# DataSearch Application Walkthrough

# .NET, SQL

Last updated: December 4, 2012

## Overview

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 development is performed within Microsoft Visual Studio. This demo environment basically involves the following;

* Add/Remove a field in a table
* Rebuilt the application using MSBUILD to produce a new .EXE file
* Deploy and Install the .EXE on to the Test Server
* Rebuild the application, deploy and install onto the Productions Server.

This process uses Community Manager to enforce additional workflow, approvals and promotions, which is described in more detail in the ‘Workflow’ section below.

## Benefits for the customer

The benefits that this configuration will demonstrate to the customer are;

* Web based facility which allows customers to log issues into a central system.
* Automated workflow to assign request to the correct teams.
* Built in approval processes for authorized users.
* Secure central repository for application code management.
* Business process management aligned with software development lifecycle.
* Automated deployment and installation to Windows machine.
* No manual intervention required for Build process or deployment.
* Easy rollback functionality.
* Satisfies auditors requirements – logs from build server, applications servers, test systems etc all stored in a central single location.
* End to end change management.

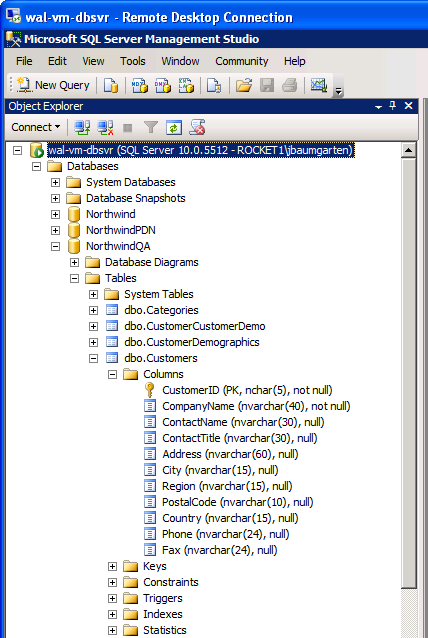
## Workflow

This is the general workflow of this demo;

* Raise an issue in CM, by logging on as a customer as ‘andy’
* This issue gets assigned to a Manager
* Sign into CM as yourself, and you should see the issue assigned to a ‘Manager’
* Approve the issue
* This assigns the issue to a developer called ‘Jon’
* Go into the issue, and link the request to LM, by creating a task with the same name as the issue in **Personal Banking\Customer Master\DataSearch (1.00)**
* Do the development – go into Microsoft Visual Studio, Add/Drop a field on the form and check in
* Promote to QA from within LMe – this builds the application by running scripts on wal-vm-build, imports the new .EXE file into LMe, deploys and installs it onto wal-vm-test. This promotion/deployment also sends the ‘control.txt’ file to the Psg-CMNew machine. This is then used for future emails back into CM.
* An email is sent back into CM, which changes the status of the issue to PASSED QA, which then starts an approval process, and changes the status to READY FOR PRODUCTION.
* Upon approval (either from email or CM), this will initiate the Promote action, which builds the application, imports the .EXE file into LMe, deploys and install it to wal-vm-prod.
* Once installed, an email is sent back into CM to change the status to COMPLETED.

## Pre Demo Preparation:

Open the following Applications/Tools on theses servers:

* WAL-VM-DEV2 :
  + LMe GUI,
  + Visual Studio 2010 (Project - DataSearch)
* WAL-VM-DBSVR :
  + Microsoft SQL Server Management Studio.
  + Click on the option to “connect” to the database.
  + Go into Databases, and expand the ‘NorthwindQA’ and ‘NorthwindPDN’ Databases.
  + Drill down to Tables, dbo.Customers, and Columns.
  + See if the email column is present or not. This determines if you want to check out the add or drop sql procedure in the demo.
  + 
* WAL-VM-TEST : Just leave the server displaying the Desktop
* WAL-VM-PROD : Just leave the server displaying the Desktop
* Open up a browser session to the Deployment Web Portal <http://aldmo1:10088/aldonlmw/>
* Build server ‘Jenkins’ [http://wal-vm-build:8080/view/DataSearch](http://psg-build:8080/view/DataSearch) If you cannot access this page, log onto the wal-vm-build server, and restart the ‘hudson’ service.

## Walkthrough

### CM

* Log into CM as ‘andy’ and create an Issue in CM (\_DEVELOPMENT MANAGEMENT Project)
* Log into CM as yourself and ‘Approve’ the issue. This will change the status to ‘DEVELOPMENT’ and the assignee to ‘Jon’
* Edit the issue, and Link the issue to LM by creating a task of the same name as the issue.

### Visual Studio

* In Visual Studio, checkout the following files
  + ControlFile.txt
  + DataSearchForm.vb
  + NorthwindDataSet.xsd
  + Add the email column?
    - **northwind\_customers\_ADD\_script.sql**
  + Drop the email column?
    - **northwind\_customers\_DROP\_script.sql**
* Update ControlFile.txt
  + On the first line, type the name of the Task created in LMe above – e.g ISSUE123
  + On the second line, type the CM Issue Number created in CM above – e.g ISSUE123
* Update NorthwindDataSet.xsd
  + RC on NorthwindDataSet.xsd and choose ‘View Designer’.
  + Dropping the ‘Email’ column?
    - Single click in the grey space just left of the word “email” to select it.
    - Right click it and choose ‘Delete’.
  + Adding the ‘Email’ column?
    - Right click anywhere in the grey box at the top that says “Columns”.
    - Select “Add”, then “Column”
    - Type over the name to rename it to ‘Email’.
  + Close the window and save the .xsd.
* Update DataSearchForm.vb
  + Double click on DataSearchForm.vb to open the edit window.
  + Single click on the ‘City’ Field in the center 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.
  + Show how it either contains or does not contain the Email column.
  + Close this window to save the .vb
* Checkin the files into LMe.
* This will move everything from Development into STAGING Environment, which can be seen using the LMe GUI.

### LMe

* Navigate to the collection tab and open the task.
* Promote the files into QA
* This deploys the source files to the build server and kicks off Jenkins to manage the build.

### Show

* Show the build being started using the Jenkins browser window.
* Show the Jenkins window next to it.
* Once the Jenkins build is done, switch to wal-vm-dbsvr
  + RC on “Columns” and select “refresh”
  + Show the Email column appearing/disappearing in the Test database.
* Return to wal-vm-dev2
* Open Web Portal and show how we are deploying the results of the build over to the wal-vm-test server.
* Open CM and show how the status and the attachments and the history have been updated.
* Open LMe and show the imported package under PersonalBanking\CustomerMaster\BuildQUA
* Switch over to wal-vm-test
  + Click on the desktop shortcut for “DataSearch”
  + Show had the Email column has appeared/disappeared
  + Click on “Show all Companies” to show the test data in this database.

**Under the Covers**

* 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 call either the ADD or DROP stored procedure to update the TEST database on WAL-VM-DBSVR.
  + Uses MSBUILD to build the application.
  + Imports the build result (.EXE), associated files back into LMe.
  + Promotes and deploys the build to QA.
* The last step of the Jenkins build is to deploy the generated .EXE to the QA server.
* This deployment will do the following;
  + Import into Build Release PersonalBanking\CustomerMaster\BuildQUA
  + Deploy the files to WAL-VM-TEST, into **C:\ProgramFiles\Aldon\TESTEnvironment**
* This deployment will 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 the stated CM issue
  + This updates the issue to Passed QA, updates the description text and attaches test documentation**.**
  + This then invokes an Escalation in CM, and changes the status to ‘**PRODUCTION READY’**.
  + This then invokes an Approval in CM.
* APP.CONFIG
  + This file tells VB which database to connect up to during the build.
  + We maintain two versions in the solution, appQA.config and appPDN.config
  + Our script substitutes either the QA or the PDN version over app.config just before the build.

### Promote to Produciton

* Go to your inbox
* Show the email sent with the test results from the automated test run in QA.
* Show the email asking for approval to promote to production.
* Show the attachments to the email.
* Reply to the email and approve,
* This automatically triggers the promotion/deployment process to production WAL-VM-PROD
* Show the emails sent from the promotion and deployments.

### Jenkins

* Show the build being started using the Jenkins browser window.

### Web Portal

* The last step of the Jenkins build is to deploy the generated .EXE to the QA server.
* Show this happening from the web portal.
* This deployment will do the following;
  + Import into Build Release ***PersonalBanking\CustomerMaster\BuildPDN***
  + Deploy the files to WAL-VM-PROD, into **C:\ProgramFiles\Aldon\TESTEnvironment**

### WAL-VM-PROD

* 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.