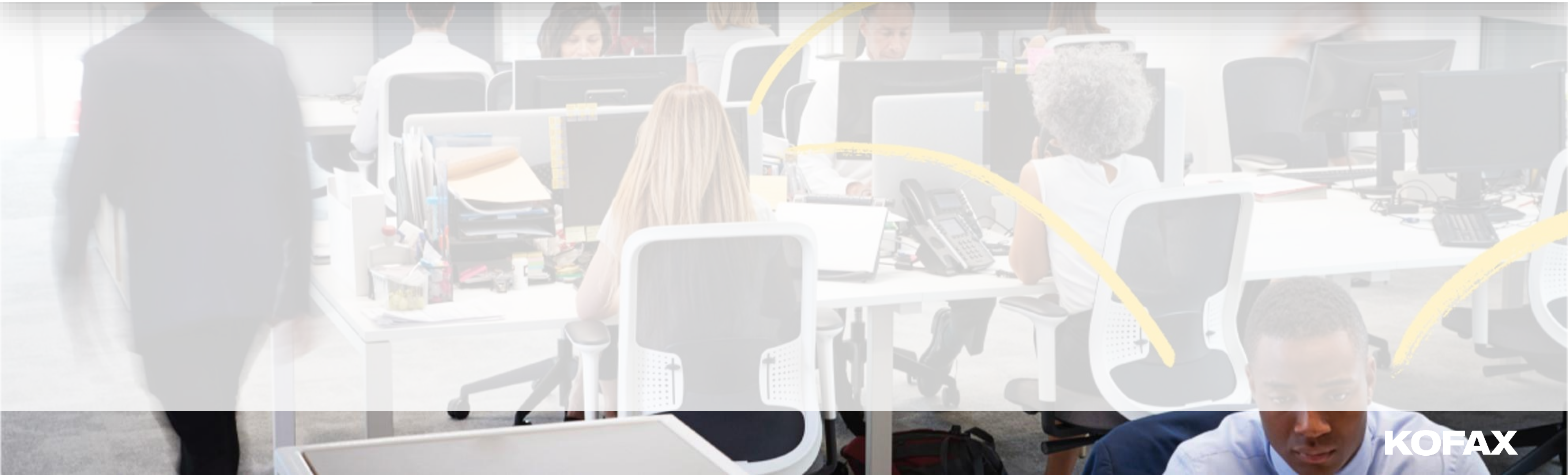
Debugging

Work Like Tomorrow.™

KOFAX

Process Debugging



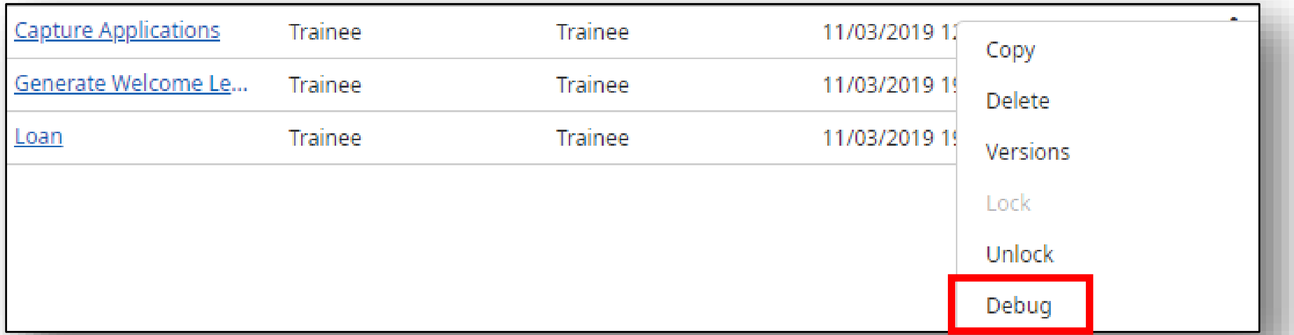
Process Debugging Overview

- You can use the Debug option to test/troubleshoot a process, case, fragment, business rule, or synchronous map with typical debug functionality
  - Set breakpoints
  - Update data
  - Run to next breakpoint
  - Step over

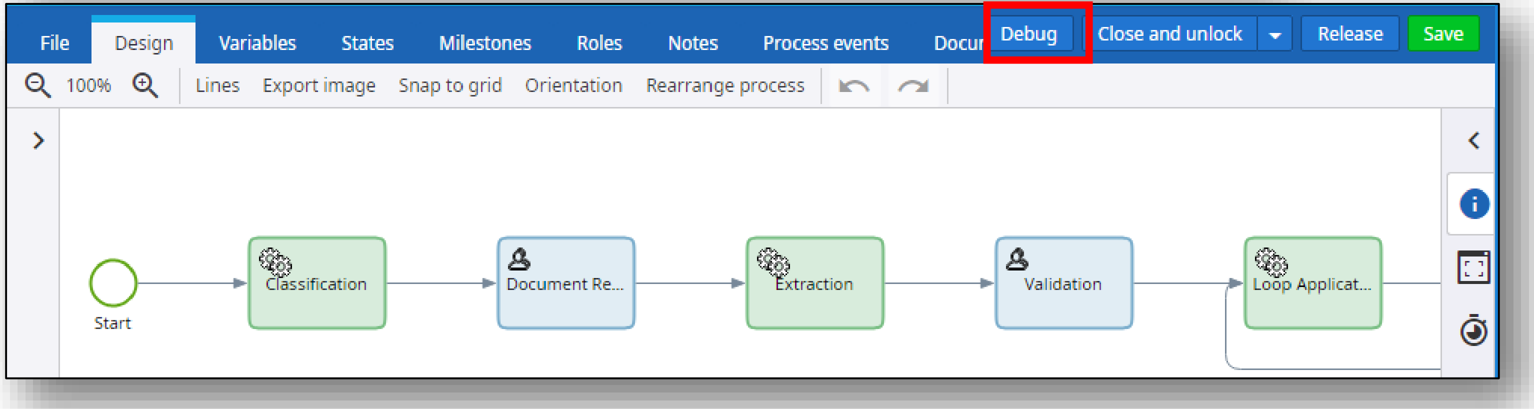
- Benefit
  - Quicker solution development time
  - More efficient trouble shooting

Debugging a Process

- You can launch the debugger from
  - Context menu on the list pages
  - Using the Debug button on the ribbon when the process is open
- This will open a new browser tab

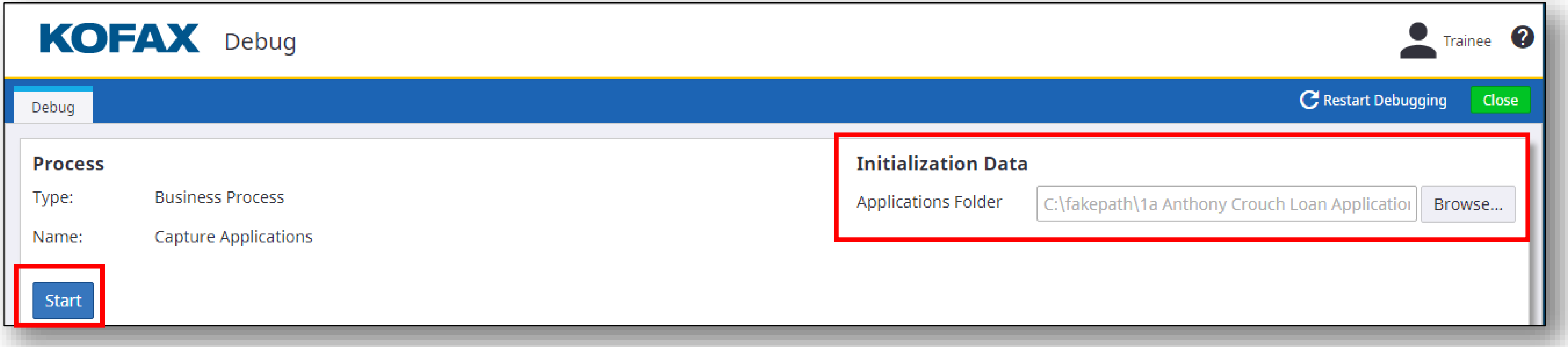


Capture Applications	Trainee	Trainee	11/03/2019 1
Generate Welcome Le...	Trainee	Trainee	11/03/2019 1
Loan	Trainee	Trainee	11/03/2019 1



Debugging a Process

- The debugger appears in a new browser tab
- The user can provide initialisation data before starting the debug job
- You can browse to the filing system if you wish to pass in a folder of documents



Process

Type: Business Process

Name: Capture Applications

Initialization Data

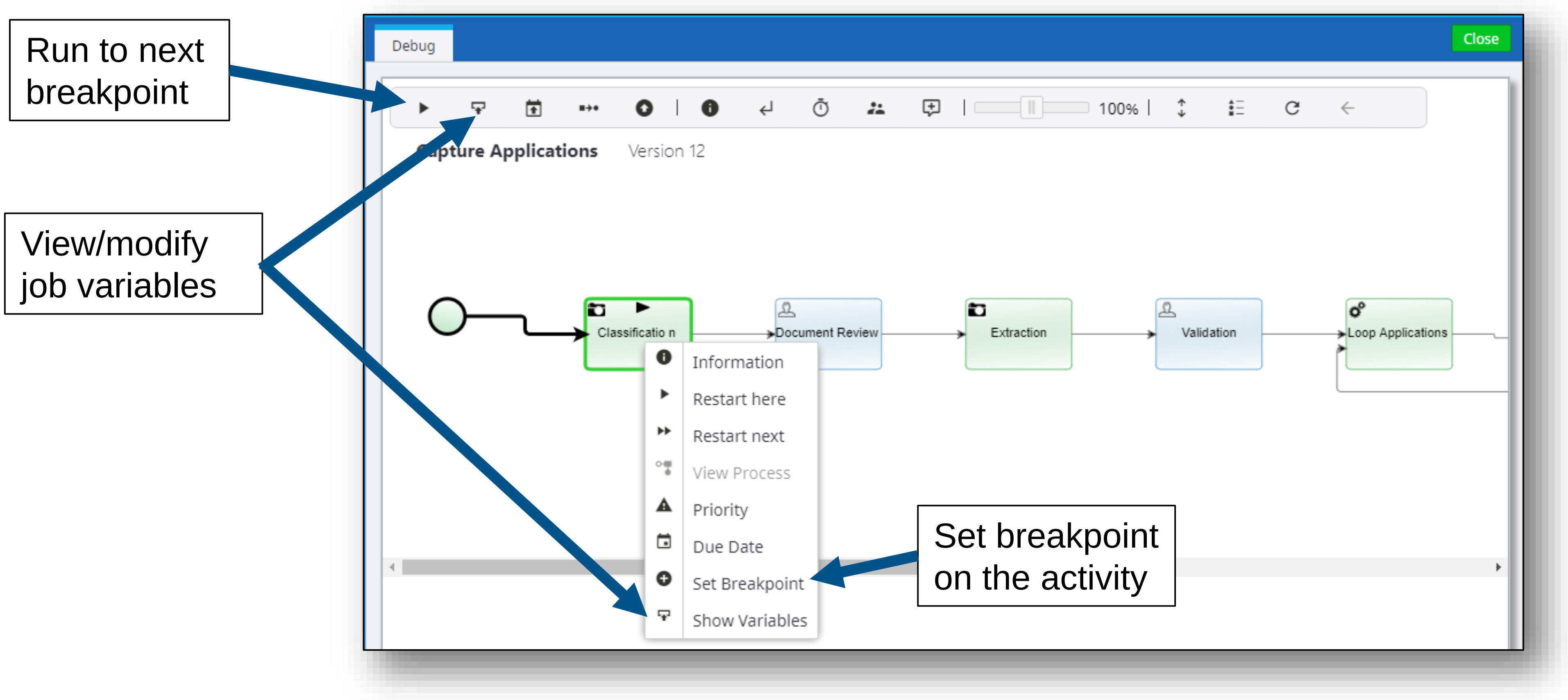
Applications Folder: C:\fakepath\1a Anthony Crouch Loan Applicatioı

Debugging a process

- When you run the start the debugger a job is created
- As the job is in a special state the user will be able to open this within the workspace and continue debugging using the View Job option in the context menu
- The debug job will not be archived
- Job clear down will include debug jobs

Debugging a Process

- The debug control works similar to the job viewer control within the workspace with some additional functionality



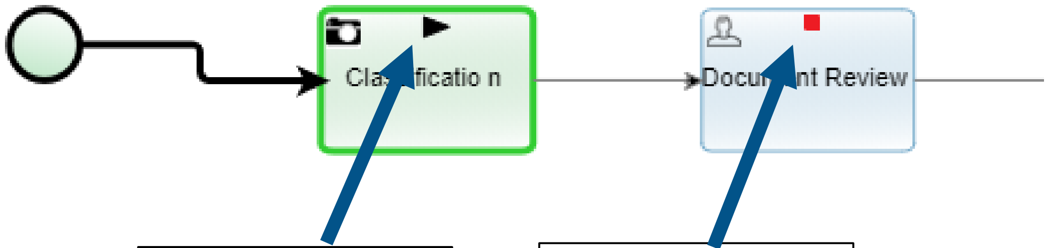
Run to next breakpoint

View/modify job variables

Set breakpoint on the activity

Setting a Breakpoint

- Click on an activity and select Set Breakpoint



Move to next activity

Breakpoint set on the activity

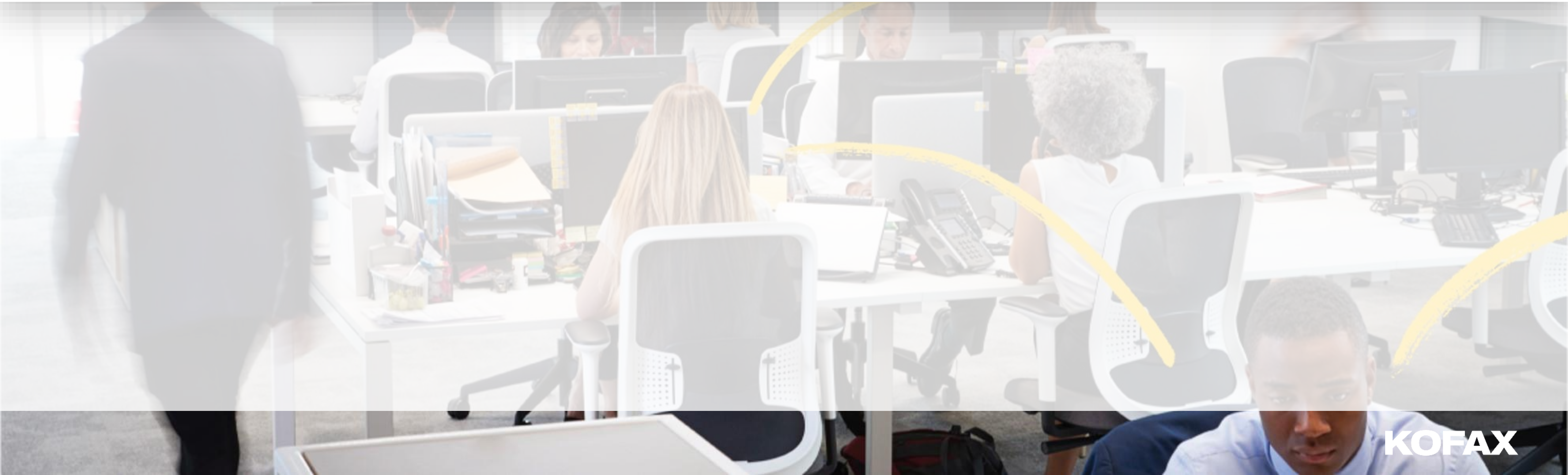
Debug Jobs

- Debugging a business rule or synchronous job will create an asynchronous job
- When the user debugs a process template and provides initialisation data that would activate the skin, the skin will be used for debugging
- Saved processes can be used for debugging

Debug Activities

- All manual activities will be executed as Synchronization activities but when the resources are viewed via the context menu the correct resources should be listed but these can’t be updated
- All automatic activities including a business rule will be executed as per runtime behaviour, there is no capability to debug further into these
- Parallel paths will execute as per runtime and only when dependents are all executed will the job progress

Form Debugging



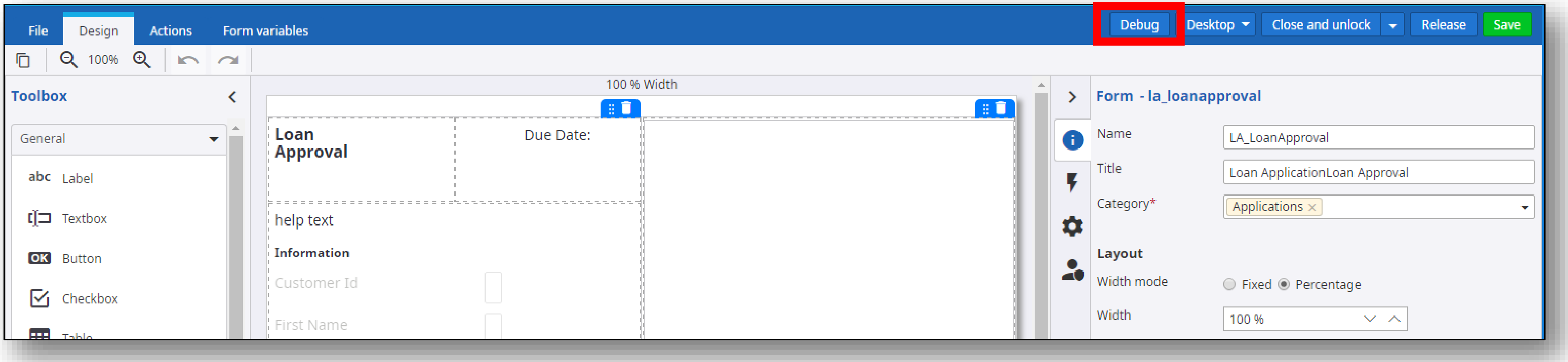
Form Debugging Overview

- You can use the Debug option to provide a means of debugging/testing a form with typical debug functionality:
  - Set breakpoints
  - Update data
  - Run to next breakpoint
  - Step over

- Benefit
  - Quicker solution development time
  - More efficient trouble shooting

Debugging a Form

- You can launch the debugger from
  - Context menu on the list pages
  - Using the Debug button on the ribbon when the process is open
- This will open a new browser tab



Loan Approval

Due Date:

help text

Information

Customer Id

First Name

Form - la_loanapproval

Name: LA_LoanApproval

Title: Loan ApplicationLoan Approval

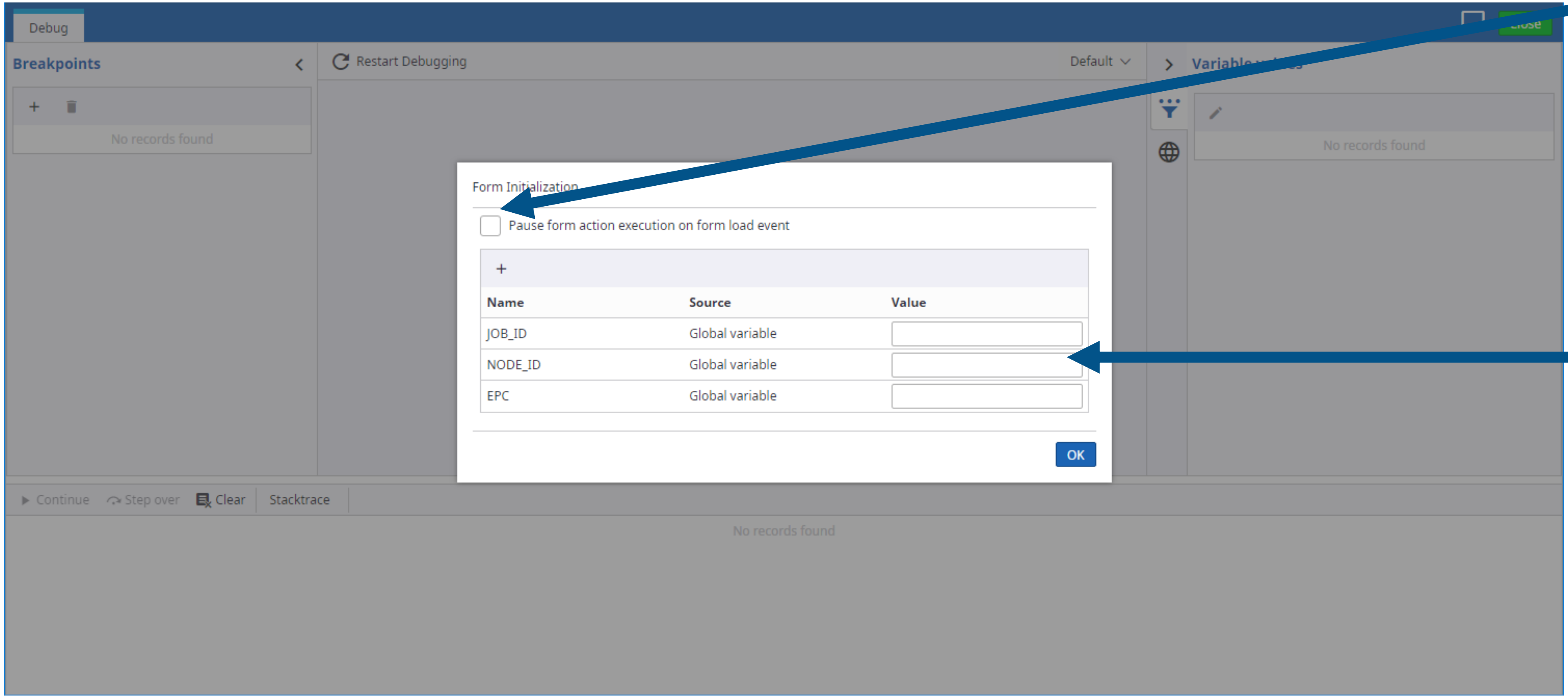
Category*: Applications

Layout

Width mode: ○ Fixed ● Percentage

Width: 100 %

Debugging a Form



Form Initialization

□ Pause form action execution on form load event

+

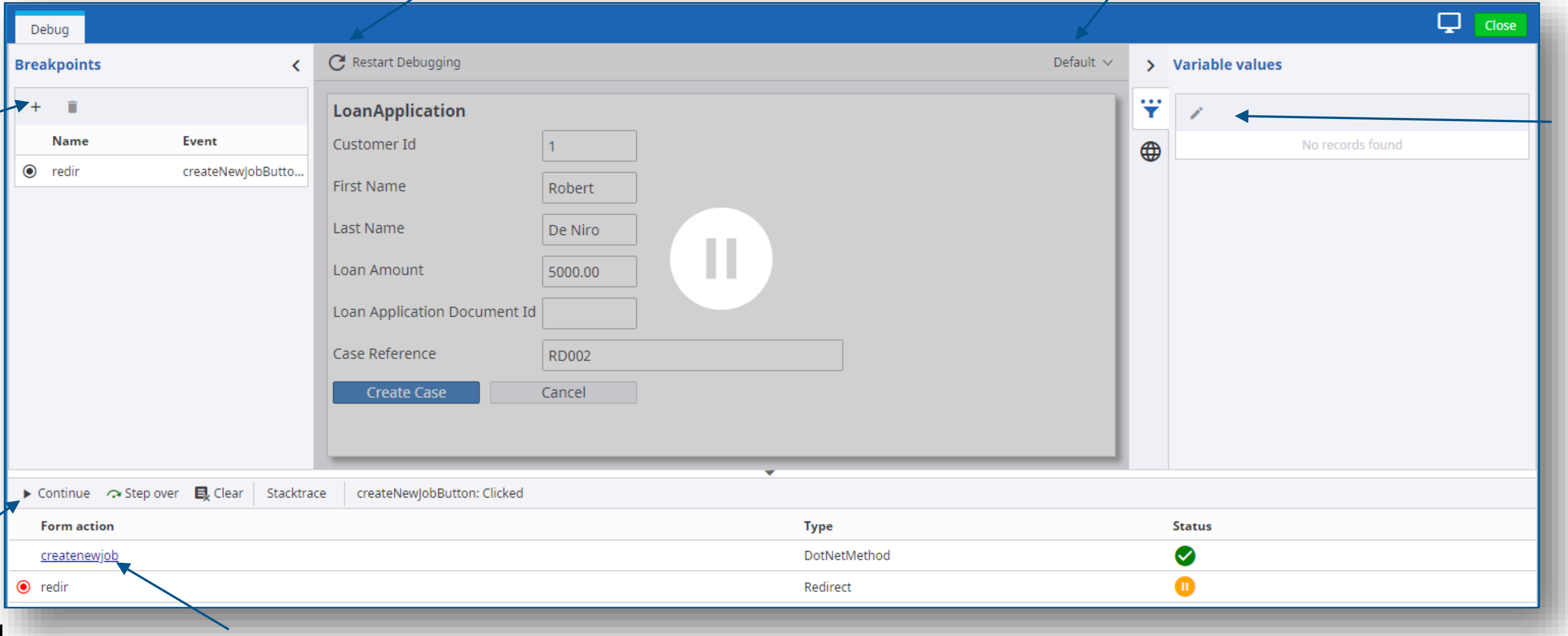
Name	Source	Value
JOB_ID	Global variable	
NODE_ID	Global variable	
EPC	Global variable	

OK

Stop the form load event from executing to enable you to set breakpoints within those actions

Initialisation variables will be listed here and you can update their values, you can also add additional initialisation data

Debugging a Form



Breakpoints

Name	Event
redir	createNewJobButto...

LoanApplication

Customer Id	1
First Name	Robert
Last Name	De Niro
Loan Amount	5000.00
Loan Application Document Id	
Case Reference	RD002

Variable values

No records found

Form action	Type	Status
createnewjob	DotNetMethod	✓
redir	Redirect	⏸

Will reload the form to restart debugging

Form can be viewed in different devices

Switch to other form factors

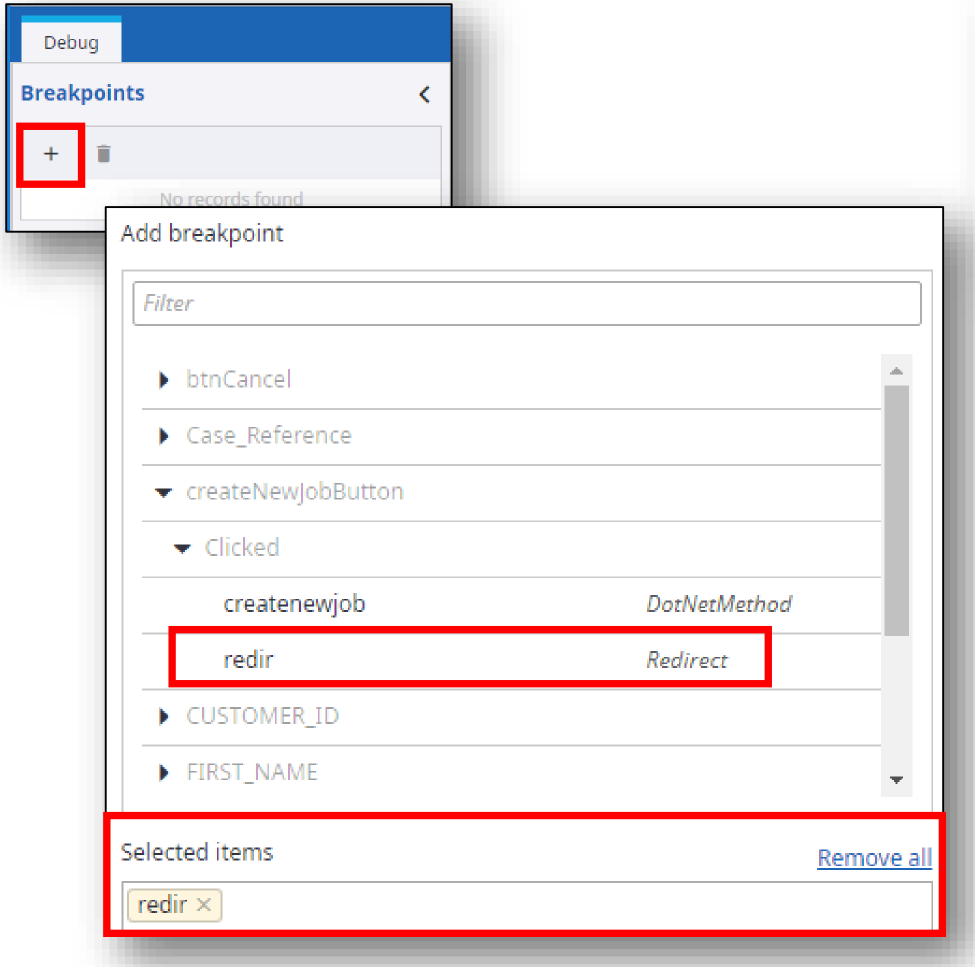
Add breakpoints here

Form and global variable values can be viewed/modified

Actions will be listed in stack trace and debug can be controlled

Server side actions can be expanded to view the inputs/outputs

Add a Breakpoint



Debug

Breakpoints

+

No records found

Add breakpoint

Filter

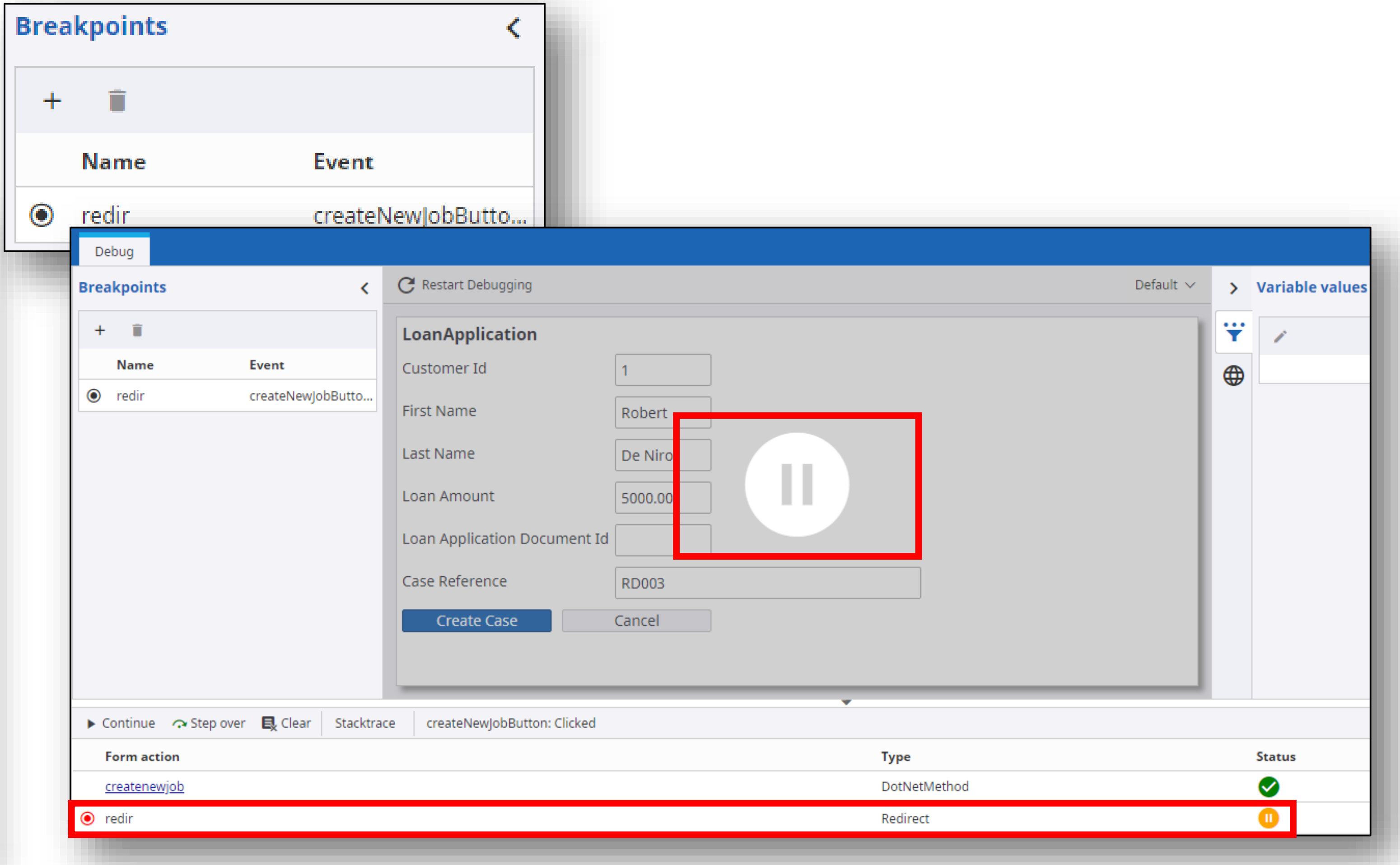
- btnCancel
- Case_Reference
- createNewJobButton
  - Clicked

createnewjob	*DotNetMethod*
redir	*Redirect*

- CUSTOMER_ID
- FIRST_NAME

Selected items — Remove all

redir ×



Breakpoints

	Name	Event
◉	redir	createNewJobButto...

Debug

Breakpoints

	Name	Event
◉	redir	createNewJobButto...

Restart Debugging — Default

Variable values

LoanApplication

Customer Id	1
First Name	Robert
Last Name	De Niro
Loan Amount	5000.00
Loan Application Document Id	
Case Reference	RD003

Create Case — Cancel

Continue — Step over — Clear — Stacktrace — createNewJobButton: Clicked

Form action	Type	Status
createnewjob	DotNetMethod	✓
redir	Redirect	⏸

Points to Note

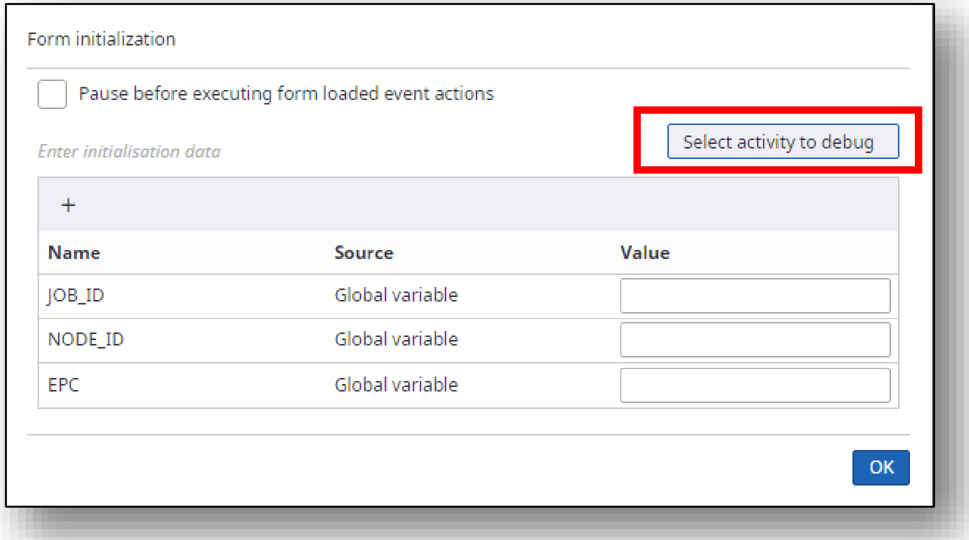
- When a form is launched into debug mode the primary form will be loaded as per the runtime behaviour
- No breakpoints are supported on the before and after render events
- Document and Folder forms are not supported for debug
- Forms must be released in order to debug

Points to Note

- Actions
  - Redirect actions will not actually redirect but when executed will display a message stating where it would have redirected to
  - If a server side action has failed the error message will be displayed on screen as per runtime behaviour and the action should be highlighted in red
  - Actions requiring a session ID will be executed using system session ID therefore form security will not be part of debugging a form
  - Actions are invoked as per runtime e.g. click the button or select a row
  - End Conditions will not be listed in the stack trace
- Viewing variable values - array form variables are not supported

Debugging Activity Forms

- As all actions are executed as per runtime, valid data must be used when debugging
- Activity Form
  - Initialisation values for Job ID, Node ID, and EPC must be provided in a take activity form
  - This can be achieved by progressing a job to the pending activity you wish to debug
  - You can then select this activity in the Debugger

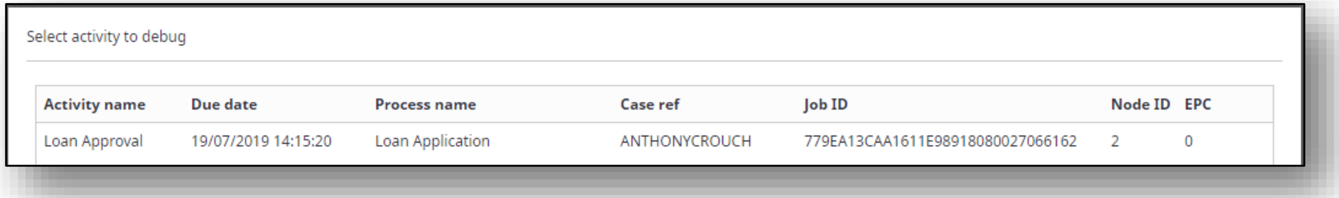


Form initialization

□ Pause before executing form loaded event actions

Enter initialisation data

Name	Source	Value
JOB_ID	Global variable	
NODE_ID	Global variable	
EPC	Global variable	



Select activity to debug

Activity name	Due date	Process name	Case ref	Job ID	Node ID	EPC
Loan Approval	19/07/2019 14:15:20	Loan Application	ANTHONYCROUCH	779EA13CAA1611E98918080027066162	2	0

You may need to add e.g. the Designers group to the Approving Officers group in order to select the runtime activity in the Debugger