# Managing LegaSuite J Walk 4 with Aldon

Last updated: April 11, 2012

# Set up on PSG-TS

* Start up LMe 6.0 on psg-lmx and set defaults to Seagull/Jwalk/
* There are two releases:
  + Seagull/Jwalk/Car(Source)
    - This release will hold a copy of the XML screens, panels, etc, that make up the current Cars application
    - It will be used by the developers only as a “reservation” system so they can see who is working on what.
    - There is only D and P in this release, nothing in-between. You checkout to reserve at the beginning. You check in to release at the end.
    - The actual source version control will be managed by Subversion
    - As an option, the Aldon OVCC could be used to keep the XML code in the LMe inventory in synch with the actual code being managed in Subversion
  + Seagull/Jwalk/Car(Batch)
    - This is the build release that will hold the imported JWR file for release management and deployment.
* J Walk 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
  + It has two components
    - Terminal Editor
      * This component shows the 5250 data stream
    - Panel Editor
      * ?
* J Walk Client
  + This is the solution that will actually show the converted GUI panels.
  + I have created shortcuts to the client and changed the properties attribute to point the client to either the QA or the Live target folders.
* All of this is installed in psg-ts.
* Start, Programs, LegaSuite
* We will be editing the sample “Car” application
  + The development environment is C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR
  + QA is here: C:\JWalk\CarsApplication\QA\Car
  + Live is here: C:\JWalk\CarsApplication\Live\Car
* 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 have created a folder on psg-ts to group all of this together. It is c:\jwalk
* I have created two short cuts to the J Walk Client. One points to QA, one to Live.
  + Create the short cuts from the Start, Programs, LegaSuite menu.
  + Change the properties attribute for QA to look like this: "C:\Program Files\SEAGULL\J Walk Windows Client\4.1038.1.550\jw9c.exe" /eeeqa
  + Change the properties attribute for live to look like this: "C:\Program Files\SEAGULL\J Walk Windows Client\4.1038.1.550\jw9c.exe" /eeelive
  + Change the jwalk.ini file (located in C:\Program Files\SEAGULL\J Walk Windows Client\4.1038.1.550 ) to have two new “ee” sections, one for QA and one for live, that point back to the C:\JWalk\CarsApplication\QA\Car and C:\JWalk\CarsApplication\live\Car production target directories.
    - For example:

[EELIVE]

; J Walk Emulator Emulator

DynamicTransformation=1

GraphicalAccessServices=0

Emulator=SeagullEmulatorEmulator

EmulatorSwapEnter=0

HidePanels=1

EmulatorSessionId=A

WideScreen=1

WorkingDirectory="C:\JWalk\CarsApplication\Live\CAR"

* + I created a shortcut for the J Walk Developer. I changed the properties attribute to point to the development directory. It looks like this: "C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\jw9dev.exe" /EEE /d"C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR\car.jwd"
* I created a J Walk batch script (.jwb) that 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 C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR\cardemo.jwb

;J Walk batchfile : CARDEMO.JWB

;create logfile

LOGFILE cardemolog.txt

; publish

TEFILEOPEN "C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR\CAR.JWE"

PEFILEOPEN "C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR\CAR.JWA"

ALBUMOPEN "C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR\CAR.JWP"

TEMPLATEPATH "C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CARNEW"

PUBLISH "C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR\CAR.JWR","CARDEMO",1,0,0,123

;CAR.JWR=package file outputname, CARDEMO=application title

;1=number of package licenses, 0=verify panels, 0=set clipping,

;123=optional version code (could be left out)

; quit Developer and go back to command line

EXIT

* I created a dos batch script that will invoke the publish command to build the .jwr, extract the xml items from the .jwr to the CARXML directory, import both of them into Seagull/Jwalk/Car(Build) on psg-lmx in LMe 6.0
  + C:\JWalk\CarsApplication\Scripts\build\_script

rem: THIS IS THE IMPORT SCRIPT FOR THE Seagull J Walk APPLICATION BUILD TO LM

rem: SET THE VARIABLES FOR THIS RUN

set taskid=CARBLD

set label=13.4

set textvals=C:\JWalk\CarsApplication\Scripts\build\_values.txt

set comments=C:\JWalk\CarsApplication\Scripts\build\_comments.txt

set release=Seagull/Jwalk/Car(Build)

set fromjwr="C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR\car.jwr"

set fromjwd="C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR\car.jwd"

set outputpath="C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CARXML"

REM: run the J Walk batch publish to create the JWR file.

REM: this will execute the cardemo.jwb jwalk batch script to create the car.jwr in batch.

cd C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR

"C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\jw9dev.exe" /EEE /d"C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550\CAR\car.jwd" /Bcardemo

REM: RUN THE LEGACONV UTILITY TO UNPACK THE JWD INTO THE CARXML FOLDER

RMDIR %outputpath% /s /q

legaconv %fromjwd% -outputpath=%outputpath% /verbose- /outputsave 1 /outputnormalize /nobuildinfoinxml+ /application jwalk

REM: set the exectution path to LM 6.0 to find the "aa" commands

path=%path%;C:\Program Files\Aldon\Aldon LM 6.0

rem: CHANGE THE DIRECTORY TO THE CURRENT ROOT PATH OF BUILD

cd C:\Program Files\SEAGULL\J Walk Developer\4.1038.1.550

rem: DEFINE THE RELEASE INTO WHICH WE WILL IMPORT THE BUILD

aa signon donc d0ncaps psg-lmx

aa currel %release%

rem: RUN THE IMPORT COMMAND TO PULL THE BUILD INTO LM

aa import %fromjwr% -R %outputpath% -a %taskid% -l %label% -v %textvals% -f %comments%

rem: aa import %fromjwr% -R %outputpath% -a %taskid%

rem: Run the promote and deploy to QA now.

aa promote -e IMP

rem: aa signoff

rem: EXIT SUCCESSFULLY

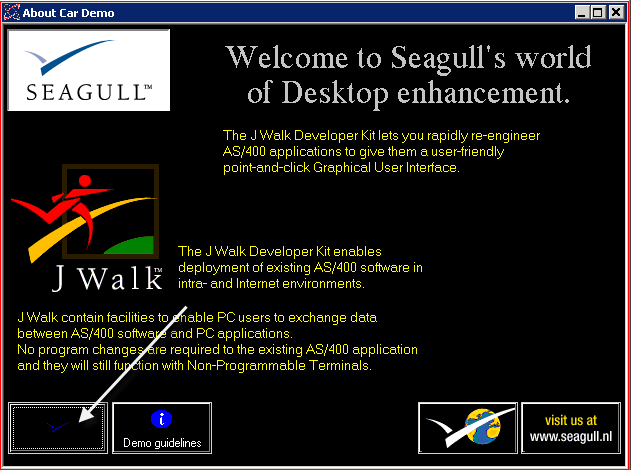
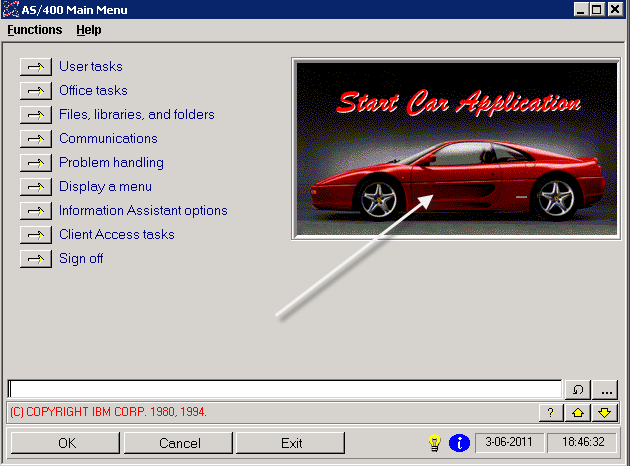
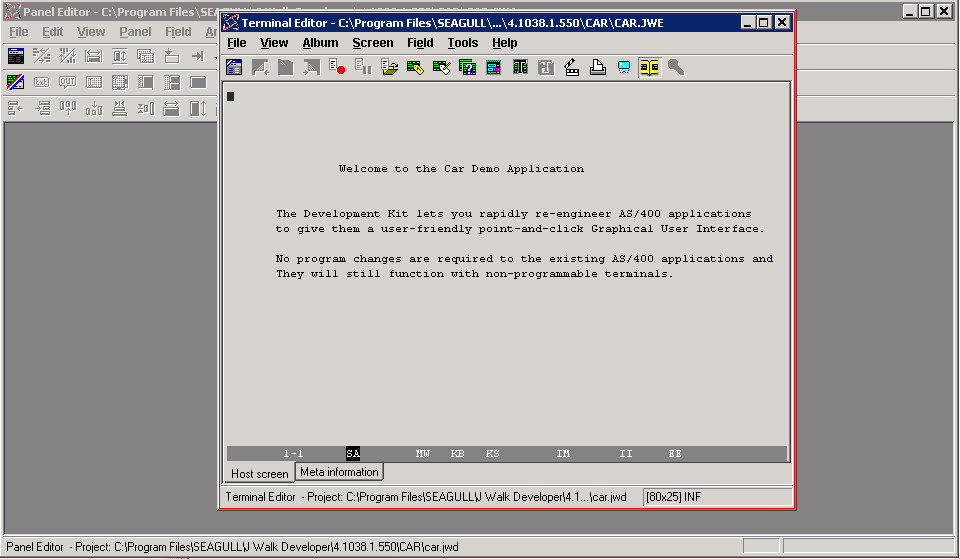
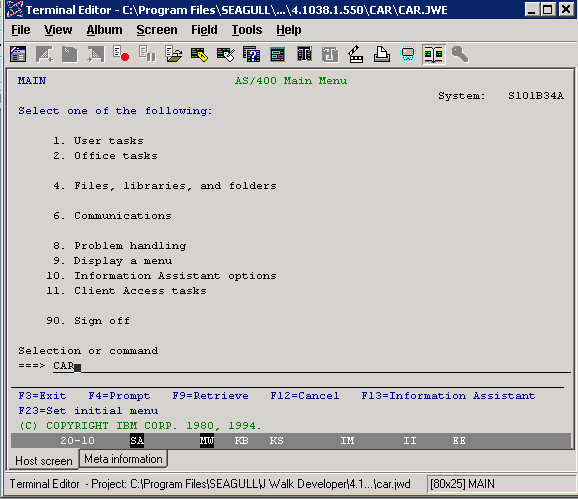
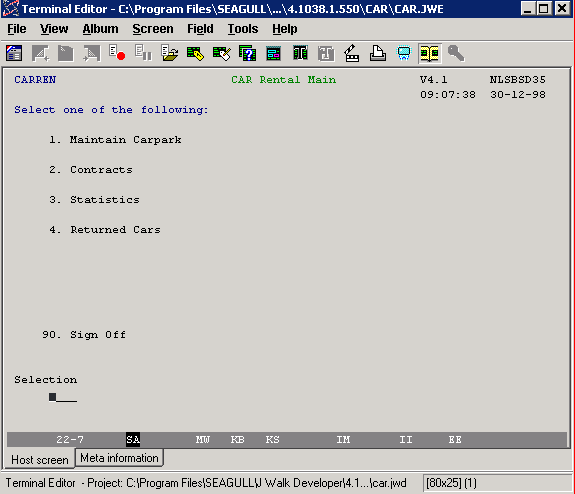
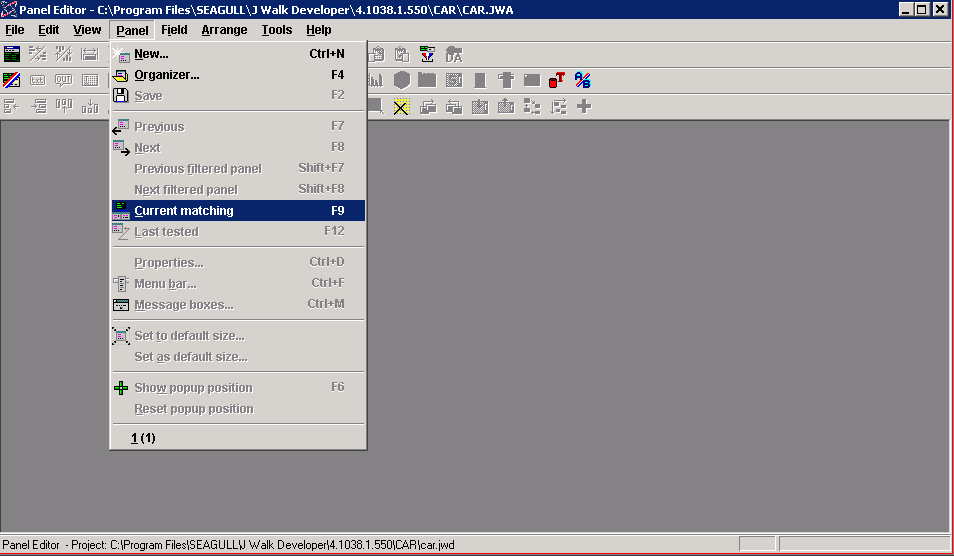
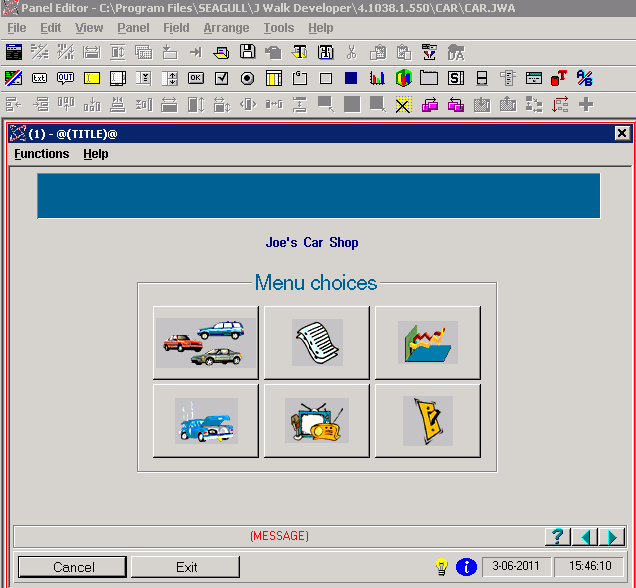
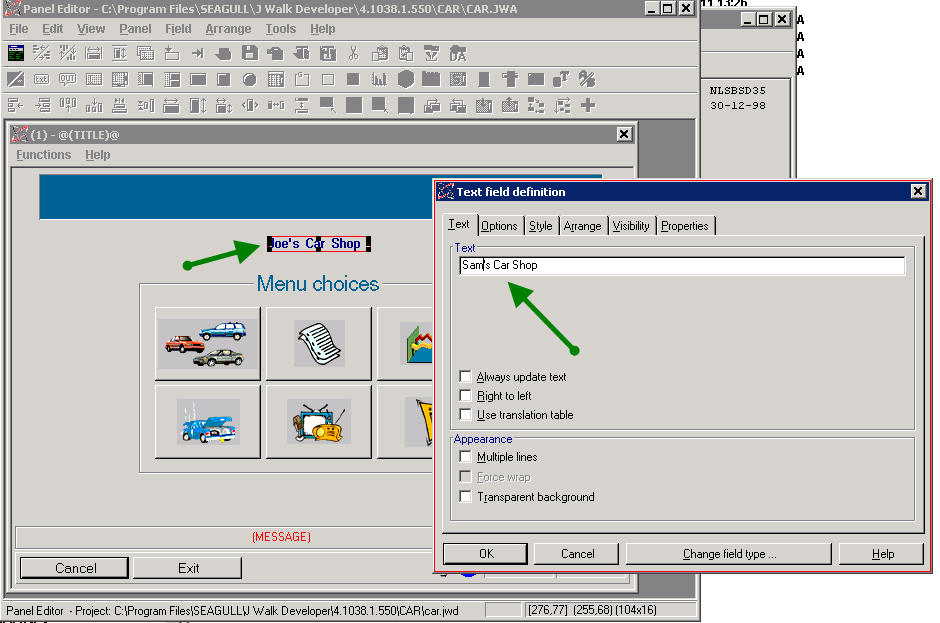
exit 0

# Demonstration of Aldon RMD

## Step 1: Reserve the Code

1. Start in LMe in the Seagull/JWalk/Cars(Source) release.
2. Check out the three items for screen (1) to your task
   1. (1).Screen
   2. (1).Panel
   3. (1)\_.Panel
3. Optional
   1. Check out a DSPF from the related LMi app to the same task.

## Step 2: Change the Code

1. Start in the C:\JWalk directory of short cuts.
2. Show the “before” live production application:
   1. DC on the “JWalk Client Cars Live” short cut to launch the JWalk client program over the live Cars application directory.
   2. Click on the icon of the Seagull in the lower left corner:
   3. Press Enter on the next page, the sign on screen
   4. Double Click on the red car to run the Cars application: 
   5. We’ll be changing the owner’s name of the Car Rental shop! 
3. Use the short cut in the c:\jwalk menu to launch the J Walk Developer application over the development directory.
4. You will see: 
5. In the Terminal Editor, press Enter twice to get to an AS/400 menu with a command line.
6. Type CAR on the command line to invoke the Car Application: 
7. This is the Car Rental Main Menu that we will be customizing with J Walk. Note the screen id in the lower right hand corner. It says (1) in this case. 
8. Now switch over to the Panel Editor. Select Panel, Current Matching. 
9. 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). 
10. 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. 
11. Save your work by clicking Panel, Save.
12. Exit the IDE.

# Step 3: Publish, Import, Promote and Deploy to QA in one step!

1. Of course, the developer will publish right from within the IDE as often as they need for their unit testing.
2. Let’s fast forward to the point when all development is done, and you are ready to build the “final” package and put it out to the QA server.
3. I created a dos batch script that will
   1. invoke the publish command to build the .jwr,
   2. extract the xml items from the .jwr to the CARXML directory,
   3. import both of them into Seagull/Jwalk/Car(Build) on psg-lmx in LMe 6.0
   4. promote the imported contents up to the QA level of the inventory
   5. deploy and install the jwr (but not the xml) onto the QA server for testing
4. Run C:\JWalk\CarsApplication\Scripts\build\_script
5. Switch over to the LMe Web Portal and show the deployment happening
6. Show LMe to show the imported package now in QA.
7. RC and show the properties. Show the batch script and the log file in the viewer.
8. Now show that we have the new changes in QA:
   1. DC on the “JWalk Client Cars Live” short cut to launch the JWalk client program over the live Cars application directory.
   2. Click on the icon of the Seagull in the lower left corner
   3. Press Enter on the signon Screen
   4. Click on the red car icon to start the Car Rental Application
   5. See the changes in QA: 
9. Return to LMe and promote to production with deployment.
10. Optional: show the production Cars application again to show it has now changed.
11. Optional: show a deployment rollback.

## Step 4: Release the Code

1. Start in LMe in the Seagull/JWalk/Cars(Source) release.
2. Work by Collection
3. Check in all the items.
4. Mention that if they had the OVCC, then the source in the LMe inventory would have been updated with the changed source. This would give them the added benefit of being able to do source compares in LMe to see what changed.
5. Mention the history of check out/check in against the task.
6. Optionally, show RM reports.