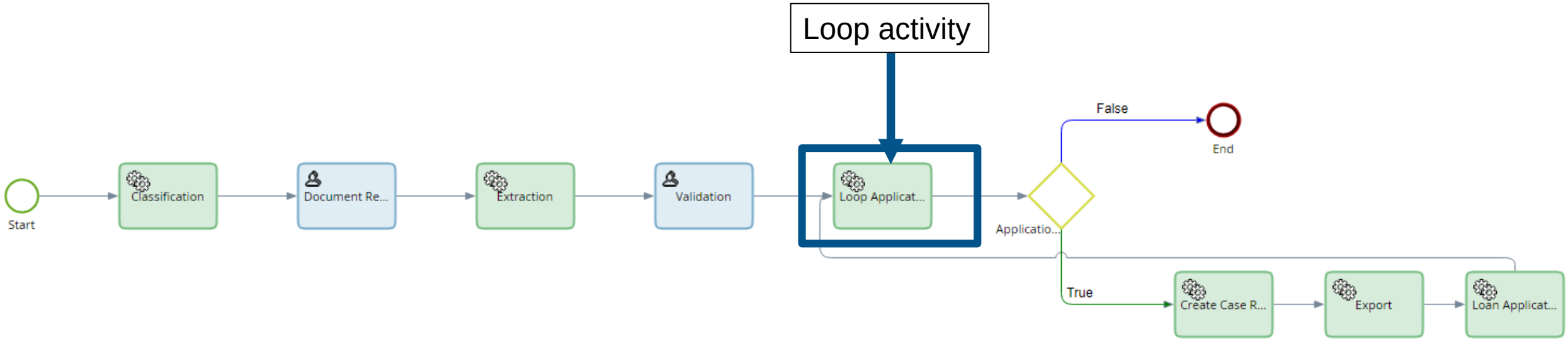
# Loop through Documents in a Folder

# Training Scenario

- In the training example when a collection of  loan applications are scanned they are stored in a folder
- For each document (application) we want to create a Loan Application process
- We will use a Loop activity to iterate through the folder and create an instance of a downstream loan application process for each document
- The downstream process is case process so you will generate a case reference for the case using an expression activity that combines the customers first and last name

# Process a Single Folder

- Use a Loop activity to loop through the Applications Folder
- Configure the process as below

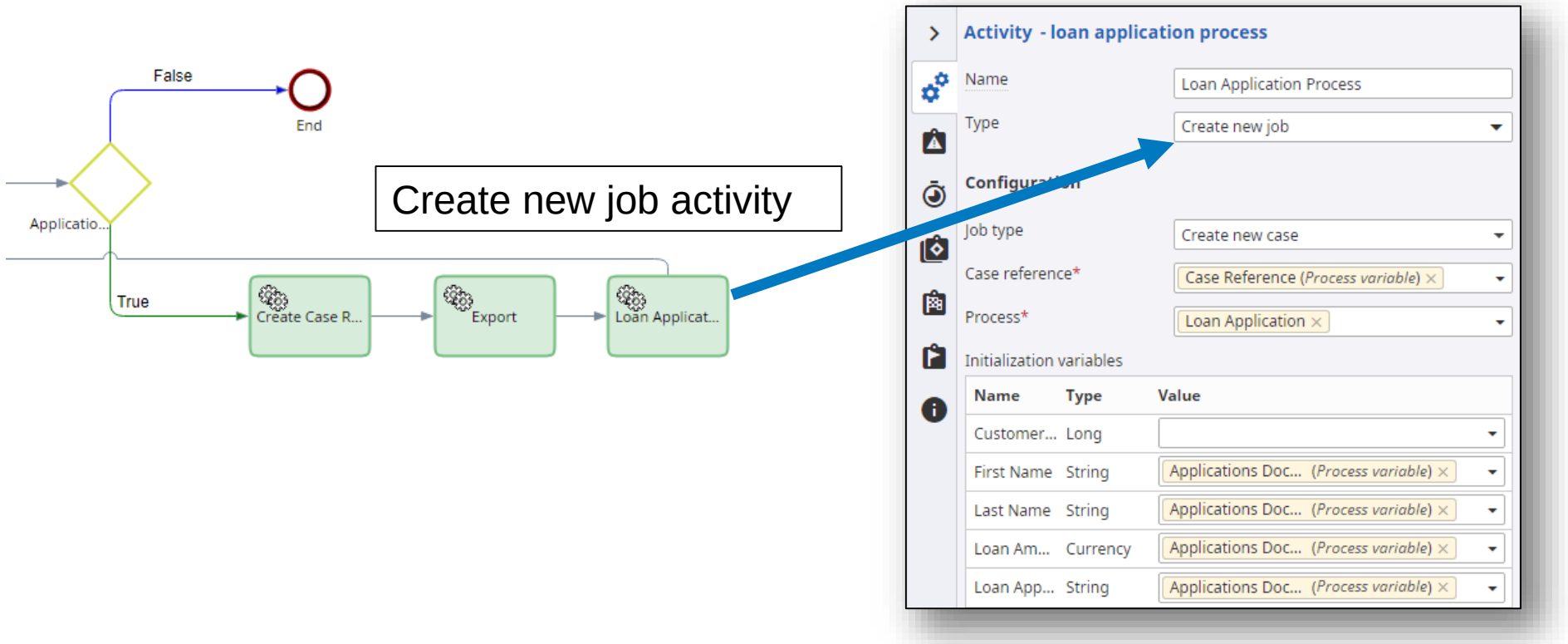





KOFAX

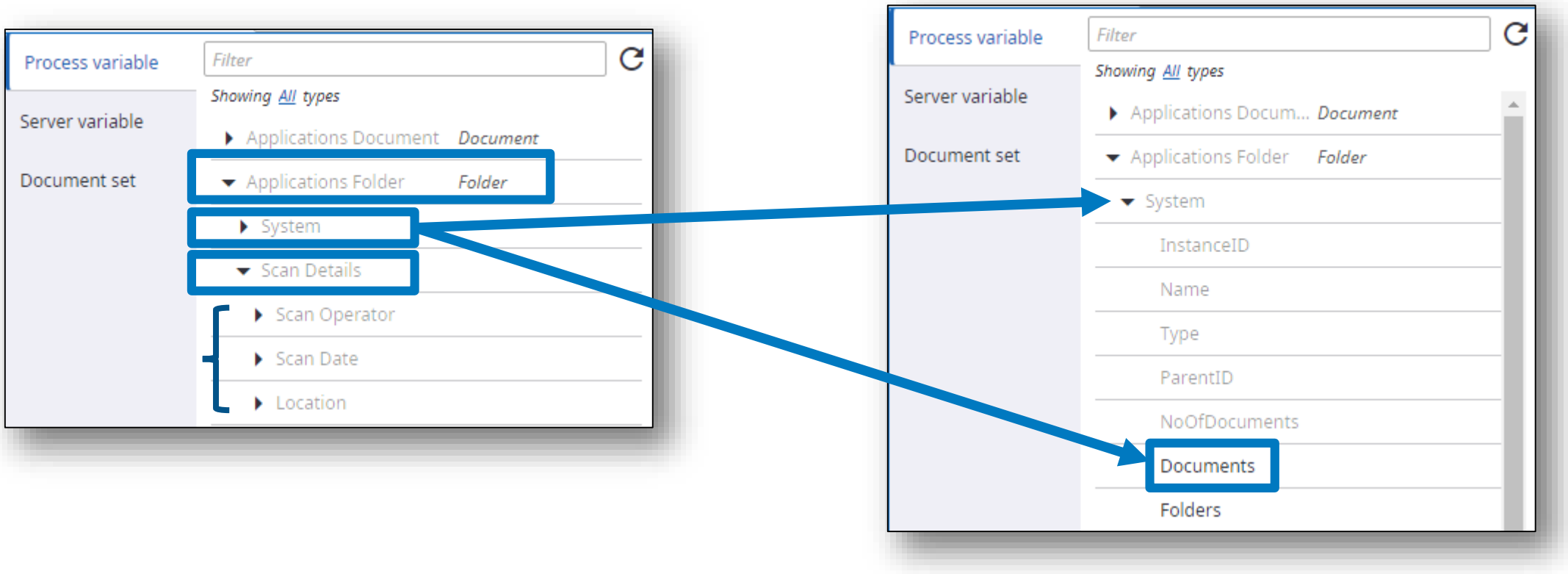
# Loop through a Folder in the Capture Process

- Use a Loop activity to loop through the Applications Folder
- When a row (Loan Application) is Found an instance of the Loan Application process is started using the Create new job activity
- The Capture Applications process has a Folder as an Initialization variable



# Loop through a Folder

- The **Loop** activity can be used to loop through the contents of a Folder variable
- The **Folder** variable has a number of **System** fields that can be accessed at design time
- The Documents field give you access to the documents in the folder



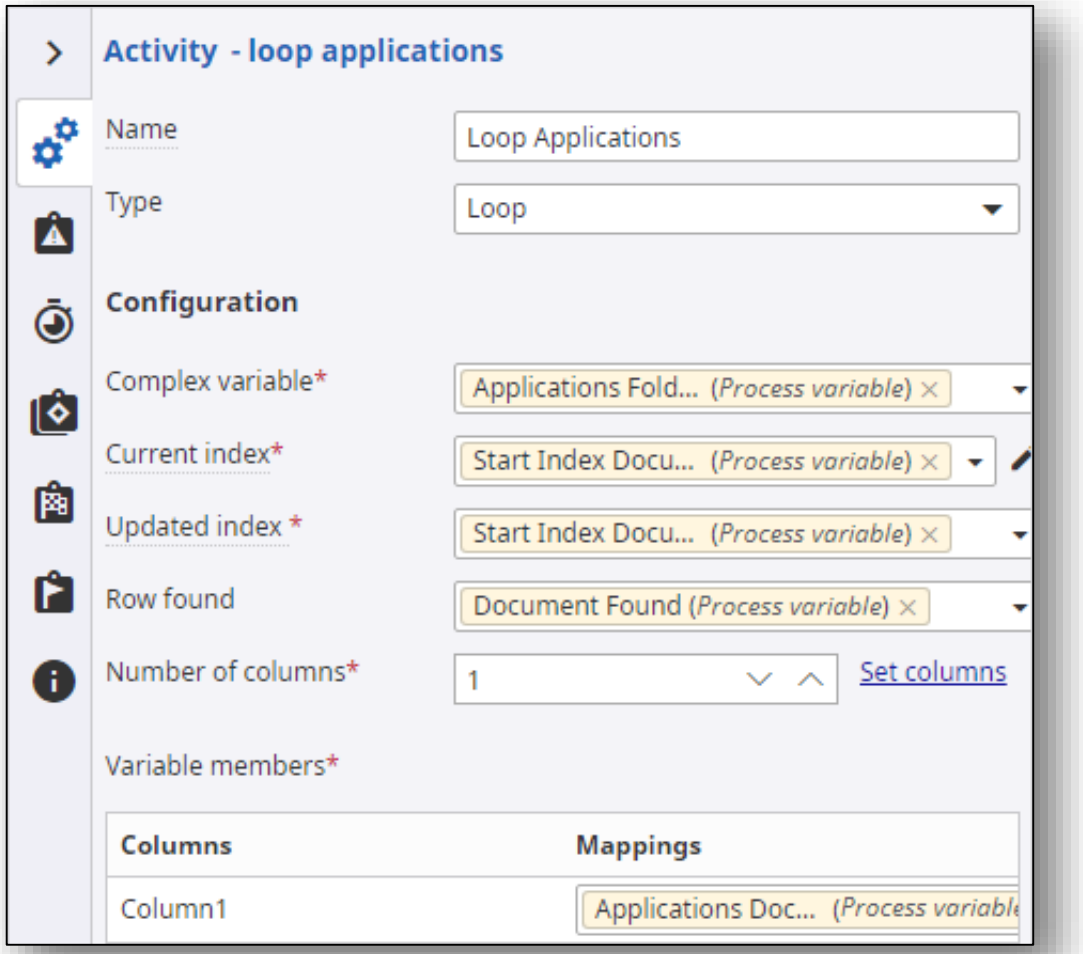



KOFAX

# Loop through Documents in a Folder

- The **Folder** variable **Documents** system field can be passed into the Loop activity (Complex variable)
- The **Current index** for the Loop activity is 1
- Each time a new row is found **Updated index** is incremented by 1
- **Row found** indicates whether  a folder has been found
- When a document is found we store the document in a new document variable called Application Document

# Loop through Documents in a Folder






KOFAX

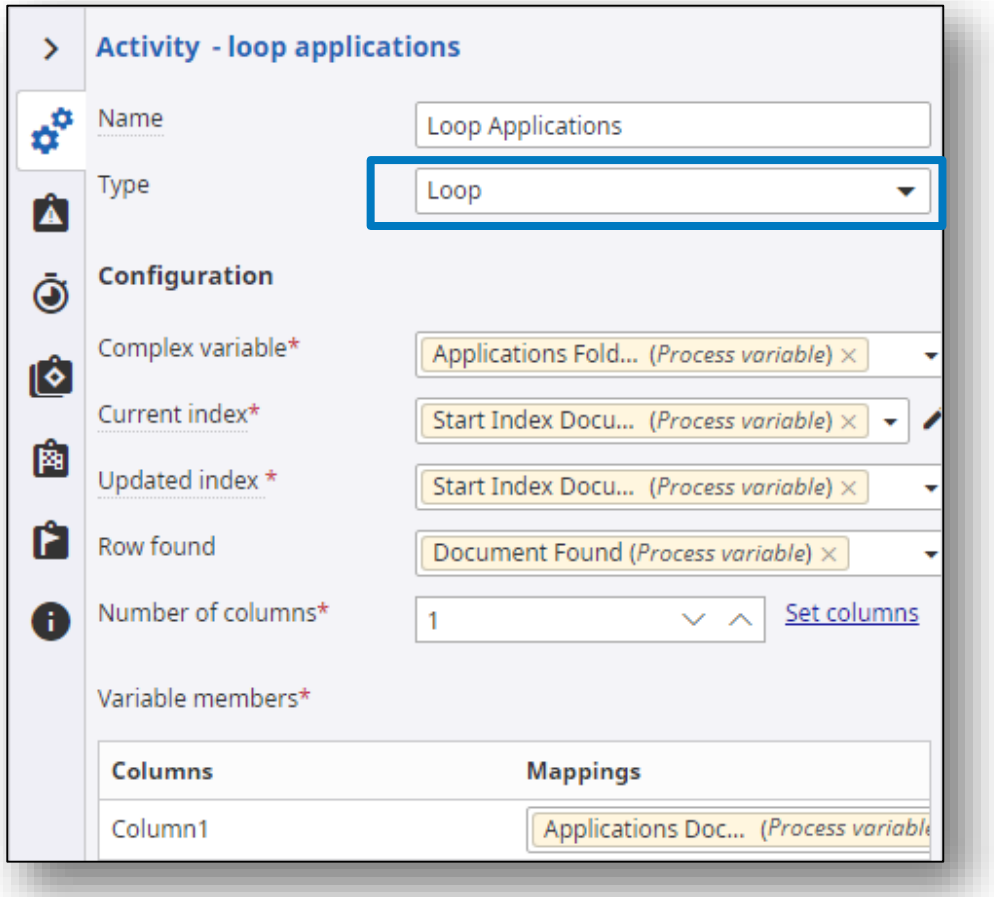
# Pass Extracted Data to a Downstream Process

- You may need to pass extracted data into a downstream process
- Although the data (Extracted fields, Folder fields, and Documents) is held in a folder you may not wish to pass the entire folder into the downstream process
- When looping through the **Documents System** field there will be different document types, so each document is stored can be stored in a document variable of type **Default Document**
- To expose the fields for a specific document type you can create a document variable with the appropriate document type and use an Expression activity to populate the variable (i.e. pass the document id from the **Default Document** variable to the **Application Document** variable

If document types share the same fields you can choose the parent document type if you wish to pass the fields to another process

# Configuring a Loop Activity

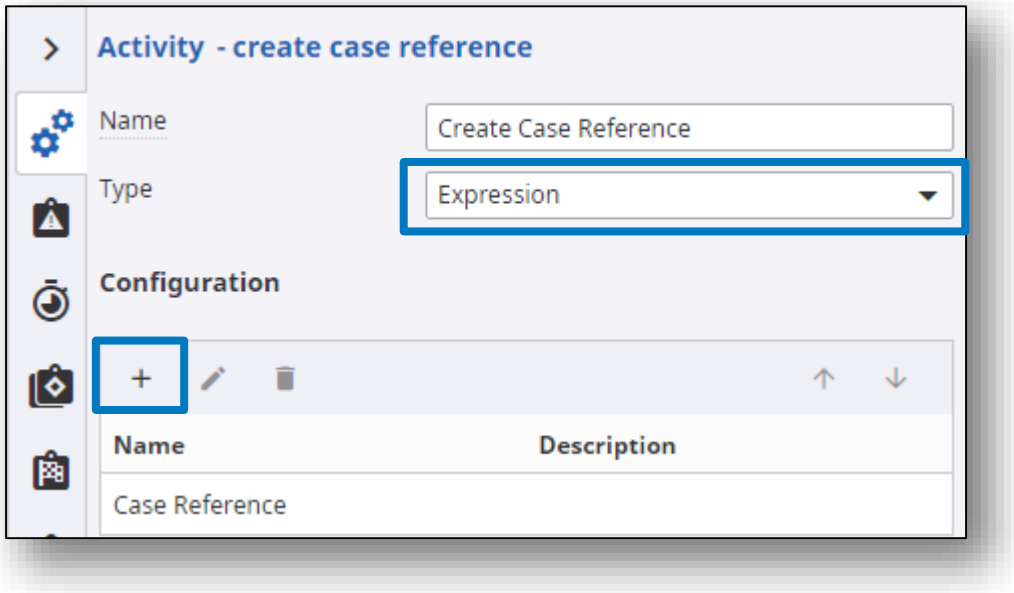
- The Loop activity helps you iterate through a list of data (array) and perform one or more steps against each row of data
- For Example, to send an email to a list of resources, you would perform a DB query to return the list.  To process each resource, use the loop activity to iterate through each resource on the list or until the end of the data




Variable members are the actual variables to be set with the required data from complex variable data columns. The number of columns value should be equal to the number of variable members that exist while configuring the loop activity. Each column of variable members must be mapped to a variable of the required type.

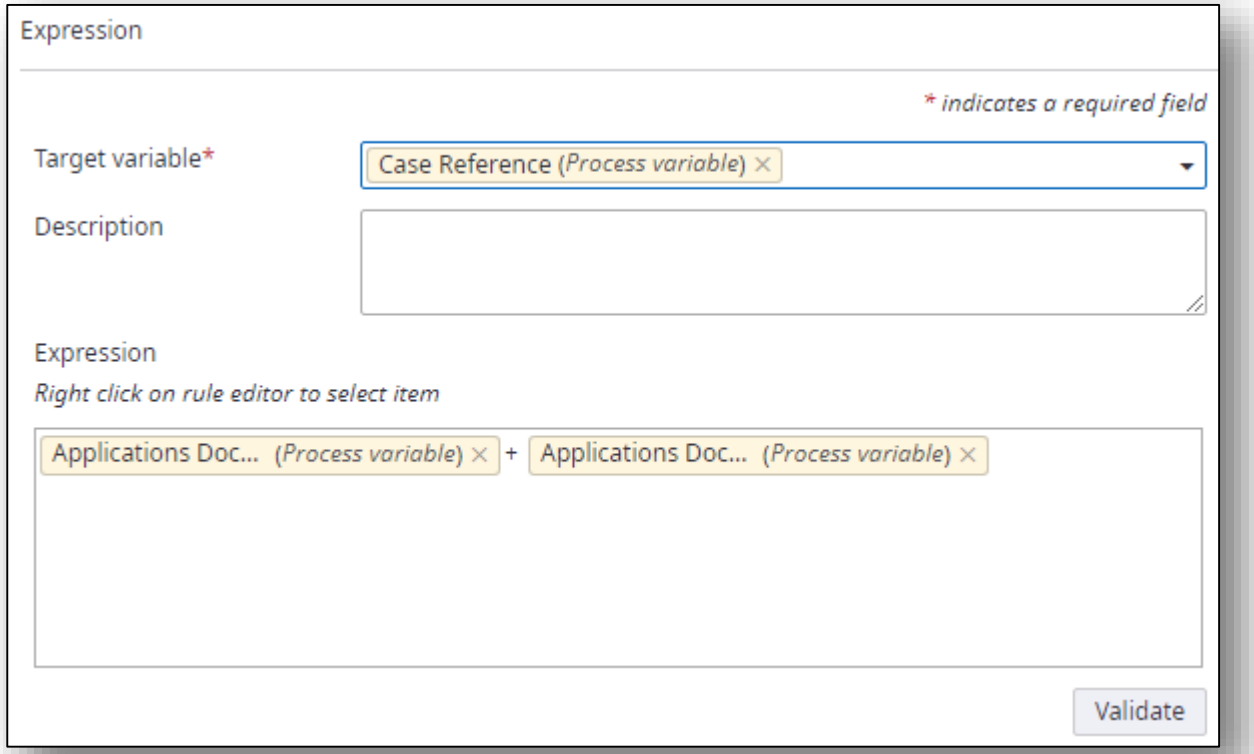
# Configuring an Expression Activity

- An Expression activity helps you set the value of one or more variables using free text, variables, operators, and functions
- For example, if the name of the customer is  Abc Xyz, and you need to display the initials, AX, the rule can be:
  - Left(FirstName,1) + Left(Surname,1)
- The value from this rule is held in an output (target) variable



# Configuring an Expression Activity (cont.)

- An Expression activity can use Text, Date and Math functions (refer to the Help for Kofax TotalAgility for more information about these functions)




An Expression activity supports the Bool (Boolean), Byte, Currency, Date, Decimal, Double, Float, Long, Nullable Date, Nullable String, Short, and String variables. It does not support Checklist, Complex, Dynamic Complex, XML, XML Expression, and System variables.

# Lab: Creating a Capture Enabled Process (Capture Applications)