

# Kofax RPA

## Getting Started with Document Transformation

Version: 11.2.0

Date: 2021-06-01

The KOFAX logo is displayed in a bold, blue, sans-serif font. The letters are all uppercase and have a consistent thickness, giving it a strong, professional appearance.

© 2019–2021 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

# Table of Contents

<b>Preface</b> .....	<b>4</b>
Related documentation.....	4
Training.....	5
Getting help with Kofax products.....	5
<b>Chapter 1: Tutorial</b> .....	<b>7</b>
Preliminary steps.....	7
Create robots and types.....	7
Add variables to the transformation robot.....	9
Upload the robot to Management Console.....	10
Create the Document Transformation workflow.....	10
Add a step to check document validity.....	14
Extract data from the document.....	15
Configure steps for human validation.....	17
Configure the robot to handle invalid and valid documents.....	29
Validate the transformed document.....	32

# Preface

This guide provides a tutorial that explains how to use Document Transformation functionality in a Kofax RPA environment. The tutorial covers functionality such as OCR, extraction, field formatting, and validation.

The instructions provided in this document assume that you have successfully downloaded and installed Kofax RPA on your computer. After you start the Management Console and Development Database from the Kofax RPA product group, ensure that "Document Transformation - Server Scheduler Service" is started in the Windows Services panel.

For a detailed explanation of the Document Transformation functionality, see *Help for Kofax RPA*.

## Related documentation

The documentation set for Kofax RPA is available here:<sup>1</sup>

[https://docshield.kofax.com/Portal/Products/RPA/11.2.0\\_ea1ydbmwk9/RPA.htm](https://docshield.kofax.com/Portal/Products/RPA/11.2.0_ea1ydbmwk9/RPA.htm)

In addition to this guide, the documentation set includes the following items:

### ***Kofax RPA Release Notes***

Contains late-breaking details and other information that is not available in your other Kofax RPA documentation.

### ***Kofax RPA Technical Specifications***

Contains information on supported operating systems and other system requirements.

### ***Kofax RPA Installation Guide***

Contains instructions on installing Kofax RPA and its components in a development environment.

### ***Kofax RPA Upgrade Guide***

Contains instructions on upgrading Kofax RPA and its components to a newer version.

### ***Kofax RPA Administrator's Guide***

Describes administrative and management tasks in Kofax RPA.

---

<sup>1</sup> You must be connected to the Internet to access the full documentation set online. For access without an Internet connection, see the *Installation Guide*.

***Help for Kofax RPA***

Describes how to use Kofax RPA. The Help is also available in PDF format and known as *Kofax RPA User's Guide*.

***Kofax RPA Best Practices Guide for Robot Lifecycle Management***

Offers recommended methods and techniques to help you optimize performance and ensure success while using Robot Lifecycle Management in your Kofax RPA environment.

***Kofax RPA Getting Started with Robot Building Guide***

Provides a tutorial that walks you through the process of using Kofax RPA to build a robot.

***Kofax RPA Desktop Automation Service Configuration Guide***

Describes how to configure the Desktop Automation Service required to use Desktop Automation on a remote computer.

***Kofax RPA Developer's Guide***

Contains information on the API that is used to execute robots on RoboServer.

***Kofax RPA Integration API documentation***

Contains information about the Kofax RPA Java API and the Kofax RPA .NET API, which provide programmatic access to the Kofax RPA product. The Java API documentation is available from both the online and offline Kofax RPA documentation, while the .NET API documentation is available only offline.

**Note** The Kofax RPA APIs include extensive references to RoboSuite, the original product name. The RoboSuite name is preserved in the APIs to ensure backward compatibility. In the context of the API documentation, the term RoboSuite has the same meaning as Kofax RPA.

## Training

Kofax offers both classroom and computer-based training to help you make the most of your Kofax RPA solution. Visit the Kofax Education Portal at <https://learn.kofax.com/> for details about the available training options and schedules.

Also, you can visit the Kofax Intelligent Automation SmartHub at <https://smarthub.kofax.com/> to explore additional solutions, robots, connectors, and more.

## Getting help with Kofax products

The [Kofax Knowledge Base](#) repository contains articles that are updated on a regular basis to keep you informed about Kofax products. We encourage you to use the Knowledge Base to obtain answers to your product questions.

To access the Kofax Knowledge Base, go to the [Kofax website](#) and select **Support** on the home page.

**Note** The Kofax Knowledge Base is optimized for use with Google Chrome, Mozilla Firefox or Microsoft Edge.

The Kofax Knowledge Base provides:

- Powerful search capabilities to help you quickly locate the information you need.  
Type your search terms or phrase into the **Search** box, and then click the search icon.
- Product information, configuration details and documentation, including release news.  
Scroll through the Kofax Knowledge Base home page to locate a product family. Then click a product family name to view a list of related articles. Please note that some product families require a valid Kofax Portal login to view related articles.
- Access to the Kofax Customer Portal (for eligible customers).  
Click the **Customer Support** link at the top of the page, and then click **Log in to the Customer Portal**.
- Access to the Kofax Partner Portal (for eligible partners).  
Click the **Partner Support** link at the top of the page, and then click **Log in to the Partner Portal**.
- Access to Kofax support commitments, lifecycle policies, electronic fulfillment details, and self-service tools.  
Scroll to the **General Support** section, click **Support Details**, and then select the appropriate tab.

## Chapter 1






# Tutorial

This guide is a step-by-step tutorial that explains how to create a Document Transformation robot similar to the sample robot included in your product. The sample robot, called *DocumentTransformationInvoiceExampleBasic*, already contains a .tif file that is ready to transform.

You can test the *DocumentTransformationInvoiceExampleBasic* robot provided with the product without creating a new robot and types. In **Design Studio**, in the **Examples** folder, simply open the robot, open the types *invoice* and *DocumentTransformationInput*, and then follow this guide to see how the robot is built.



If you decide to create a new robot from scratch, use the procedure in this guide and be sure to assign the robot and types unique names that differ from the examples.


The tutorial covers some of the most commonly used Document Transformation functions and consists of four main parts:

1. Create a transformation robot (with a Basic Engine Robot ) , a transformation workflow (with a Robot ) , and types to store data.
2. Call the Robot  from the Basic Engine Robot  using the Call Robot step.
3. Create the Document Transformation workflow in the Robot  , which includes setting up the Document Transformation step, extracting data from the .tif document, and configuring human and non-human validation of the transformed document.
4. Validate the transformed document.

## Preliminary steps

### Create robots and types

First, you need to create a Basic Engine Robot and create a Robot required to build a Transformation Workflow. You also need to create types to store the extracted data. As opposed to Basic Engine Robots that are identified by a blue icon  , Robots are identified by a green icon .

1. **Create a Basic Engine Robot **
  - a. In **Design Studio**, click **File > New Basic Engine Robot**.
  - b. Name the robot **DocumentTransformationInvoiceExampleBasic**, select a project, and then click **Next > Next**.  
By default, the Smart Re-execution (Full) execution mode is selected.

- c. Click **Finish**.  
By default, the End step is selected in the newly created robot.
- d. Insert an **Action step** in the new robot.
- e. Rename the step to **Transform Document**.
- f. Save the changes.

**2. Create a Robot** 🤖

- a. Click **File > New Robot**.
- b. Name robot **DocumentTransformationInvoiceExample** and click **Finish**.  
The new robot appears on a new tab in the editor window.

**3. Call a Robot from a Basic Engine Robot**

- a. Open the tab with the **DocumentTransformationInvoiceExampleBasic** Basic Engine Robot.
- b. In the inserted step, click **Select an Action** on the **Action** tab and choose **Call Robot**.
- c. In the **Robot** drop-down list, select the **DocumentTransformationInvoiceExample** robot.
- d. Save the changes.

**4. Create new types**

With this robot, you extract the ID of the vendor, invoice number, invoice date, and total sum. To store the extracted data, create the following types.

- a. Click **File > New Type**.
- b. Name the type **invoice**, select a project, and then click **Finish**.
- c. Click the plus sign to add new attributes to the type. Add the following attributes and specify their types:

Name	Attribute Type
VendorID	Short Text
InvoiceNumber	Short Text
InvoiceDate	Short Text
Total	Number

- d. Create another type called **DocumentTransformationInput**.
- e. Add the following attribute and specify its type:

Name	Attribute Type
doc	Binary

- f. Save the changes.

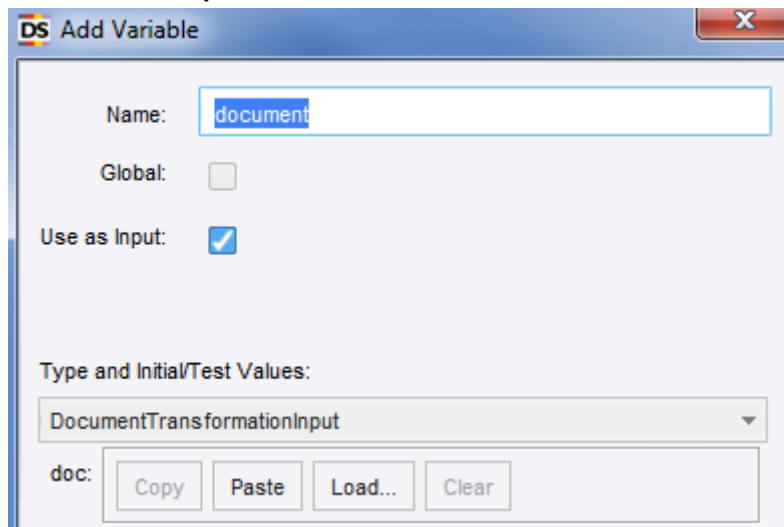
When finished, proceed to the next section.



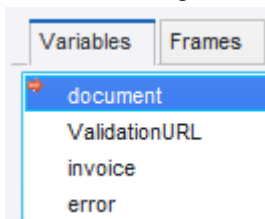
## Add variables to the transformation robot

In this procedure, you add variables of complex and simple types. A complex variable can contain several named values, which is useful for extracting diverse data such as from an invoice. A simple variable can only contain a single value such as a text string.

1. In **Design Studio**, right-click in the **Variables** field and click **Add Variable of Complex Type > DocumentTransformationInput**.
  - a. In the new dialog box, rename the **DocumentTransformationInput** variable to **document** and select **Use as Input**.



- b. Click **Load** and open the invoice document to transform. Navigate to the Kofax RPA **Examples** folder of the sample project and select the file **InvoiceExampleDocument.tif**. This invoice will be read with the built-in Invoice VAT Transformation project.
2. Right-click in the **Variables** field and click **Add Variable of Complex Type > invoice**.
3. Right-click in the **Variables** field and click **Add Variable of Simple Type > Short Text (simple)**. Name the new variable **ValidationURL**.
4. Right-click in the **Variables** field and click **Add Variable of Simple Type > Long Text (simple)**. Name the new variable **error**.
5. Save the changes.



6. Open the **Action** tab of your **Transform Document** step and configure variables as follows.
  - a. In the **Input Value** section, click the plus sign.  
In the drop-down list, click **Variable** and select **document**.
  - b. Click **OK**.
  - c. In the **Output Mapping** section, add the **invoice**, **ValidationURL**, and **error** variables.

When finished, proceed to the next section.


## Upload the robot to Management Console

In the **Projects** tree view, right-click the **DocumentTransformationInvoiceExampleBasic** robot and click **Upload** to upload it to the Management Console.

When finished, proceed to the next section.

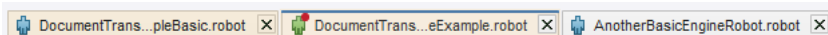
## Create the Document Transformation workflow



In this procedure, you configure variables, set up a Document Transformation step, and add a guard to wait for the transformed document.

To start editing and executing your Basic Engine Robot, you need to prepare it for execution by clicking **Prepare Execution**  in the Applications view or on the toolbar. By clicking this action, you put the robot into execution mode, which enables you to execute it while editing. You can execute action steps right after you insert them in the robot workflow and immediately see the result. When a Basic Engine Robot is not prepared for execution, you can still perform some basic editing, such as add steps, but you will not be able to execute the steps and see the result.

**Important** Only one Basic Engine Robot at a time can have the execution privilege, so to take the execution privilege from one robot to another, open the tab with the required robot and click **Prepare Execution**.

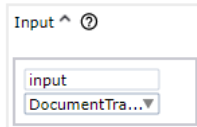
When a Basic Engine Robot has the execution privilege, the editor tab of this robot is highlighted. When a Basic Engine Robot is calling a Robot, the tabs of both robots are highlighted for convenience as shown below. The robot where execution is currently located is marked with a red dot.



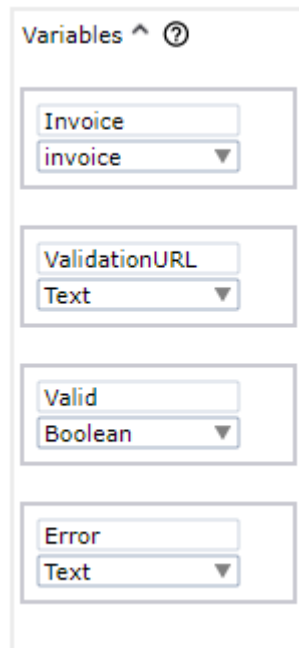
1. When the execution is allowed, open the Robot . To do so, execute the workflow to the **Call Robot** step and then click **Step Into Robot**  on the toolbar.

The tab with your **DocumentTransformationInvoiceExample** robot is opened and the editor is now active.

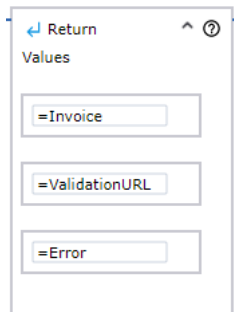
- 2. In workflow, follow these steps.
  - a. Ensure that the **Input** box is configured in this way.



- b. This variable will contain the transformed document after it is validated.
    - c. In the **Variables** box, add the following variables and specify their types.



- d. In the workflow, expand the **Return** step and add the following variables:
        - 1. **Invoice**. Ensure that you start the name with a capital letter.
        - 2. **ValidationURL**.
        - 3. **Error**.



3. To add a Document Transformation step, right-click the flow point (small circle) before the **Return** step and click **Document Transformation**.

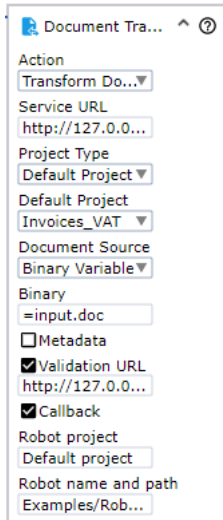
This step will extract information from InvoiceExampleDocument.tif.

**Tip** After you execute the Document Transformation step as shown later in this section, you may get an error stating that the document transformation failed and a message pointing to the transformation project. This may mean that the folder to store the Document Transformation training files does not yet exist.

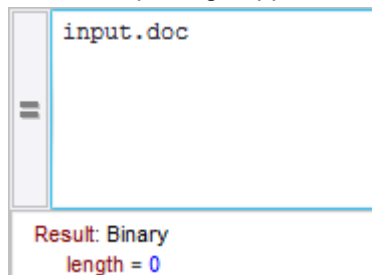
To resolve this problem, in the **Project Builder** for the **Invoice VAT** project, open the **Project Settings** and on the **General** tab, click **OK**. If the folder does not exist, you receive a notification prompting you to create it. To proceed, click **Yes**.


For more information, see the section "Set up Online Learning" in the *Kofax RPA Installation Guide*.

4. Set the properties as follows. For more information on these properties, see "Document Transformation" in *Help for Kofax RPA*.

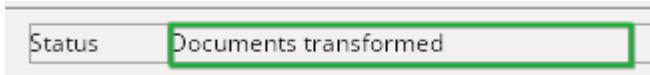


- a. In **Service URL**, specify the URL for the computer running the Document Transformation Service. In the example robot, the URL is: `http://127.0.0.1:50081`
- b. In **Document Source**, select **Binary Variable**, click the **Binary** field and type **input.doc**. Make sure the equal sign appears on the left, so that the variable can be evaluated.



- c. Select **Validation URL** and specify the URL for the Thin Client Service. This property is required to send processed documents for validation. In the example robot, the URL is: `http://127.0.0.1:50082`.
- d. Select **Callback** and specify the information about the transformation robot. The Thin Client Service will call this robot after a document is validated. When the validation is completed, the robot will be queued for execution on the Management Console. In the **Document Transformation** step of the example robot, the **Robot project** and **Robot name and path** are set to:
- Default project
  - `Examples/Robots/DocumentTransformationInvoiceExampleBasic.robot`
- e. Click **Step Over**  to execute this step.
- The status is changed to "Waiting for transformed documents." Wait until it changes to "Documents transformed."

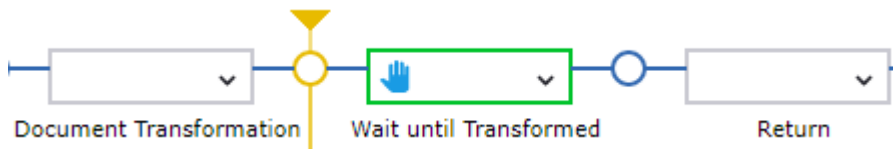
- Click **Documents transformed** to add a guard to wait for the transformed document.



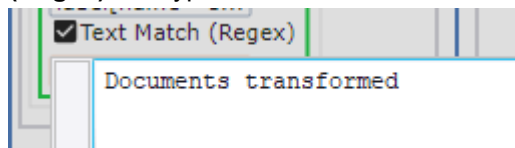
- To ensure that the guard looks for the correct text, expand the following label in the XML tree.

```
<label name="status_message" der_x="494" der_y="56" der_width="230" der_height="20">Documents transformed</label>
```

Right-click the text "Documents transformed" and click **Guard > Location Found**. The Guard step is inserted in your workflow. Rename it to **Wait until Transformed**.



- In the **Wait until Transformed** step, expand the **Location Found** guard, select **Text Match (Regex)** and type **Documents transformed**.



- Click **Step Over** to execute this step.
- Save the changes.

When finished, proceed to the next section.

## Add a step to check document validity

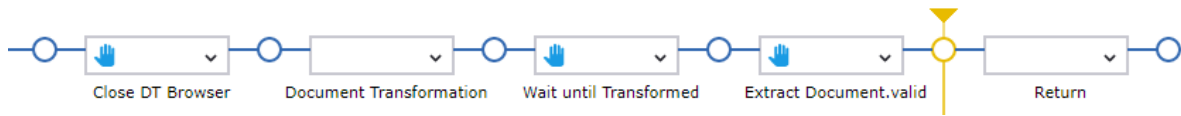
To check if the transformed document is valid, add the following step to the Document Transformation workflow. If the document appears to be invalid, it can be sent for human validation (see [Configure steps for human validation](#)).

- In the XML tree view, locate and right-click the **document** element, and then click **Extract Value From > Attribute > valid Into > Valid: Boolean**.

```
<folder valid="false" der_x="0" der_y="84" der_width="0" d
  <xvalues />
  <document class="Invoice" classificationUnconfident="
    indexInFolder="0" name="document" nextValidationInsta
    timeExtraction="2361" timeOCR="2089" valid="false" val
    der_height="0">...</document>
```

A new step is inserted in your workflow. Rename it to **Extract Document.valid**.

2. Click  **Step Over** to execute this step.



3. Save the changes.

When finished, proceed to the next section.

## Extract data from the document

In this procedure, you extract data from InvoiceExampleDocument.tif into the variables VendorID, InvoiceNumber, InvoiceDate, and Total. At the end, you add a Return Value step to the robot, so it returns the extracted values.

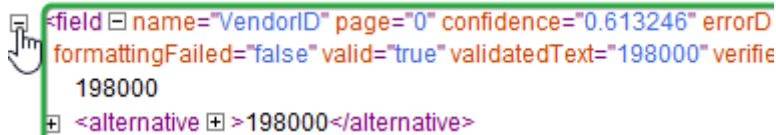
1. In the XML tree view, expand the **fields** element under **application/folder/document**.

```

<dts +>
  - <application +>
    + <toolbar +>...</toolbar>
    - <folder +>
      <xvalues + />
      - <document +>
        + <fields - der_x="0" der_y="84" der_width="0" der_height="0">...</fields>
        + <xvalues +>...</xvalues>
        + <pages +>...</pages>
    
```

2. To extract values:
  - a. Expand the second field, which is for **VendorID**, to see its values and alternatives.

- b. Right-click the **field** element and click **Extract Value From > Attribute > validatedText Into > Invoice: invoice > VendorID: Text**.



**Note** Ensure that you do **not** click the text 198000.

A new Extract Value step is inserted in your workflow. Rename it to **Extract Vendor ID**.

- c. Repeat this action for field 17, **InvoiceNumber**, and for field 18, **InvoiceDate**.  
New Extract Value steps are inserted in your workflow. Rename them to **Extract Invoice Number** and **Extract Invoice Date**, respectively.
- d. Right-click the third from the last field, **TotalAmount**, and click **Extract Value From > Attribute > doubleValue Into > Invoice: invoice > Total: Number**.  
The transformation project has a number formatter on this field and converts it to a numerical value.  
Another Extract Value step is inserted in your workflow. Rename it to **Extract Total Number**.
3. Click **Start Execution** ▶ to execute the workflow to the end.  
After the Extract Value steps are executed, you can check the extracted values. In the **State** pane on the right, expand the **Variables** branch.
4. Save the changes and step out of the robot.  
To step out of the robot and switch to working on the Basic Engine Robot, execute the entire workflow and then click **Step Out** ⚙ on the toolbar. In the Basic Engine Robot, the **Transform Document** step is now shown as executed.  
Your Document Transformation robot now has all of the available invoice details. Check the **invoice** variable in the **Variables** section in Design Studio.
5. In Design Studio, add another **Action step** to the Basic Engine Robot and make it a **Return Value** step. On the toolbar, click **Insert Step After Selected Step** ➕ > **Action step** and then select **Return Value** on the **Action** tab.  
On the **Basic** tab, in the **Variable** list, select **invoice**.  
This way the robot debugger will show the invoice parameters.
6. Save the changes.

You now have a valid robot that runs in the Management Console. This robot will be called by the Document Transformation Client.

**Important** Ensure that you have a backup copy of your document before the transformation is completed. If the transformation is completed or an error occurs, the document and data are not stored.

When finished, proceed to the next section.



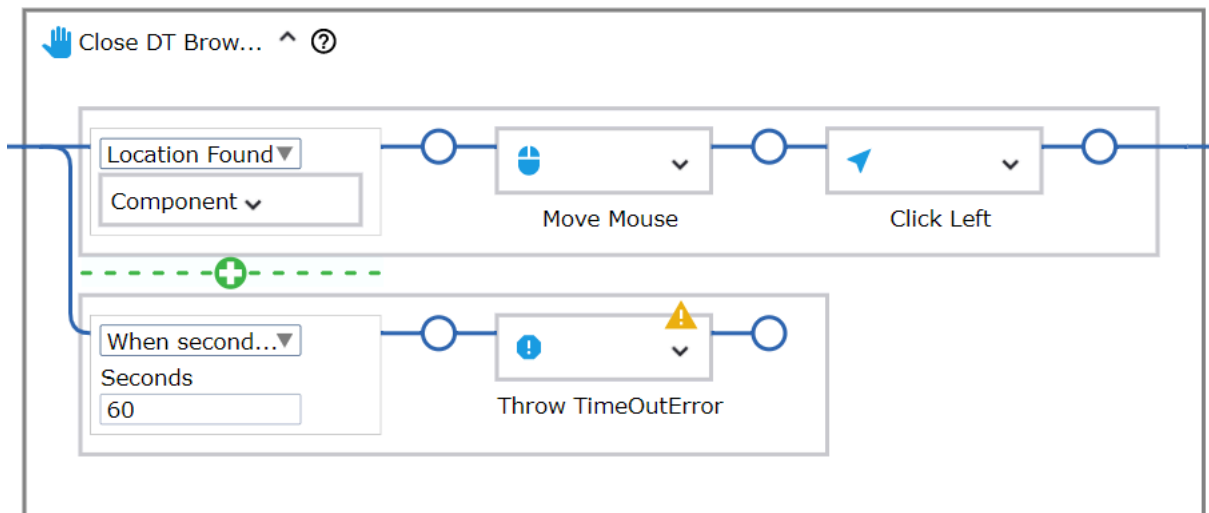
## Configure steps for human validation

In this procedure, you configure guards to handle a scenario for sending the document for human validation and add a special conditional branch in the workflow for human validation.

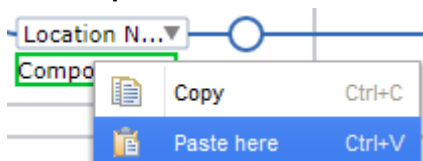
You need to re-open the Transform Document step of your robot again. Before doing so, refresh the robot to ensure that the Document Transformation browser window opened in the previous robot run is now closed.

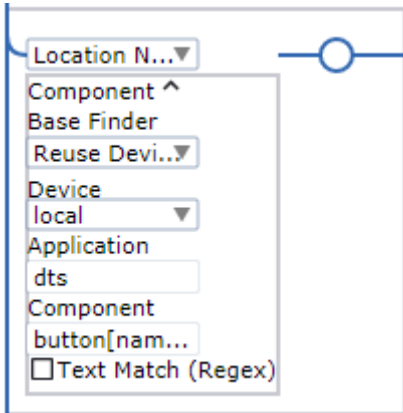
**Tip:** Alternatively, you may add a Click step to close the browser window after the robot finishes executing. To do so, in the **DocumentTransformationInvoiceExample** robot, right-click the Close button for the browser and click **Click > Left**. A new step is added to the workflow.

Expand the new step, click the green plus icon in the middle, and add a new **Guard**.



Set the new guard to **Location Not Found** and copy the **Component** from the **Location Found** guard to the **Component** box for the **Location Not Found** guard.





The created step closes the Document Transformation Browser immediately if it is open, or continues to run. Rename the step to **Close DT Browser**.









When finished, proceed to the following steps to configure human validation. In this procedure, we handle the case where the validated document is returned through the Validation Callback from the Document Transformation Client.

1. Double-click the flow point after the **Document Transformation** step.

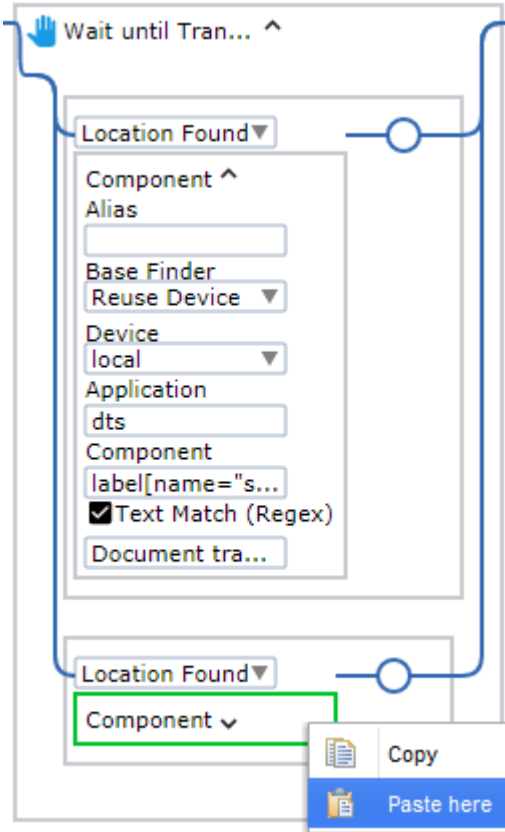
The Location Found guard in the **Wait until Transformed** step is waiting for the transformed document.

When Document Transformation Client calls the robot specified in the **Callback** option, after human validation, it passes the transformed document as a .zip file to the **document.doc** variable in the robot. The .zip file looks similar to the following:

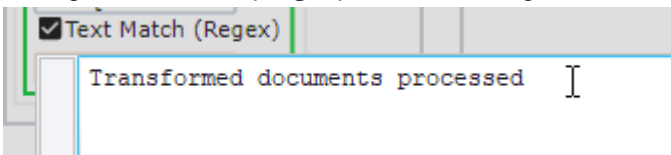
 project	File folder	
 config.xml	XML Document	1 KB
 document.tif	TIF File	42 KB
 document.xdc	Kofax Transformation XDocument File	37 KB
 Folder.xfd	XFD File	1 KB
 tree.dat	DAT File	97 KB

When the Document Transformation Browser opens the .zip file, the status is changed to "Transformed document processed."

- To configure the guard to wait until the transformed documents are processed, in the **Wait until Transformed** step, create an additional **Location Found** guard and then copy the **Component** box of the first **Location Found** guard and paste it in the **Component** box for the second guard.



Change **Text Match (Regex)** of the second guard to **Transformed documents processed**.



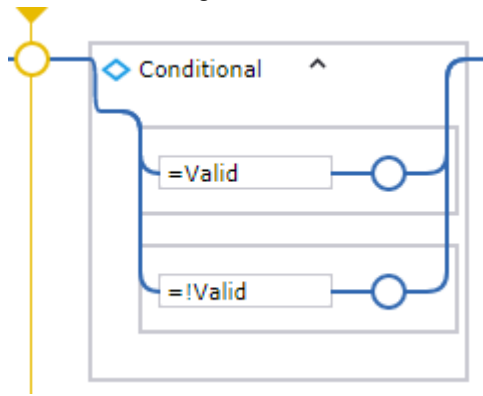
With the two guards, the workflow now handles two scenarios: with human validation and without it.

**Tip** In this step, you can optionally configure additional guards to handle errors that may occur during transformation. For more information, see [Configure error handling](#).

- Double-click the flow point after the **Extract Document.valid** step to execute to this point.

4. Add a special branch for human validation. Right-click the flow point and click **Conditional step**. A new step is inserted in your workflow.

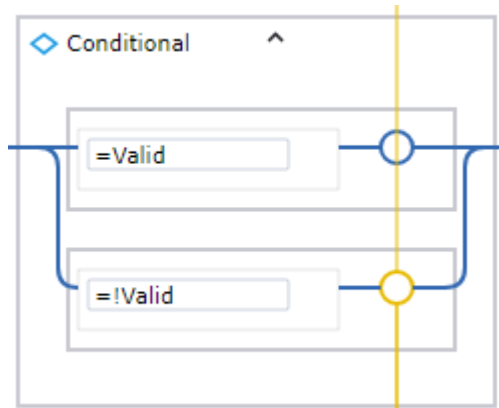
- a. Add the following two conditions to the **Conditional** step.



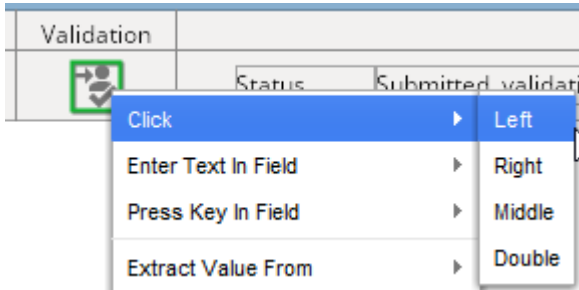
The exclamation mark in **=!Valid** denotes "not" ("invalid").


- b. Click **Step Into**  to step into the **Conditional** step.

As the document in the robot is not valid at this point, the second condition (branch) is executed.



5. Now you need to send the document to the Document Transformation Client. In the **Recorder View**, right-click the validation button and click **Click > Left**.

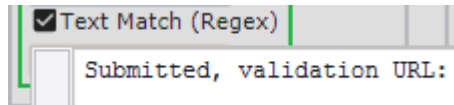


A new step is inserted in the second branch of the **Conditional** step. Rename it to **Send to Validation**. Click **Step Over**  to execute it.

- a. When the status is changed to **Submitted**, **validation URL:**, click it, and then click **Guard > Location Found**.

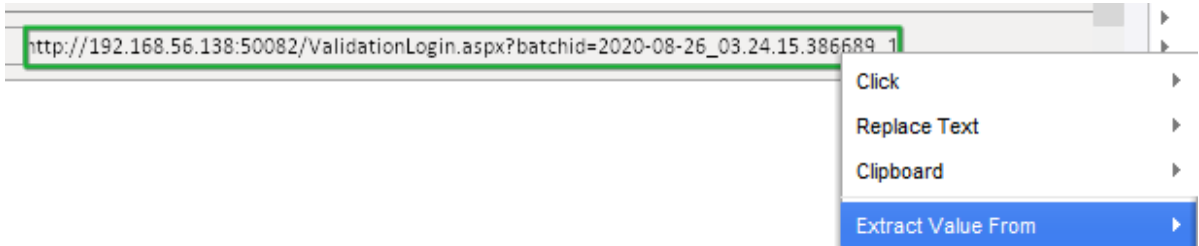
A new step is inserted in the second branch of the **Conditional** step. Rename it to **Wait for URL**.

- b. In the **Wait for URL** step, expand the **Component** box, select **Text Match (Regex)** and type **Submitted, validation URL:**.

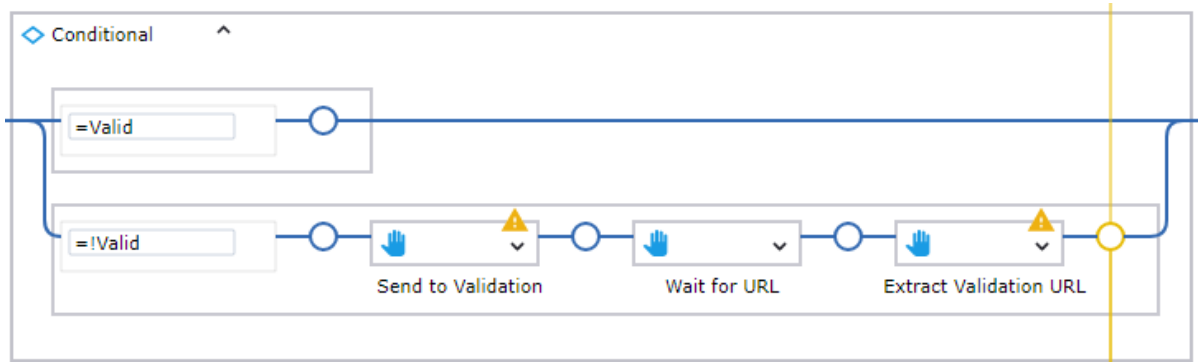


- c. Click **Step Over**  .

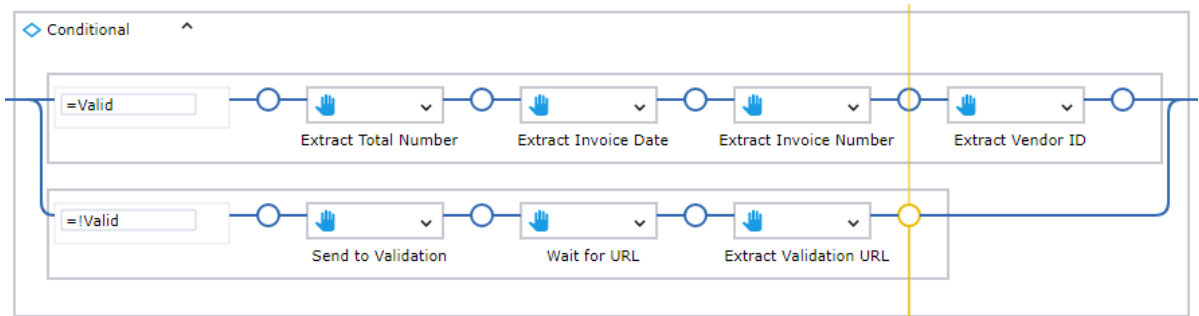
- Right-click the following URL and click **Extract Value From > Text Into > ValidationURL: Text**.



The Extract Value step is inserted in the **Conditional** step. Rename it to **Extract Validation URL** and then execute it.



- Select the four **Extract Value** steps after the **Conditional** step, and then cut and paste them into the **=Valid** branch of the **Conditional** step.



- Save the changes.

Your transformation workflow is finished. The workflow now has three possible paths:

- New document is sent to the Document Transformation Service for extraction and classification and returns "Valid." The document does not require human validation.
- New document is sent to the Document Transformation Service for extraction and classification and returns "Invalid." The document requires human validation.
- Document .zip package is sent from the Document Transformation Client after human validation and returns "Valid." The Validation URL is retrieved.

Click **Start Execution** ▶ to run the workflow to the end. When finished, step out of the robot and proceed to [Configure the robot to handle invalid and valid documents](#).

- To step out of the robot and switch to the Basic Engine Robot, click **Step Out** ⏪ on the toolbar after the entire workflow is executed. In the Basic Engine Robot, the **Transform Document** step is now shown as executed.
- To close the robot without executing it to the end or returning a result, instead of clicking **Start Execution** ▶, click **Leave Robot** 🚪 on the toolbar. The tab with the Basic Engine Robot is now opened. The **Transform Document** step is now shown as *not* executed.

## Configure error handling

You can optionally configure error handling, so that if something goes wrong during the transformation process, an error is reported and the information about this error is stored in variables. Later, you can configure to [send an email](#) containing the error information to a human.

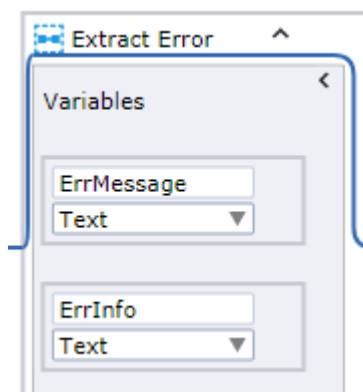
The following procedure shows how to configure guards for error handling in the **Wait until Transformed** step. The *DocumentTransformationInvoiceExampleBasic* robot provided in your installation additionally contains the "When seconds have passed" guards in the **Close DT Browser** and **Extract Document.valid** steps, which are configured in a similar way.

1. In the **Wait until Transformed** step, create another **Location Found** guard under the second **Location Found** guard and then copy the **Component** box from the second guard and paste it in the **Component** box for the third guard.
2. Expand the copied **Component** box and make the following changes:
  - In the **Component** field, change the value to `label[name="status_title"]`.
  - Change **Text Match (Regex)** to **Error**.

With this guard, the workflow handles another scenario: if an error is reported. The other two scenarios are described in [Configure steps for human validation](#).

3. Right-click the flow point next to this **Location Found** guard and add a Group step. Rename it to **Extract Error**.

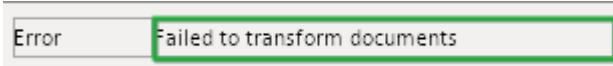
Expand the step, expand the **Variables** box, and then add two variables of type Text as follows.



These variables will store the error message name and information if an error occurs.

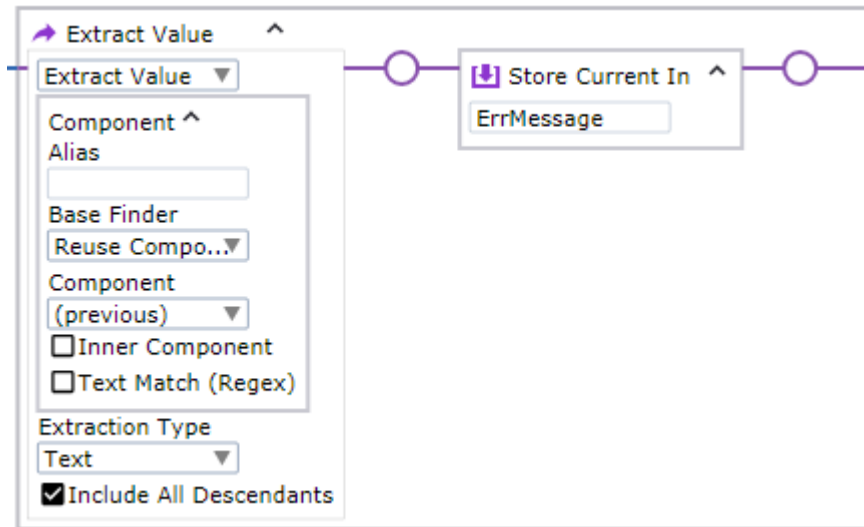
4. Double-click the flow point next to the Variables box to execute to this point.

5. In the **Recorder View**, right-click the button **Failed to transform documents** and click **Guard > Location Found**.



A new guard is inserted in your workflow. This is the guard that extracts the error message name. Rename it to **Extract Message**.

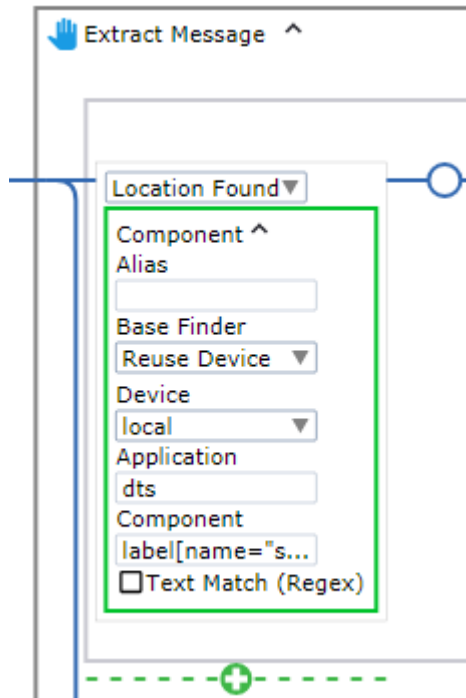
- Right-click the flow point next to the **Location Found** box, add an **Extract Value** step, and then configure it as follows.



With this step, if an error is reported, the error name is stored in the variable **ErrorMessage**.

- Click the plus sign below the **Location Found** box to add a new guard.

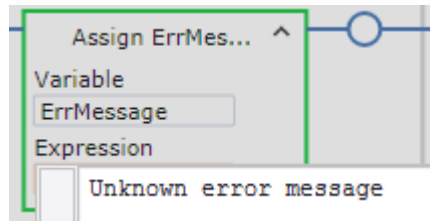




By default, the "When seconds have passed" type of guard is inserted. Set it to **5** seconds.

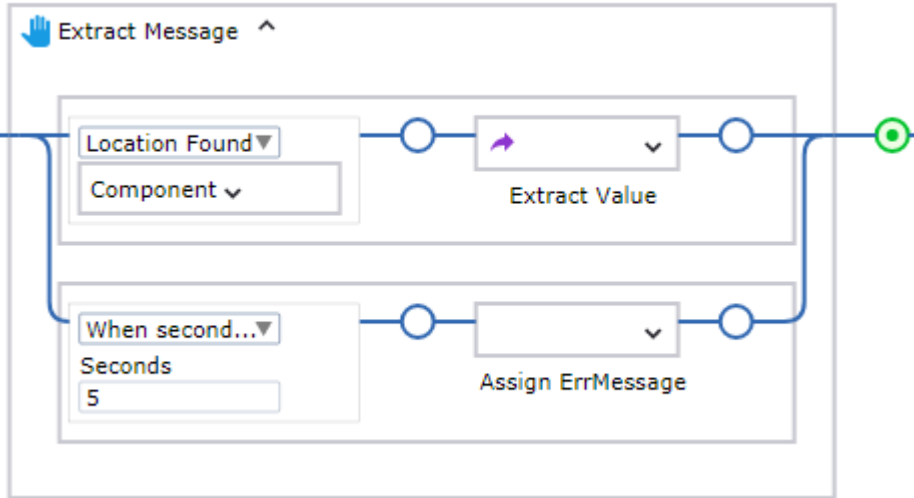
- Right-click the flow point next to the **When seconds have passed** guard, add an **Assign step**, and then rename it to **Assign ErrorMessage**.

Configure the Assign step as follows.



With these two steps, if an unknown error is encountered, the respective expression is stored in the variable **ErrorMessage**.

6. Double-click the flow point next to the **Extract Message** step to execute to that point.



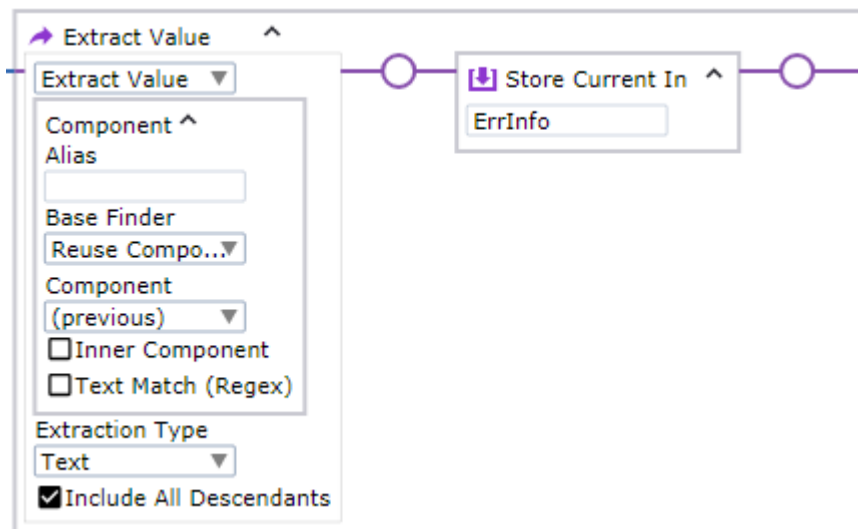
7. In the **Recorder View**, right-click the button with the error description and click **Guard > Location Found**.

Error: connect ECONNREFUSED 127.0.0.1:50081

This is a connection refused error that is shown when the Document Transformation Service is not running on the target computer.

A new guard is inserted in your workflow. This is the guard that extracts the information about the error. Rename it to **Extract Info**.

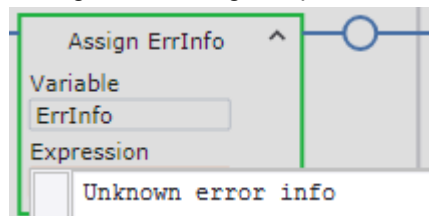
- Right-click the flow point next to the **Location Found** box, add an **Extract Value** step, and then configure it as follows.



With this step, the error message information is stored in the variable **ErrInfo**.

- Click the plus sign below the **Location Found** box to add a new guard. Set "When seconds have passed" to **5** seconds.
- Right-click the flow point next to the **When seconds have passed** guard, add an **Assign step**, and then rename it to **Assign ErrInfo**.

Configure the Assign step as follows.

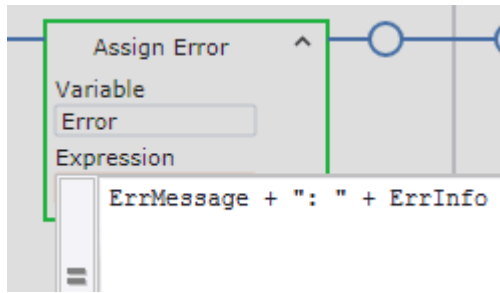


With these two steps, if an unknown error is encountered, the respective expression is stored in the variable **ErrInfo**.

8. Double-click the flow point next to the **Extract Info** step to execute to that point.

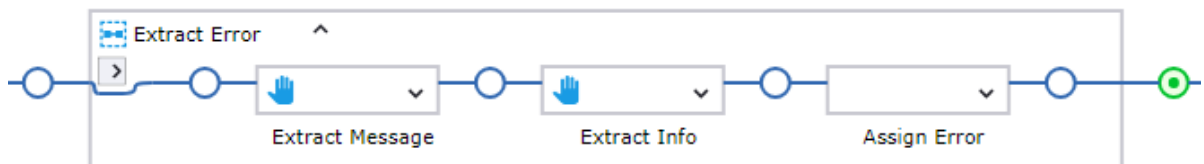
9. Right-click the flow point next to the **Extract Info** step, add an **Assign step**, and then rename it to **Assign Error**.

- In the **Variable** field, enter **Error**. This is one of the variables you specified in [Create the Document Transformation workflow](#).
- In the **Expression** field, enter the following expression: **ErrMsg + ": " + ErrInfo**  
Click the gray bar on the left so the equal sign appears, and the expression can be evaluated.

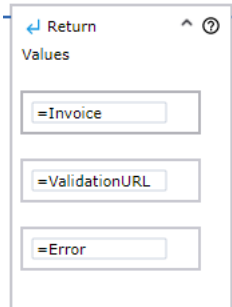


With this step, the **Error** variable will return both the error message name and the error information.

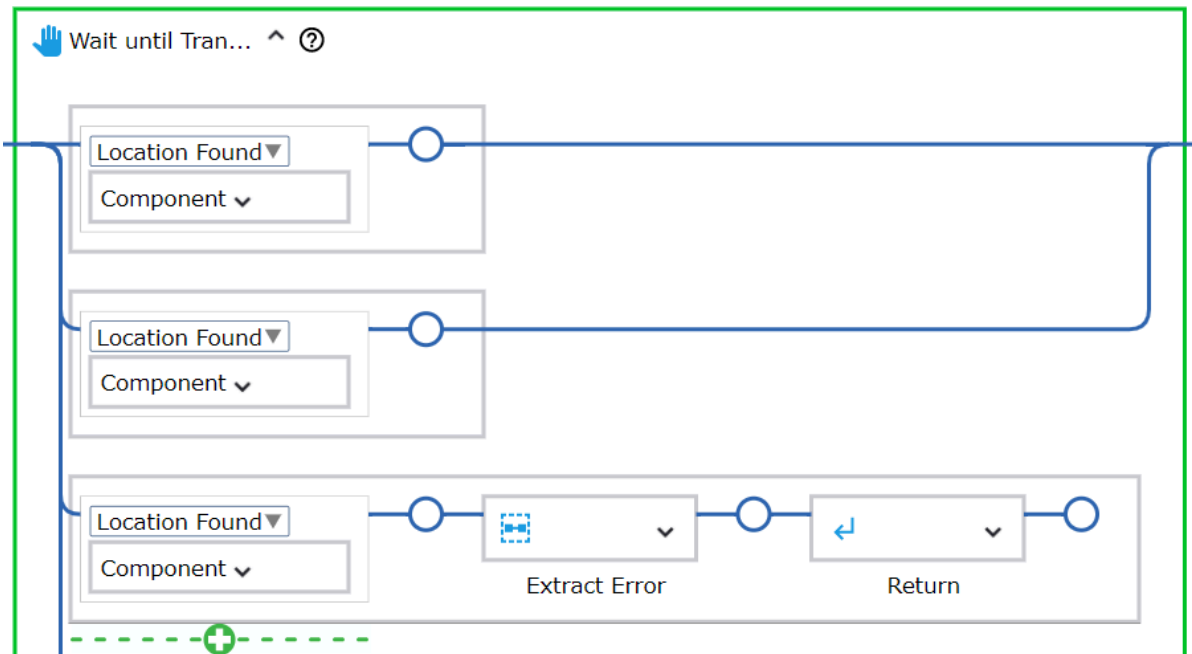
The **Extract Error** group should look similar to the following.



10. Right-click the flow point next to the group and add a **Return step**. Configure it as follows.



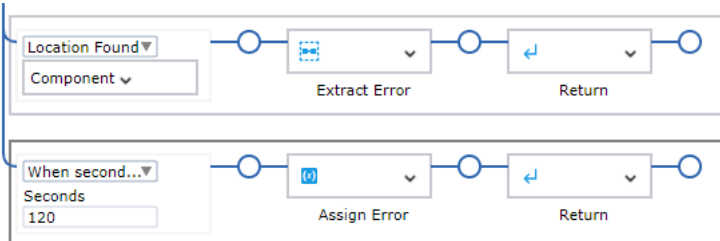
11. Now click the plus sign below the third **Location Found** guard to add another guard.



- Set "When seconds have passed" to **120** seconds.
- Next to this guard, add an **Assign** step and rename it to **Assign Error**.
- In the **Variables** field, enter **Error**.
- In the **Expression** field, enter the following text:  
**Waiting for transformation timed out.**
- After the **Assign Error** step, add the same **Return** step as in Step 10. You can copy and paste the step.

With these steps, if the transformation process takes too long, the expression specified above is stored in the return variable **Error**.


The error handling part in the **Wait until Transformed** step should look similar to the following.



## Configure the robot to handle invalid and valid documents

In this procedure, you configure branches in the robot in Design Studio to handle three scenarios: when the document is valid or invalid, or if an error occurs during transformation. If the document is valid, the robot sends the values extracted in the Transform Document step in an email to a human. If the

transformed document is invalid, it is sent for human validation. If an error occurs, an email with error information is sent to a human.

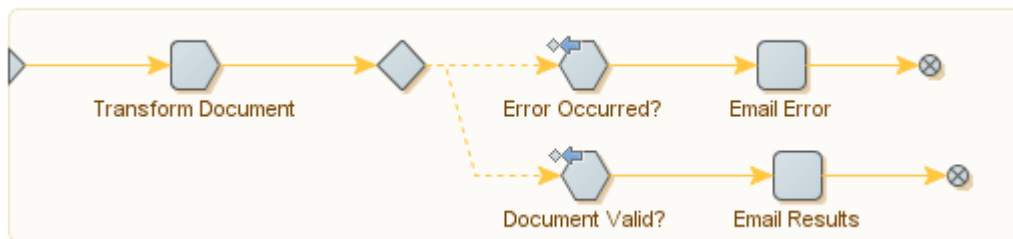
1. In **Design Studio**, add a **Try step** after the Transform Document step.  
Click the **Transform Document** step and, on the toolbar, click **Insert Step After Selected Step**  > **Try step**.
2. On the first branch, add an **Action step** and make it a **Test Value** step.  
Right-click the **Return Value** step, click **Insert Step Before** > **Action step** and then select **Test Value** on the **Action** tab.
  - a. In the **Condition** section, type `error==""`. Use straight quotation marks.
  - b. Set **If** and **Do** to **Condition is Satisfied** and **As Specified Under "Error Handling"**, respectively.
  - c. On the **Error handling** tab, in the **Then** list, select **Try Next Alternative**.
  - d. In the robot, rename the **Test Value** step to **Error Occurred?**.
3. After the **Error Occurred?** step, add an **Action step** and make it a **Send Email** step. On the **Action** tab, select **Output** > **Send Email**.
4. Configure the Send Email step as follows.
  - a. In the **FROM Address** and **TO Address** fields, enter the sender and recipient email addresses, respectively.
  - b. Set the **Message** field to **Expression** and add the following expression:  
`"Something went wrong during transformation or subsequent processing.\n" + error`
  - c. Set **Message Type** to **Text**.
  - d. In the **Subject** field, enter text similar to the following: "Transformation has failed."
  - e. On the **Server** tab, configure your SMTP server settings.
  - f. In the robot, rename the step to **Email Error**.



5. On the second branch, add an **Action step** and also make it a **Test Value** step.
  - a. In the **Condition** section, type `ValidationURL==""`. Use straight quotation marks.
  - b. Set **If** and **Do** to **Condition is Not Satisfied** and **As Specified Under "Error Handling"**, respectively.
  - c. On the **Error handling** tab, in the **Then** list, select **Try Next Alternative**.
  - d. In the robot, rename the **Test Value** step to **Document Valid?**.
6. Click the **Return Value** step and also make it a Send Email step.

7. Configure the second Send Email step as follows.
  - a. In the **FROM Address** and **TO Address** fields, enter the sender and recipient email addresses, respectively.
  - b. Set the **Message** field for this step to **Expression** and add the following expression:
 

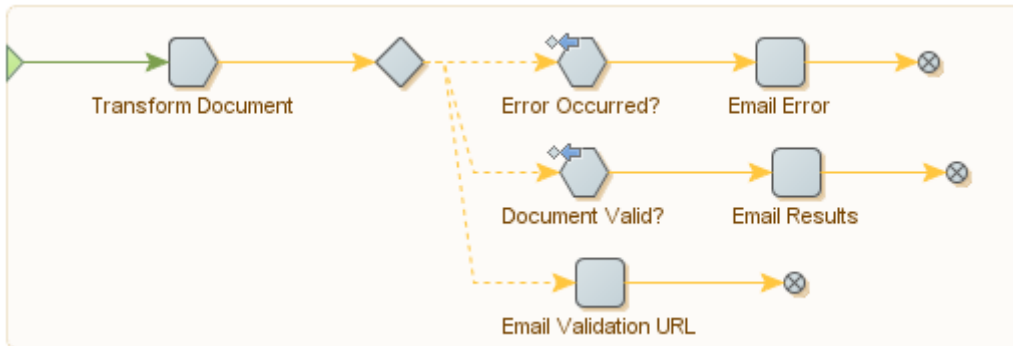
```
"VendorID=" + invoice.VendorID + "\n" + "InvoiceNumber=" + invoice.InvoiceNumber +
          "\n" + "InvoiceDate=" + invoice.InvoiceDate + "\n" + "Total=" + invoice.Total
```
  - c. Set **Message Type** to **Text**.
  - d. In the **Subject** field, enter text similar to the following: "This is the result of the robot execution."
  - e. On the **Server** tab, configure your SMTP server settings and save the changes.
  - f. In the robot, rename the step to **Email Results**.



8. On the third branch, add an **Action step** and also make it a **Send Email** step.
9. Configure the second Send Email step as follows.
  - a. In the **FROM Address** and **TO Address** fields, enter the sender and recipient email addresses, respectively.
  - b. Set the **Message** field for this step to **Expression** and add the following expression:
 

```
>><a href="<<+ValidationURL+>>">Click me!</a><<
```
  - c. Set **Message Type** to **HTML**.
  - d. In the **Subject** field, enter text similar to the following: "Please validate this invoice."
  - e. On the **Server** tab, configure your SMTP server settings.
  - f. In the robot, rename the step to **Email Validation URL** and save the changes.

The robot workflow is now ready for use, and it looks similar to the following:



**Note** The **Send Email** steps cannot be run in Design mode and you need to execute them in Debug mode. To switch to this mode, on the toolbar, click **Debug**  and then click **Single Step** .

It is not possible to execute in Design and Debug modes at the same time.

If the document is valid, the robot emails the values from the transformed document: VendorID, Number, Date, and Total. If the transformed document is invalid, an email containing the Validation URL is sent to a human who can click it to validate the document. If an error occurs during the transformation process, an email containing the error information is sent to a human.

Upload the robot to the Management Console. You can now test your robot in Design Studio.

All available documents that require validation can be now found at the following URL:

<http://127.0.0.1:50082/Validation.aspx>

In this URL, **127.0.0.1** is the IP address of the Thin Client Server, which can be different, depending on where it was installed.

For an introduction to the human validation procedure, see the next section.

## Validate the transformed document

This topic describes basic steps that you can perform to validate the transformed invoice.

After you open the Validation URL <http://127.0.0.1:50082/Validation.aspx> in a browser, open the batch containing your transformed document. In the Classification Result pane on the left, you see the Document Transformation results.



**Tip** To facilitate and accelerate the process of validating the document, we recommend that you use the keyboard whenever possible. For example, press Ctrl+O to open a batch or Ctrl+S to suspend a batch (save and close until later). To use a shortcut, press the keys consecutively (one after the other), not simultaneously.

1. The field next to the ID Number field is selected requiring you to confirm the ID.

ID Number	<input type="text" value="198000"/>		<input type="text"/>
Company Code	<input type="text" value="3"/>		

To confirm that the correct ID is extracted, press Enter.

ID Number	<input type="text" value="198000"/>		<input type="text" value="Confirmed"/>
Company Code	<input type="text" value="3"/>		

2. The next potentially invalid field, Invoice Number, is now selected.

- Invoice -

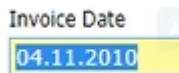
Invoice Number

You can check the number in the top right corner of the document. It is marked red to notify you that it needs validation.

Date: 06.11.2010  
Invoice No: 67-90943

To validate the field and confirm that the correct invoice number is extracted, press Enter.

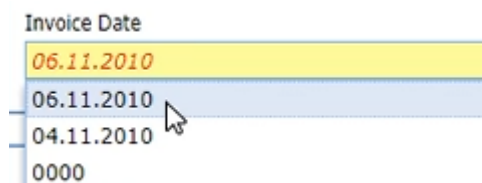
3. The next potentially invalid field, Invoice Date, is shown.



In the top right corner of the document, you can see that the actual date is 06.11.2011. The extracted date was deliberately made invalid for training purposes.

To fix the date, you can follow one of the following methods.

- Use automatic field completion. First, clear the incorrect date using the Backspace key. Type "0" to see all of the numbers in the document that start with a zero. Press the Down Arrow key to select "06.11.2010".



The invoice date in the document is now marked green.



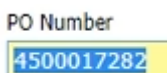
Press Enter to confirm the new date.

**Tip** When validating documents, you can enable the Online Learning feature to increase the rate of field recognition on similar documents. This feature is based on remembering the layout of a sample document, such as an invoice. By using automatic field completion, manually typing or selecting the correct value in the document (such as the correct invoice date), you contribute to the knowledge base. As a result, when you work on a similar document next time, the invoice date will be correct automatically. The feature is enabled on the General tab in the Project Settings.

- In the document, click or lasso the invoice date to select it, and then press Enter.

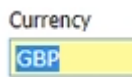


4. The next potentially invalid field, PO Number, is now shown.



Confirm that the extracted number is correct. After you confirm that the PO number is correct, the value of the Document Type field, which used to be marked as invalid, is automatically confirmed.

5. The Currency field is now selected.



Confirm that the extracted currency is correct.

After you confirm all of the values, you receive a notification that the batch is valid. You can now close it. The extracted values will be sent in an email to a human as you configured in [Configure the robot to handle invalid and valid documents](#).

You have now completed the tutorial.