

ABBYY FlexiCapture

Connector for Automation Anywhere Guide

Table of Contents

About the ABBYY FlexiCapture Connector for Automation Anywhere 3

System requirements 3

Installation 3

Usage scenarios 4

Configuring ABBYY FlexiCapture 4

Configuring Automation Anywhere 10

Additional information 12

 Verification in the attended service scenario 12

 Passing registration parameters to ABBYY FlexiCapture 12

 FlexiCapture MetaBot logics 13

 Export logic 13

 Import logic 15

 Automation Anywhere sample tasks 16

 ExportToFlexiCaptureExample 17

 ImportFromFlexiCaptureExample 18

About the ABBYY FlexiCapture Connector for Automation Anywhere

ABBYY FlexiCapture enables users to perform high-quality full-text recognition, classify documents, detect fields and capture their data, and convert files into various formats (e.g. searchable PDF).

The ABBYY FlexiCapture Connector for Automation Anywhere is intended for sending document files from Automation Anywhere to ABBYY FlexiCapture and loading document files and captured data into the Automation Anywhere robotic process automation software.

ABBYY FlexiCapture Connector for Automation Anywhere is implemented as a MetaBot for ABBYY FlexiCapture.

System requirements

The ABBYY FlexiCapture Connector for Automation Anywhere supports:

- ABBYY FlexiCapture 12 Distributed and ABBYY FlexiCapture 12 Cloud;
- Automation Anywhere Enterprise 11.2.

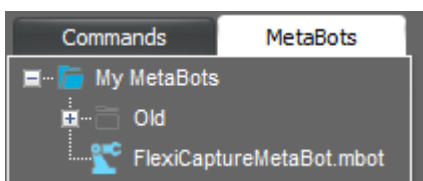
Required External Components:

- Microsoft .NET Framework 4.5.

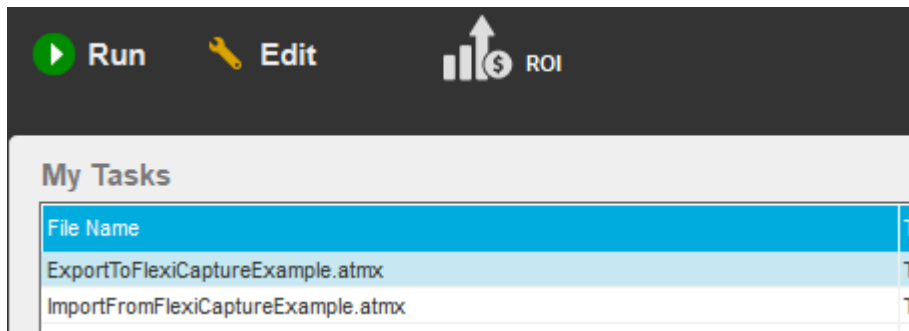
Important! To be able to send documents from Automation Anywhere to ABBYY FlexiCapture, you need to open the following port in the firewall to the Application Server: port 80 (if HTTP is used) or port 443 (if HTTPS is used).

Installation

ABBYY FlexiCapture Connector for Automation Anywhere is distributed as a self-extracting RAR archive that includes a MetaBot file and sample tasks. To install the FlexiCapture MetaBot, copy the **FlexiCaptureMetaBot.mbot** file into this folder: %Userprofile%\Documents\Automation Anywhere Files\Automation Anywhere\My MetaBots. The FlexiCapture MetaBot will appear in the list of available MetaBots in the Task Editor:



To install the [sample tasks](#), copy the **ExportToFlexiCaptureExample.atmx** and **ImportFromFlexiCaptureExample.atmx** files into this folder: %Userprofile%\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks. The task will appear in the **My Tasks** list in the main window of the Automation Anywhere Enterprise Client:



Note: If, when installing the Automation Anywhere Client, you chose to place the runtime files folder in the **Program Data** folder, be sure to replace %Userprofile%\Documents with %ProgramData% in all the paths in this guide.

Usage scenarios

The ABBYY FlexiCapture Connector for Automation Anywhere can typically be used in the following scenarios.

1. [Unattended service scenario](#), where files are sent synchronously from Automation Anywhere to ABBYY FlexiCapture 12; no verification is required.

The FlexiCapture MetaBot component, which runs on the Automation Anywhere side, uses the [Export](#) logic to send a file to ABBYY FlexiCapture 12 and gets the name of the matching Document Definition and the processing results from ABBYY FlexiCapture 12.

2. [Attended service scenario](#), where files are sent asynchronously from Automation Anywhere to ABBYY FlexiCapture 12; verification is required.

The FlexiCapture MetaBot component, which runs on the Automation Anywhere side, uses the [Export](#) logic to send a file to ABBYY FlexiCapture 12 and gets back a URL linking to a document verification page on the Web Verification Station. The user clicks the provided link and verifies the document. Once the verification is completed, the [Import](#) logic is applied to get the name of the matching Document Definition and the processing results from ABBYY FlexiCapture 12.

To implement a usage scenario, you need to:

1. [Configure ABBYY FlexiCapture](#).
2. [Configure Automation Anywhere](#).

Configuring ABBYY FlexiCapture

Before you start configuring ABBYY FlexiCapture, make sure that the following requirements are met:

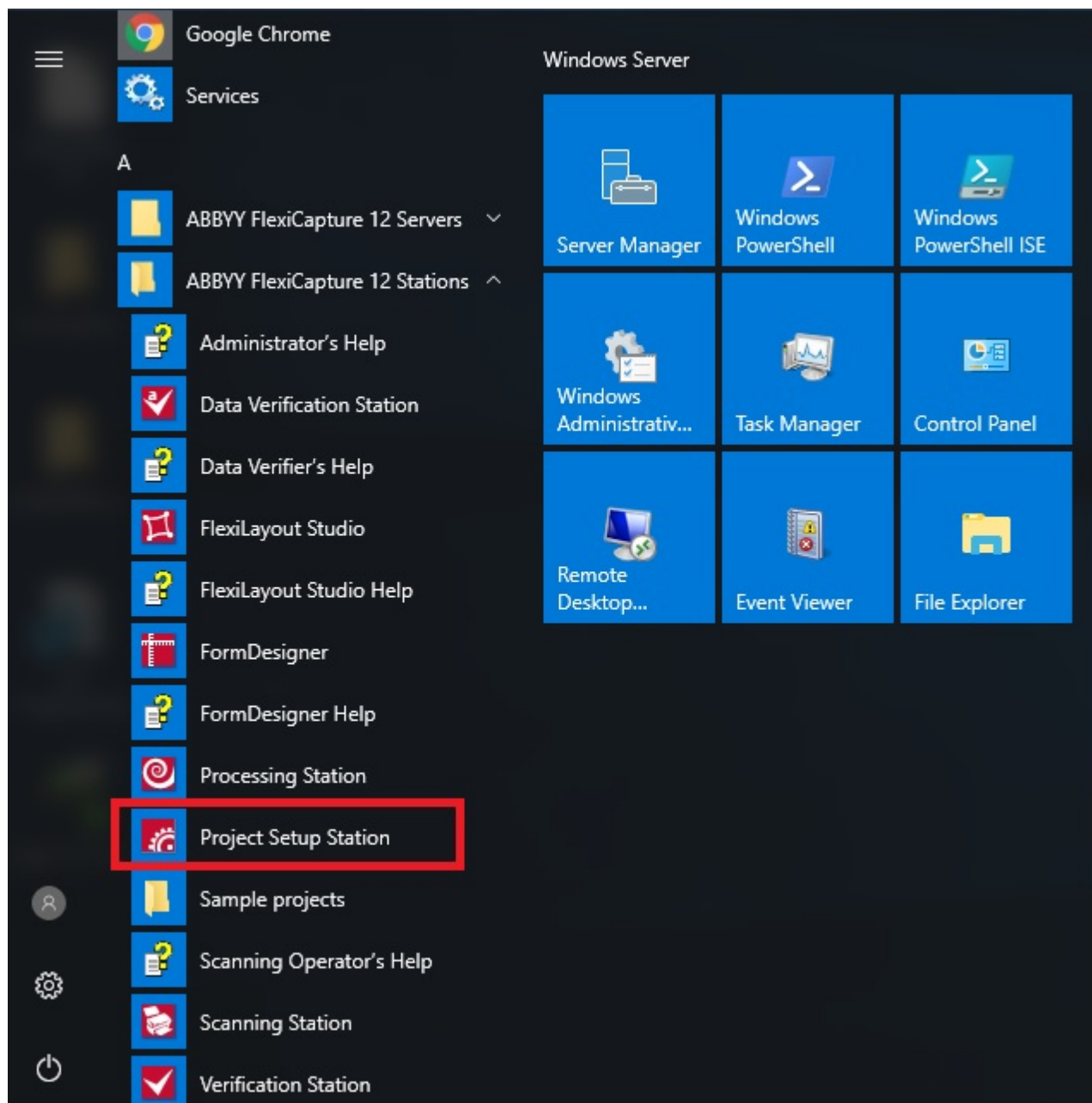
- The ABBYY FlexiCapture 12 Project Setup Station is installed.
- The ABBYY FlexiCapture 12 Application Server administrator's account is configured.

Uploading the sample ABBYY FlexiCapture project to the Application Server

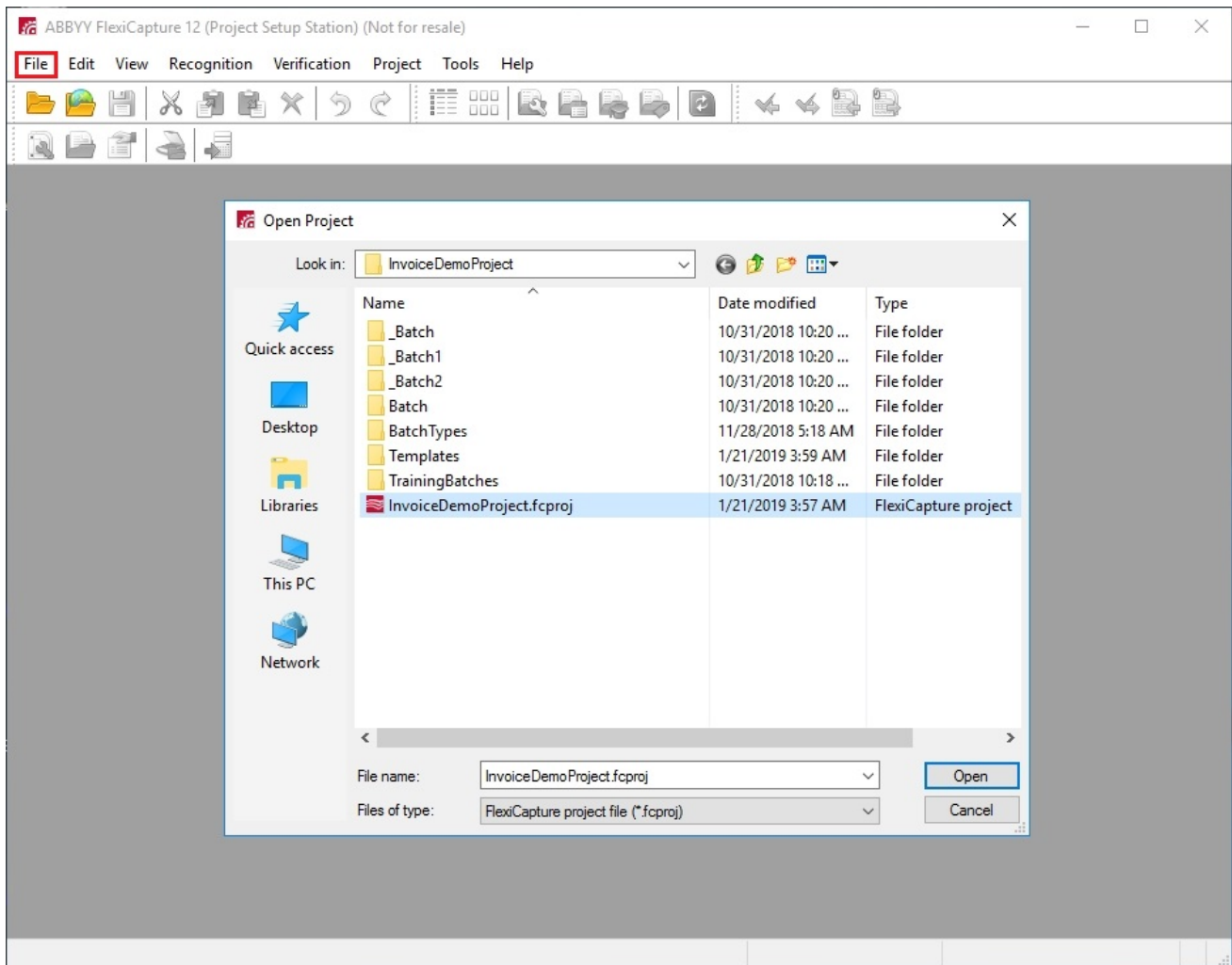
To help you get to grips with ABBYY FlexiCapture Connector for Automation Anywhere, a sample ABBYY FlexiCapture project is included in the distribution package, which demonstrates how the

connector can be used to extract data from invoices. To upload this sample project to the ABBYY FlexiCapture server, complete the following steps:

1. Open the Project Setup Station.



2. Click **File** → **Open Project...**, browse to ABBYY FlexiCapture Connector for Automation Anywhere\Samples\FlexiCapture Projects\InvoiceDemoProject, and select the project named **InvoiceDemoProject.fcproj**.



3. Click **File** → **Upload Project to Server...** to upload the project to the ABBYY FlexiCapture server.

4. In the dialog box that opens, specify the address of the ABBYY FlexiCapture server and the ABBYY FlexiCapture administrator credentials.

Note: If you are using a tenant, specify the name of the tenant in the corresponding field.

5. Click **OK**.

Now you can configure the sample project for the usage scenario of your choice.

Unattended service scenario

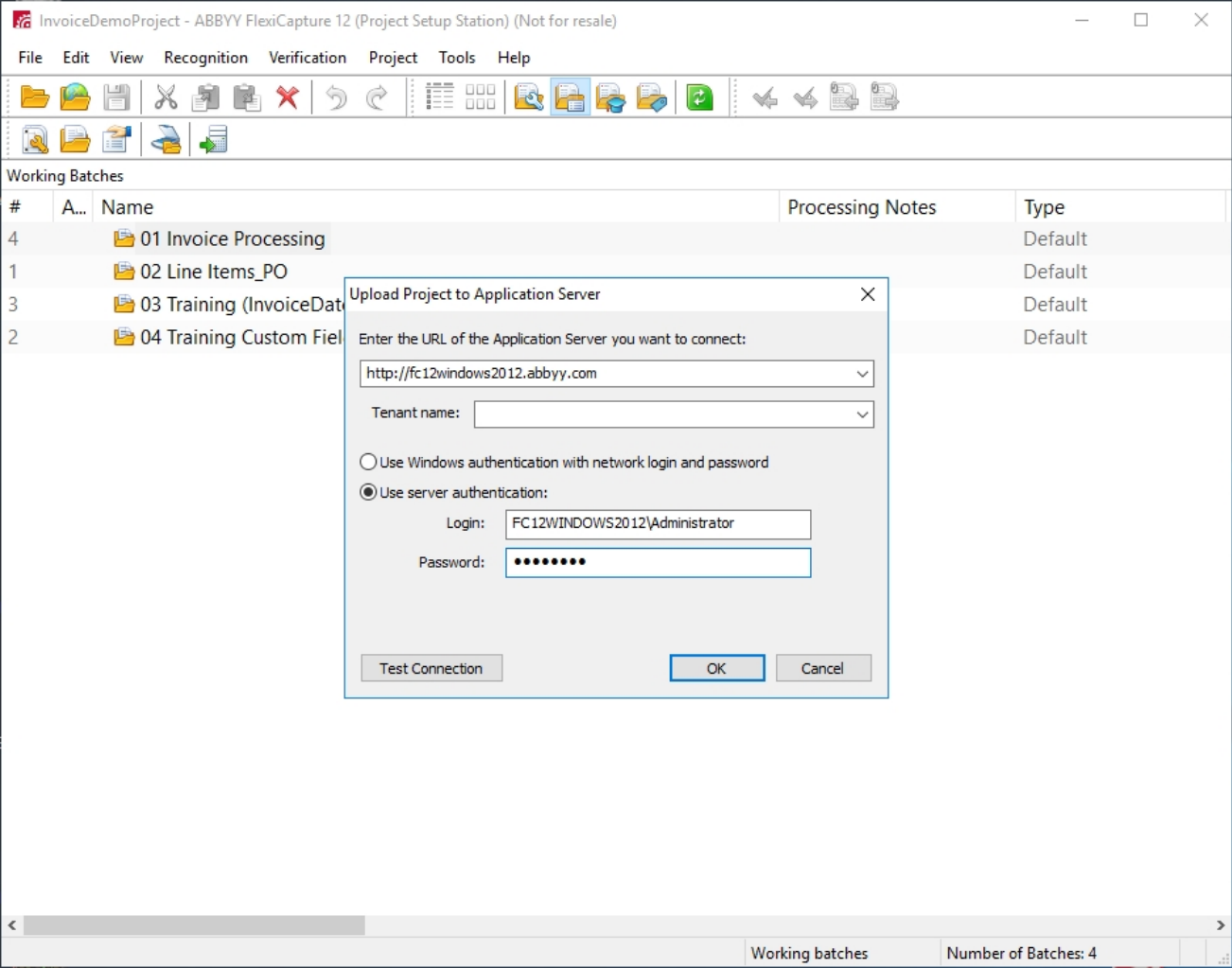
To configure ABBYY FlexiCapture as an unattended service, complete the following steps:

1. Open the Project Setup Station.

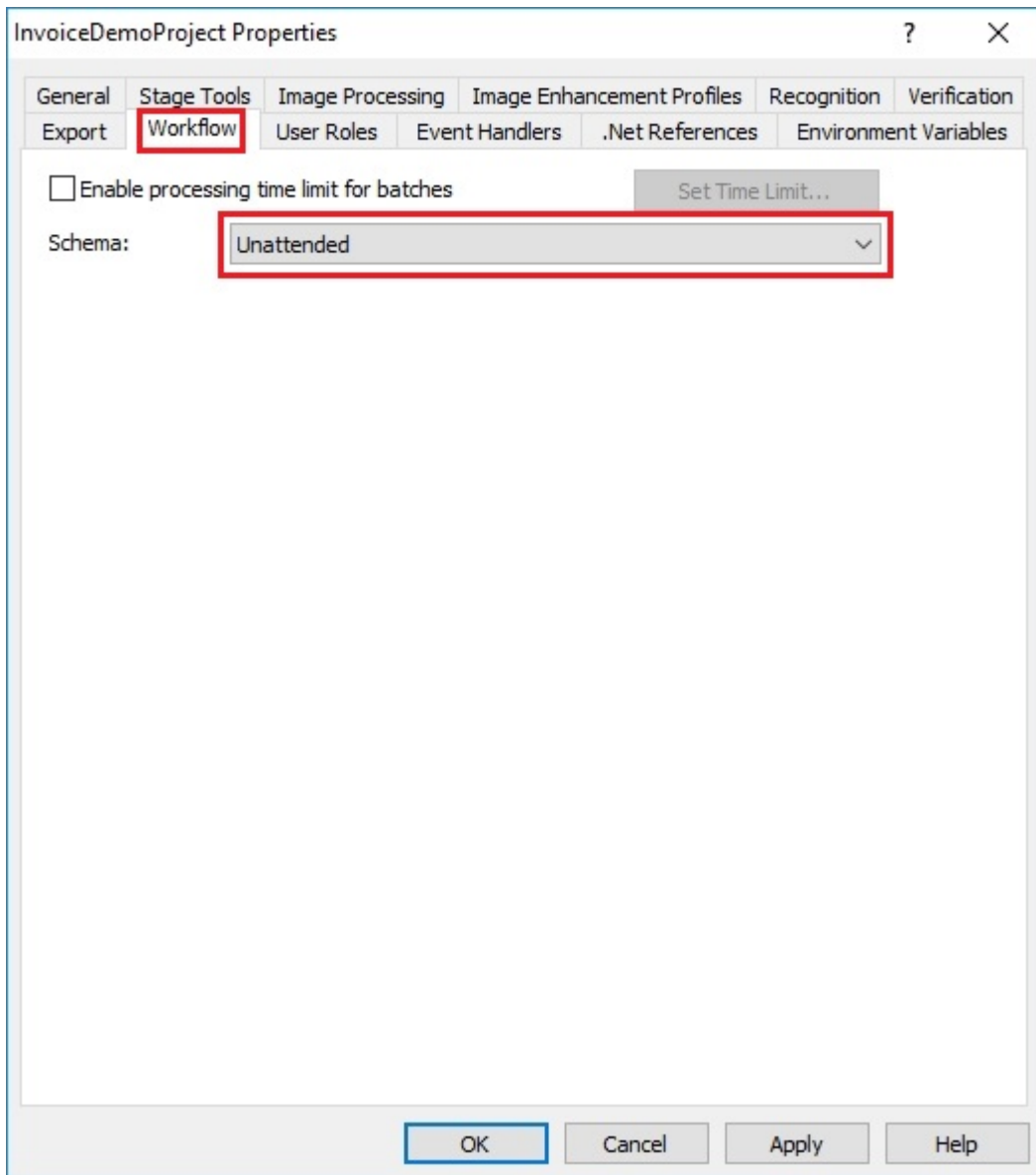
2. Click **File** → **Open Project from Server...**

3. In the **Open Existing Project from Application Server** dialog box, specify the address of the ABBYY FlexiCapture server and the FlexiCapture administrator credentials.

Note: If you are using a tenant, specify the name of the tenant in the corresponding field.



- 4. In the dialog box that opens, select the desired project.
- 5. Click **Project** → **Project Properties...** and click the **Workflow** tab. From the **Schema** drop-down list, select **Unattended**.




6. Click **OK**.

Next, you need to [configure Automation Anywhere](#).

Attended service scenario

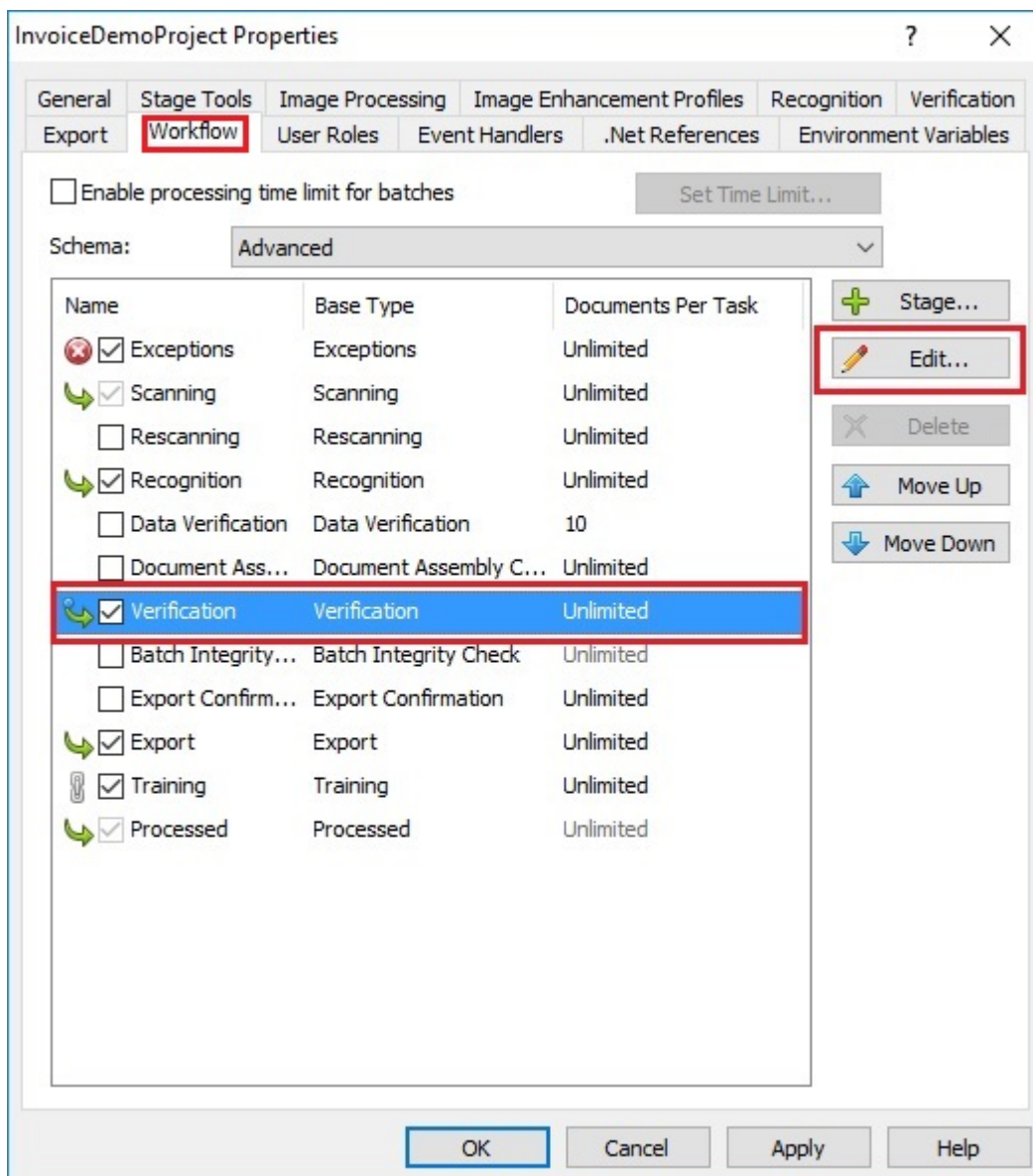
To configure ABBYY FlexiCapture as an attended service, complete the following steps:

1. Open the Project Setup Station.
2. Click **File** → **Open Project from Server....**
3. In the **Open Existing Project from Application Server** dialog box, specify the address of the ABBYY FlexiCapture server and the FlexiCapture administrator credentials.

 **Note:** If you are using a tenant, specify the name of the tenant in the corresponding field.

4. In the dialog box that opens, select the desired project.
5. Click **Project** → **Project Properties...** and click the **Workflow** tab. From the **Schema** drop-down list, select **Advanced**.

6. Select the **Verification** option.



7. Click **Edit....**

8. In the dialog box that opens, click the **General** tab and select the **Enable web stations** option.

9. Click **OK**.

10. Click **OK** again to close the **Project Properties...** dialog box.

Next, you need to [configure Automation Anywhere](#).

Configuring Automation Anywhere

Before you start configuring Automation Anywhere, make sure that the following requirements are met:

- The Automation Anywhere user has a Bot Creator license to be able to configure input parameters.
- The FlexiCapture user has roles of Scanning Operator and Verification Operator or the role of Administrator.


Sending documents to ABBYY FlexiCapture (attended and unattended service scenarios)

To send a file to ABBYY FlexiCapture:


1. Open the Automation Anywhere Enterprise Client.
2. In the list of available tasks, select **ExportToFlexiCaptureExample** and click **Edit**.
3. Specify the path to the ABBYY FlexiCapture server. To do this, double-click the **Variable Operation** action of the **\$ServerURL\$** and type in the server address in the dialog box that opens.

4. Specify the values of the other required parameters:

- a. **\$Username\$** – the name of the user on the ABBYY FlexiCapture server
- b. **\$Password\$** – the password of the user on the ABBYY FlexiCapture server
- c. **\$ProjectName\$** – the name of the uploaded project (i.e. **InvoiceDemoProject**)
- d. **\$InputFilePath\$** – the path to one of the sample images provided with the MetaBot


 **Note:** The activity only works with files containing one document each (i.e. each document must be placed in a separate file).

- e. **\$OutputDirPath\$** – the path to the folder where processing results will be stored

 **Note:** If you are using tenants, **\$Company\$** is also a required parameter. To enable this parameter, right-click the **Variable Operation** action and click **Enable** for the **\$Company\$** parameter. Next, specify the name of your tenant.

5. To test-run the task, save the changes by clicking **Save** in the **Task Editor** and click **Run**.

The image file will be sent to ABBYY FlexiCapture and you will get back either a link to the verification page (in the case of **an attended scenario**) or processing results (in the case of **an unattended scenario**).

 **Note:** If an error occurs while the task is running, information about the error will be recorded in a log file, which can be found here: %Userprofile%\Documents\Automation Anywhere Files\Log\My Tasks\Bot Store\FC-MetaBot\ExporterErrorLog.txt.

Getting processing results in the attended service scenario

To get verified processing results:


1. Complete the steps in [Sending documents to ABBYY FlexiCapture \(attended and unattended service scenarios\)](#).
2. Open the Automation Anywhere Enterprise Client.
3. In the list of available tasks, select **ImportFromFlexiCaptureExample** and click **Edit**.
4. In the **Task Editor** dialog box, specify the values of the following required parameters:
 - a. **\$ServerURL\$** – the address of the ABBYY FlexiCapture server
 - b. **\$Username\$** – the name of the user on the ABBYY FlexiCapture server
 - c. **\$Password\$** – the password of the user on the ABBYY FlexiCapture server
 - d. **\$ProjectName\$** – the name of the ABBYY FlexiCapture project
 - e. **\$OutputDirPath\$** – the path to the folder where processing results will be stored

 **Note:** If you are using tenants, **\$Company\$** is also a required parameter.

5. To test-run the task, save the changes by clicking **Save** in the **Task Editor** and click **Run**.

The ABBYY FlexiCapture server will receive a request to check the output folder for suitable processing results. Since the previously sent document has not yet been verified, a message will be displayed saying that there are no suitable processing results.

For instruction on verifying documents in ABBYY FlexiCapture, see [Verification in the attended service scenario](#).

 **Note:** If an error occurs while the task is running, information about the error will be recorded in a log file, which can be found here: %Userprofile%\Documents\Automation Anywhere Files\Log\My Tasks\Bot Store\FC-MetaBot\ImporterErrorLog.txt

Additional information

Verification in the attended service scenario

In the attended service scenario, documents must be verified after they are sent to ABBYY FlexiCapture.

To verify a document:

1. Open the verification URL you received for the document in any browser.
2. Enter your ABBYY FlexiCapture user credentials.
3. Verify the document.
4. Click the **Complete Task** button.

To load document processing results, complete the steps described in [Getting processing results in the attended service scenario](#).

Passing registration parameters to ABBYY FlexiCapture

Registration parameters are textual properties identifying a batch or document.

In order to assign registration parameters to a batch that is to be sent to ABBYY FlexiCapture, specify the path to the file containing the registration parameters in the **\$RegistrationParametersFilePath\$** parameter.

The file containing the registration parameters must meet the following requirements:

- The file must have the *.csv extension and must contain only name-value pairs.
- The file must not have a header line.
- The comma (,) character must be used as a separator between a parameter name and its value.
- If a value contains a comma, the value must be enclosed in quotes.
- The value field of a registration parameter may be empty but may not be missing.

Example of a correctly formatted *.csv file:

Id,10

Company,Vector

Department,"Finance,Marketing"

FlexiCapture MetaBot logics

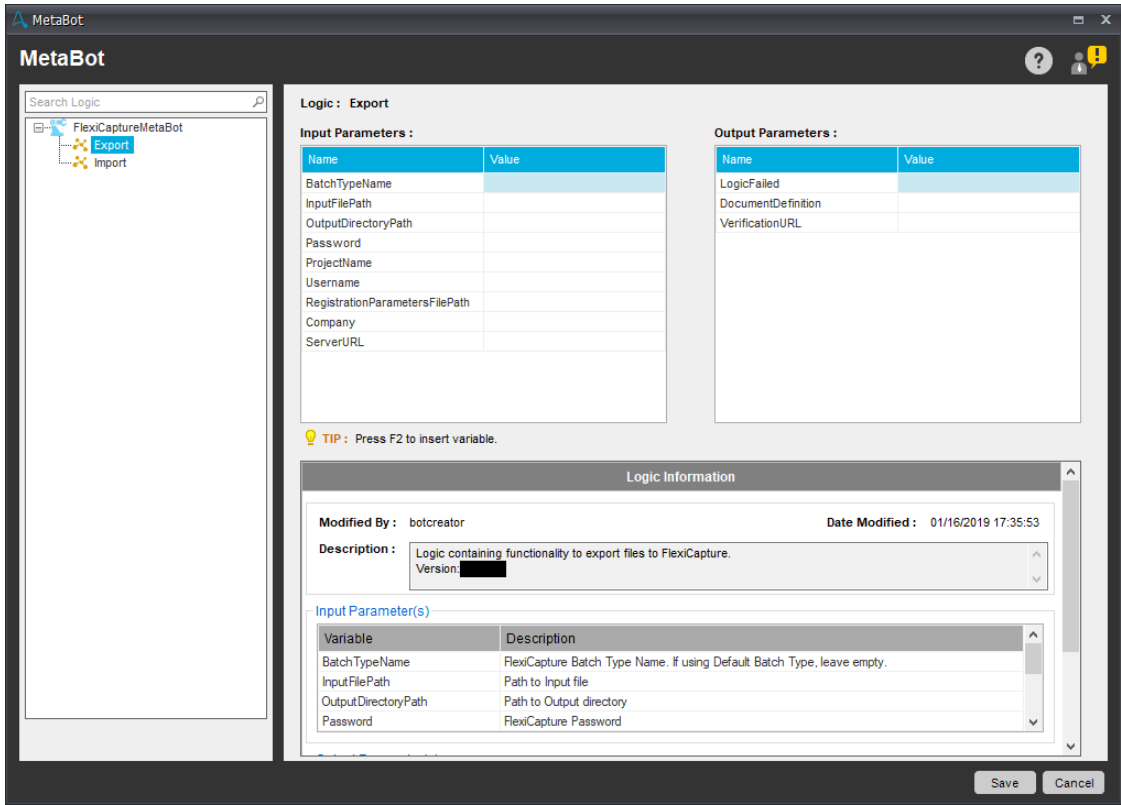
[Export](#)

[Import](#)


Export logic

The Export logic is used for sending files to ABBYY FlexiCapture for processing. Running this logic will either save the processing results and get the name of the matching Document Definition (in the unattended service scenario) or get a URL linking to the verification page (in the attended service scenario).

 **Note:** The Export logic uses the BatchOwner parameter to interact with the Import logic.



Input parameters:

Name	Description
BatchTypeName (optional)	The name of the batch type to be used when connecting to ABBYY FlexiCapture. To use the default batch type, specify an empty string.
Company (optional)	The name of the ABBYY FlexiCapture tenant to be used for processing. If no tenants are to be used, specify an empty string.
InputFilePath	The path to the file to be processed.  Note: The activity only works with files containing one document each (i.e. each document must be placed in a separate file).
OutputDirectoryPath	The path to the folder where processing results are to be stored.
Password	The password of the ABBYY FlexiCapture user.
ProjectName	The name of the ABBYY FlexiCapture project.
RegistrationParametersFilePath (optional)	The path to the CSV file that contains the batch registration parameters.
ServerURL	The address of the ABBYY FlexiCapture server.
Username	The name of the ABBYY FlexiCapture user.

Output parameters:

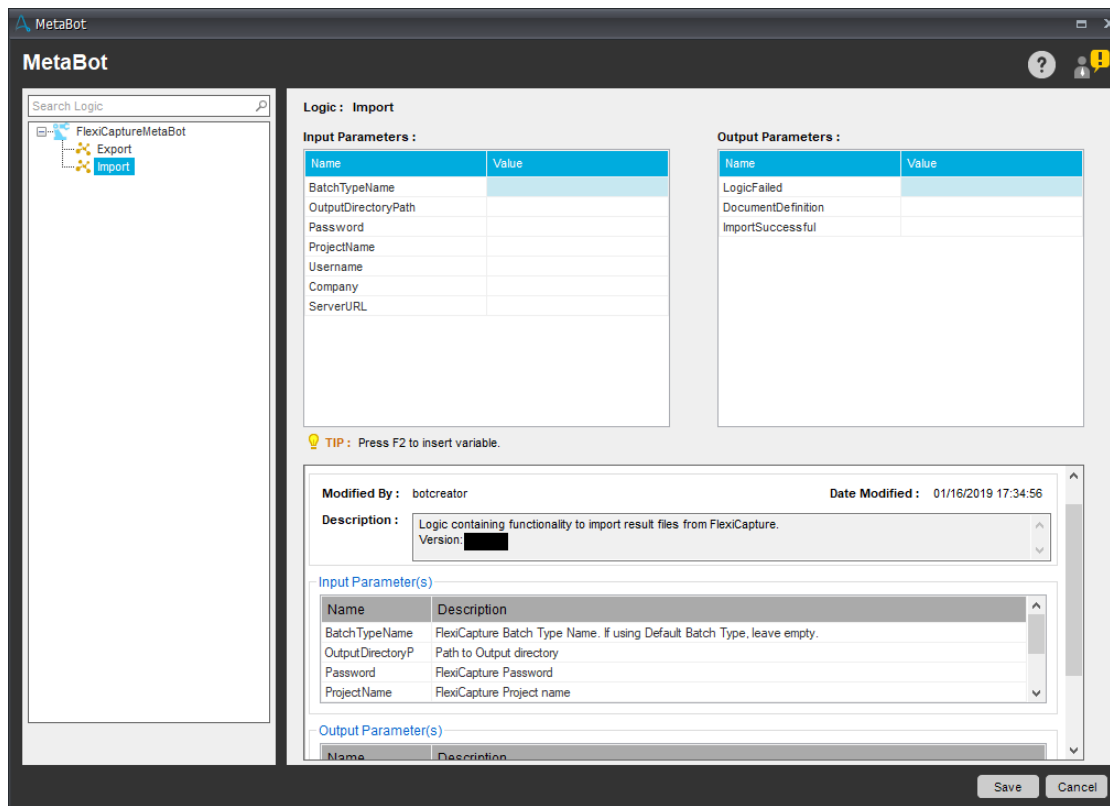
Name	Description
VerificationURL	A URL linking to the verification page on the ABBYY FlexiCapture Web Verification Station. Contains an empty string if no verification is required.
LogicFailed	<i>True</i> if an error occurred while the logic was running. <i>False</i> if the logic was executed successfully.
DocumentDefinition	The name of the matching Document Definition.

 **Note:** You can find the log file for the Export logic here: %Userprofile%\Documents\Automation Anywhere Files\Log\My Tasks\Bot Store\FC-MetaBot\ExporterErrorLog.txt.

Import logic

The **Import** logic is used for getting the results of processing files sent to ABBYY FlexiCapture by the Export logic. Running this logic saves the processing results into a user-specified folder.

 **Note:** The Import logic uses the BatchOwner parameter to interact with the Export logic.



MetaBot

Search Logic

FlexiCaptureMetaBot

Export

Import

Logic : Import

Input Parameters :

Name	Value
BatchTypeName	
OutputDirectoryPath	
Password	
ProjectName	
Username	
Company	
ServerURL	

Output Parameters :

Name	Value
LogicFailed	
DocumentDefinition	
ImportSuccessful	

TIP : Press F2 to insert variable.

Modified By : botcreator **Date Modified :** 01/16/2019 17:34:56

Description : Logic containing functionality to import result files from FlexiCapture.
Version: [redacted]

Input Parameter(s)

Name	Description
BatchTypeName	FlexiCapture Batch Type Name. If using Default Batch Type, leave empty.
OutputDirectoryP	Path to Output directory
Password	FlexiCapture Password
ProjectName	FlexiCapture Project name

Output Parameter(s)

Name	Description
------	-------------

Save Cancel

Input parameters:

Name	Description
BatchTypeName (optional)	The name of the batch type to be used when connecting to ABBYY FlexiCapture. To use the default batch type, specify an empty string.
Company (optional)	The name of the ABBYY FlexiCapture tenant to be used for processing. If no tenants are to be used, specify an empty string.
OutputDirectoryPath	The path to the folder where processing results are to be stored.
Password	The password of the ABBYY FlexiCapture user.
ProjectName	The name of the ABBYY FlexiCapture project.
ServerURL	The address of the ABBYY FlexiCapture server.
Username	The name of the ABBYY FlexiCapture user.

Output parameters:

Name	Description
ImportSuccessful	<i>True</i> if a suitable batch has been found and the processing results have been successfully saved to OutputDirectoryPath . <i>False</i> if no suitable batch has been found.
LogicFailed	<i>True</i> if an error occurred while the logic was running. <i>False</i> if the logic was executed successfully.
DocumentDefinition	The name of the matching Document Definition.


 **Note:** You can find the log file for the Import logic here: %Userprofile%\Documents\Automation Anywhere Files\Log\My Tasks\Bot Store\FC-MetaBot\ImporterErrorLog.txt.

Automation Anywhere sample tasks

The ABBYY FlexiCapture Connector for Automation Anywhere distribution package includes the following sample tasks:

- [ExportToFlexiCaptureExample](#)
- [ImportFromFlexiCaptureExample](#)

These sample tasks use of the FlexiCapture MetaBot logics.

 **Note:** The values of the input parameters in the sample tasks are for demonstration only and can be changed.

ExportToFlexiCaptureExample

The ExportToFlexiCaptureExample sample task includes the following actions:

The screenshot displays the 'ExportToFlexiCaptureExample' task configuration in Automation Anywhere. The 'Actions List' tab is active, showing a sequence of 24 actions. The actions include initializing variables, setting up logic parameters, running the 'Export' logic from a MetaBot, and handling the results based on the 'LogicFailed' variable. Conditional logic checks if verification is required and copies the URL to the clipboard if needed. Finally, it displays a message box indicating success or failure and stops the task.

Line	Action
1	Comment: Initializing variables for logic parameters. Initialization of optional logic parameters is disabled by default.
2	Variable Operation: FC-VM To \$ServerUrl\$
3	Variable Operation: TestTenant To \$Company\$
4	Variable Operation: TestBatchType To \$BatchTypeName\$
5	Variable Operation: FC-VM\Administrator To \$Username\$
6	Variable Operation: Pass@123 To \$Password\$
7	Variable Operation: SingleEntryPoint To \$ProjectName\$
8	Variable Operation: C:\Temp\Metabot_Input\RegParams.csv To \$RegParamFilePath\$
9	Variable Operation: C:\Temp\Metabot_Input\Images\example images\Invoices_eng_04.pdf To \$InputFilePath\$
10	Comment: Note: Output directory path contains timestamp.
11	Variable Operation: C:\Temp\Exporter_Output_\$Year\$_\$Month\$_\$Day\$_\$Hour\$\$Minute\$\$Second\$ To \$OutputDirPath\$
12	Run Logic "Export" from MetaBot "My MetaBots\FlexiCaptureMetaBot.mbot"
13	If \$LogicFailed\$ Not Equal To (<>) "true" Then
14	If \$VerificationUrl\$ Not Equal To (<>) "" Then
15	Assign value of variable "\$VerificationUrl\$" to Clipboard
16	Message Box: "Export successful, but verification is required.Verification URL was copied to clipboard."
17	Else
18	Message Box: "Export successful!Document Definition was copied to clipboard.Output was saved to '\$OutputDirPath\$'"
19	Assign value of variable "\$DocumentDefinition\$" to Clipboard
20	End If
21	Else
22	Message Box: "FlexiCapture Metabot failed to export file. Check error logs for details."
23	Stop The Current Task
24	End If

The actions at the start of this sample task set the values of the variables. Next, the Export logic is run and the input parameters are passed using the variables. The LogicFailed variable will be assigned a value indicating whether any errors occurred when running the logic (*true* means that an error occurred and you should check the MetaBot error log; *false* means that the logic was executed successfully without any errors). The VerificationURL variable will contain a URL linking to the verification page. The DocumentDefinition variable will contain the name of the matching Document Definition.

Should any file submitted by this task require verification, the corresponding URL will be copied to the Clipboard. If no verification is required, a message will be displayed saying that the processing results have been successfully saved and the name of the matching Document Definition will be copied to the Clipboard.

Note: To process errors of the Export logic, the program will check the LogicFailed output parameter, whose value is saved to the LogicFailed variable.

ImportFromFlexiCaptureExample

The ImportFromFlexiCaptureExample sample task includes the following actions:

ImportFromF...

Actions List [NORMAL] [VISUALIZE] [ANALYZE]

FILTERS ☒ Mouse Moves ☒ Keystrokes ☒ Mouse Clicks ☒ Delays ☒ Other Windows **All**

- 1 Comment: Initializing variables for logic parameters. Initialization of optional logic parameters is disabled by default.
- 2 Variable Operation: FC-VM To \$ServerUrl\$
- 3 Variable Operation: TestTenant To \$Company\$
- 4 Variable Operation: TestBatchType To \$BatchTypeName\$
- 5 Variable Operation: FC-VM\Administrator To \$Username\$
- 6 Variable Operation: Pass@123 To \$Password\$
- 7 Variable Operation: SingleEntryPoint To \$ProjectName\$
- 8 Comment: Note: Output directory path contains timestamp.
- 9 Variable Operation: C:\Temp\Importer_Output_Year\$_Month\$_Day\$_Hour\$Minute\$\$Second\$ To \$OutputDirPath\$
- 10 Run Logic "Import" from MetaBot "My MetaBots\FlexiCaptureMetaBot.mbot"
- 11 IF \$LogicFailed\$ Not Equal To (<>) "true" Then
- 12 IF \$ImportSuccessful\$ Equal To (=) "true" Then
- 13 Assign value of variable "\$DocumentDefinition\$" to Clipboard
- 14 Message Box: "Import successful!Output was saved to '\$OutputDirPath\$'Document definition was copied to clipboard"
- 15 ELSE
- 16 Message Box: "Import unsuccessful. No available batches to import found."
- 17 End If
- 18 ELSE
- 19 Message Box: "FlexiCapture Metabot failed to import. Check error logs for details."
- 20 Stop The Current Task
- 21 End If

The actions at the start of this sample task set the values of the variables. Next, the Import logic is run and the input parameters are passed using the variables. The LogicFailed variable will be assigned a value indicating whether any errors occurred when running the logic (*true* means that an error occurred and you should check the MetaBot error log; *false* means that the logic was executed successfully without any errors). The ImportSuccessful variable will be *true* if a suitable batch has been found and *false* if no suitable batch has been found. The DocumentDefinition variable will contain the name of the matching Document Definition if import was a success.

If the import succeeds, a message will be displayed saying that the processing results have been successfully saved and the name of the matching Document Definition will be copied to the Clipboard. Otherwise, an failure message will be displayed.

Note: To process errors of the Import logic, the program will check the LogicFailed output parameter, whose value is saved to the LogicFailed variable.