUBVERSION

This demo uses the PlantsbyWebsphere application , which is stored in the Subversion repository.

***Server locations;***

The SVN Repository is installed on PSG-DBSVR, in C:\Program Files (x86)\WANdisco\uberSVN. The URL is <http://PSG-DBSVR.hq.aldon.com:9880/psg-svn>

The LMe Client is also installed on this machine. The Development machine is PSG-DEV, Using SubClipse Plugin for SVN within Eclipse IDE.

***User Setup (SVN and LMe)***

User creation is performed by an SVN administrator. Access the Admin logon page here <http://psg-dbsvr.hq.aldon.com:9890/ubersvn/views/platform/shared/welcome.jsf> .

Log in using the following details;

User ID: admin

Pwd: admin1

Create a Userid in SVN which matches the same Userid defined in LMe/Security Server. Please note, the passwords defined for the SVN user and LMe user do not have to match.

***Encrypt the LMe Passwords for the SVN Developers***

On the PSG-DBSVR Machine, do the following;

Open up a command prompt using ‘Run as Administrator’

CD to the LMe install directory

Execute the following command;

aldonsvn.exe --encrypt LM(e)\_username

where LM(e)\_username is the SVN developer’s registered log-in name for LM(e).

***Configuring a Project***

This only needs to be performed if you are creating adding a new project to SVN and LMe. If you are using an existing SVN project, skip to ‘Adding a project into Eclipse’.

On the PSG-DBSVR machine, navigate to the ***aldonsvn.conf*** file, located in the LMe installation directory.

Locate the section named svn\_lme\_map, and specify the name of the SVN repository and project, with the corresponding Group/App/Release defined within LMe. E.g;

[svn\_lme\_map]

psg-svn:trunk\PlantsbyWeb = psg-lm:Retail/eCommerce/PlantsByWeb(1.00)::$PlantsByWeb

***Adding a project into Eclipse***

The development in this demo application is performed from within Eclipse, using the Subclipse plugin, which enables users to work on files, and commit them back to the SVN repository, directly from within Eclipse.

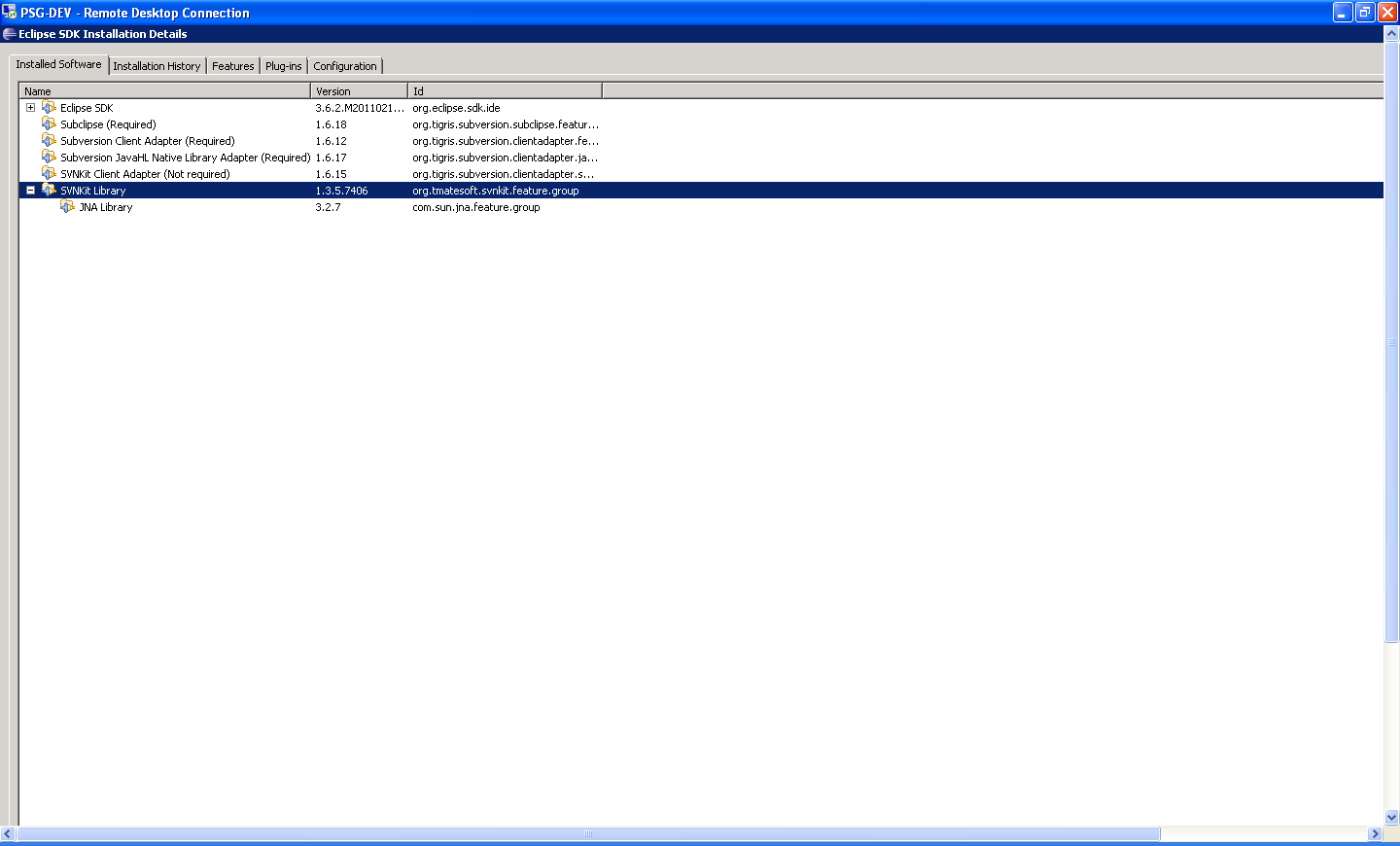
***Install Subclipse Plugin (if necessary)***

Open Eclipse Helios at c:\program files\eclipse\eclipse.exe on the PSG-DEV machine. To see if the Subclipse plug-in already installed, select HELP – INSTALL SOFTWARE, What’s Installed. If the plugin is already installed, it will be listed. If the SVN Subclipse Plugin has already been installed onto this machine, skip to ‘Select or create a new workspace’.

If not, click HELP, INSTALL SOFTWARE. Add the following link;

Subclipse 1.6.x Update Site - <http://subclipse.tigris.org/update_1.6.x/>

Install the following components;



**Select an existing, or create a new Workspace.**

First, add the SVN repository to your workspace. To do this, select WINDOW – Open Perspective – Other – SVN Repository Exploring. Open up the perspective, then in the window on the left hand side, right click and select New – Repository Location.

In the ‘Add SVN Repository’ box, type in the URL of the SVN repository, and select ‘Finish’. E.g;

[**http://psg-dbsvr.hq.aldon.com:9880/psg-svn/**](http://psg-dbsvr.hq.aldon.com:9880/psg-svn/)

***If you are prompted to sign-on to the SVN repository, you must enter the User Name that is configured in Subversion, which must also match your UserID that is set in Aldon LMe. You can ignore the error that warns of “incompatible JavaHL”.***

Now go back into the ‘Java’ perspective window, and select File, Import.

Select SVN – Checkout Projects from SVN.

Select the correct SVN repository, and click Next, then go into the ‘Trunk’ folder, and select the ‘PlantsbyWeb’ project.

This will add the project into the workspace.

You may now work on the required files.

Open up the ‘Plantsbywebsphere’ project in the Java Perspective. Drill down through the following folders;

PlantsbyWeb

Plantsbywebspheresample

Src

PlantsbywebsphereWeb

WebContent

Double click on PROMO.HTML to open it up, and edit the following lines of code – basically, just change the ‘Prepare your pond for Spring’ to ‘Prepare your pond for Summer’;

<td bgcolor="#DCEBCD"><p class="tips">Tip of the day:: **Prepare your pond for Spring.** Preserve extra grass seed by keeping it dry.

Tape boxes and bags closed, or seal them into plastic bags. Be sure to remove extra air

from the bags. Store all seed in a cool, dry area such as a garage or basement.</td>

</tr>

Save the changes.

Right Click and select TEAM – COMMIT.

This will Commit the change back to the Subversion repository, and also add the source files into the LMe G/A/R – Retail/ecommerce/Plantsbyweb.

Deployment is configured to automatically deploy the files to the PSG-BUILD server once you perform the ‘Commit’. This then starts the build job, builds the application, and then imports the build result back into LMe, and deploys it to the PSG-TEST server, and installs it onto the Websphere Application Server. This can be seen by browsing to the following link;

<http://psg-test:9080/PlantsByWebSphere/>

Show the change which you made in Eclipse is now running live on the WAS. Once you are ready, go into LMe, and Promote the files to send them to the production environment. This basically does the same thing again, sends the files to the build server, builds the application, imports and sends the build result to the production system, PSG-APPSVR. This can be seen by browsing to the following link;

<http://psg-appsvr:9080/PlantsByWebSphere/>

# ‘PlantsByWebsphere’ Application in LMe

This demo uses the PlantsbyWebsphere application , which is stored in the LMe repository.

***Server locations;***

The LMe Repository is installed on PSG-LM. The LMe client is installed on the Development machine, which is PSG-DEV. The Eclipse IDE to be used is Rational Developer for Power Systems.

**Open the PlantsbyWebshpere Workspace.**

Open Rational Developer for Power Systems IDE.

Open up the ‘Plantsbywebsphere’ project in the Java Perspective. Drill down through the following folders;

PlantsbyWeb

Plantsbywebspheresample

Src

PlantsbywebsphereWeb

WebContent

Double click on PROMO.HTML to open it up, and edit the following lines of code – basically, just change the ‘Prepare your pond for Spring’ to ‘Prepare your pond for Summer’;

<td bgcolor="#DCEBCD"><p class="tips">Tip of the day:: **Prepare your pond for Spring.** Preserve extra grass seed by keeping it dry.

Tape boxes and bags closed, or seal them into plastic bags. Be sure to remove extra air

from the bags. Store all seed in a cool, dry area such as a garage or basement.</td>

</tr>

Save the changes.

Right Click and select CHECKIN.

This will check the files into LMe G/A/R – Retail/ecommerce/Plantsbyweb, into the first Interim staging environment.

Promote the files, which will deploy the files to the PSG-BUILD server. This then starts the build job, builds the application, and then imports the build result back into LMe, and deploys it to the PSG-TEST server, and installs it onto the Websphere Application Server. This can be seen by browsing to the following link;

<http://psg-test:9080/PlantsByWebSphere/>

Show the change which you made in Eclipse is now running live on the WAS by browsing to the above URL.

Once you are ready to promote to Production, go into LMe, and Promote the files to send them to the production environment. This basically does the same thing again, sends the files to the build server, builds the application, imports and sends the build result to the production system, PSG-APPSVR. This can be seen by browsing to the following link;

<http://psg-appsvr:9080/PlantsByWebSphere/>