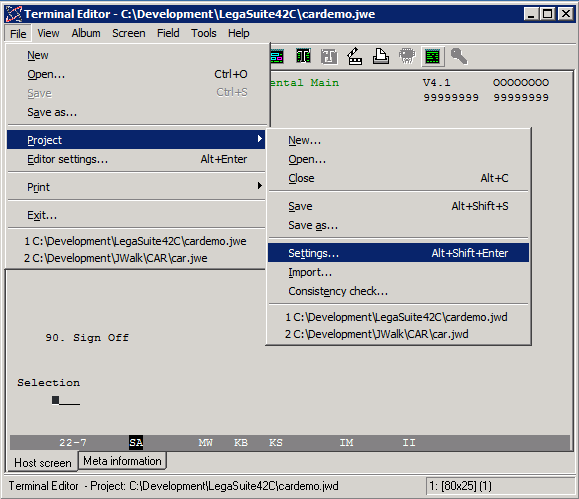
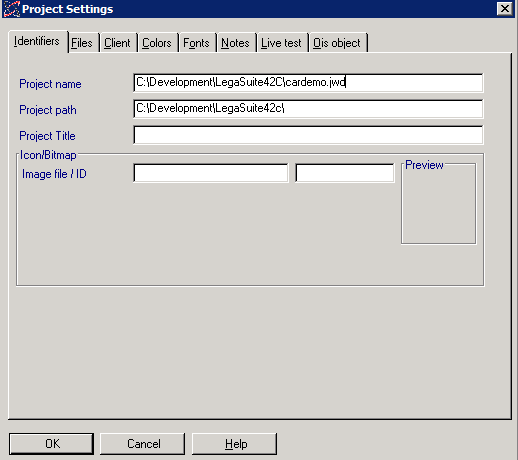
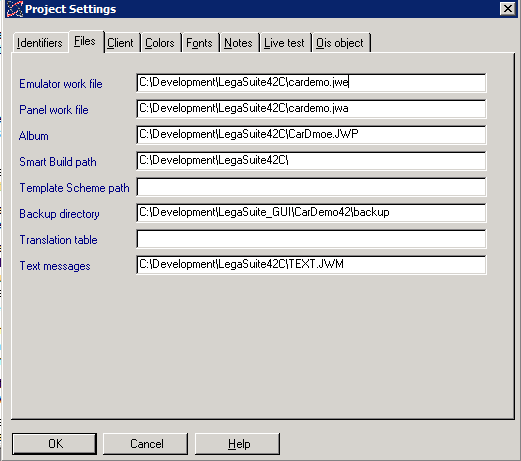
# Managing LegaSuite LegaSuite 4.2C

# with Aldon and Subversion

Last updated: October 24, 2012

# Background:

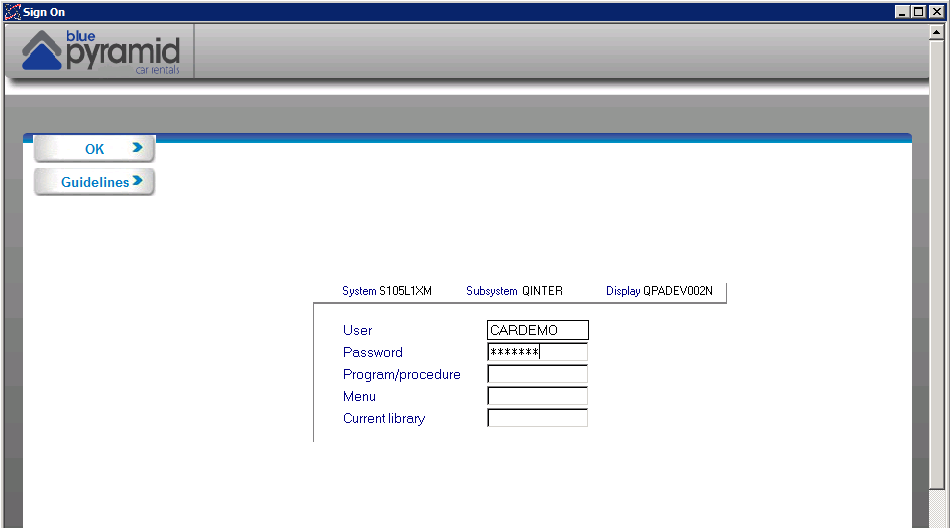
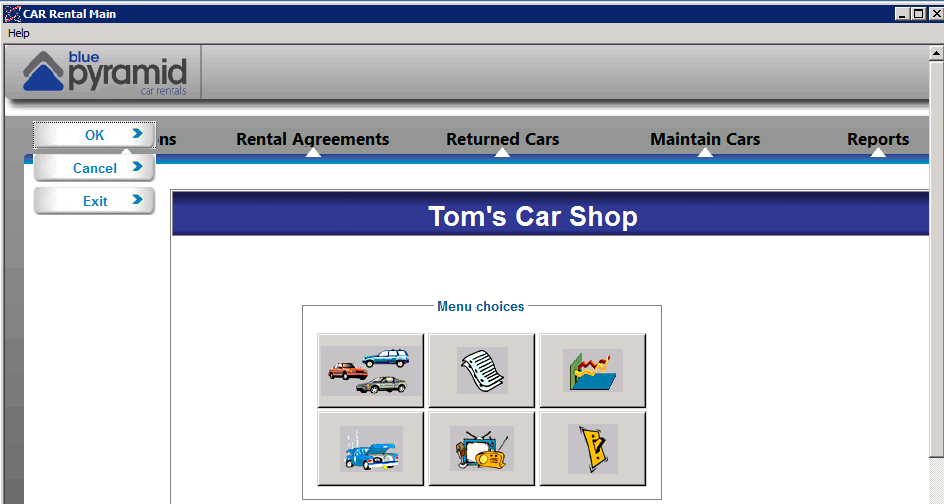
* The environment:
  + Most work is done from PSG-DEV.
  + LMe 6.2 on PSG-LM
  + Version control by SVN and the Aldon SVN connector
  + Car Rental Application on PSG-DEV
    - Development: C:\Development\LegaSuite42c
  + A QA version of the “Car Rental” application is on PSG-TEST.
  + The live version of the Car Rental application is on PSG-APPSVR
  + LegaSuite 4.2C Developer Client on PSG-DEV
  + LegaSuite 4.2C Developer Client on PSG-BUILD to do the automated builds for QUA and PDN
  + LegaSuite 4.2C Windows Client on PSG-TEST and PSG-APPSVR to show the final results.
* LegaSuite Developer
  + This is the programmer’s IDE used to generate and change the gui screens from the 5250 displays.
  + It is a tool set that runs on your PC and helps you build, customize, and enhance GUI panels
* LegaSuite Client
  + This is the solution that will actually show the converted GUI panels.
  + Create shortcuts to the client and changed the properties attribute to point the client to either the QA or the Live target folders.
  + On PSG-TEST and PSG-APPSVR, create a “Cars Demo” shortcut with the following target:
    - "C:\Users\JBaumgarten\AppData\Roaming\SEAGULL\J Walk Windows Client\4.1038.1.759\jw9c.exe" /etelnet /has400148.seagullsoftware.com /glicsp=1291
    - Start in: C:\LegaSuite42C
* Important file extensions:
  + .JWR: This is the file that contains the published changes. It is what needs to be imported, deployed and installed on the target servers to affect the change.
  + .JWD: This is the project file. It is holds all the scripts, objects, components, and screens that make up the application. There is a LegaSuite utility available to “decompress” this file into the individual components.
  + .JWB: A jwalk batch script that can execute jwalk commands in batch mode
* I created a build entry in Jenkins for both Car42CBuildQUA and Car42CBuildPDN. They call their respective build scripts.
* I created a dos batch script that will invoke the publish command to compile the application and import all of it AldonDemo/LegaSuite4/Build(4.2C)
  + See PSG-BUILD, C:\BUILD\LegaSuite42C\QUA\Scripts\Car42BuildQUA.bat
* I created a LegaSuite batch script (.jwb) that is invoked by the build script. It enables the publish (creation of the .jwr package) process happen in batch mode. This is very handy for those shops that have separation of duties on who can do the final create, import and deployment.
  + See PSG-BUILD, C:\BUILD\LegaSuite42C\QUA\Scripts \cardemo.jwb
  + See PSG-BUILD, C:\BUILD\LegaSuite42C\PDN\Scripts\cardemo.jwb
* To invoke the running application on PSG-TEST and PSG-APPSRV, you need to configure the client to know where to find the project and the files.
  + 
  + 
  + 

# Pre-Demo

1. Start up LMe 6.2 and set defaults to AldonDemo/LegaSuite4/
2. Open up browsers for the LMe web portal and Jenkins (LegaSuite 4.2C tab).
3. Start up sessions to PSG-TEST and PSG-APPSVR that have shortcuts on the desktop that launch the respective Cars Demo application for that environment.
4. \*\*IMPORTANT\*\* Do not leave the Cars Demo application up and running while you do the demo or the new versions cannot be updated on that target!

# Demonstration of Aldon

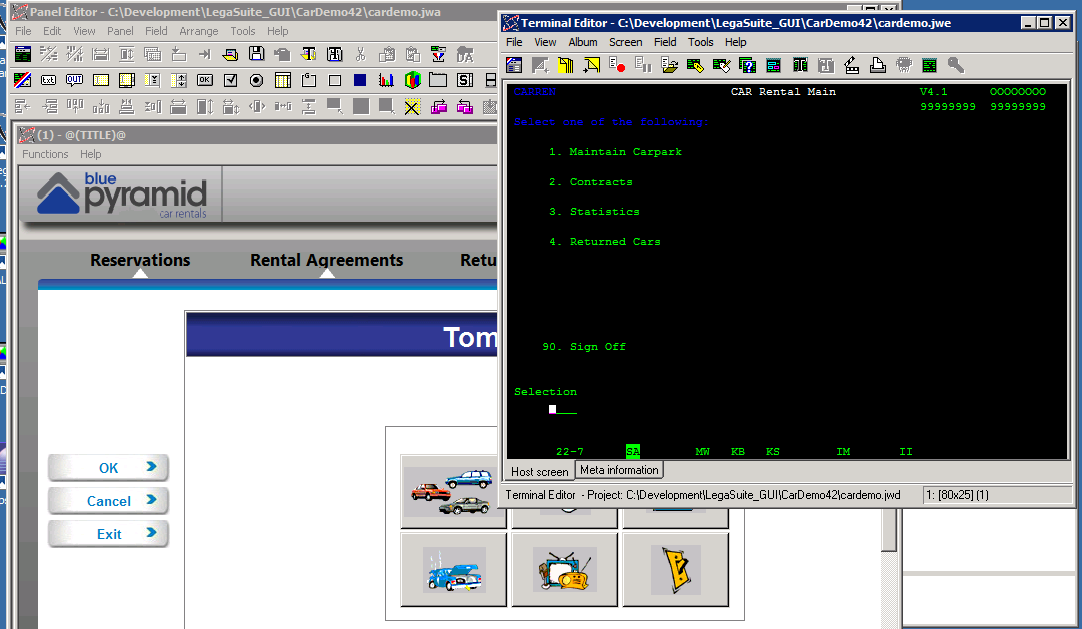
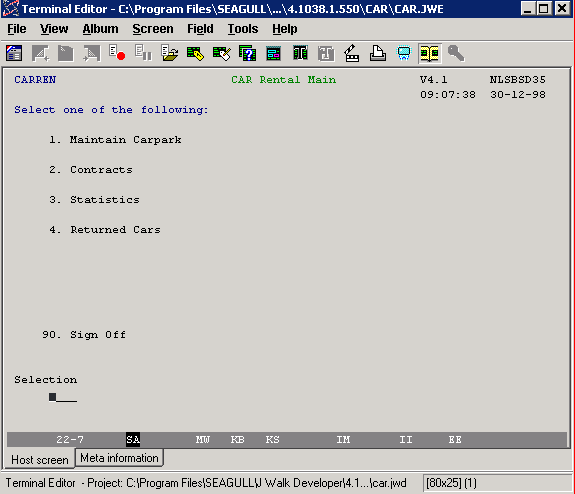
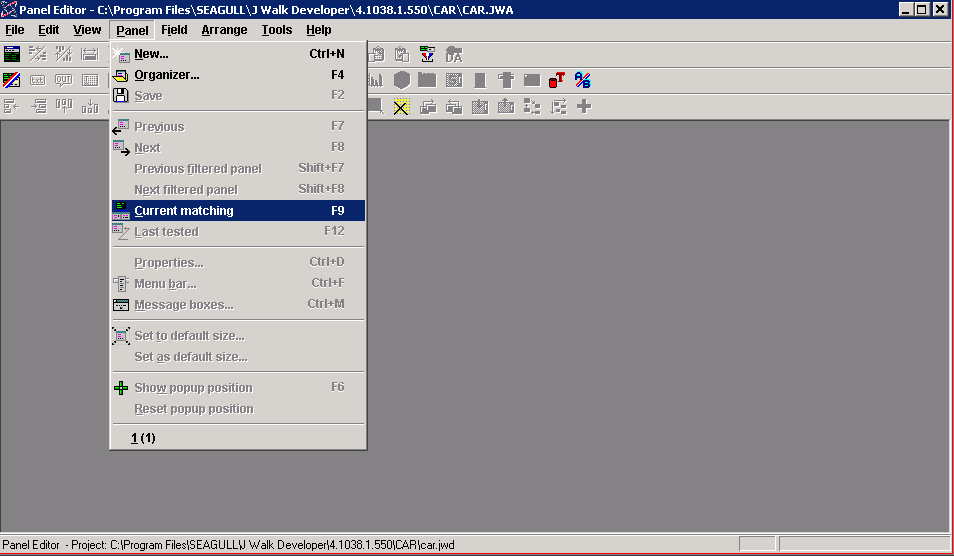
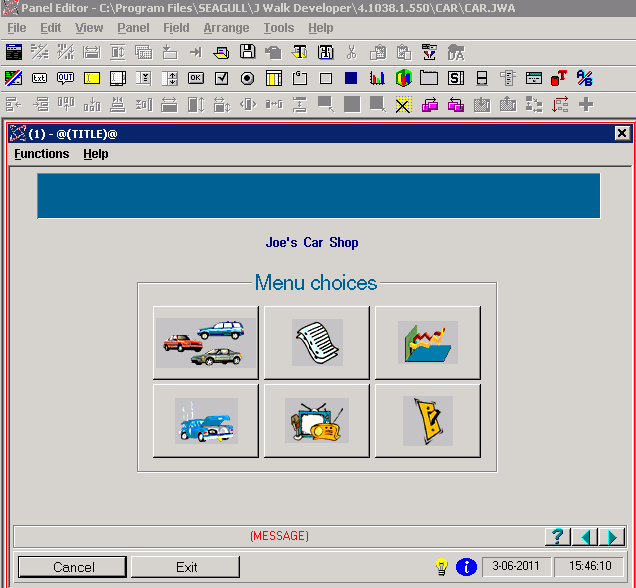
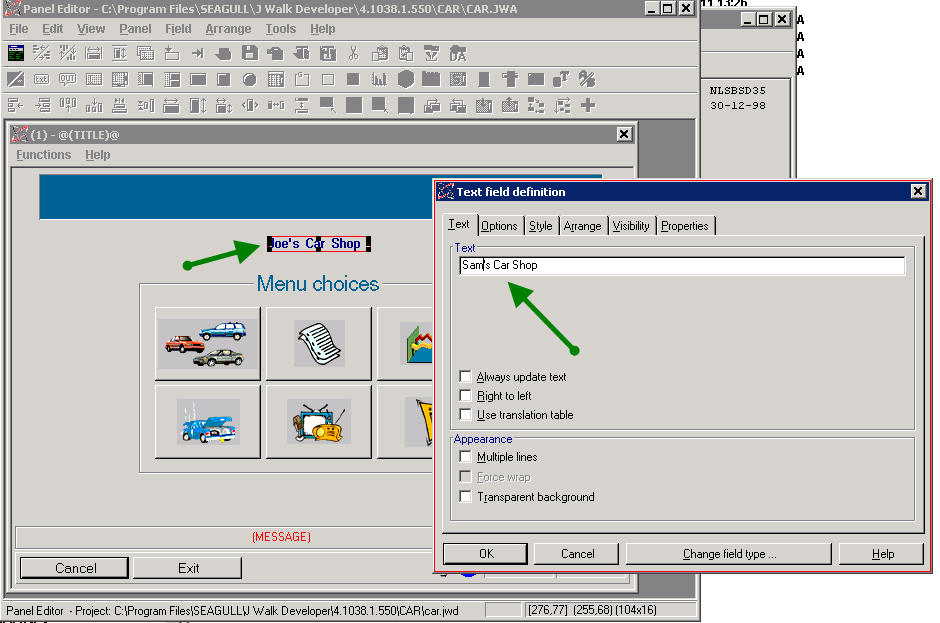
## Step 1: Show what we will be changing:

1. Switch to PSG-APPSVR and DC on the “Cars 4.2C Live” short cut to launch the LegaSuite client program over the live Cars application directory.
2. 
3. On the main panel, type in the user of “cardemo” and the password of “seagull”.
4. Click the OK button with the mouse (pressing enter doesn’t work)
5. We’ll be changing the owner’s name of the Car Rental shop! 
6. Exit out of the application back to the desktop.
7. \*\*IMPORTANT\*\* Do not leave the Cars Demo application up and running while you do the demo or the new versions cannot be updated on that target!

## Step 2: Create the Issue in CM

1. Create issue as end user
   1. From the Customer CM portal create an Issue in CM (\_DEVELOPMENT MANAGEMENT Project)
   2. Title: Change Car Shop Ownership
   3. Type: Enhancement
   4. Product: Car Rental
   5. Version: 2
   6. Urgency: Not Urgent
   7. Set the XD flag to Yes
   8. LMe release is **AldonDemo/LegaSuite4/Base(4.2C)**
2. Approve the request from Agent
   1. Switch to your agent CM tab
   2. See ISSUEnnn in the “Dev Request” status.
   3. Edit the issue, and Link the issue to LM by creating a task of the same name as the issue.
   4. Approve the issue. This will change the status to ‘DEVELOPMENT’ and the assignee to ‘Jon’

## Step 3: Change the Code

1. Return to PSG-DEV.
2. Explain that developers are using Subversion to version control the Car Rental application.
3. Show subversion running over C:\Development\LegaSuite4C
4. Use the short cut in the c:\jwalk menu to launch the LegaSuite 4.2C Developer application over the development directory.
5. You will see: 
6. This is the Car Rental Main Menu that we will be customizing with LegaSuite. Note the screen id in the lower right hand corner. It says (1) in this case. 
7. Now switch over to the Panel Editor. Select Panel, Current Matching. 
8. This shows the converted and customized version of the current 5250 screen. Note the screen id in the upper right hand corner of the menu is (1). 
9. Double click on the text “Joe’s Car Shop”. This brings up the properties of the Text Field. Change the name of the car shop to something new and press OK in the Text field dialog box. 
10. Save your work by clicking Panel, Save.
11. Exit the IDE.
12. Change C:\Development\LegaSuite42C\ControlFile.txt
    1. Change the first line Issue Number
    2. Change the second line Issue Number

# Step 4. Commit via Subversion

1. Browse to C:\Development\Jwalk
2. RC, then Tortoise, then Check for Modifications
3. See (1).JWW, the changed format for Screen 1 and the ControlFile.txt
4. Add a comment of “:ISSUEnnn” and commit.
5. Switch to LMe and show that the (1).JWW and ControlFile.txt have been automatically checked in and promoted up to INT (no deployment at this stage).
6. This automatically triggers the start of the build and deployment process.

# Step 5: Publish, Import, Promote and Deploy to QA in one step!

1. Switch to LMe collections tab.
2. Find the ISSUEnnn
3. Promote all of them from INT to QUA.
4. Switch to the LMe Web Portal and show the deployment of the code over to PSG-BUILD via profile Car42C\_Q\_Bld.
5. This will send the source code over to PSG-BUILD and place it in c:\development\LegaSuite42C and call Jenkins to a do a build and import.
6. Switch to the Jenkins web browser and show the QA job running.
7. This calls a batch script on PSG-BUILD that will
   1. invoke the JWalk publish command to build the .jwr,
   2. import both of them into AldonDemo/LegaSuite4/Build(4.2C) QUA environment
   3. deploy and install the package onto the QA server for testing
   4. Send emails.
   5. Update (via email) the CM ticket to “Passed QA” with updated subject and attachment.
8. Switch over to the LMe Web Portal and show the deployment happening to PSG-TEST via profile Jwalk\_CAR\_Q\_SVR.
9. Switch to Outlook and show emails to CM and Promote folders
10. Now show that we have the new changes in QA:
    1. Switch to PSG-TEST.
    2. DC on the “Cars QUA” short cut to launch the JWalk client program over the live Cars application directory.
    3. Sign on.
    4. See the changes in QA: 
11. Switch to CM and show:
    1. Status of task is now Production Ready
    2. Attachments from “automated testing”
    3. Approval pending for promote into Production

## Step 6: Promote to Production

1. Switch to Outlook and find the CM approval waiting message.
2. Reply to approve.
3. Switch to Web Portal and Jenkins.
4. Show the corresponding deployments to the build server in web portal, the build job running in Jenkins and then the final deployment of the JWR to PGS-APPSVR.
5. Switch to PSG-APPSRV and show the production Cars application to show it has now changed.
6. Optional: show a deployment rollback.
7. Optionally, show RM reports.