IFIX – PCM INTEGRATION USING QTP AUTOMATION TOOL

A Dissertation submitted to the University of Hyderabad in partial fulfillment of

the degree of

MASTER OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE

By

MOHAMMAD ASLAM BAIG

DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

SCHOOL OF MATHEMATICS COMPUTER & INFORMATION SCIENCES

University of Hyderabad
(P.O.) Central University, Gachibowli
Hyderabad – 500 046
Andhra Pradesh
India

CERTIFICATE

This is to certify that the dissertation entitled "**IFIX – PCM INTEGRATION USING QTP AUTOMATION**" submitted by **MOHAMMAD ASLAM BAIG** bearing Reg. No **09MCMI28** in partial fulfillment of the requirements for the award of **Master of Technology** in **ARTIFICIAL INTELLIGENCE** is a bonafide work carried out by him under my supervision and guidance.

The dissertation has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

D. Karuna
Project Supervisor,
GE India Exports Pvt. Ltd.,
Hyderabad.

Dr.Atul Negi
Project Supervisor,
Department of CIS,
University of Hyderabad.

Head of the Department,
Department of CIS,
University of Hyderabad.

Dean,
School of MCIS,
University of Hyderabad.

DECLARATION

I **MOHAMMAD ASLAM BAIG** hereby declare that this Dissertation entitled "**IFIX-PCM INTEGRATION USING QTP AUTOMATION TOOL**" submitted by me under the guidance and supervision of **Dr. ATUL NEGI & D.KARUNA** is a bonafide work. I also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date: **MOHAMMAD ASLAM BAIG**

09MCMI28

Acknowledgement

Apart from the efforts of me, the success of my project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project

*I would like to express my greatest gratitude to my project supervisor **D. Karuna**, for her valuable and continuous support throughout the progress of the project. Her suggestions helped me to improve my work a lot.*

*I am extremely thankful to my project supervisor **Dr. Atul Negi** for his valuable and continuous support throughout the progress of the project*

I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go my own way, without whom I would be unable to complete my project.

At last but not the least I want to thank my friends who appreciated me for my work and motivated me.

Finally, I would like to thank Allah, who made all the things possible.

MOHAMMAD ASLAM BAIG

Dedicated to

My Sweet Mother & my family…

Abstract

The Project Entitled “iFIX-PCM Integration using QTP Automation Tool” is used in checking the changes in existing code due to the process of applying patch (new release) to the existing code.

Change Management is a feature that’s part of a product called as Proficy Machine Edition (ME). The feature being described in this project is integration from iFIX 4.0 to v5.5 of ME allowing iFIX users to take advantage of Change Management as well as allow ME users running the Change Management server to view iFIX files stored in the Change Management system.

This project is a subset of the full feature Integration test plan and is meant to be an Interproduct regression test plan for future iFIX releases when we need to verify that the iFIX to ME integration is still functioning correctly

Contents

List of Figures

List of Tables

ABBREVIATIONS:

HMI – Human Machine Interface

SCADA – Supervisory Control and Data Acquisition

IP – Intelligent Platform

QTP – Quick Test Professional

PCM – Proficy Change Management

SCU – Security Configuration Unit

ROI – Return on Investment

CHAPTER 1

INTRODUCTION

1.1 Introduction to iFIX :

Proficy-iFIX is windows -based tool HMI (Human Machine Interface) / SCADA (Supervisory Control and Data Acquisition) component of GE IP(Intelligent Platform) family of software automation products. Based on open, component based technology Proficy-iFIX is designed to allow easy integration and interoperability between plant floor and business systems. It includes functional and architectural features that reduce the design time for automation projects, allow simple systems upgrades and maintenance provides seamless integration with third party applications, and increase productivity.

The SCADA portion of Proficy-iFIX provides monitoring, supervisory control, alarming and control functions. It guarantees the absolute integrity of data and provides complete distributed networking capabilities.

The HMI portion of Proficy-iFIX is the window in to Industry process. It provides all the tools to develop pictures. These pictures are used by operators to monitor the industry process.

1.2 About PCM:

Proficy Change Management is powerful software that enables team to manage, store and retrieve automation systems software, configuration, code and plant floor documentation easily. Proficy Change Management has powerful capabilities including user based security, version control, audit trails, e-signatures, central storage, automated & manual backup and recovery for any file, configuration or code base.

Proficy Change Management offers an automated way to help the organization assess, monitor, manage, control—and lower—project costs as well as ongoing maintenance costs. In addition, by providing traceability, notification of changes and regulatory compliance, it drives increased efficiency of the personnel, and as a result, the productivity.

1.2.1 PCM Features:

- **Version Control -** ensures that only one person at a time is making changes to the system and archives the versions once those changes are made—providing the ability to revert to a previous version.

- **Security -** enables you to monitor and control who has access to what function—reducing the number of errors that occur due to unauthorized access. It also enables you to set up a permission hierarchy by roles, which eliminates the need to establish permissions for every employee.

- **Audit Trails and Reports -** allow you to track what has been happening with programs and devices in the plant, automatically providing the "who", "what", "when", "where" and "why" information of events taking place throughout the operation.

1.3 QTP (Quick Test Professional) :

The Integration between iFIX and PCM is done by the QTP Tool.

Quick Test Professional software provides functional and regression test automation for software applications and environments. Quick Test Professional can be used for enterprise quality assurance.

Quick Test Professional supports keyword and scripting interfaces and features a graphical user interface. It performs functional and regression testing through a user interface such as a native GUI or web interface. It works by identifying the objects in the application user interface or a web page and performing desired operations (Such as mouse clicks or keyboard events) it can also capture object properties like name or handler ID.

Quick Test Professional uses a VBScript scripting language to specify the test procedure and to manipulate the objects and controls of the application under test. To perform more sophisticated actions, users may need to manipulate the underlying VBScript.

In the Next Chapter we will discuss about the work till now happened in this field and shortcomings of the existed work and about the proposed work.

CHAPTER - 2

RELATED WORK SURVEY

2.1 Survey:

Before the project “iFIX-PCM Integration using QTP Automation Tool” there exists one Project to perform the same operations. But those scripts are not working now as large changes in code takes place. All the previous dialogs are been now changed to new names and also new features are added to the product iFIX.

So to meet with the huge changes to the code, added features this solution is proposed.

2.2 Generic View of the Problem:

For every 8-10 months there is new release of Software happens in the every big Industries like GE, Microsoft etc. For the next 4-5 months project is put in Test mode. As in this phase, bugs from clients are considered and release new patches to solve the customer problems.

As a new Patch released, new code is added to the existing code. Changes to the existing code as a process of new patch release in the software is been checked by this Project.

2.3 Reasons for the Existing Method to Fail:

1) Added new features after that, it is not supporting the new features.

2) Enhancement is not considered. I.e. it will not be used in all platforms.

3) It will not be able to work on any Browser.

4) Difficult to Understand for the new users.

5) Not easy to execute.

6) Difficult to add new functions.

CHAPTER - 3

PROPOSED WORK SETUP

In this chapter we will discuss about the Settings to be performed before we start working.

3.1 CM User Setup

- We must use the Change Management Server Configuration tools to manually add iFIX users to the CM server. The iFIX user name needs to match the CM user name.
- We will also need to assign permissions to those users. The permissions we are interested in using are: Check In and Out, Get, Label, Override Checkout, Change Password and Edit Project List.
- Verifying file management – iFIX files will be able to be added to CM projects, checked out, checked-in etc.

3.2 iFIX SCU

- Security must be enabled to use Change Management.
- One new Application Feature will be added to SCU named “Change Management”. We must add Change Management to iFIX group in order to use Change Management to manage iFIX projects and files.

3.3 Proposed work Advantages

1) It will support all the existing features.

2) Enhancement is taken care.

3) Can be work on any browser, any OS.

4) Easily understood by new users.

5) Flexible to the changes in the code.

CHAPTER - 4

DEVELOPING AUTOMATION FRAMEWORK AND TEST SCRIPT

4.1 Folder Structure

- It's suggested that there should be a folder structure when developing Quick Test Pro.

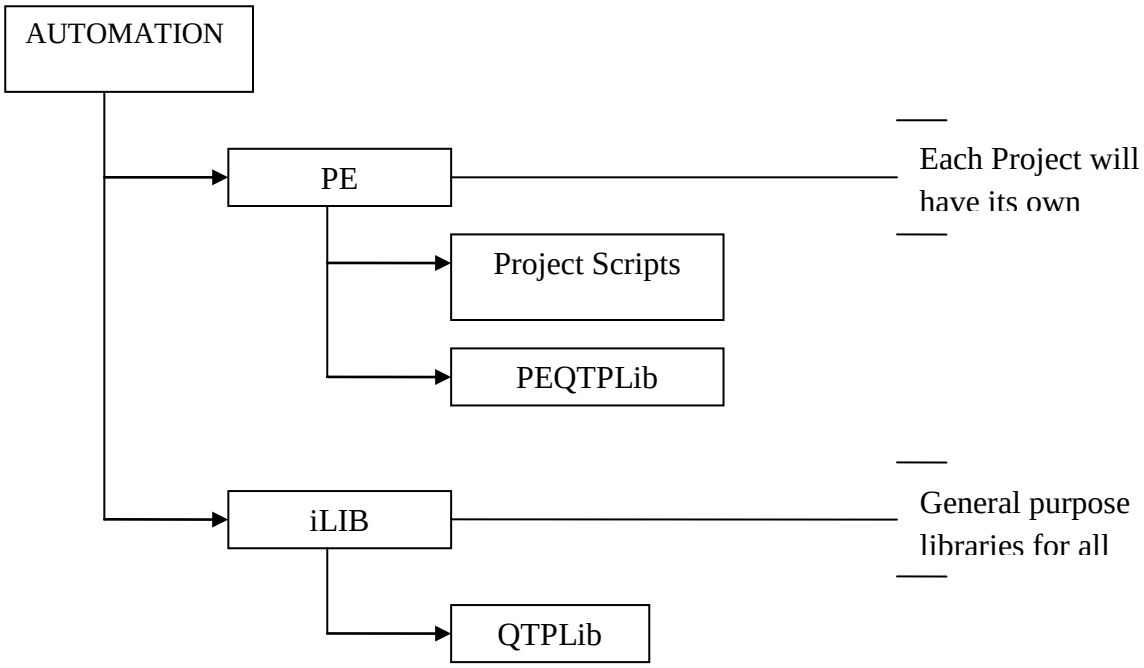

Figure 4.1: Folder Structure

1) **PE -** Project folder for PE. It will contain the following folders:
    - **Project Scripts** – Tests scripts for the project.
    - **PEQTPLib** – QTP files used for the project such as library files, object repositories, and recovery scenarios.

2) **iLIB -** General purpose library files used in automation testing that are not project specific. The following folders are:
    - **QTPLib** – General purpose QTP library files.

By this folder structure, easier to find shared libraries and more structured. It wouldn't make sense if a general purpose library is located in the folder of a test script.

4.2 Determining the test scripts

- For the test plan given, identify how best it can be broken down to write scripts individually and effectively. Note: The automated test plan does not need to literally follow the manual test plan.
- Determine whether you can break down the entire test plan to scripts either by test suite wise or by test case wise based upon the complexity and length
- Make sure scripts are not too long to understand and they are easy to identify the functionality which they are going to perform
- Avoid continuous and many functionality tests in a single test script. Try to split the scripts in to various actions and use action calls appropriately

4.3 Usage of Script Libraries/Functions

- The most important thing to be considered while developing test scripts is to initially identify the code/functionality which will be re-used throughout the test plan many number of times. This helps in reducing the effort of creating the script many times and as well supports easy maintenance
- It is required to identify the functionality/code in the early stages of developing test scripts, which can be re-used throughout the testing while creating scripts for other test plans even.

- Place the code in script libraries and you can use them in appropriate test scripts you need. Add location of library files in Folders tab of the Options dialog (menu “Tools/Options”) (see Figure 4.2)
- When adding Resources (menu “File/Settings”), DO NOT specify the path, only the file name. (see Figure 4.3)

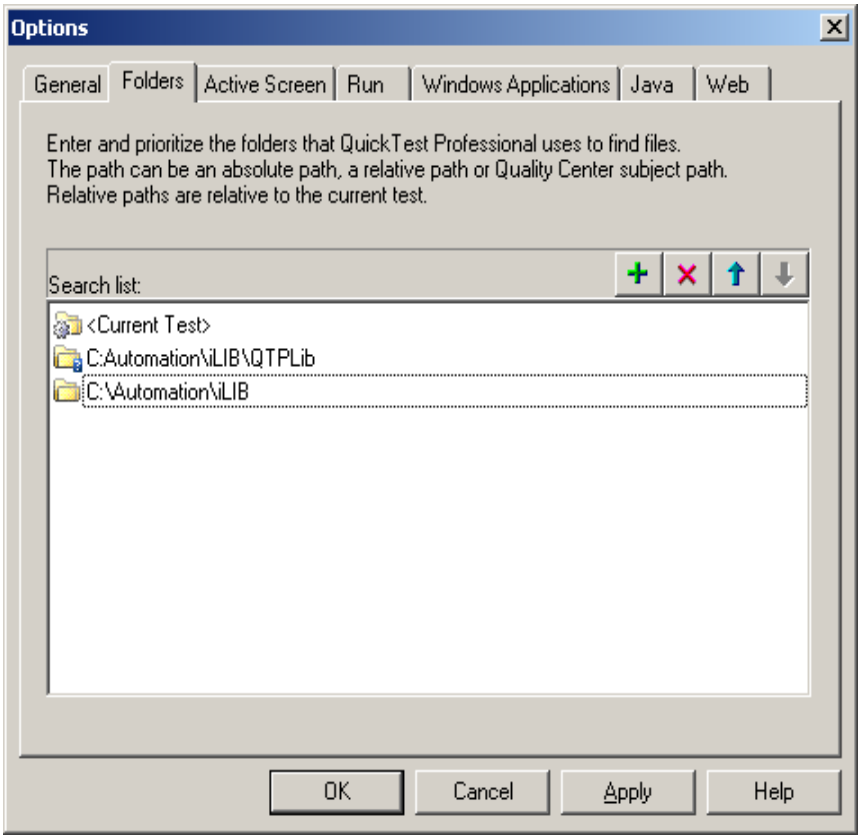

Figure 4.2: Folders tab of the Options dialog -Adding location of library files in Folders tab of the Options dialog – This allows library files to be located anywhere on the machine. This goes hand in hand with why we don’t specify paths when adding libraries and shared object repositories to the test.

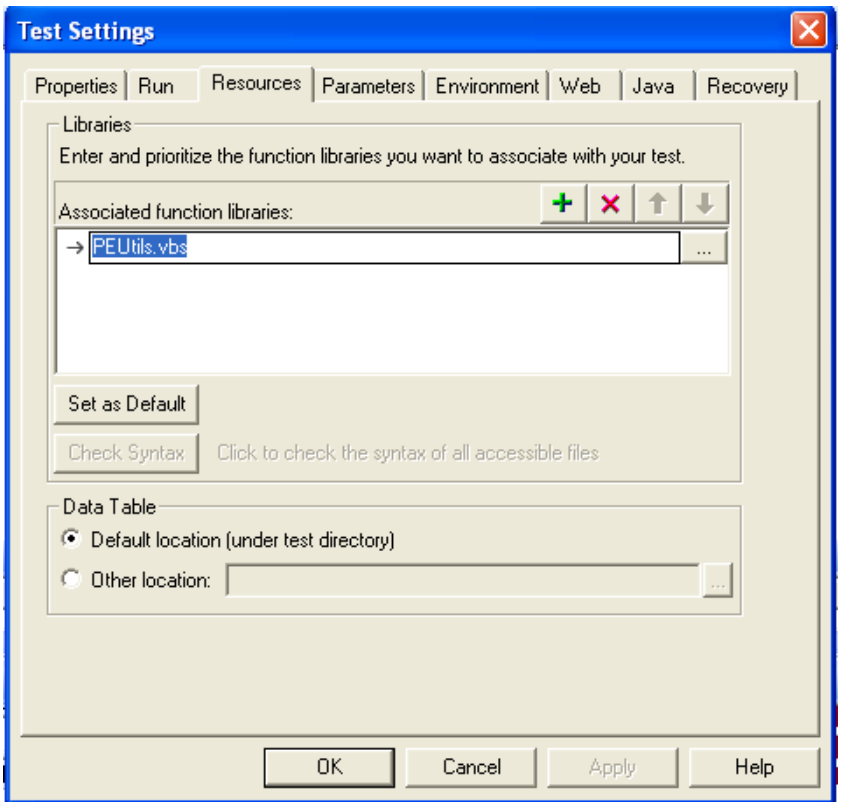

Figure 4.3: Resources tab of the Test Settings dialog -Not specifying paths when adding library resources to test. The path should be specified in the Folders tab. This allows more flexibility when developing/running the tests.

4.4 Repository Maintenance

- Use Shared Object Repository to simplify maintenance.
- By this changes to the repository only have to be made in one location as compared to if all test scripts had their own repository.
- Before actually starting recording/creating a script add all the objects to a shared object repository which are likely to be opened
- Similar kind of objects need not be added to the repository again and again. It is recommended to use Descriptive Programming in scripting. This helps in easy maintenance of scripts as well as the repository.

4.5 Parameterization

This is the most important concept which needs to be understood and made use of throughout for effective writing of test scripts. When we write a script to test an application, we may want to check how the application performs the same operations with multiple sets of data. And sometimes there could be scenarios where in the test script created on one machine may not run on another machine due to change in the machine names.

So, in the test script where ever the machine name is being used if it is parameterized by using a Global Sheet then it would make the job easier. Where ever possible try to use either a Global Sheet or a Local Sheet to parameterize things.

4.6 Reasons for Structured Scripts

- **Using Shared Object Repository –** Changes to the repository only have to be made in one location as compared to if all test scripts had their own repository.

- **Using Libraries –** Allows the re-use of test code used in logging in, creating schema (in RTIP), etc. In addition, it will be easier to maintain.

- **Using Folder Structure –** Easier to find shared libraries and more structured. It wouldn't make sense if a general purpose library is located in the folder of a test script.

- **Adding location of library files in Folders tab of the Options dialog –** This allows library files to be located anywhere on the machine. This goes hand in hand with why you don't specify paths when adding libraries and shared object repositories to the test.
- **Not specifying paths when adding library resources to test –** The path should be specified in the Folders tab. This allows more flexibility when developing/running the tests.

CHAPTER - 5
PROBLEM DESCRIPTION AND IMPLEMENTATION

5.1 iFIX User Preferences (Logon and Project info)

- Logon on Workspace startup checkbox :

  Positive Test:
  Log into iFIX`s Workspace. Verify CM is logged in and a CM session is started. Note – upon initial successful CM logon for a Project, behind the scenes in ME a complete iFIX project with the entire iFIX directory structure (just directories not files within directories) is created as a project in ME/Source Safe.

  Verify Work Space system tree displays current check out status for individual files.

  Negative Test:
  Verify if the user does not exist (i.e. user not defined in CM admin), then receive an error at startup that the user is invalid and have to manually create this user in CM in order to continue

5.2 Change Management Logon

- User Preference – Prompt User = FALSE
  - If user/pwd/server are valid, the user is logged in to CM
  - If the currently logged in iFIX user is an invalid CM user, then receive an error that the logon failed due to invalid user.

- If the iFIX/CM user is valid but password is invalid, then receive an error that the logon failed due to invalid password.

- If the iFIX/CM user and password are all valid but the CM server specified in the User preferences is invalid, you should receive an error that the logon failed due to invalid server name.

5.3 Security and Configuration

- Configure a CM user's Check In/Out rights.
  - Verify a user in a group that does not have Check In/Out rights cannot Check in any files from within iFIX

  - Verify a user in a group that DOES have Check In/Out rights is allowed to Check in a file from within iFIX.

5.4 Diff Tool available from the History Report Dialog

- Compare PDB only binary diff is done
  - Verify the diff for the following:
    - Local file against the latest version in PCM
    - Two different labeled versions within PCM
  - Verify a diff where the files match, the report should indicate no diff.
  - Verify a diff where the files don't match, the report should show a Diff Exists

5.5 History Report available for each file in the System tree

- Verify the following information is correct on the History Status Dialog box:
  - Version
  - User
  - Label
  - Time
  - Date
  - Action
  - Comment
  - Path – full path of server. Path is relative to project.

5.6 Check out Status Dialog

- File
  - Verify the file for which we select the Check Out status from the RMM displays in the list box
  - Verify Security Files don't display in the File field
  - Verify Historical Assign Files don't display in the File field

5.7 Get Project Dialog and Previous Project Location Dialog

- Project Destination Field
  - Verify Project Destination will show previous project locations
  - Verify when enter a value in this field and the Project is copied to this Destination when we select the Get button.

CHAPTER - 6

RESULTS

6.1 Setting up Batch Files

Tests should be setup to run in batches. It is suggested to Setup the batches by functional area, test plan, test suite.

They can be setup using the following Mercury tools:

- Test Batch Runner
- Multi Test Manager

6.2 Setting up scripts for Logging Test Results to Common Test Results Database

The following information is needed for logging test results to the Common Reporting system:

- TestCase
- Suite
- BuildNumber
- OperatingSystem
- ComputerName
- ScriptName
- Result
- StartTime
- EndTime
- ResultMessages
- LogFile
- LogTime

- TestDesc
- ModuleName
- SuiteType

6.3 Return On Investment(ROI) :

- '**Return on Investment'** is defined as amount of effort i.e. human effort, time, and money will be saved by Automating this Test Plan.

- **Practically for the iFIX-PCM TestPlan :**
  The effort needed to execute this test plan by a person is 7 days.
  If in a year the test plan is executed 10 times then the time spent in a year is 70 days.
  If the effort spend to Automate this test plan is say 30 days
  In the First year of project completion effort saved will be 70-30 = 40 Days
  After that for each year the savings will be equivalent to 70 days.

Below Table shows the Return On Investment details for the product "iFIX-PCM Integration using QTP Tool".Waht is the actual time taken to check each of the Testcases manually,what is the time saved from Automation Scripts all the details are enclosed in given table.

Table 6.1 Return on Investment of Project

Automation type	Test Plan/Script Name	Manual Test Cases	Test Cases automated	Planned Effort in days	Actual Effort in days	Maintenance effort days (include the time taken to modify any scripts + execution effort etc.)	Total Automation Effort in days = Actual Effort + Maintenance Effort	Manual Effort days (i.e. Effort needed if automated cases are to be done manually)	No. of Times Script Executed	Defects Found by Automated Testing	Total Manual Effort days = Manual Effort *execution runs =	Total Effort Saved in days = Total Automation Effort - Total Manual Effort
	CM iFIX	57	40	30	40	10	50	7	10	2	70	20

CHAPTER - 7

CONCLUSION

The Project Entitled “iFIX-PCM Integration using QTP Automation Tool” is used in checking the changes in existing code due to the process of applying patch (new release) to the existing code.

By running the Scripts in the project we will get the result as each test case is successfully executed or any error occurred during the execution. While doing this project challenges faced are parameterizing the variables, i.e. the values like user name password are taken from Global Sheet, export the values to document.

REFERENCES

1) IFIX Cockpit page – http://fxbcockpit:8080/synergy/themes/metal/frames.html

2) Edmonton Cockpit page (client API feature) - http://gefedmpolarisbd:8080/synergy/themes/metal/frames.html

3) GE Fanuc Coding Standards for Quick Test Pro.doc – http://albnydev13fage.cho.ge.com/svn2/proficy/cimplicity/Non-Distributables/7.1 PPS/QA Files/Automation Files/GE Fanuc Coding Standards for Quick Test Pro.doc

4) Mercury Quick Test Pro User's Guide.doc - http://albnydev13fage.cho.ge.com/svn2/proficy/cimplicity/Non-Distributables/7.1 PPS/QA Files/Automation Files/Mercury Quick Test Pro User's Guide.doc

5) 7.5 QA Automation Strategy Document - http://albnydev13fage/svn2/proficy/cimplicity/Non-Distributables/7.5 General Release/QA Files/Automation Documentation/CIMPLICITY 7.5 QA Automation Strategy.docx

6) www.ge-ip.com/products/2820