X-FLOW CONFIGURATION AND SETUP

Original Software

J MCDOUGALL

JULY 2021

BERTIE 4

TABLE OF CONTENTS

Original Software

SYNOPSIS

The X-flow functionality allows tests to be grouped under a test header entity. The tests have a concept of dependency on previous tests, this means if there is a dependency on a previous test then a test would not become 'Ready to Test' until the previous test had finished.

The X-flow functionality removes much of the manual interaction with tests, once set up tests will remain inactive until X-Flow is started via the test header. Tests will then become 'Ready to Test' and available to execute.

This can be used for end to end tests where the order the tests are completed in a strict order, or tests that are grouped in a meaningful way.

This is configured via the management of the Application definition (model), therefore to make any changes you will need to have a Qualify Admin licence.

Original Software

X-FLOW CONFIGURATION

This functionality will allow you to automate when tests are ready to be executed, this can be based on a dependency where another test or tests need to have been completed and passed before the test changes to a state that can be executed.

When the test becomes ready to be executed, the test will then appear in a testers layout or 'To Do' list.

From a Test or Team lead position, they are then able to create work and assign it to testers. When the group of tests is planned to be started, the test header record simply starts the X-Flow process.

APPLICATION DEFINITION CONFIGURATION

To set up the X-Flow functionality, you will need to log into the system as the administrator or a user that has access to edit the Application Definitions. There is potential to damage and corrupt the Application Definition (Model) in a way that may not be recoverable. Before editing the Application Definition ensure you have a full backup of the SQL databases.

If you are unsure please contact Original Software support for assistance.

1. X-Flow has the concept of a Test Header and Executable Tests (The piece of work allocated to a tester). The relationship needs to be that the Test Header is a direct parent of the Executable Tests. The following diagram shows the simplest imple-mentation of an X-Flowed system.
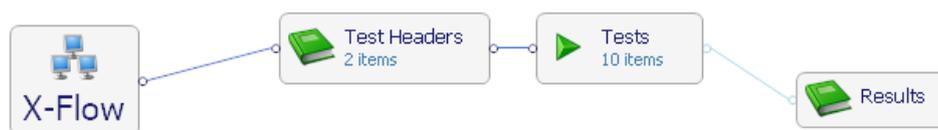
2. The Test Header requires the following additional properties as a minimum:

    a. A boolean value is required to be True to switch on the functionality, you can elect to have the default as True or False.

    | Type: | Boolean |
    | Name: | X-Flow ? |
    | Alignment: | Default |

    ☐ Required ☐ Use as Caption ☐ Card Caption [
    ☐ Inc in Comments ☐ Use for Planning ☐ Card Colour [
    ☐ Id for Import ☐ Use for De-duplication ☐ Import Can Extend [

    Details | Help Text

    Default Value: ☑

3. The executable test entity (Test in this example) requires to have the following additional properties.

    a. A boolean value is required to mark the test as dependent on the previous test has completed, this can be set to either true or false.

    | Type: | Boolean |
    | Name: | Dependent ? |
    | Alignment: | Default |

    ☐ Required ☐ Use as Caption ☐ Card Caption
    ☐ Inc in Comments ☐ Use for Planning ☐ Card Colour
   ☐ Id for Import ☐ Use for De-duplication ☐ Import Can Extend

    Details | Help Text

    Default Value: ☐

Original Software

b. A Single Value Select will need to be set as a required field. This will contain the status of the test.

Type:	Single Value Select		
Name:	Status		
Alignment:	Default		

✓ Required ☐ Use as Caption ☐ Card Caption ☐ Read Only
☐ Inc in Comments ☐ Use for Planning ☐ Card Colour ☐ Primary Search
☐ Id for Import ☐ Use for De-duplication ☐ Import Can Extend ☐ Agile Board

Details | Help Text

Pick List:	Test Status
Default Value:	Not Ready

i. The picklist needs to contain the following as a minimum, this can be extended and we discuss this in later chapters.

Not Ready
Ready to be Tested
Testing in Progress
Passed
Failed
Aborted
Cancelled
X-Flow Error

c. A Multi-Line Text field. This will be used to pass communication from one test to the next.

Type:	Multi Line Text	
Name:	Handover Note	
Alignment:	Default	

☐ Required ☐ Use as Caption ☐ Card Caption
☐ Inc in Comments ☐ Use for Planning ☐ Card Colour
☐ Id for Import ☐ Use for De-duplication ☐ Import Can Extend

Details | Help Text

| Maximum Length:	500
Default Value:	

4. For the executable entity (Test) we will need to create a workflow

   a. From the Test entity, on the setting page check both 'Support WorkFlow' and Workflow compulsory
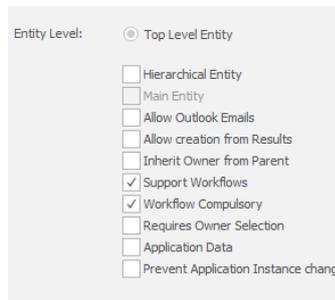
   b. From the top ribbon, menu select Workflows
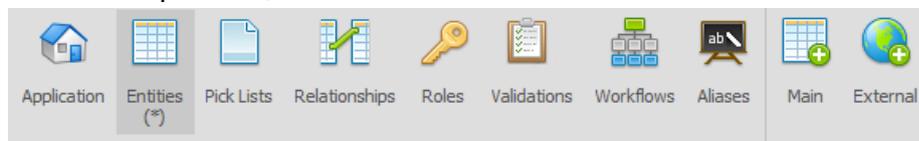
   c. From the Workflow screen, create a new Workflow. On the settings page selects the 'Entity' as the test entiry and the controlling Property as the picklist that was created.
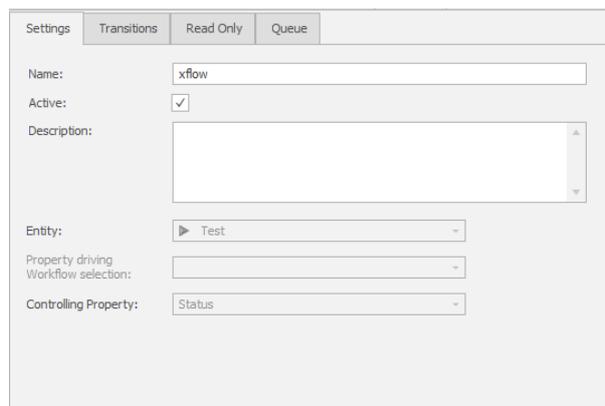
   d. From the 'Transitions' screen, you can plot the normal transitions. However, the following are required:

      i. From the initial status, we have to be able to progress to an 'X-Flow Error' status. This is to ensure that if permissions or configuration stop the status change, we are then notified of an issue.

      ii. From <any> status we need to be able to progress to 'Aborted'.' cancelled' or x-flow Error.

iii. This is a typical basic X-Flow workflow transition setup.

From	Not Ready	Ready to b...	Testing in P...	Passed	Failed	Aborted	Cancelled	X-Flow Error
<New>	⊘							
Not Ready		⊘						⊘
Ready to be Tested			⊘					
Testing in Progress				⊘				
Passed					⊘			
Failed		⊘						
Aborted								
Cancelled								
X-Flow Error								
<Any>						⊘	⊘	⊘

5. From the Test Header entry, we will need to finalise the setup. This is accessed from the X-Flow tab on the entity.

a. Support x-Flows

i. This is required to be checked to activate the functionality

b. Within X-Flow Property:

i. This is the boolean value on the Test Header, when this is checked on a record, the 'Test Header' and Child 'Tests' will act with the X-Flow functionality.

c. Child Entiry

i. This is the entiry that is used for the executable tests, 'Test' in this instance.

d. Handover Property

i. This is the property on the 'Test' that users will use to pass on notes to the subsequent tests. In this example, we called in 'Handover Note'.

e. Workflow

  i. This is the Workflow we created, although an entity can have multiple Workflows, in this example, we have a single workflow.

f. Initial Status

  i. This is the status that we require the Test entity to be before we start the X-Flow process.

g. Transition to Start

  i. This is the status we would like the Test entiry to change to when it's ready to be executed.

h. Transition to Error

  i. This is the status we would like the Test entity to change too if permissions or other configurations prevent the test from becoming active.

i. Transition to Abort

  i. This is the status we would like the Test entity to change too when it's aborted

j. Transition to Cancel

  i. This is the status we would like the Test entity to change to cancel the test.

k. Completion – Passed

  i. These are the status in the picklist where we want to mark the test as Passed

l. Completion – Failed

  i. These are the status in the picklist where we want to mark the test as Failed. Please note with this configuration the next test would become active.

ADVANCED CONFIGURATION

The following configurations are listed as examples of what can be achieved with X-Flow.

### Limiting the X-Flow Error and Cancel options

It is often desired that testers are not given the option to be able to set the status to 'Cancelled' or 'X-Flow Error'
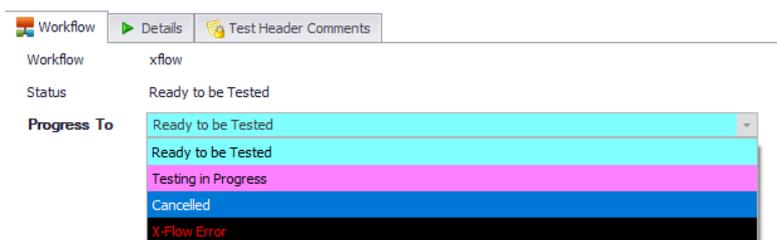
We can limit this by editing our transitions within the Workflow, it is suggested that these are restricted to a 'Team Lead' type role.

To restrict this edit the Application Definition and edit the Workflow that was created for the Test entry.

From the 'Transitions' screen, right mouse click on the transition you would like to restrict, and select properties. From the 'Restrict To' tab, select Roles and select the Test Lead (or similar) role.
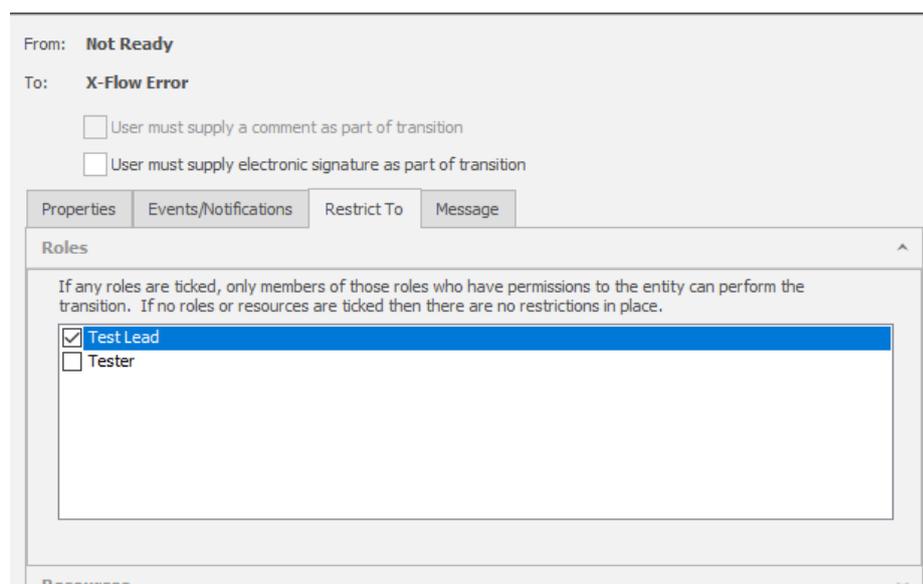
From our example, this would need performing on the following transitions.

1. 'Not Ready' to 'X-Flow Error'

Original Software

2. 'Any' to 'Aborted'

3. 'Any' to 'Cancelled'

4. 'Any' to 'X-Flow Error

The tester would then see only the status that is relevant to the process flow.
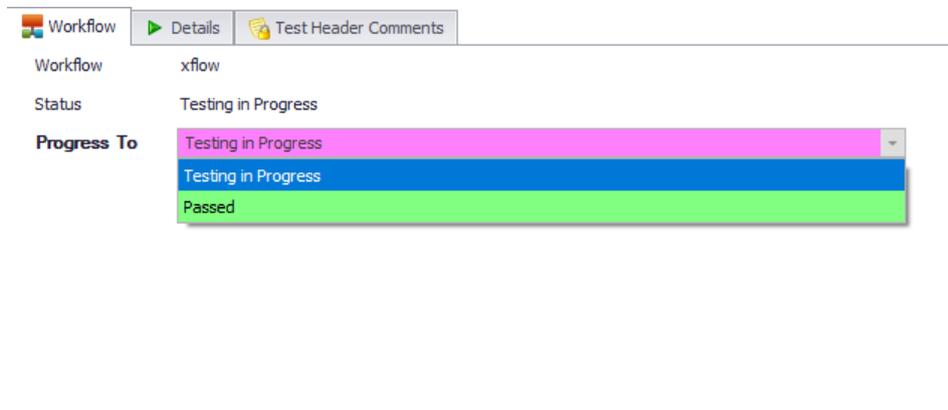
Preventing Sequential Tests starting after a failed test

With the basic set-up, if a user fails a Test, the next Test will become 'Ready' and executable immediately. This means the Test could appear in a different users work queue before handover notes have been completed, or it may not be possible to perform the test. For example, if the previous test created a new customer, we would be unable to invoice that customer.

In this example, I have created additional status's that will prevent this from occurring.

1. Instead of having just the one 'Failed' status, we need to have three.

    a. Failed Creating Defect: This is set when the test fails, using the automatic status settings, gives the tester time to create the defect and decide if testing can continue.

    b. Failed Continue: This works as the previous 'Failed' status, the next test will become active.

    c. Failed Stop: This will not make the test active.

2. In the workflow, we need to create a transition between the 'Testing in Progress' and 'Failed creating defect', we would also then need to create a transition between 'Failed Creating Defect' and both 'Failed Continue and 'Failes Stop'.

Original Software

3. On the X-Flow tab of the 'Test Header' Entity, we will need to the 'Completion-Failed' to be 'Failed Creating Defect' and 'Failed Continue'