KOFAX Education

TECHNICAL TRAINING LAB INSTRUCTIONS

In this lab you will learn how to create and use variables. Variables are containers for data. Data can be passed into a job when it is first created (Initialization data), retrieved from an external source while the job is running, and can be assigned as inputs (read only), and outputs (updateable), to an activity. As an example, your process may require customer type data e.g. name, address, etc. if you wish to generate a letter and send it to the customer as part of the process.

Lab 11-1

Variables

Create Process Variables

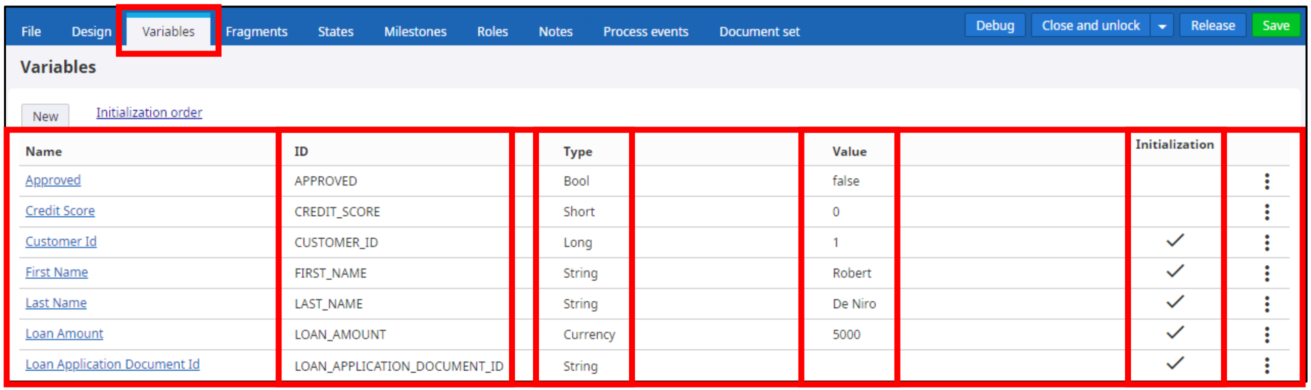
In this part, you will create process variables in the loan application process.

Process variables are used to store data or to reference documents/folders, for example when a loan is applied for, the **Loan Amount** variable holds the amount of the loan being applied for, e.g. 5000. When a loan decision is required the **Approved** variable is set to True or False.

When a variable is assigned as a **Process Initialization** variable its value is provided at the beginning of the process.

1. With the **Loan Application** case definition already open, from the **Ribbon**, select **Variables**.

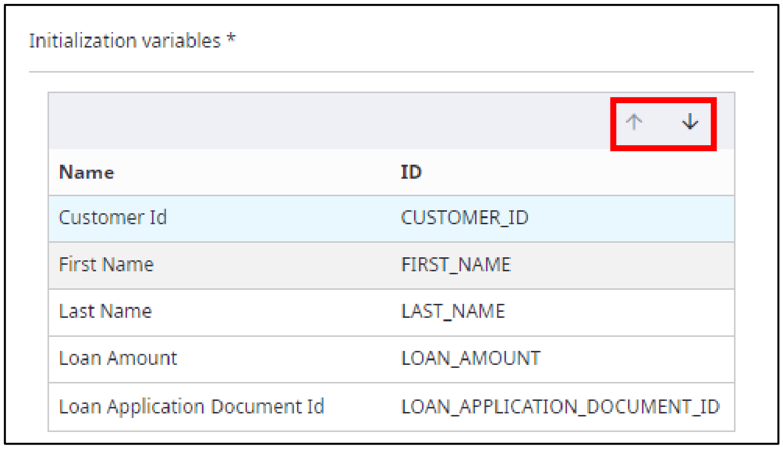
2. Add the following new process variables and default **Values** as shown below.



Name	ID		Type		Value		Initialization	
Approved	APPROVED		Bool		false			⋮
Credit Score	CREDIT_SCORE		Short		0			⋮
Customer Id	CUSTOMER_ID		Long		1		✓	⋮
First Name	FIRST_NAME		String		Robert		✓	⋮
Last Name	LAST_NAME		String		De Niro		✓	⋮
Loan Amount	LOAN_AMOUNT		Currency		5000		✓	⋮
Loan Application Document Id	LOAN_APPLICATION_DOCUMENT_ID		String				✓	⋮

Notes: (i) Loan Amount, Customer Id, First Name, Last Name, and Loan Application Document Id have been set as process initialialization variables i.e. the process expects to be initialized with this data.
(ii) The variable name and Id are different for Credit Score, Loan Amount, First Name, Last Name, Loan Application Document Id, and Customer Id (underscores between words in the ID field).

3. With the **Variables** list page still open, click **Initialization order** and set the Initialization order to **Customer Id, First Name, Last Name, Loan Amount,** and **Loan Application Document Id**.
Note: TotalAgility uses the initialization order when you build a Create new job form later in the course. The fields will be created on the form in the order you specify.

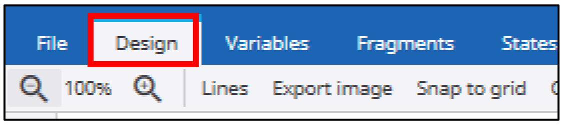


Initialization variables *

Name	ID
Customer Id	CUSTOMER_ID
First Name	FIRST_NAME
Last Name	LAST_NAME
Loan Amount	LOAN_AMOUNT
Loan Application Document Id	LOAN_APPLICATION_DOCUMENT_ID

4. Click **OK** to return to the **Variables** list page.

5. From the **Ribbon**, select the **Design** tab.



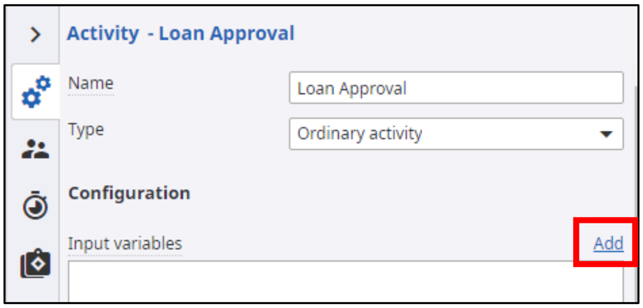
Note: Once you create a process variable, you cannot modify its ID or type. You must delete the process variable and re-create it.

Add Input and Output Variables to Activities

In this part, you will add variables to activities in the **Loan Application** process. For the **Loan Approval** activity an approver will need to know the customer's details and credit score (inputs), before they can approve or reject the loan (outputs).
Also notice the use of the **Credit Score** variable, a variable specific to this activity, which cannot be provided by the loan applicant when the loan application is first applied for. We will retrieve the customer's credit score from a **Customer** database later on in the labs.

6. Select the **Loan Approval** activity.

7. From the **Properties** panel, from the **Input variables** area, click **Add**.



Activity - Loan Approval

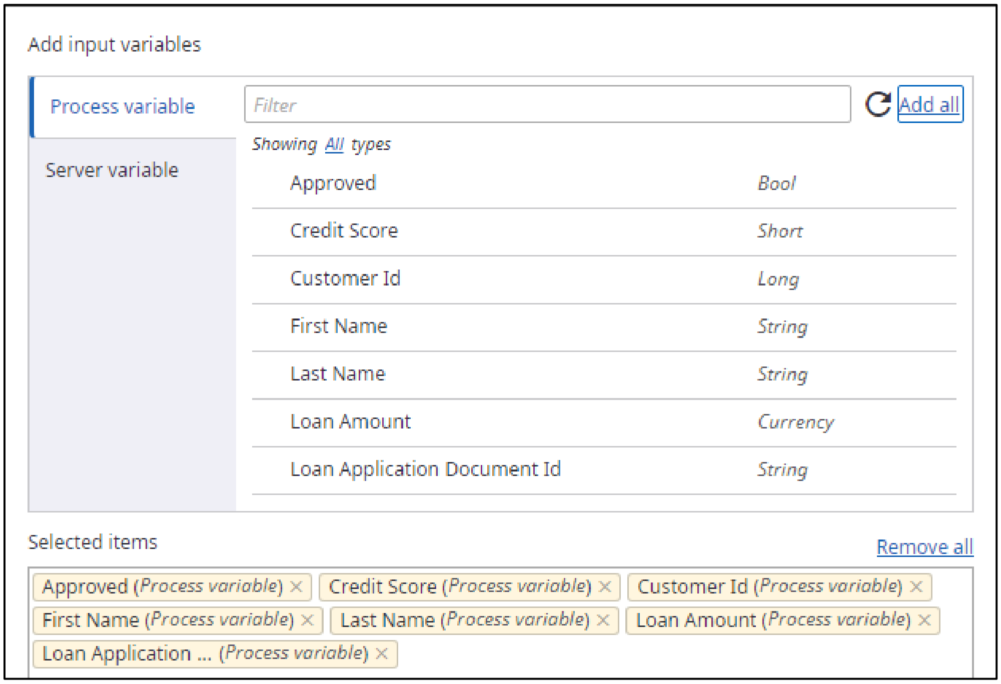
Name: Loan Approval

Type: Ordinary activity

Configuration

Input variables

8. Click **Add all** to add all the variables to the **Selected items** list (you can also click each variable individually to add to the Selected items list).



Add input variables

Showing All types

Approved	*Bool*
Credit Score	*Short*
Customer Id	*Long*
First Name	*String*
Last Name	*String*
Loan Amount	*Currency*
Loan Application Document Id	*String*

Selected items

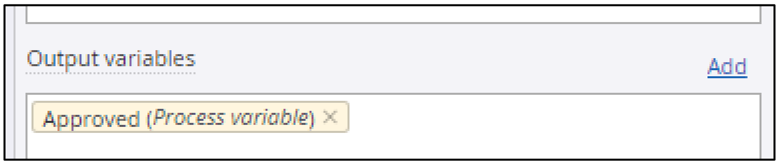
Approved (*Process variable*) × Credit Score (*Process variable*) × Customer Id (*Process variable*) × First Name (*Process variable*) × Last Name (*Process variable*) × Loan Amount (*Process variable*) × Loan Application ... (*Process variable*) ×

Note: The Loan Application Document Id will be used in a later lab to display a customer's Loan Application document in a Web Capture Control on the Loan Approval form.

9. **Delete** the **Approved** variable from the **Selected items** area.

10. Click **Done.**

11. From the **Properties** panel, from the **Output variables** area, click **Add**, click **Approved**, and click **Done**.



Output variables

Approved (*Process variable*) ×

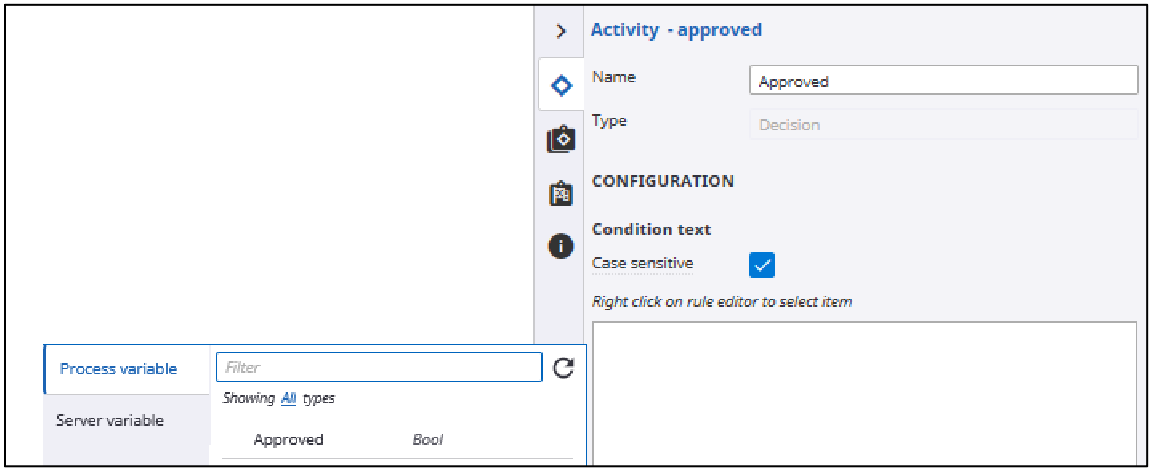
12. **Save** the process.

Test a Variable Value in a Decision

In this part, you will test the value of a variable in a Decision.

13. Select the **Approved** decision activity.

14. In the **Condition text** rule editor, delete the word **True**. Right click and from the **Process variable** list select **Approved**.


Activity - approved

Name: Approved

Type: Decision

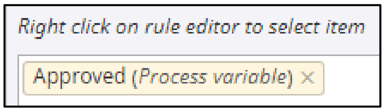
CONFIGURATION

Condition text

Case sensitive ☑

Right click on rule editor to select item

Notes: (i) [Approved] and [Approved]=True are equivalent.
(ii) You can also use the following loigical operators/keywords: <, >, =, <>, >=, <=, ! (not), And, Or etc.


Right click on rule editor to select item

Approved (*Process variable*) ×

15. **Save** the process (leave the process open for the next part).

In this lab, you will add a data access activity to the Loan Application Case Definition. In this example we will use the data access activity to retrieve the applicant's credit score from an external database.

Lab 11-2

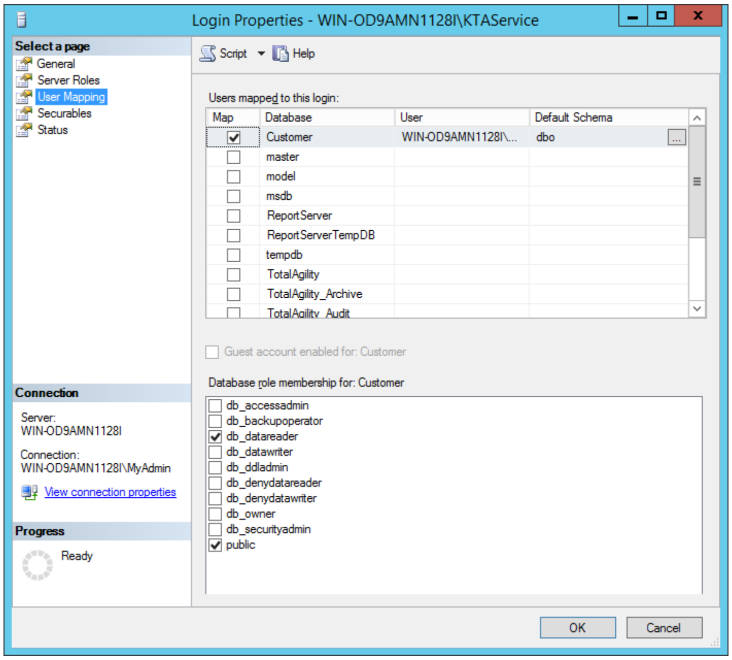
Data Access Activity

Add a Data Access Activity

In this part, you will retrieve a credit score from an external SQL database called Customer. To connect to an external database, a connecting string is required. A server variable holding the connection string already exists on your environment. Note that the KTAService account will execute the data activity. The service account must be granted read access to the external database in SQL Server Management Studio.

The connection string is: *Server=(local);Trusted_Connection=Yes;Database=Customer;*
Note*: This server variable can be found by going to the* **Main** menu and clicking **System data > Server variables** and *filtering by the* ***XYZ Bank*** *category.*

1. Skip to step 7 if you are using a Kofax training machine:
2. From the Taskbar open **SQL Server Management Studio** and click **Connect**.
3. Expand the **Security** folder and expand the **Logins** folder.
4. **Double click** the **KTAService** account and select the **User Mapping** page.
5. Select the **Customer** database and grant the **db_datareader** role to the service account.



Server:
WIN-OD9AMN1128I

Connection:
WIN-OD9AMN1128I\MyAdmin

Users mapped to this login:

Map	Database	User	Default Schema
☑	Customer	WIN-OD9AMN1128I\...	dbo
☐	master		
☐	model		
☐	msdb		
☐	ReportServer		
☐	ReportServerTempDB		
☐	tempdb		
☐	TotalAgility		
☐	TotalAgility_Archive		
☐	TotalAgility_Audit		

☐ Guest account enabled for: Customer

Database role membership for: Customer

☐ db_accessadmin
☐ db_backupoperator
☑ db_datareader
☐ db_datawriter
☐ db_ddladmin
☐ db_denydatareader
☐ db_denydatawriter
☐ db_owner
☐ db_securityadmin
☑ public

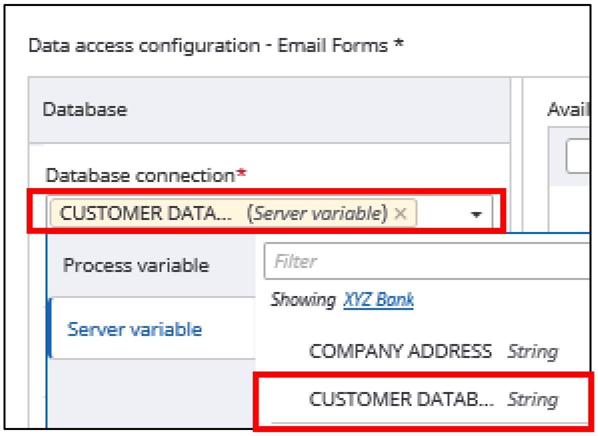
6. Click **OK** and **close** SQL Server Management Studio.

7. Select the **Credit Check** activity in the **Loan Application** process.

8. From the **Type** dropdown, select **Data access**.

9. Click **Configure**.

10. Click the **Database connection** dropdown, select the **Server variable** tab, and change the **Category** to **XYZ Bank**. Select the **CUSTOMER DATABASE** server variable.
Note: You can learn about server variable encryption and whitelisting in the Lab Notes.



Data access configuration - Email Forms *

Database

Database connection*

CUSTOMER DATA... *(Server variable)* ×

Process variable

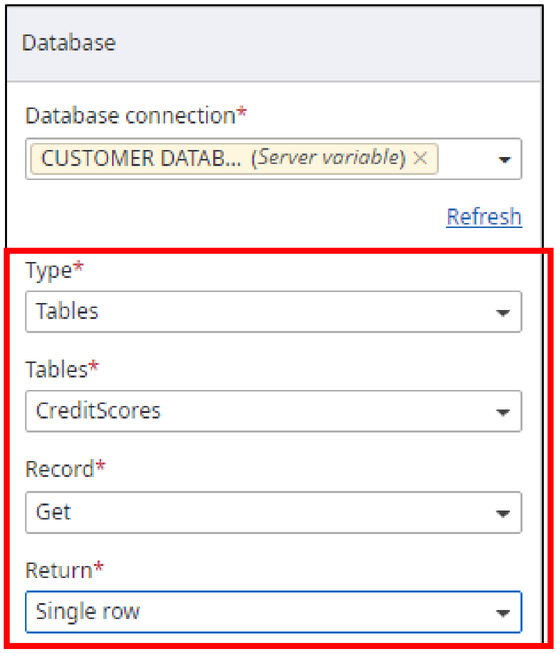
Server variable

Showing XYZ Bank

COMPANY ADDRESS *String*

CUSTOMER DATAB... *String*

11. From the **Type** dropdown, select **Tables**, for **Tables** select **CreditScores**, for **Record** select **Get** and for **Return** select **Single Row.**



Database

Database connection*

CUSTOMER DATAB... *(Server variable)* ×

Type*

Tables

Tables*

CreditScores

Record*

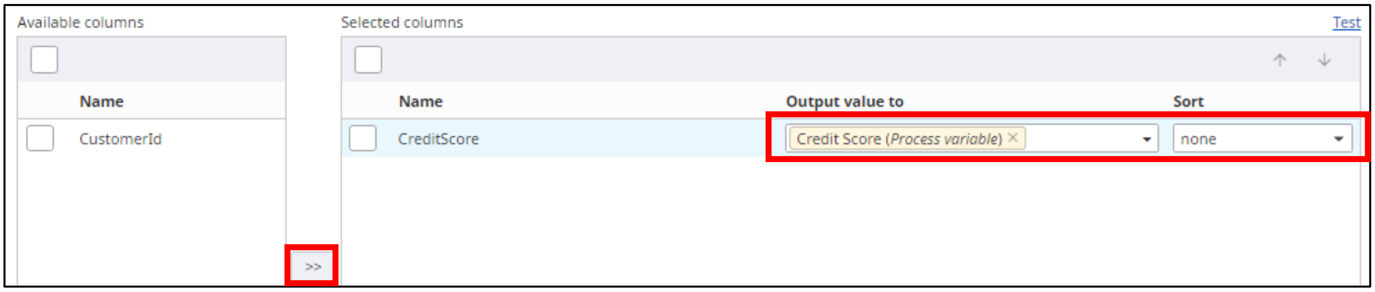
Get

Return*

Single row

12. In the **Available columns** list, check the **CreditScore** checkbox, and click the **Add >>** button to move the column to the **Selected columns** list.

13. From the **Output value to** dropdown, select **Credit Score** (process variable).



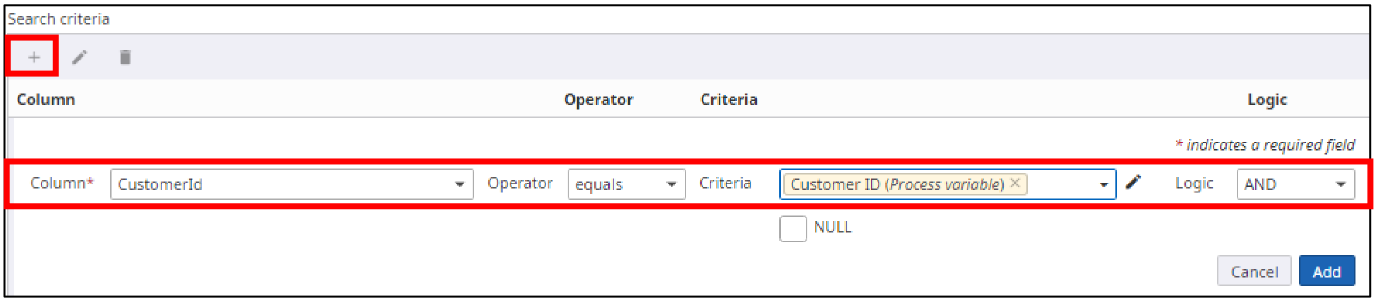
Available columns

Name
CustomerId

Selected columns

Name	Output value to	Sort
CreditScore	Credit Score (*Process variable*)	none

14. In **Search criteria** click **Add** (+ icon) and configure the following:



Search criteria

Column	Operator	Criteria	Logic

* *indicates a required field*

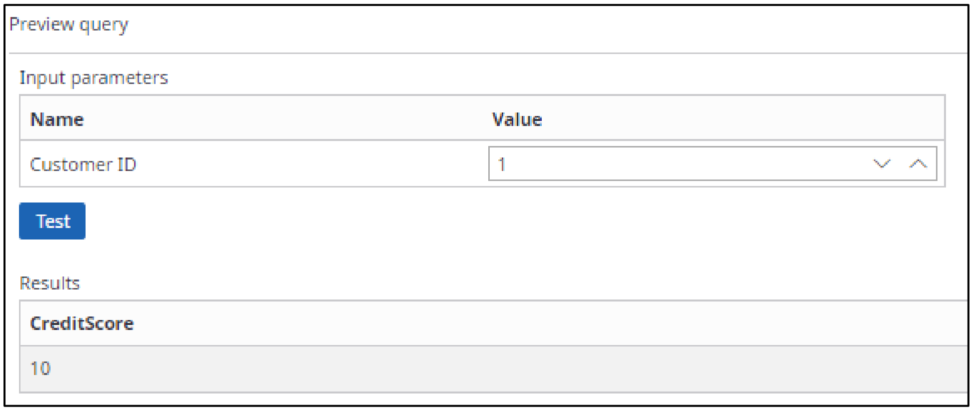
Column* CustomerId Operator equals Criteria Customer ID (*Process variable*) Logic AND

□ NULL

15. Click **Add** to add the criteria.

16. Click **Test** (top right).

17. Enter **1** as the **Customer Id** and click **Test**. The Results for the test should return **10**.



Preview query

Input parameters

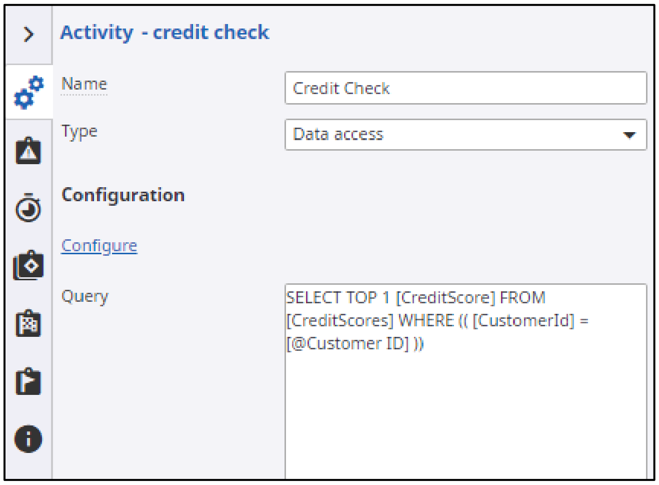
Name	Value
Customer ID	1

Results

CreditScore
10

Hint: To verify results, you can open Microsoft SQL Server Management Studio (from Start menu), login and open the **CreditScores** table in the **Customer** database.

18. Click **Close** on the Test window and **OK** to close the data access configuration. The Query is now listed in the **Query** field in the **Properties** panel.



Activity - credit check

Name: Credit Check

Type: Data access

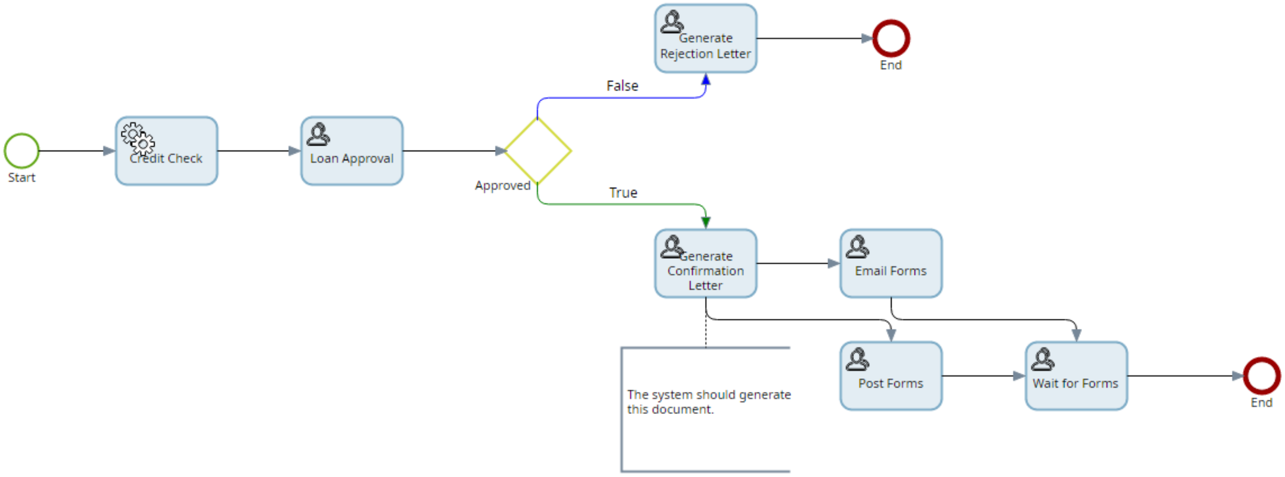
Configuration

Configure

Query:

```
SELECT TOP 1 [CreditScore] FROM
[CreditScores] WHERE (( [CustomerId] =
[@Customer ID] ))
```

19. **Release** the **Loan Application** process. The process looks like this:



Start
Credit Check
Loan Approval
Approved
False
True
Generate Rejection Letter
End
Generate Confirmation Letter
Email Forms
Post Forms
Wait for Forms
End
The system should generate this document.

Important Note

You will test the credit check activity at run-time in a later lab. In the next lab, to save time, you will delete the process you have created and import an identical process which has some extra variables and some additional configuration. You can learn how to export a process as part of package in the Export Solution as a Package self-learning lab.

Lab 11-3

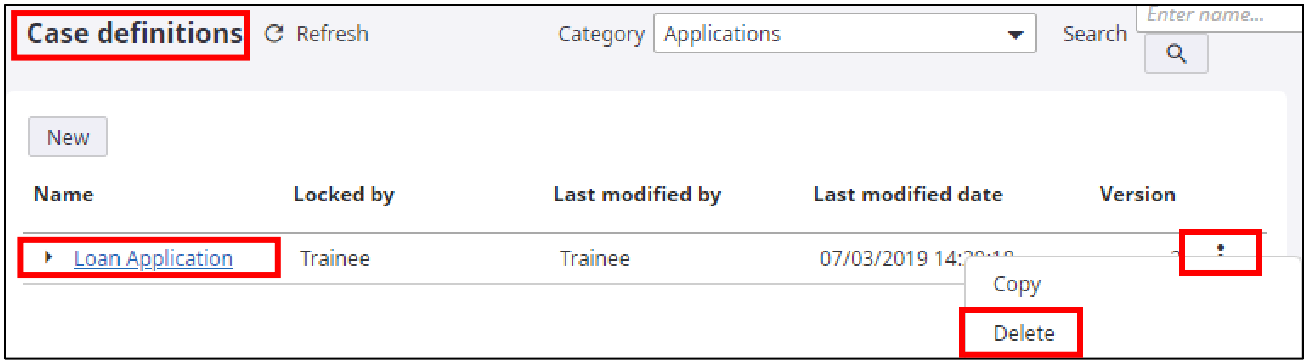
Import New Version of Loan Application

Import Process

In this part, you will delete the existing **Loan Application** process. You will then import a new version of the **Loan Application** process that has pre-configured activity notification in the **Loan Approval** activity as well as pre-defined variables for sending emails.

Delete Loan Application Process

1. **Close and unlock** the **Loan Application** case process.
2. From the **Main** menu navigate to **Workflow > Case definitions**.
3. Locate the **Loan Application** case definition in the list page.
4. Click the ⋮ and from the context menu, select **Delete**.



Category: Applications

Name	Locked by	Last modified by	Last modified date	Version
Loan Application	Trainee	Trainee	07/03/2019 14:	

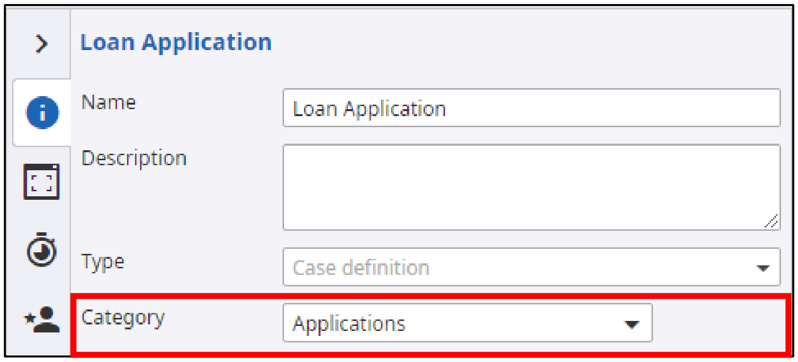
5. Click **Yes** to delete all versions of the process and any associated fragments.
   Note: You cannot delete a process if there are live jobs in the system based on that process. Run job clear down first to remove live jobs before deleting the process.

Import New Version of Loan Application and Configure

In this part, you will import a new version of the Loan Application process. Because the version is assigned to the Default category you will reassign the process to the Applications category you created in an earlier lab.

6. From the **Main** menu navigate to **Import…**
7. **Browse** to and open **C:\KTAEssentials\LabFiles\Processes\Loan Application V2.zip**
8. Click **Import**.
9. From the **Main** menu navigate to **Workflow > Case definitions** and from the **Category** dropdown, select **Default Category**.
10. Open the **Loan Application** process.

11. In the **Properties** panel, from the **Category** dropdown, select **Applications**.



Loan Application

Name: Loan Application

Description:

Type: Case definition

Category: Applications

12. **Release** and **Close and unlock** the case definition.

Lab Notes

Sever Variables

1. It is possible to encrypt Server String variable types by setting the **Secure** property of the server variable to checked.

2. You can whitelist connection strings. From the **Main** menu navigate to **System** > **System settings**, and from the **System** card, click **Whitelist**.

Data Access Activity

3. The **Data Access** activity also supports aggregate functions (Max, Min, and Count) and calling Views and Stored Procedures.

4. You can also connect to Oracle databases using an Oracle connection string. KTA supports the **OLEDB** driver. An example connection string is shown below:
   Provider=OraOLEDB.Oracle;Data Source=SecureOracle;User Id=KTA_USER;Password=W0rkstation;

Revision Questions

1. What is a process variable used for?

   ____

   ____

2. What is a process initialization variable?

   ____

3. A process variable can only be used within the process that it has been created in?

   ____

4. A server variable can be accessed by?

   ____

5. What is an Input variable?

   ____

   ____

6. What is an Output variable?

   ____

   ____

7. You can change the ID of a variable after it has been created?

   ____

8. The initialization order of variables in a process determines the order of fields that are displayed on a create new job form?

   ____

9. Which account must be given database access when using the data access activity to interact with a non TotalAgility database?

   ____

10. What type of variable would be required if you are returning multiple rows from a data access query?

    ____

11. TotalAgility allows you to run queries against which object types in SQL Server?

    ____

12. Can you add multiple conditions to a query?

    ____

13. To connect to an external database, you will need to define a?

    ____