

**BPT Components creation standards and process guidelines**

17 Dec, 2015

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No | Version Number | Author/Reviewer | Date Created/Modified (DD/MM/YYYY) | Remarks |
| 1 | 0.1 | Suri Babu Mylapilli | 12/11/2015 | Initial Version |
| 2 | 0.2 | Deepak Kumar | 12/14/2015 | Added web for Test Plan and Test Lab section |
| 3 | 0.3 | Deepak Kumar | 12/17/2015 | Added folder structure and naming convention |

|  |
| --- |
| CONFIDENTIALITY STATEMENT |

The data contained here in shall not be disclosed, duplicated, or used in whole or in part for any purpose other than Internal Distribution ofCardinalHealth

Table of Contents

[1. Purpose and scope 1](#_Toc437632861)

[1.1 Purpose of this Document 1](#_Toc437632862)

[1.2 Audience 1](#_Toc437632863)

[1.3 Objectives 1](#_Toc437632864)

[1.4 Scope 1](#_Toc437632865)

[2. Business components 2](#_Toc437632866)

[2.1 What is a Business Components 2](#_Toc437632868)

[2.2 Component Folder 2](#_Toc437632869)

[2.3 Component Naming 3](#_Toc437632870)

[3. Component parameters 3](#_Toc437632871)

[3.1 Parameters General Information 3](#_Toc437632873)

[3.2 Component Parameters 4](#_Toc437632874)

[3.3 Parameter Names 4](#_Toc437632875)

[3.4 Parameter Type values 4](#_Toc437632876)

[3.5 Parameter Default Values 5](#_Toc437632877)

[3.6 Parameter Description 5](#_Toc437632878)

[3.7 Input File 6](#_Toc437632879)

[4. Components created in hp alm test plan module 6](#_Toc437632880)

[4.1 Final Business Component Assembly 6](#_Toc437632881)

[5. Test Execution (Test Lab) 9](#_Toc437632882)

[6. Component design 12](#_Toc437632887)

[6.1 Component Steps 12](#_Toc437632889)

[6.2 Component Step Description 13](#_Toc437632890)

[6.3 Component Step Expected Result 13](#_Toc437632891)

[6.4 Parameters Used In Component Steps 13](#_Toc437632892)

[7. Component Attachments 15](#_Toc437632893)

[7.1 Attachments 15](#_Toc437632895)

[8. Component Review process 16](#_Toc437632896)

[8.1 Intended of Review 16](#_Toc437632900)

[8.2 When the Review Process Takes Place 17](#_Toc437632901)

[8.3 Who Reviews the Process 17](#_Toc437632902)

[8.4 Component Review Documentation 18](#_Toc437632903)

1. Purpose and scope

## Purpose of this Document

This document is designed to convey the standards and guidelines around Business Process Testing and Automation.

This will be tool-specific, focusing on utilizing HP ALM, UFT, cFactory and its capabilities.

## Audience

This document is intended for **Internal Distribution** CardinalHealth testing team.

## Objectives

The objectives of this document are:

* To ensure that components are written in a manner that is conducive to the development of automation scripts for those components.
* To identify the manner in which components, test cases, and test sets will be stored and maintained from a Business Process Testing perspective

## Scope

The scope of this document is limited to the development of components, test cases, and test sets within the framework of Business Process Testing and automating the same.

1. Business components

## What is a Business Components

Business Components are granular functional units that are reusable and perform a specific task (i.e. navigation, validation, data entry, etc.) They are the building block of a Business Process Test.

## Component Folder

A folder structure exists for each application under test as well as a “Generic Components” folder structure for common components.

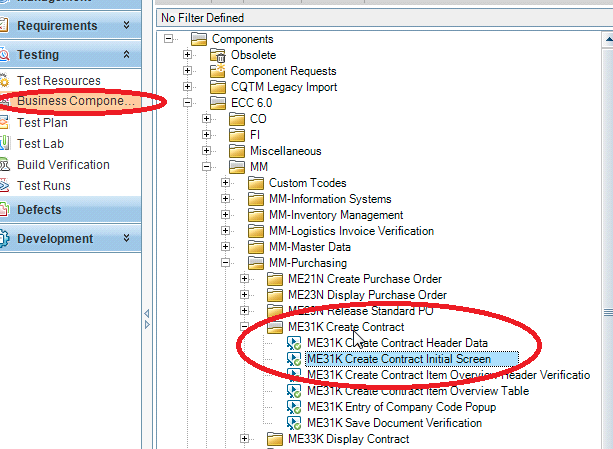
Application folder structures may have their own “generic” folder structures as well for components that are usable in multiple areas of the application. Additionally, a “Workspace” folder structure also exists for each member of the team. This area is for experimenting with components. Important no live test assets should be utilizing components in Workspace

For cFactory the folder name should start with Application Name and then followed by the module name, from the below sample figure, ECC is Application, MM Purchasing is a Module Name where we have to place all the functions or TCodes under the Module Name

**SAP:**

Folder names are Tcode name

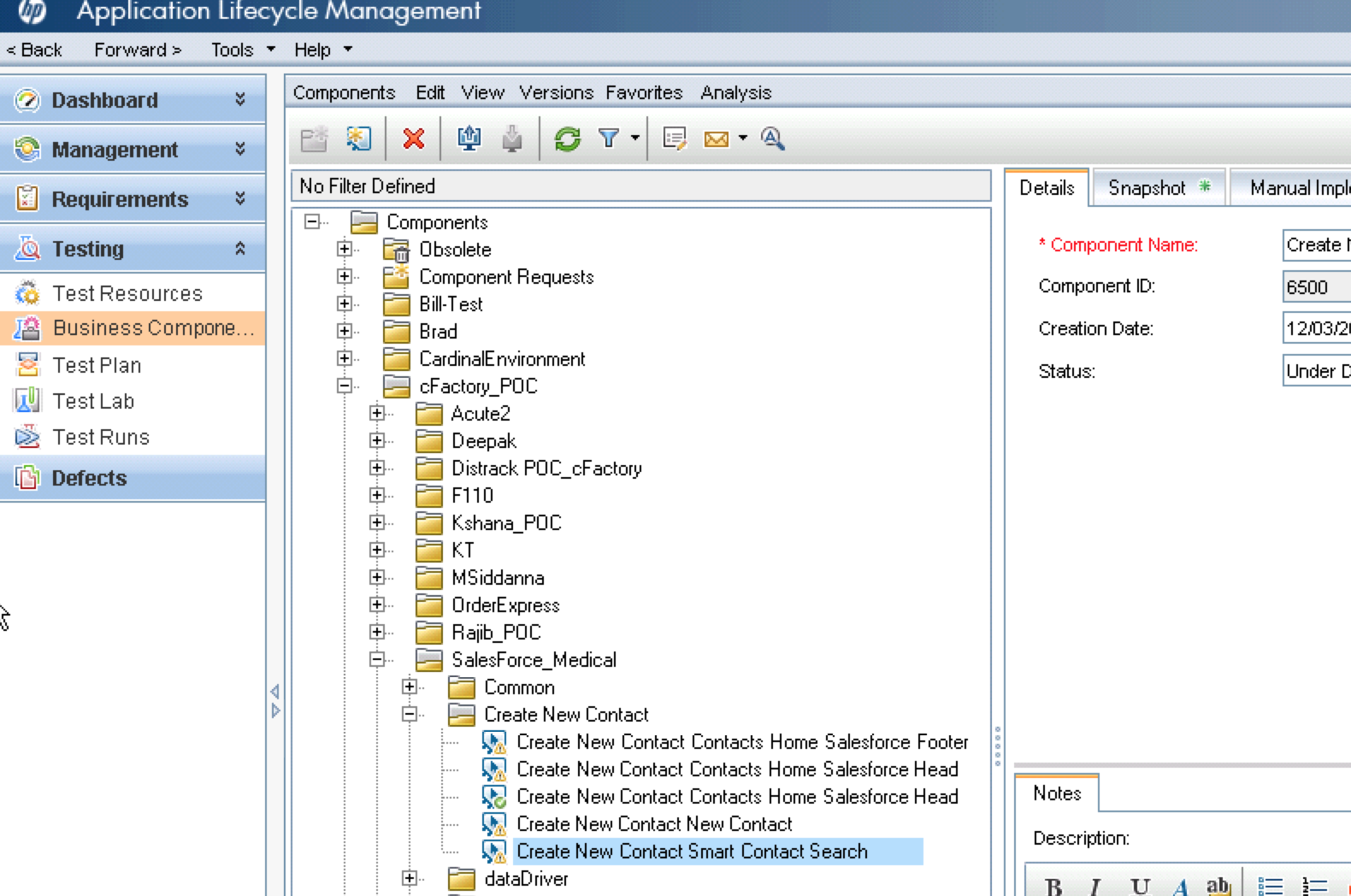
Example: FB120 ScreenName



**Web:**

Folder names are Action + Functionality name

Example: Create (Action) + New Contact (Functionality)



## Component Naming

Components must be named in a way that description of what action is being taken at that time in a test script and then followed by function

For Example: **Create Shopping Cart, Create** is an action and **Shopping Cart** is functionality

These components would be part of the “Action + Functionality” folder and also needs to be balanced with making sure that the names are not excessively long.

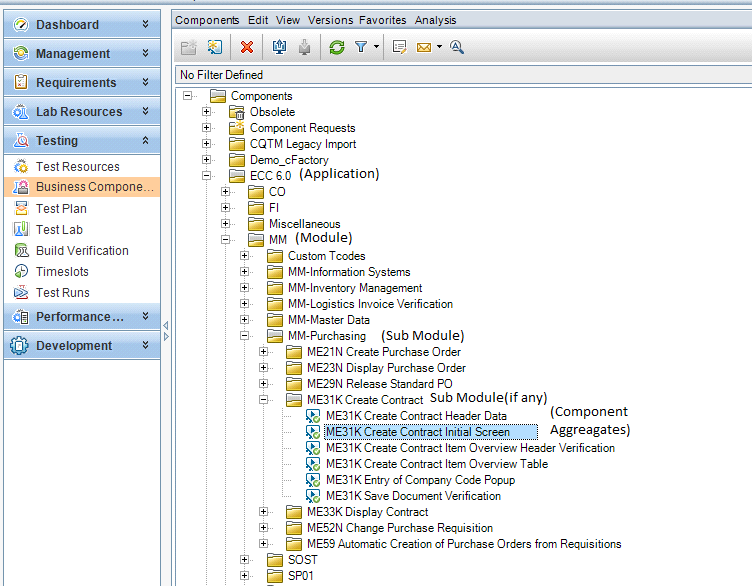
For **cFactory** the BPT components names are auto created. (which can be updated as per this document)

**Folder Structure: - Application\Module or Functional Area\Sub Module or Sub Functional Area\Component Aggregates**

**SAP:**

**Naming Convention: -** Tcode + Screen Name

Example: ME31K (Tcode) + Create Contract Initial Screen (Screen Name)



**Web:**

**Naming Convention: -** Action + Functionality + Screen name

Example : Create(Action) + New Contact (Functionality) + Smart Contact Search (Screen name)

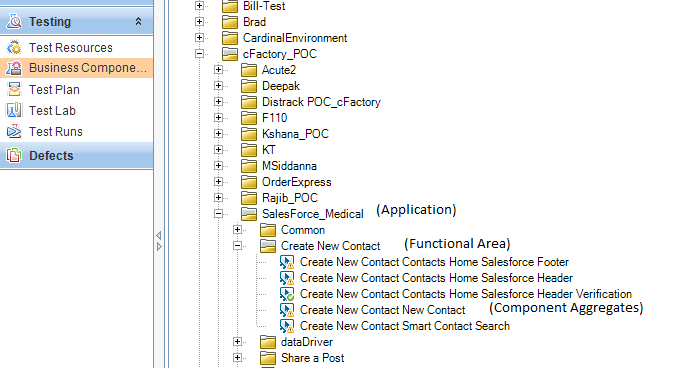


Figure: 1

1. Component parameters

## Parameters General Information

***This section is for normal BPT components***

The parameters are fields that receive or return data values. By parameterizing data, the same component can be used in different ways: positive testing, negative testing, boundary testing, or even entering specific data within a particular Component, Test Case, or Test Set.

The Parameters tab allows a user to define the parameters for a particular component:

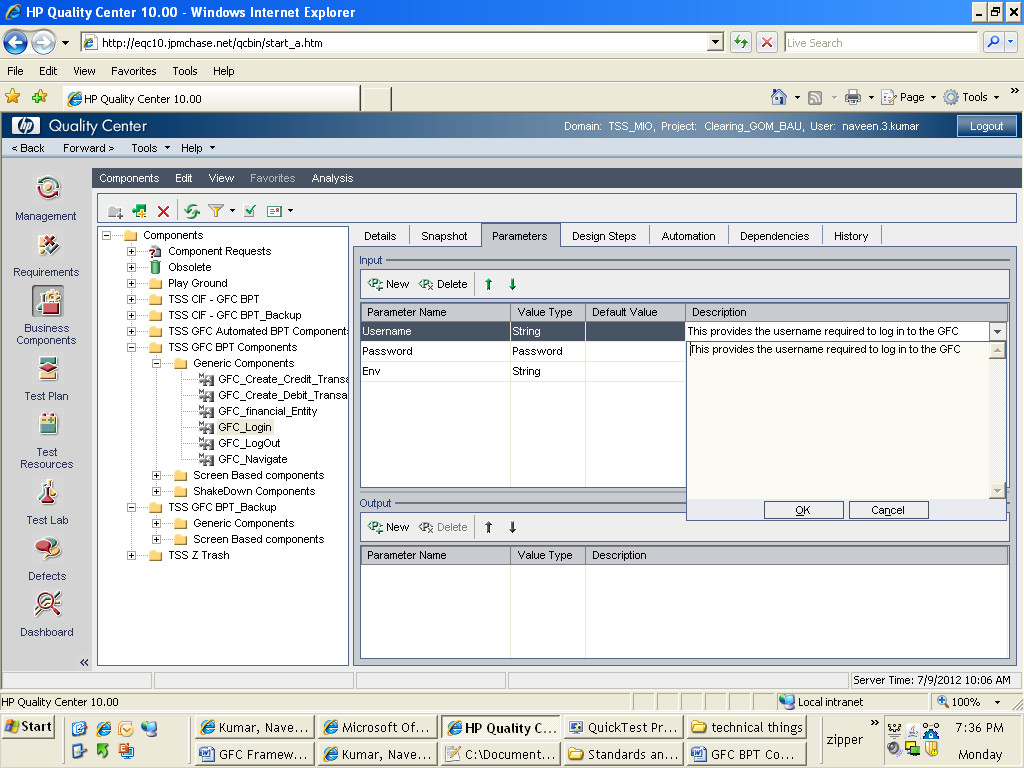


Figure: 2

## Component Parameters

Input parameters enable components to use data provided by an external source. Output parameters enable data from one component to be passed to an input value to a subsequent component in the business process test flow.

Input parameters contain data that is brought into the system. These could be field values, filenames of files containing data, or the data file parameters inside those files.

Output parameters contain data that comes OUT of the system and needs to be stored in a variable.

## Parameter Names

Parameter names should utilize Hungarian notation. The name must give an indication as to what data is being held by the parameter

Ex: AccountNumber, InputDataSheet

## Parameter Type values

The parameter is further defined by what type of data it holds. The types are:

* String
* Boolean
* Date
* Number
* Password
* Date and Password value types should not be used. Use the type of String instead

## Parameter Default Values

Even though the parameters are defined at the component level and default values can be assigned, the actual values of the parameters are found at the Test Case or Test Set level. The default value will be the value assigned to the parameter when the component is added to a test case. Sometimes a parameter will need a special value.

Defined special values are:

|  |  |
| --- | --- |
| Value | Meaning |
| \*NA\* | Not Applicable – The field, table cell, etc. does not exist so there is no value to validate. |
| \*Null\* | Null/Empty Value – The field, table cell, etc. exists, but does not have a value. For example if an edit box appears on the screen and there is no data in it, its value is \*Null\*. |
| \*IgnoreData\* | Ignore Value – The field exists, has a value (which could be null), but should be ignored at this time and should be considered as having been validated and deemed as PASSED. |
| \*Today\* | \*Today\* A value of the current system date (in mm/dd/yyyy format) will be utilized. |
| \*Time\* | A value of the current system time (in hh:mm:ss xM format) will be utilized |
| \*Now\* | A value of both the current system date and time in mm/dd/yyyy hh:mm:ss xM format will be utilized. |

## Parameter Description

The “Description” under the parameters tab will be used to provide the details and usage of the parameter

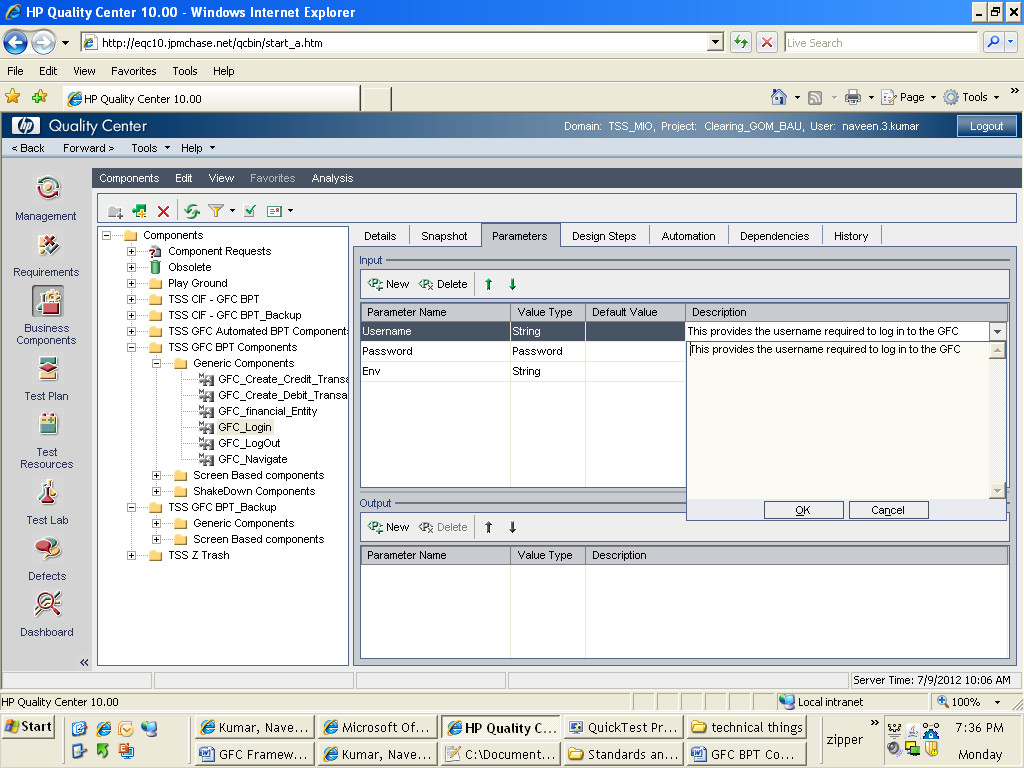


Figure: 3

## Input File

If the component contains complex data and cannot handle through parameters, such data will be handled through CSV files using the Import and export option available in the test lab -> Test sets

Apart from the normal required input data at component level if there is a specific data that changes for every run will be maintained in a customized Global Input Excel called master Global input sheet and will be stored in the QC Test Resources tab

1. ComponeNts CREATED in HP ALM TEST PLAN MODULE

## Final Business Component Assembly in HP ALM Test Plan

The Functionality components at page level are created in the Test Plan module.

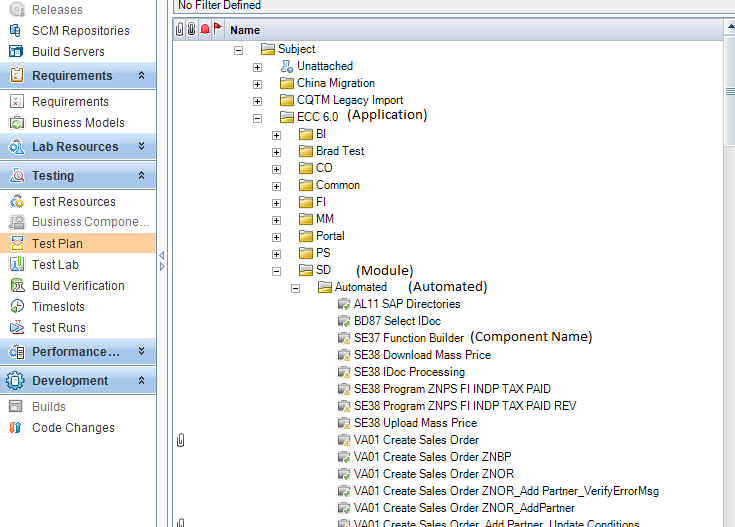
For **cFactory** follow the below steps for final component assembly.

**Folder Structure: Application\Module or Functional Area\Automated\Component Name**

**SAP:**

**Naming Convention: -** Tcode + Screen Name

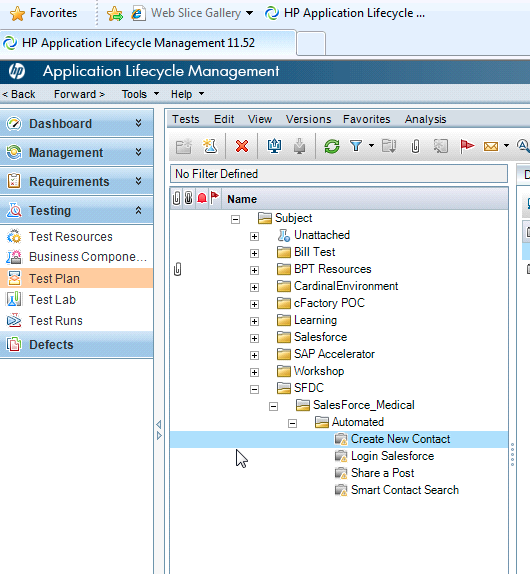
Example: VA01 (Tcode) + Create Sales Order (Screen Name)



**Web:**

**Naming Convention: -** Action + Functionality

Example: Create (Action) + New Contact (Functionality)



1. Test Execution (Test Lab)

The automated components are executed under the Test lab folder

For cFactory follow the below steps

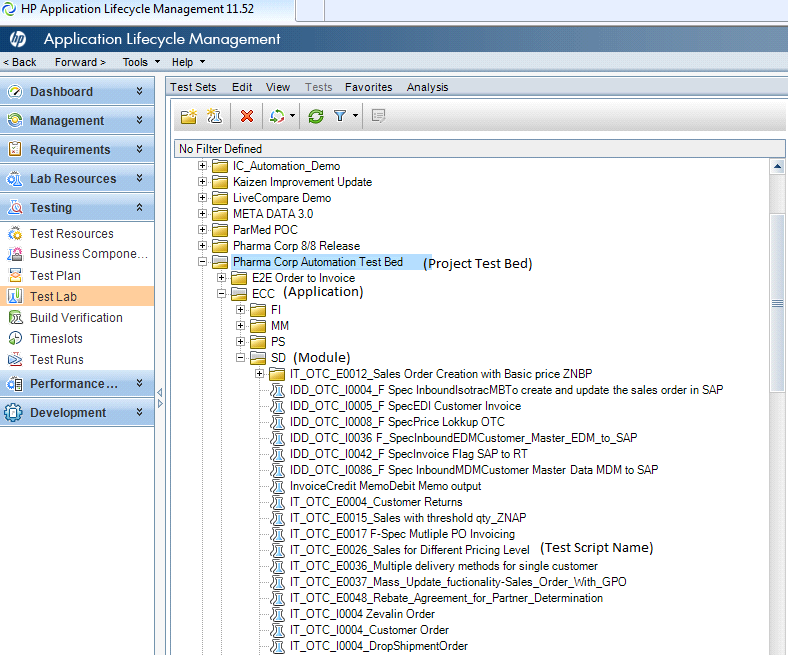
Inside module-wise folder, test sets should be present which are created using the t-codes in test plan module by drag –drop method.

**Folder Structure (Recommended): - Project Test Bed\Application\Module or Functional Area\Test Script Name**

**SAP:**

**Naming Convention: -** Manual Script Name

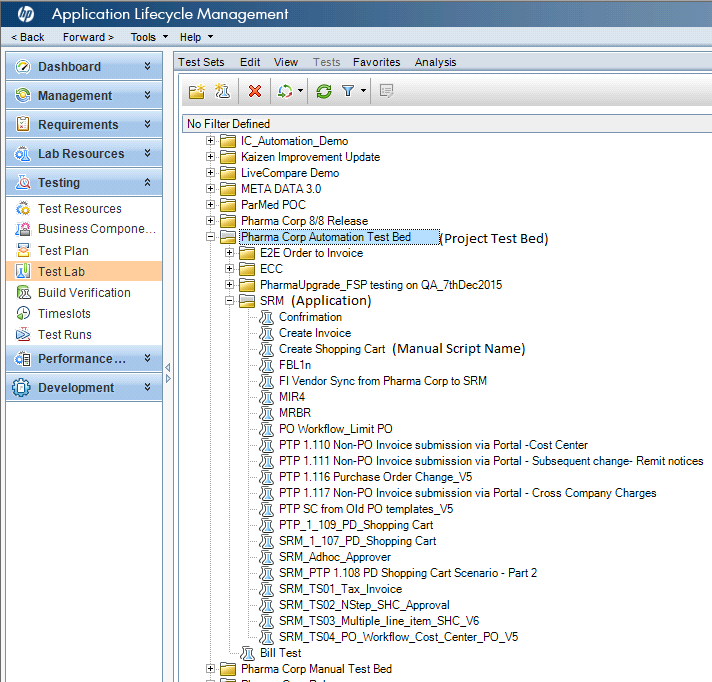
Example: IDD\_OTC\_10005\_F SpecEDI Customer Invoice



**Web:**

**Naming Convention: -** Manual Script Name

Example: Create Shopping Cart



5. Component design

## Component Steps

Action words will be used along with the object name to create step names.

Example: Enter Username

Select Account Number

Below is the list of action words and will continue to update the list as on required

|  |  |
| --- | --- |
| Enter | Entering of text or other information into a field. |
| Edit | Modify the text or data in a field |
| Delete | Delete text or data from a field |
| Clear | Clearing of a selection |
| Click | Click on any object available on the screen |
| Select | Used to show the selection of the object |
| De-Select / Un-Select | Used to un-select the object |
| Verify | verification of the object either appearance, state, of contents |
| Navigate | Changing of the current display either to a new module or another area of the current module. |
| Double Click | Double click on an object |
| Update | Modify the data being displayed or in the system |
| Delete | Remove data |
| Open | Open any screen or table etc |

## Component Step Description

An explanation of what is to be done in this step. If any data is to be entered on this step, it should be added into the description of a component step. The parameters should be entered after the text of the description.

## Component Step Expected Result

This describes the outcome of the action in the Description. All the results should be located here. The parameters should be entered after the text of the expected result

## Parameters Used In Component Steps

Steps that utilize parameters will have the name of the parameter in triple angle brackets:

<<<Like This>>>. The text inside triple angle brackets indicates the name of a parameter.

A parameter must be created before it can be selected. Quality Center has a control that ensures only valid component parameters can be used in a step.

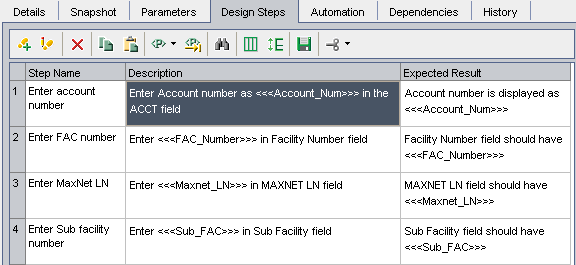


Figure: 4

1. Component Attachments

## Attachments

Attachments are not stored at the test case level. If any data files that are used at test runtime are stored at the test set level.

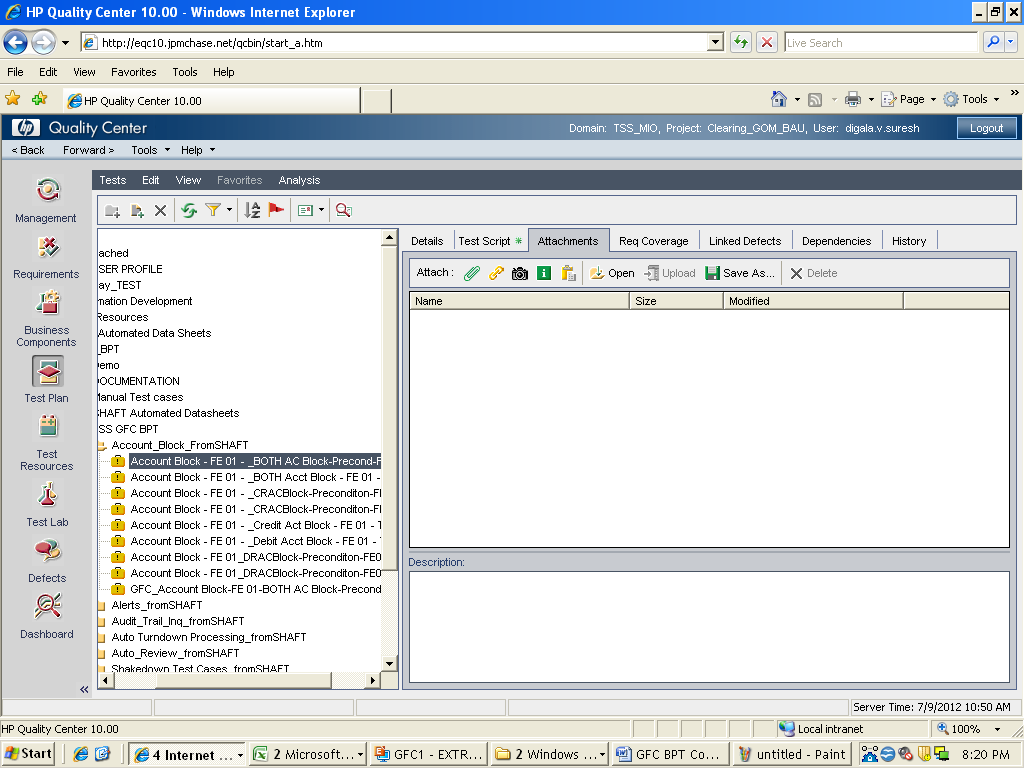


Figure: 5

1. Component Review process

## Intended of Review

The purpose of the review of the manual component is to ensure that all engineers that are creating and updating components in accordance with the standards described in this document.

**Manual Component Review Guidelines**

|  |  |
| --- | --- |
| **Item Name** | **Description** |
| Folder name is meaningful. | * Is the folder setup for the component descriptive and easy to navigate? |
| Component Name | * Is the component name descriptive of the component? |
| Component: System Fields | Are the following editable system fields correct?   * Assigned To * Created By * Creation Date * Status |
| Details: Component Description | * Does the Description field conform to the template and spell out for what the component is used?   **Summary**  **Pre-Condition**  **Post-Condition** |
| Parameters | * Are all parameters named correctly? * Do all parameters have the correct Value Type? * Do all input parameters have a default value? * Do all parameters have a description that fully details what and how it is used? * Do all input file parameter descriptions contain data file parameter information? * Are return code parameters only input parameters? |
| Component Step Name | * Does each Step Name clearly show what the test step represents and meet the standards? |
| Component Step Description | * Does each Step Description clearly explain what action is to be taken? * Do parameters in the Description column represent the data that is going to be entered into the application? |
| Component Expected Results | * Does each Step Expected Result clearly explain the outcome performed by the application when the step has completed? * Do parameters in the Expected Result column represent the data that is expected to be returned by or gathered from the application? |
| Component Parameter Values Displayed at Execution | * When the test case is to be executed, will the text of each step with a parameter value be understandable on its own, i.e. does the parameter value have text associated text to explain it? |
| Attachment | * Are all input files are attached * Are the files attached named for what they are used for? * Are input files prefixed with “Input” in their filenames??   For example:  InputCustDetailsData.xls (Indicates an input file that contains customer details.) |

**Automation Scripts Review Check List**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Description | Yes/No/Not Applicable | Additional Remarks |
| 1 | The Test scripts are complete with respect to the Specifications document on which they are based. |  |  |
| 2 | Test data is mentioned explicitly for each test condition. |  |  |
| 3 | Test data for a test condition is complete in the sense that values for all inputs required for testing that condition are specified |  |  |
| 4 | ‘Expected result’ section of each test script is complete and unambiguous. |  |  |
| 5 | Test scripts exist for covering error conditions that can occur. |  |  |
| 6 | All the messages / error codes specified are correct and same as in the Specification document. |  |  |
| 7 | Pre-conditions for executing a test script or a set of test scripts are specified. |  |  |
| 8 | It is specified which test scripts are to be executed together (or in a specific order). |  |  |
| 9 | Database object (e.g. tables, columns, stored procedures, indexes) names wherever used are correct and qualified if required. |  |  |
| 10 | Test scripts exist for checking Performance of various operations e.g. data retrieval and updating, report printing. |  |  |
| 11 | Test scripts have been written to provide planned coverage in testing. |  |  |
| 12 | If branch or condition coverage, then there are at least two test scripts each covering the TRUE and FALSE processing of the branch/condition being tested. |  |  |
| 13 | Behavior at boundary values in loops is being tested for |  |  |
| 14 | Test scripts have been written to test all error handling code in the program. |  |  |

For detailed Functional Automation User Guide refer to

***Cardinal Health Functional Automation Framework document***

**For Automation technical framework refer to**

***Cardinal Automation technical Framework***

## When the Review Process Takes Place

Once the component was created, it is tested by the author to make sure that it works correctly then the reviewer is notified

## Who Reviews the Process

The component will ideally be reviewed by an another engineer of the same team who is knowledgeable of the application and requirement and has knowledge of BPT

## Component Review Documentation

The review Checklist needs to be used to review the components and all the review comments needs to be documented and share with the component creator