Type of Submission: Article

Title: IBM Case Manager
Subtitle: Add documents with an Entry Template using ICM 5.2. Pass properties into the Entry Template and add the committed document into the current Case.

Keywords: icm, entry template, add document

Author: Dave Ward
Job Title: Solution Architect
Email: dave.ward@insight2value.co.uk
Company: Insight 2 Value, IBM Business Partner based in the UK

Abstract: This article builds upon an excellent article found here (https://www.ibm.com/developerworks/community/blogs/e8206aad-10e2-4c49-b00c-fee572815374/entry/learn_how_to_customize_your_case_manager_client_using_scripts?lang=en) and expands upon it to deliver the following:

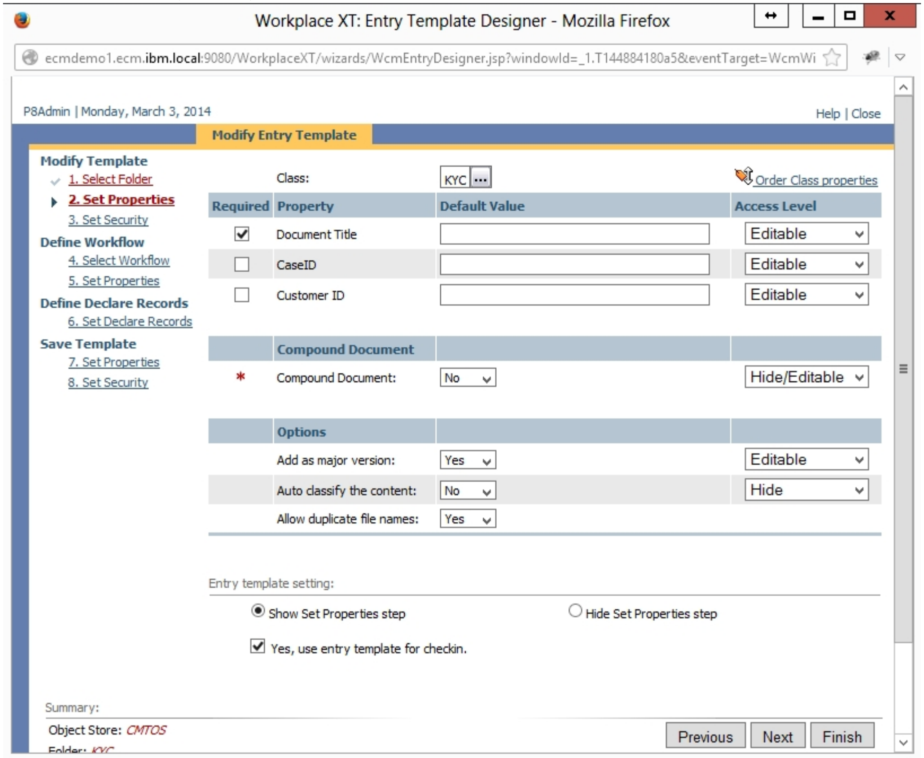
- Adding menus to the Case Info widget.
- Use of Script Actions.
- Use of the Properties Controller Manager to get the most up to date property values.
- Use of ICN API to invoke an Entry Template.
- Pass values from the current ICM case / work object into the Entry Template.
- Make Entry Template property fields read-only (meaning the user cannot amend a value that has been passed into the Entry Template from the corresponding Case property).
- Add the document back into the current Case.

Pre-Requisites

Open Case Builder.
Create a new solution / Edit an existing solution.
Create a Role.
Create a Case Type / Task / etc.
Create some test cases / work items.

Create Entry Template

Create an Entry Template that contains at least one “Editable” field.



Class: KYC

Required	Property	Default Value	Access Level
☑	Document Title		Editable
□	CaseID		Editable
□	Customer ID		Editable
	Compound Document		
*	Compound Document:	No	Hide/Editable
	Options		
	Add as major version:	Yes	Editable
	Auto classify the content:	No	Hide
	Allow duplicate file names:	Yes	

Entry template setting:

● Show Set Properties step ○ Hide Set Properties step

☑ Yes, use entry template for checkin.

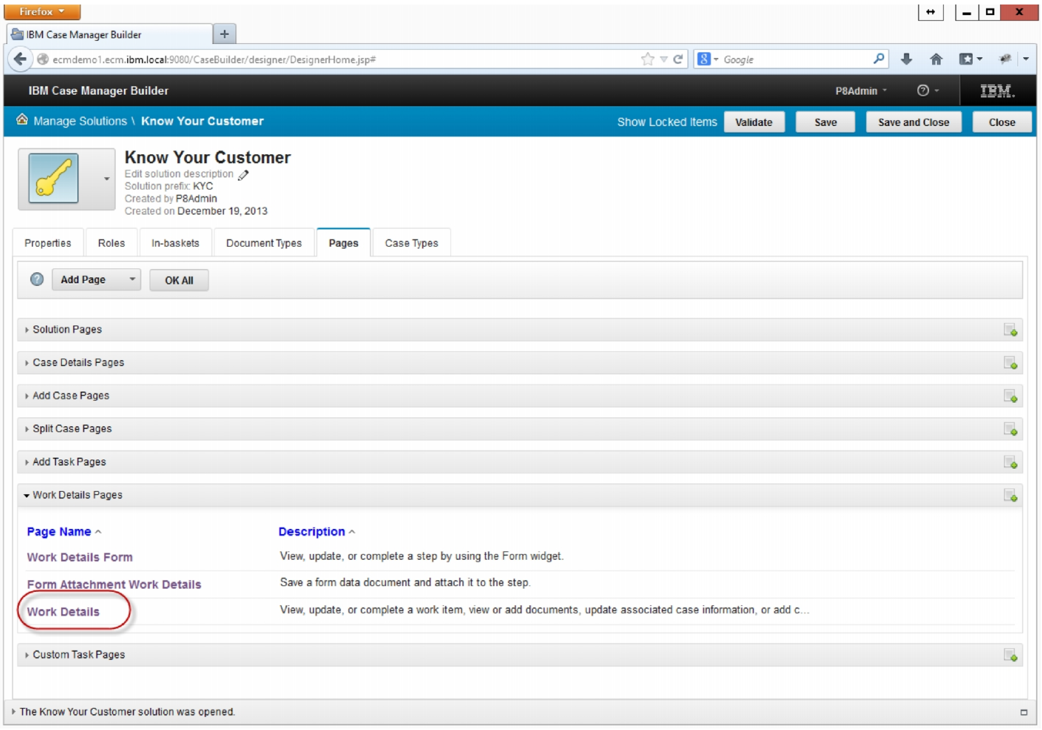
Summary:
Object Store: *CMTOS*
Folder: *KYC*

You need to get the Version Series Id of this Entry Template as it will be used in the JavaScript throughout this example.

Work Details Integration

Create the Work Details menu event action

Open Case Builder.
Edit your solution.
Click on the “Pages” tab.
Expand the “Work Details Pages” group.
Open the “Work Details” page.



Know Your Customer
Edit solution description
Solution prefix: KYC
Created by P8Admin
Created on December 19, 2013

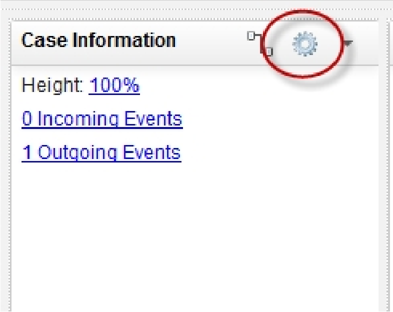
- Solution Pages
- Case Details Pages
- Add Case Pages
- Split Case Pages
- Add Task Pages
- Work Details Pages

Page Name	Description
Work Details Form	View, update, or complete a step by using the Form widget.
Form Attachment Work Details	Save a form data document and attach it to the step.
Work Details	View, update, or complete a work item, view or add documents, update associated case information, or add c...

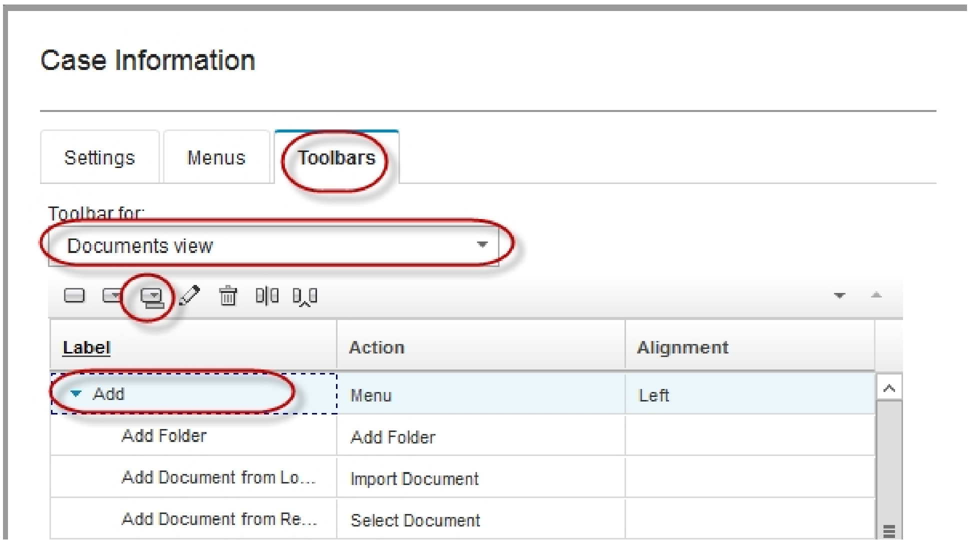
- Custom Task Pages

The Know Your Customer solution was opened.

On the Case Information widget, click the “Edit Settings” icon.


Case Information
Height: 100%
0 Incoming Events
1 Outgoing Events

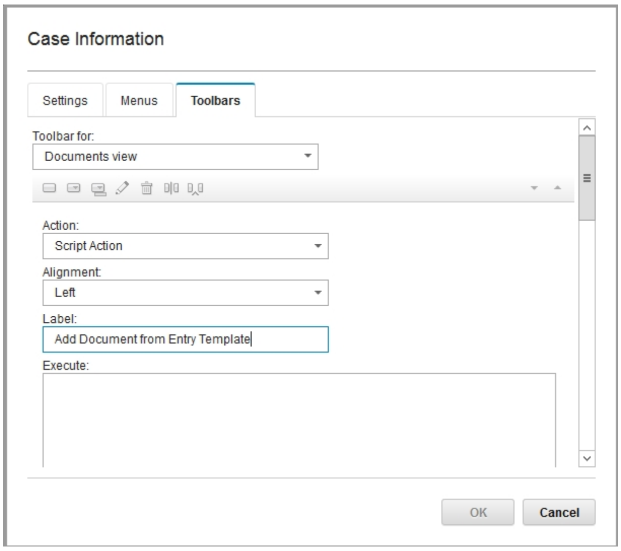
Click the “Toolbars” tab.
Make sure “Documents view” is selected in the “Toolbar for:” list.
Select the existing “Add” row in the Grid.
Click the “Add Menu Item” toolbar button.


Case Information

Toolbar for:
Documents view

Label	Action	Alignment
Add	Menu	Left
Add Folder	Add Folder	
Add Document from Lo...	Import Document	
Add Document from Re...	Select Document	

Create a new toolbar item similar to the following:



Toolbar for: Documents view

Action: Script Action

Alignment: Left

Label: Add Document from Entry Template

Execute:

In the “Execute:” box, paste the following code:

NOTE: You need to add the vsId of your Entry Template to line 9 of the script (the line beginning “`var entryTemplateId =`”)

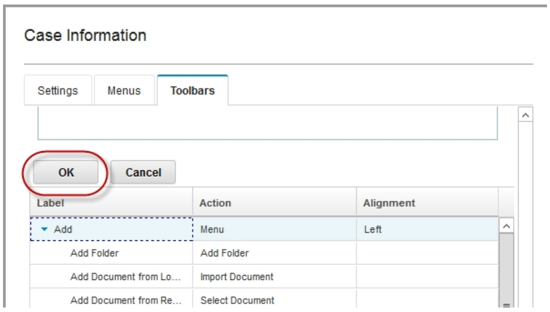
NOTE: You need to add the Id of your Entry Template property to line 53 of the script (the line beginning “`var etPropId =`”)

NOTE: You need to add the Id of your Work Item property to line 54 of the script (the line beginning “`var modelPropId =`”)

```
try {
    var ScriptAdaptorScope = this;

    /*Get the currently selected folder in the Case Info widget*/
    var currentFolder = this.getActionContext().CurrentFolder[0];

    /*The vsId of the Entry Template*/
    /*===== NOTE - CHANGE THIS LINE TO THE VSID OF YOUR ENTRY TEMPLATE =====*/
    var entryTemplateId = "{D4DFCDA1-C9DC-4240-A197-B3036CFFDDD6}";
    /*===== NOTE - CHANGE THIS LINE TO THE VSID OF YOUR ENTRY TEMPLATE =====*/

    var repositoryId = this.getWidget().getSolution().getTargetOS().id;

    /*Get the Work Item Editable*/
    var myWorkItemEditable = this.getWidget().model.payload.workItemEditable;

    require([
        "dojo/_base/declare",
        "dojo/_base/lang",
        "ecm/widget/dialog/AddContentItemDialog"
    ], function(
        declare,
        lang,
        AddContentItemDialog
        )
    {
        /*Get Repository*/
        var repository = ecm.model.desktop.getRepository(repositoryId);

        /*Retrieve the Entry Template*/
        repository.retrieveItem(entryTemplateId, function(item) {
                /*Check the MIME Type to make sure it is an Entry Template*/
                if (item && item.mimetype) {
                    switch (item.mimetype) {
                        case "application/x-filenet-documententrytemplate":
                        case "application/x-filenet-folderentrytemplate":
                        case "application/x-filenet-entrytemplate":
                        case "application/x-filenet-customobjectentrytemplate":
                        case "application/x-filenet-declarerecordentrytemplate":

                            /*Get Entry Template*/
                            var entryTemplate = repository.getEntryTemplateById(item.id,
item.name, item.objectStore);

                            /*Retrieve Entry Template Handler*/
                            var entryTemplateRetrievedHandler = lang.hitch(this,
function(entryTemplate) {
```

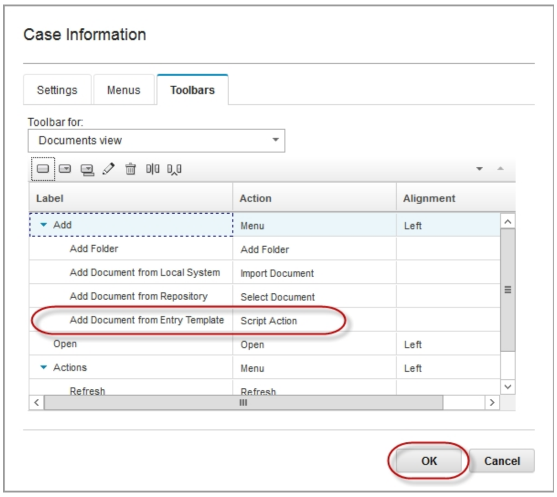
```
                                /*Create AddContentItemDialog*/
                                var myDialog = new AddContentItemDialog();

                                if (entryTemplate) {

                                    /*===== NOTE - CHANGE THIS SECTION TO REFLECT THE ENTRY
TEMPLATE FIELD ID AND THE CORRESPONDING CASE PROPERTY =====*/
                                    var etPropId = "Customer_ID";
                                    var modelPropId = "Customer_ID";
                                    var modelCurrentValue;
                                    /*===== NOTE - CHANGE THIS SECTION TO REFLECT THE ENTRY
TEMPLATE FIELD ID AND THE CORRESPONDING CASE PROPERTY =====*/

                                    /*Get the current values in the work item or case to pass
into the Entry Template*/

require(["icm/model/properties/controller/ControllerManager"], function(ControllerManager){
                                        /*Retrieve prop value using property controller API*/
                                        var theController =
ControllerManager.bind(myWorkItemEditable);
                                        var propController =
theController.getPropertyController(modelPropId);
                                        modelCurrentValue = propController.get("value");
                                        /*Unbind is necessary to prevent mem leak.*/
                                        ControllerManager.unbind(myWorkItemEditable);

                                        /*Set the values in the Entry Template*/
                                        setDefaultValue
(entryTemplate,etPropId,modelCurrentValue, true);

                                        var folder = entryTemplate.folder;
                                        if (folder && !folder.hasPrivilege("privAddToFolder")
&& !entryTemplate.allowUserSelectFolder) {
                                            var message =
ecm.model.Message.createErrorMessage("entry_template_folder_restricted_error", [
                                                entryTemplate.name,
                                                folder.name
                                            ]);
                                            if (message) {
                                                ecm.model.desktop.addMessage(message);
                                            }
                                        } else {
                                            myDialog.show(
                                                repository,
                                                null,
                                                (entryTemplate.type ==
ecm.model.EntryTemplate.TYPE.DOCUMENT),
                                                false,
                                                function(newItem) {
                                                    try {
                                                        /*Add new content item / document to
the current case root/sub folder*/
                                                        currentFolder.addToFolder(newItem,
function() {
                                                            console.log('Document added to
folder');
                                                        });
                                                    }
                                                    catch (Error) {
                                                        alert ("Oops! Error in callback");
                                                    }
                                                },
                                                null,
                                                true,
                                                entryTemplate);
                                        }

                                    });

                                }
                            });
                            if (!entryTemplate.isRetrieved) {
entryTemplate.retrieveEntryTemplate(entryTemplateRetrievedHandler, false, false);
                            } else {
                                entryTemplateRetrievedHandler(entryTemplate);
                            }
                            break;
                    }
                }
            },
            "EntryTemplate",
            "current",
            entryTemplateId);
    });
}
catch (Error) {
    alert ("Source Module: "+ ScriptAdaptorScope.arguments.label +" Script
Adaptor\r\n\r\n"+Error.name+" - "+Error.description+"\r\n"+Error.message);
}

function setDefaultValue (entryTemp,propIdName,propDefaultValue,makeReadOnly) {
    try {
        var i;
        var len;

        if (entryTemp) {
            len = entryTemp.propertiesOptions.length;

            for(i=0; i<len; i++) {
                if (entryTemp.propertiesOptions[i].id == propIdName) {
                    entryTemp.propertiesOptions[i].defaultValue = [propDefaultValue];
                    if (makeReadOnly) {
                        /*Make the field read only*/
                        entryTemp.propertiesOptions[i].readOnly = true;
                    }
                    break;
                }
            }
        }

    }
    catch (Error) {
        alert ("Source Module: "+ ScriptAdaptorScope.arguments.label +" Script Adaptor -
setDefaultValue Function\r\n\r\n"+Error.name+" - "+Error.description+"\r\n"+Error.message);
    }
}
```

Click “OK” to add the menu item.



Label	Action	Alignment
Add	Menu	Left
Add Folder	Add Folder	
Add Document from Lo...	Import Document	
Add Document from Re...	Select Document	

The new toolbar item will be visible in the “Toolbar for: Documents View” list.
Click “OK” to close the Case Information settings dialog.



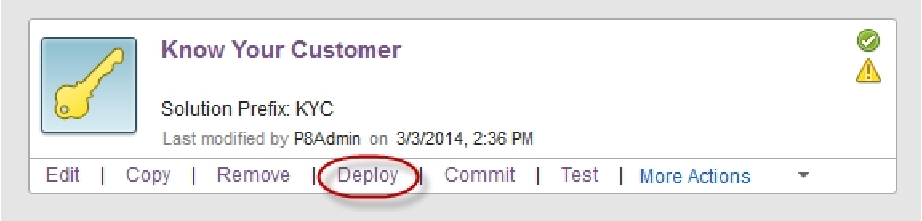
Toolbar for: Documents view

Label	Action	Alignment
Add	Menu	Left
Add Folder	Add Folder	
Add Document from Local System	Import Document	
Add Document from Repository	Select Document	
Add Document from Entry Template	Script Action	
Open	Open	Left
Actions	Menu	Left
Refresh	Refresh	

In the Page Designer, Click “Save” and then “Close”.

In your Solution click “Save and Close”.

In the Case Builder home page, “Deploy” your solution.



Know Your Customer

Solution Prefix: KYC

Last modified by P8Admin on 3/3/2014, 2:36 PM

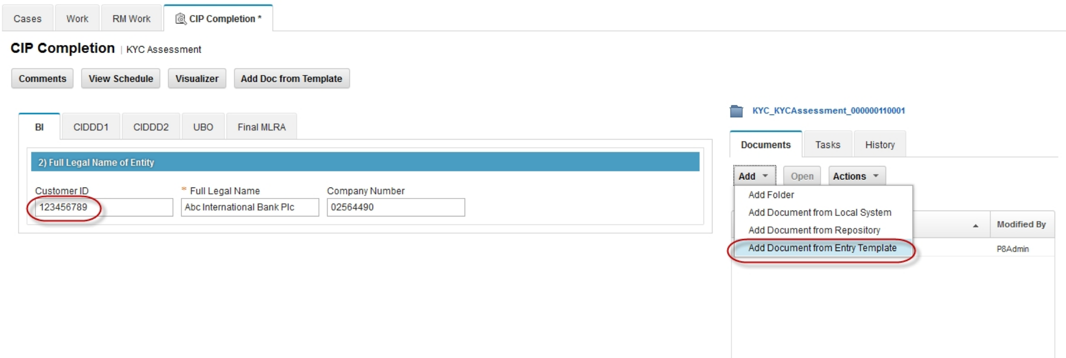
Edit | Copy | Remove | Deploy | Commit | Test | More Actions

Test the Script

Open the Work Details page

Make a change to the Customer ID field (or whatever the field was called in your solution) in the Properties widget.

In the Case Info widget, click “Add” → “Add Document from Entry Template”



CIP Completion | KYC Assessment

2) Full Legal Name of Entity

Customer ID: 123456789

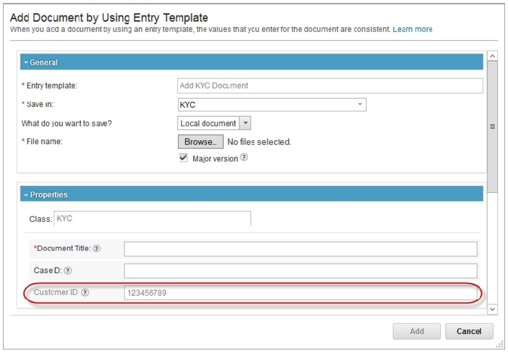
* Full Legal Name: Abc International Bank Plc

Company Number: 02564490

KYC_KYCAssessment_000000110001

Documents
Tasks
History
Add
Open
Actions
Add Folder
Add Document from Local System
Add Document from Repository
Add Document from Entry Template
Modified By
P8Admin

The Entry Template is displayed. The Customer_ID value has been passed from the Work Item into the Entry Template and the field has been made read-only.



Add Document by Using Entry Template

When you add a document by using an entry template, the values that you enter for the document are consistent. Learn more

General

* Entry template: Add KYC Document

* Save in: KYC

What do you want to save? Local document

* File name: Browse... No files selected.

☑ Major version

Properties

Class: KYC

*Document Title:

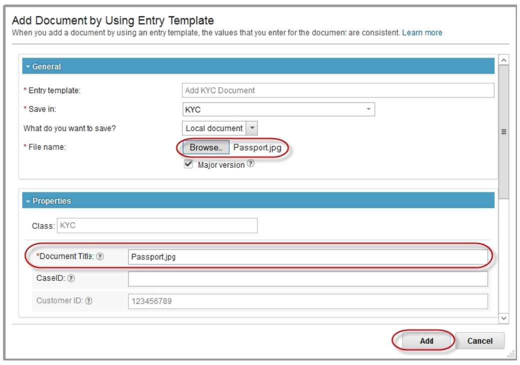
CaseID:

Customer ID: 123456789

Browse to a suitable file name and select it.

Add a suitable Document Title to the Entry Template.

Click the “Add” button.



Add Document by Using Entry Template

When you add a document by using an entry template, the values that you enter for the document are consistent. Learn more

General

* Entry template: Add KYC Document

* Save in: KYC

What do you want to save? Local document

* File name: Browse... Passport.jpg

☑ Major version

Properties

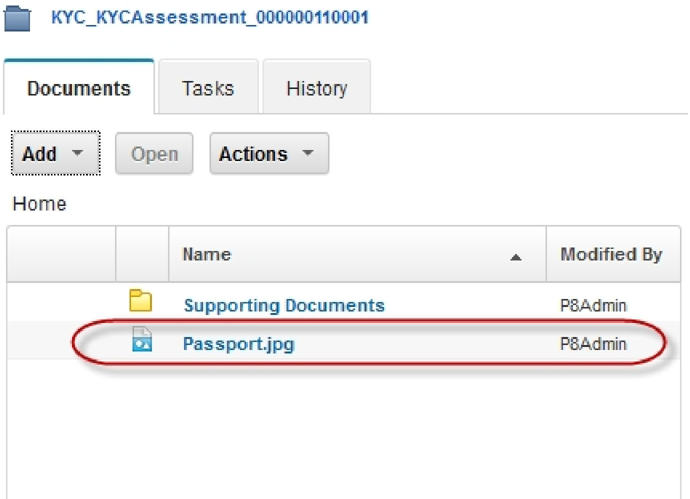
Class: KYC

*Document Title: Passport.jpg

CaseID:

Customer ID: 123456789

The document has been added to the Case folder / sub-folder that was selected in the Case Information widget when the Entry template was invoked.



Name	Modified By
Supporting Documents	P8Admin
Passport.jpg	P8Admin

Case Details Integration

You can achieve the same results in the Case Details page with one slight amendment.
Use this JavaScript instead, which references the current case properties and not the current work item properties.

In the "Execute:" box, paste the following code:
NOTE: You need to add the vsId of your Entry Template to line 9 of the script (the line beginning "`var entryTemplateId =`")
NOTE: You need to add the Id of your Entry Template property to line 53 of the script (the line beginning "`var etPropId =`")
NOTE: You need to add the Id of your Work Item property to line 55 of the script (the line beginning "`var modelPropId =`")

```
try {
    var ScriptAdaptorScope = this;

    /*Get the currently selected folder*/
    var currentFolder = this.getActionContext().CurrentFolder[0];

    /*The vsId of the Entry Template*/
    /*===== NOTE - CHANGE THIS LINE TO THE VSID OF YOUR ENTRY TEMPLATE =====*/
    var entryTemplateId = "{D4DFCDA1-C9DC-4240-A197-B3036CFFDDD6}";
    /*===== NOTE - CHANGE THIS LINE TO THE VSID OF YOUR ENTRY TEMPLATE =====*/

    var repositoryId = this.getWidget().getSolution().getTargetOS().id;

    /*Get current Case*/
    var currentCase = this.getActionContext().Case[0];

    require([
        "dojo/_base/declare",
        "dojo/_base/lang",
        "ecm/widget/dialog/AddContentItemDialog"
    ], function(
        declare,
        lang,
        AddContentItemDialog
        )
    {
        /*Get Repository*/
        var repository = ecm.model.desktop.getRepository(repositoryId);

        /*Retrieve the Entry Template*/
        repository.retrieveItem(entryTemplateId, function(item) {
                /*Check the MIME Type to make sure it is an Entry Template*/
                if (item && item.mimetype) {
                    switch (item.mimetype) {
                        case "application/x-filenet-documententrytemplate":
                        case "application/x-filenet-folderentrytemplate":
                        case "application/x-filenet-entrytemplate":
                        case "application/x-filenet-customobjectentrytemplate":
```

```
                        case "application/x-filenet-declarerecordentrytemplate":

                            /*Get Entry Template*/
                            var entryTemplate = repository.getEntryTemplateById(item.id,
item.name, item.objectStore);

                            /*Retrieve Entry Template Handler*/
                            var entryTemplateRetrievedHandler = lang.hitch(this,
function(entryTemplate) {

                                /*Create AddContentItemDialog*/
                                var myDialog = new AddContentItemDialog();

                                if (entryTemplate) {

                                    /*===== NOTE - CHANGE THIS SECTION TO REFLECT THE ENTRY
TEMPLATE FIELD ID AND THE CORRESPONDING CASE PROPERTY =====*/
                                    var etPropId = "Customer_ID";
                                    var modelPropId = "Customer_ID";
                                    var modelCurrentValue;
                                    /*===== NOTE - CHANGE THIS SECTION TO REFLECT THE ENTRY
TEMPLATE FIELD ID AND THE CORRESPONDING CASE PROPERTY =====*/

                                    /*Get the current values in the work item or case to pass
into the Entry Template*/

require(["icm/model/properties/controller/ControllerManager"], function(ControllerManager){
                                        /*Retrieve prop value using property controller API*/
                                        var theController =
ControllerManager.bind(currentCase);
                                        var propController =
theController.getPropertyController(modelPropId);
                                        modelCurrentValue = propController.get("value");
                                        /*Unbind is necessary to prevent mem leak.*/
                                        ControllerManager.unbind(currentCase);

                                        /*Set the values in the Entry Template*/
                                        setDefaultValue
(entryTemplate,etPropId,modelCurrentValue, true);

                                        var folder = entryTemplate.folder;
                                        if (folder && !folder.hasPrivilege("privAddToFolder")
&& !entryTemplate.allowUserSelectFolder) {
                                            var message =
ecm.model.Message.createErrorMessage("entry_template_folder_restricted_error", [
                                                entryTemplate.name,
                                                folder.name
                                            ]);
                                            if (message) {
                                                ecm.model.desktop.addMessage(message);
                                            }
                                        } else {
                                            myDialog.show(
                                                repository,
                                                null,
                                                (entryTemplate.type ==
ecm.model.EntryTemplate.TYPE.DOCUMENT),
                                                false,
                                                function(newItem) {
                                                    try {
                                                        /*Add new content item / document to
the current case root/sub folder*/
                                                        currentFolder.addToFolder(newItem,
function() {
                                                            console.log('Document added to
folder');
                                                        });
                                                    }
                                                    catch (Error) {
                                                        alert ("Oops! Error in callback");
                                                    }
                                                },
                                                null,
                                                true,
                                                entryTemplate);
                                        }
                                    });
                                }
                            });
                            if (!entryTemplate.isRetrieved) {

entryTemplate.retrieveEntryTemplate(entryTemplateRetrievedHandler, false, false);
                            } else {
                                entryTemplateRetrievedHandler(entryTemplate);
                            }
                            break;
                    }
                }
            },
            "EntryTemplate",
            "current",
            entryTemplateId);
    });
}
catch (Error) {
    alert ("Source Module: "+ ScriptAdaptorScope.arguments.label +" Script
Adaptor\r\n\r\n"+Error.name+" - "+Error.description+"\r\n"+Error.message);
}

function setDefaultValue (entryTemp,propIdName,propDefaultValue,makeReadOnly) {
    try {
        var i;
        var len;

        if (entryTemp) {
            len = entryTemp.propertiesOptions.length;

            for(i=0; i<len; i++) {
                if (entryTemp.propertiesOptions[i].id == propIdName) {
                    entryTemp.propertiesOptions[i].defaultValue = [propDefaultValue];
                    if (makeReadOnly) {
                        /*Make the field read only*/
                        entryTemp.propertiesOptions[i].readOnly = true;
                    }
                    break;
                }
            }
        }

    }
    catch (Error) {
        alert ("Source Module: "+ ScriptAdaptorScope.arguments.label +" Script Adaptor -
setDefaultValue Function\r\n\r\n"+Error.name+" - "+Error.description+"\r\n"+Error.message);
    }
}}
```