# Calculate Client Security Hash

# Contents

# 1.Prerequisites

1. The automation will need to process DataRows for each transaction so the data type for TransactionItem will be DataRow.

# 2.Configuration in UiPath Studio

## 2.1. Create a new project

1. Create a new project using the Robotic Enterprise Framework template.
2. Set a proper name for the project.
3. Provide a proper description.

## 2.2. Whiteboard your workflows

| Module Name | Description | Pre-condition | Post-action | Arguments |
|---|---|---|---|---|
| System1_Login | Opens a Microsoft Edge browser to access System 1 and log in using provided credentials | N/A | Opens Edge System 1 Dashboard if credentials are correct. | in_System1URL - String<br>in_System1Credential - String |
| System1_Close | Closes System 1 in Microsoft Edge browser | System 1 Opened, Log Out button accessible | Logs Out and Closes the browser | N/A |
| System1_NavigateTo_WIDetails | Navigates to the WI Details Page using the WIID input argument | System 1 is open at any page | WI Details opened | in_WIID - string<br>in_System1_WorkItemURL - String |
| System1_NavigateTo_WorkItems | Navigates to Work Items Page | System 1 Dashboard is opened | Work Items Page opened | N/A |
| System1_NavigateTo_UpdateWorkItem | Navigates to Update Work Item Page | System 1 Work Items Page is opened | Update Work Item Page opened | in_WIID - String<br>in_System1_UpdateWorkItemURL – String |
| System1_Extract_WIsDataTable | Extracts the entire Table of all Work Items in the output argument | Work Items Page is opened | None | out_DataTable - DataTable |
| System1_Extract_ClientInformation | Extracts the ClientID, ClientName and ClientCountry and writes to the output arguments | Work Item Details Page is opened | None | out_ClientID - String<br>out_ClientName - String<br>out_ClientCountry - String |
| System1_UpdateWorkItem | Updates the Work Item to the Status and Comment using input arguments. Status can be "Open", "Completed" or "Rejected" | System 1 Update Work Item Page is opened | None | in_Comment - String<br>in_Status - String<br>in_WIID - String |
| SHA1_Open | Opens SHA1 generator from the URL argument | None | Opens Edge SHA1 generator | in_SHA1URL - String |
| SHA1_Close | Closes SHA1 generator | SHA1 generator Main Page is opened | Closes Edge SHA1 generator | N/A |

| Module Name | Description | Pre-condition | Post-action | Arguments |
|---|---|---|---|---|
| SHA1_ProcessHash | Retrieves the HashResult based on the input HashFormula | SHA1 generator is opened | None | in_HashFormula - String<br>out_HashResult - String |
| SendEmail | Sends Exception Email using Outlook and the Input Arguments | None | None | in_To - String<br>in_Subject - String<br>in_Body - String |

## 2.3. Develop your workflows

1. Create three new folders in the project folder. Name them System1, SHA1 and Common.

### *2.3.1. System1_Login.xaml*

1. Inside the System1 folder, create a new Sequence type workflow called System1Login.
2. Provide a description to the workflow.
3. Open the Arguments panel and create two in arguments of type string:
    a. in_System1URL
    b. in_System1Credential
4. Add a Log message activity to mark the start of the workflow execution.
5. Add a Get Credential activity. Configure the following properties:
    a. Asset Name – Provide the in_System1Credential argument.
    b. Password – Create a new variable called Password of type SecureString.
    c. Username – Create a new variable called Username of type String.
6. Open Acme System1 in Edge Browser.
7. Add Use Application/Browser in the System1_Login workflow and indicate the opened browser and replace the URL with the argument in_System1URL.
8. Set properties as below:
    a. Close: Never
    b. Open: IfNotOpen
9. Inside the Do Container of the Use Application/Browser activity add:
    a. A Type Into activity where you indicate the email field in the browser and provide the Username variable for the Text.
    b. A Type Into activity where you indicate the password field in the browser and provide the Password variable for the Secure text.
    c. A Click activity where you indicate the Login button in the browser as target.
10. Log into ACME in your browser.
11. After the Use Application/Browser activity, add a Check App State activity.
    a. Indicate the Dashboard header in the browser as a target.
    b. Create a new Boolean variable LogInSuccessful. And set to true in “Target appears” section and to false in “Target does not appear” section.
12. Add an If activity.
    a. Set the condition to Not LogInSuccessful.
13. Go to ACME, return to the login page, try to log in with the wrong credentials.
14. Return to Studio. Inside the Then block of the If activity:

a. Add a Throw activity with the Exception property set to: new Exception("Incorrect Credentials supplied to System1")
b. Note that later during development, we will invoke a workflow here to send an email to the process owner.

15. After the If activity, add a Log Message to mark the end of the workflow.

### *2.3.2. System1_Close.xaml*

1. Ensure you are logged into ACME System1 in your web browser.
2. In Studio, include a Log Message activity to indicate the start of the workflow.
3. Add a Use Application/Browser activity and specify the browser that is logged into ACME.
4. Set the properties as below:
    a. Close: Always
5. Within the Do container of the Use Application/Browser activity:
    a. Add a Click activity and indicate the Log Out button.
6. Add a Log Message activity to signify the end of the workflow.

### *2.3.3. System1_NavigateTo_WIDetails.xaml*

1. Open the Arguments panel and establish a new in argument of String type named "in_WIID" and "in_System1_WorkItemURL."
2. Ensure that the ACME System1 Work Items page is open in your browser.
3. In Studio, add a Log Message activity to indicate the start of the workflow.
4. Add a Use Application/Browser activity and specify the browser logged into ACME.
5. Within the Do container of the Use Application/Browser activity:
    a. Add a Go To URL activity and set the URL property to in_System1_WorkItemURL + "/" + in_WIID.
6. Add a Log Message activity to signify the end of the workflow.

### *2.3.4. System1_NavigateTo_UpdateWorkItem.xaml*

1. Open the Arguments panel and establish a new in argument of String type named "in_WIID" and "in_System1_UpdateWorkItemURL."
2. Ensure that the ACME System1 Work Items page is open in your browser.
3. In Studio, add a Log Message activity to indicate the start of the workflow.
4. Add a Use Application/Browser activity and specify the browser logged into ACME.
5. Within the Do container of the Use Application/Browser activity:
    a. Add a Go To URL activity and set the URL property to in_System1_UpdateWorkItemURL + "/" + in_WIID.
6. Add a Log Message activity to signify the end of the workflow.

### *2.3.5. System1_NavigateTo_WorkItems.xaml*

1. Add a Log Message activity to mark the beginning of the workflow.
2. Ensure that the ACME System 1 Dashboard page is open in your web browser.
3. Add a Use Application/Browser activity and indicate the ACME System 1 Dashboard page.
4. In Studio, insert a Click activity and specify the "Work Items" button.
5. Add a Log Message activity to mark the conclusion of the workflow.

### *2.3.6. System1_Extract_WIsDataTable.xaml*

1. Add a Log Message activity to signify the start of the workflow.
2. Open the Arguments panel and establish a new out argument of DataTable type named "out_DataTable."
3. In your web browser, navigate to the Work Items page on ACME System1.
4. In Studio, utilize the Data Scraping wizard to extract the work items data. Configure its properties as follows:
    a. For the Output DataTable property, input "out_DataTable."
    b. Set the Number of items property to '0'.
5. Add a Log Message activity to signify the conclusion of the workflow.

### *2.3.7. System1_Extract_ClientInformation.xaml*

1. Add a Log Message activity to signify the start of the workflow.
2. Open the Arguments panel and create three out arguments of String type:
    a. out_ClientID
    b. out_ClientName
    c. out_ClientCountry
3. Navigate to the Work Items page in your web browser.
4. In Studio, add a Use Application/Browser activity and indicate the Work Item Details page.
5. Inside the Do container of the Use Application/Browser activity, add a Get Text activity, indicate the Client Information Details text box of a WI5 details page and configure the following properties:
    a. Selector - "<webctrl tag='P' idx='1' />"
    b. Save to – Create a new String type variable called ClientInformation
6. After the Use Application/Browser activity, add a Multiple Assign activity with the following rows:
    a. out_ClientID = ClientInformation.Substring(ClientInformation.IndexOf("Client ID: ") + "Client ID: ".Length).Split(Environment.NewLine.ToCharArray)(0)
    b. out_ClientName = ClientInformation.Substring(ClientInformation.IndexOf("Client Name: ") + "Client Name: ".Length).Split(Environment.NewLine.ToCharArray)(0)
    c. out_ClientCountry = ClientInformation.Substring(ClientInformation.IndexOf("Client Country: ")+"Client Country: ".Length).Split(Environment.NewLine.ToCharArray)(0)
7. Add a Log Message activity to note value of each out argument and mark the end of the workflow.

### 2.3.8. System1_Update_WorkItem.xaml

1. Add a Log Message activity to mark the start of the workflow.
2. Open the Arguments panel and create three in arguments of type string:
    a. in_Comment
    b. in_Status
    c. in_WIID
3. Ensure that the System1 Update Work Item page is opened on your browser.
4. In Studio, add a Use Application/Browser activity, indicate the current page, and modify the selector to attach to only Update Work Item pages.
5. In Studio, add a Type Into activity, indicate the Comments field and provide in_Comment for the type of this field.
6. Add a Click activity and indicate the status drop-down. Configure the following properties:
    a. Selector - "<webctrl aaname='"+in_Status+"' tag='A' />"
7. Add a Click activity and indicate the Update Work Item button.
8. In your browser, click Update Work Item.
9. In workflow, add a Click activity and indicate the OK button.
10. In your browser, close the Work Item update message window.
11. Add a Log Message activity to mark the end of the workflow.

### 2.3.9. SHA1_Open.xaml

1. Open the SHA1 generator page in your browser.
2. In the SHA1 folder in the project panel, create a new sequence type workflow called SHA1_Open.
3. In the Arguments panel, create a new in argument of type string called in_SHA1URL.
4. Add a Log Message activity to mark the start of the workflow.
5. Add a Use Application/Browser activity, indicate the SHA1 generator page and provide in_SHA1URL for the URL.
6. Set properties as below:
    a. Close: Never
    b. Open: IfNotOpen
7. Add a Log Message activity to mark the end of the workflow.

### 2.3.10. SHA1_Close.xaml

1. Open the SHA1 generator page in your browser.
2. In Studio, add a Log Message activity to mark the start of the workflow.
3. Add a Use Application/Browser activity and indicate the SHA1 page.
4. Set property as below:
    a. Close: Always
5. After the Use Application/Browser activity, add a Log Message activity to mark the end of the workflow.

### 2.3.11. SHA1_ProcessHash.xaml

1. Open the Arguments panel and create an out and an in argument of type string:
    a. out_HashResult
    b. in_HashFormula
2. Add a Log Message activity to mark the start of the workflow.
3. Add an Use Application/Browser activity and indicate the SHA1 page.
4. Inside the Do Container of the Use Application/Browser activity add the logic to:
    a. Input the in_HashFormula value into the designated field of the web page.
    b. Generate the Hash code.
    c. Retrieve the resulting value and store it in the out_HashResult argument.
    d. Return to the initial state of the page.
5. After the Use Application/Browser activity, add a Log Message activity to mark the end of the workflow.

### 2.3.12. SendEmail.xaml

1. Inside the Common folder, create a new workflow file called SendEmail.xaml.
2. Open the Arguments panel and create three in arguments of type String.
    a. in_To
    b. in_Subject
    c. in_Body
3. Add a Log Message activity to mark the start of the workflow.
4. Add a Send Outlook Mail Message activity with the following properties:
    a. To = in_To
    b. Subject = in_Subject
    c. Body = in_Body
5. Add a Log Message activity to mark the end of the workflow.
6. Go to System1_Login.xaml.
7. Invoke SendEmail.xaml as the first activity inside the Then block of the If activity.
8. Click Import Arguments and map the following values:
    a. in_To = “exceptions@acme-test.com”
    b. in_Subject = "Incorrect Credentials"
    c. in_Body = "Hello,"+Environment.NewLine+"The email or the password is incorrect. Please check and restart"+Environment.NewLine+"Thank you"

## 2.4. Edit the Configuration file

1. Provide the following values in the Settings sheet:

| Name | Value |
|---|---|
| System1_URL | http://www.acme-test.com/ |
| System1_Credential | System1_Credential |
| SHA1URL | The URL for your preferred SHA1 generator. |
| logF_BusinessProcessName | ACME - Calculate Client Security Hash |
| WorkItemType | WI5 |

| | |
|---|---|
| WorkItemStatus | Open |
| ACMESystem1_WorkItemURL | https://acme-test.uipath.com/work-items |
| ACMESystem1_UpdateWorkItemURL | https://acme-test.uipath.com/work-items/update |

2. In the Constants sheet, set the MaxRetryNumber to 2.
3. In Orchestrator, create a Credential type Asset for ACME System 1 with the name System1_Credential.

# 3. Change the TransactionItem Data Type

1. In Main.xaml, locate the TransactionItem variable in the Variables panel; change the Variable type to the DataRow type;
2. The dt_TransactionData variable does not need to be changed as by default it is of type DataTable.
3. In the Get Transaction Data state:
    a. Locate the End Process (Stop process requested) activity and set the value to Nothing
    b. Locate the Invoke GetTransactionData workflow activity and open the Arguments list; change the data type for the out_TransactionItem argument from QueueItem to DataRow. Click OK to save the changes and close the dialog.
    c. Expand the Exception section in the Try GetTransactionData section and locate the End Process (Could not get new transaction) activity; the change here is like the one performed at the beginning of this list.
4. In the Process Transaction state:
    a. Locate the Invoke Process workflow activity and open the Arguments list; change the data type for the in_TransactionItem argument from QueueItem to DataRow. Click OK to save the changes and close the dialog;
    b. Locate the Invoke SetTransactionStatus workflow activities (including the Business Exception and System Exception scenarios) and open the Arguments list; locate the entry for the in_TransactionItem argument and change the value not the data type!) from TransactionItem to Nothing.

# 4. Applications Used: open/close/kill

## 4.1. Edit the InitiAllApplications.xaml workflow

1. Add a Log Message activity to mark the start of the workflow.
2. Invoke the System1\System1_Login.xaml workflow.
    a. Click Import Arguments and make the following changes:

| Name | Direction | Type | Value |
|---|---|---|---|
| in_System1URL | In | String | in_Config("System1_URL").ToString |

| | | | |
|---|---|---|---|
| in_System1Credential | In | String | in_Config("System1_Credential").ToString |

3. Invoke the SHA1\SHA1_Open.xaml workflow.
    a. Click Import Arguments and make the following changes:

| Name | Direction | Type | Value |
|---|---|---|---|
| in_SHA1URL | In | String | in_Config("SHA1URL").ToString |

4. Add a Log Message activity to mark the end of the workflow.

## 4.2. Edit the Framework/CloseAllApplication.xaml workflow

1. Add a Log Message activity to mark the start of the workflow.
2. Invoke the System1\System1_Close.xaml workflow.
3. Invoke the Invoke the System1\SHA1_Close.xaml workflow.
4. Add a Log Message activity to mark the end of the workflow.

## 4.3. Edit the Framework/KillAllProcesses.xaml workflow

1. Add a Log Message activity to mark the start of the workflow.
2. Add a Kill Process activity.
    a. For the ProcessName property, enter the process name for your browser (for example, “msedge” for Edge)
3. Add a Log Message activity to mark the end of the workflow.

# 5. Business Process: Transaction Data and Process

## 5.1. Edit GetTransactionData.xaml workflow

1. Change the data type for the out_TransactionItem argument from QueueItem to DataRow.
2. Delete the Get Transaction Item activity along with the Retry Scope and Try Catch activities that surrounds it.
3. Add an If activity at the start of the workflow.
    a. Name it “If first execution”
    b. Set the Condition to: io_dt_TransactionData is Nothing
4. Inside the Then Branch of the If activity:
    a. Invoke System1\System1_NavigateTo_WorkItems.xaml
    b. Invoke System1\System1_Extract_WIsDataTable.xaml
        i. Click Import Arguments and map the io_dt_TransactionData to the out_DataTable argument.

c. Add a Filter Data Table activity.
    i. Set the input and output DataTable to io_TransactionData
    ii. Set the Filtering Mode to keep
    iii. Set “Type” = in_Config("WorkItemType").ToString “WI5”
    iv. Add another row and set “Status” = in_Config("WorkItemStatus").ToString
    v. Add an Info level Log Message activity with the message: "No. of Transactions: "+ io_TransactionData.Rows.Count.ToString

5. Add another If activity
    a. Name it: “If rows to process”
    b. Set the Condition to: in_TransactionNumber <= io_dt_TransactionData.Rows.Count
6. Inside the Then Branch of the If activity, add an Assign activity:
    a. To = out_TransactionItem
    b. Value = io_dt_TransactionData(in_TransactionNumber - 1)
7. Inside the Else Branch add another Assign activity:
    a. To = out_TransactionItem
    b. Value = Nothing

## 5.2. Edit Process.xaml workflow

1. Change the data type for the in_TransactionItem argument from QueueItem to DataRow.
2. Create four string type variables:
    a. ClientID
    b. ClientName
    c. ClientCountry
    d. HashResult
3. Invoke System1\System1_NavigateTo_WIDetails.xaml.
    a. Click Import Arguments and make the following changes:

| Name | Direction | Type | Value |
|---|---|---|---|
| in_WIID | In | String | in_TransactionItem("WIID").ToString |
| in_System1_WorkItemURL | In | String | in_Config("ACMESystem1_WorkItemURL").ToString |

4. Invoke System1\System1_Extract_ClientInformation.xaml.
    a. Click Import Arguments and make the following changes:

| Name | Direction | Type | Value |
|---|---|---|---|
| out_ClientID | Out | String | ClientID |
| out_ClientName | Out | String | ClientName |
| out_ClientCountry | Out | String | ClientCountry |

5. Invoke SHA1\SHA1_ProcessHash.xaml

a. Click Import Arguments and make the following changes:

| Name | Direction | Type | Value |
|---|---|---|---|
| out_HashRestult | Out | String | HashResult |
| in_HashFormula | In | String | ClientID+"-"+ClientName+"-"+ClientCountry |

6. Invoke System1\System1_NavigateTo_UpdateWorkItem.xaml.
    a. Click Import Arguments and make the following changes:

| Name | Direction | Type | Value |
|---|---|---|---|
| in_WIID | In | String | in_TransactionItem("WIID").ToString |
| in_System1_UpdateWorkItemURL | In | String | in_Config("ACMESystem1_UpdateWorkItemURL").ToString |

7. Invoke System1\System1_Update_WorkItem.xaml
    a. Click Import Arguments and make the following changes:

| Name | Direction | Type | Value |
|---|---|---|---|
| in_Comment | In | String | HashResult |
| in_Status | In | String | “Completed” |
| in_WIID | In | String | in_TransactionItem(“WIID”).ToString |