

How to...

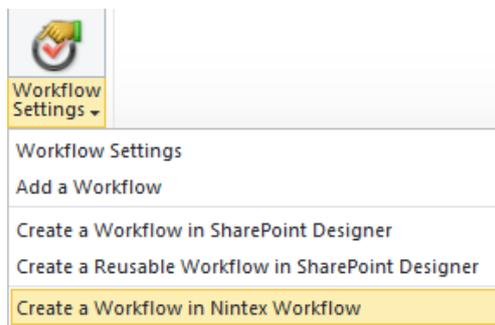
How to design a workflow that cycles through all items within a list and starts a child workflow

Background

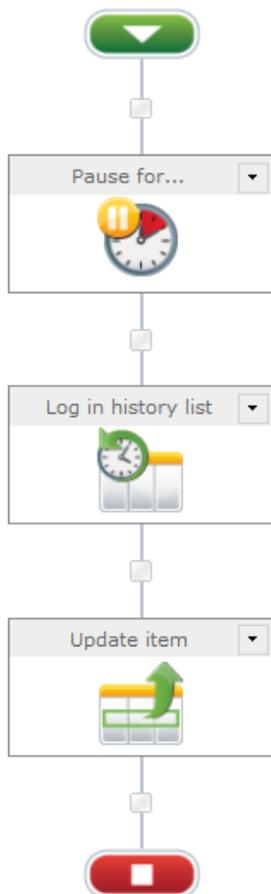
In this example, you will create a parent workflow that queries a List, filtering on a specific SharePoint column, and then determines whether or not a child workflow should be started.

Creating the Child Workflow

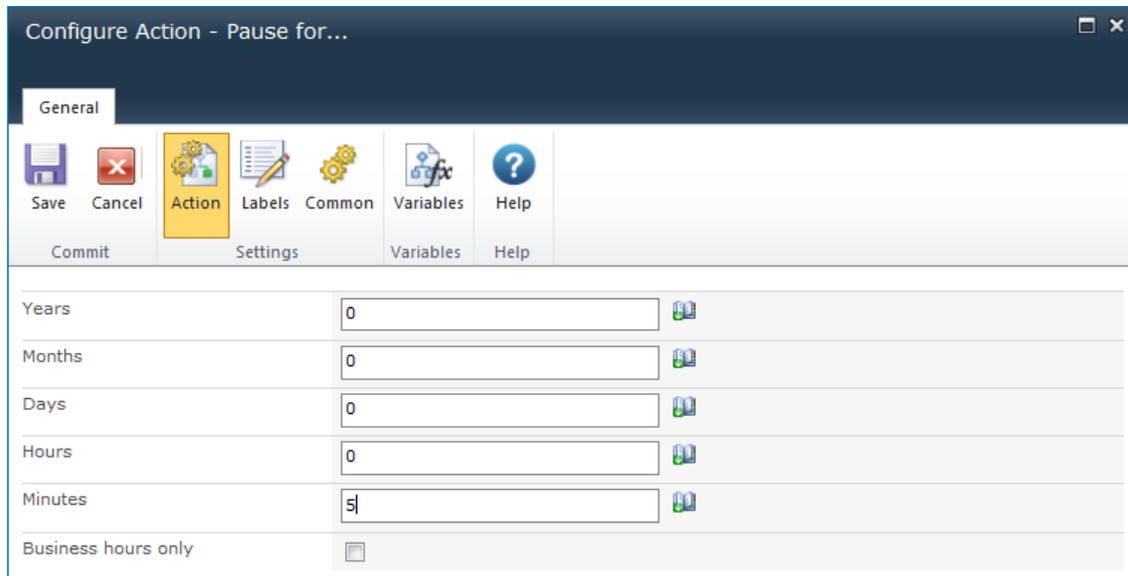
1. Create a SharePoint list titled **wfCycle**.
2. Create a Yes/No column titled **Run**.
3. Select the **Workflow Settings** toolbar button then the **Create a Workflow in Nintex Workflow** drop-down selection.



4. Next, add a **Pause for**, **Log in history list**, and **Update item** action to the workflow designer. The finished workflow will look similar to the following design.

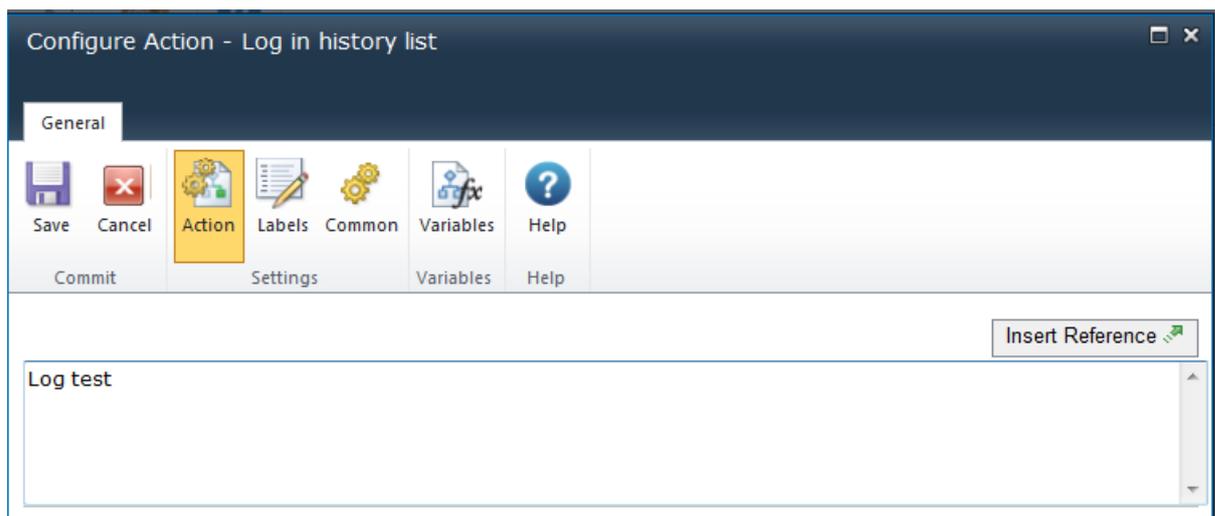


5. Configure the **Pause for** action so that it's set to 5 minutes.

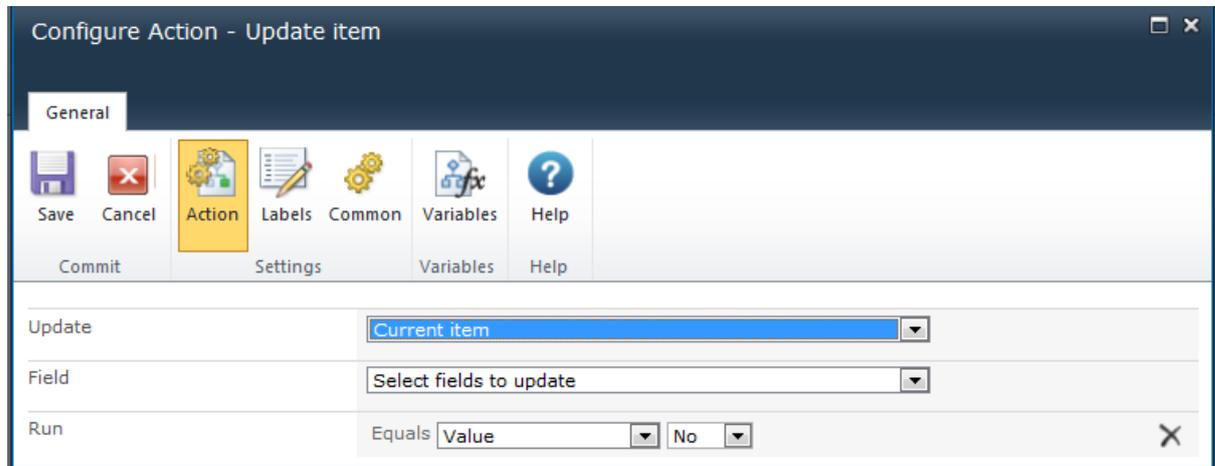


This delay will allow previous actions in the parent workflow to 'commit' prior to the child workflow kicking off. See <http://connect.nintex.com/forums/post/7827.aspx> for details. (We recommend setting this to 5 minutes to begin with, though you may be able to reduce this time by a few minutes.)

6. Configure the **Log in history list** action with some data as shown below. (This action was placed here as an example though you may want the child workflow to do something else here.)



- Configure the **Update item** action to modify the Run column to **No**.

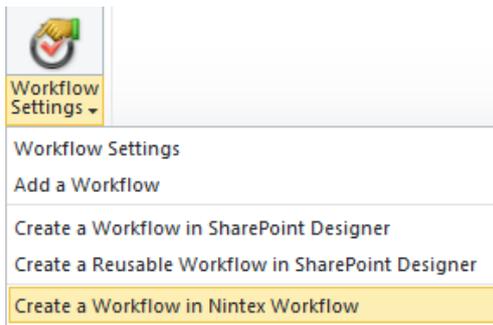


- Publish the workflow as **wfCycle_log**.

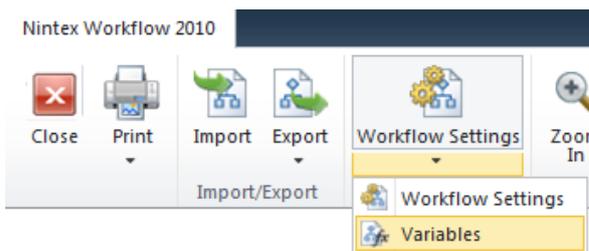
Creating the Parent Workflow

Once the above steps are completed, you can now create your parent workflow.

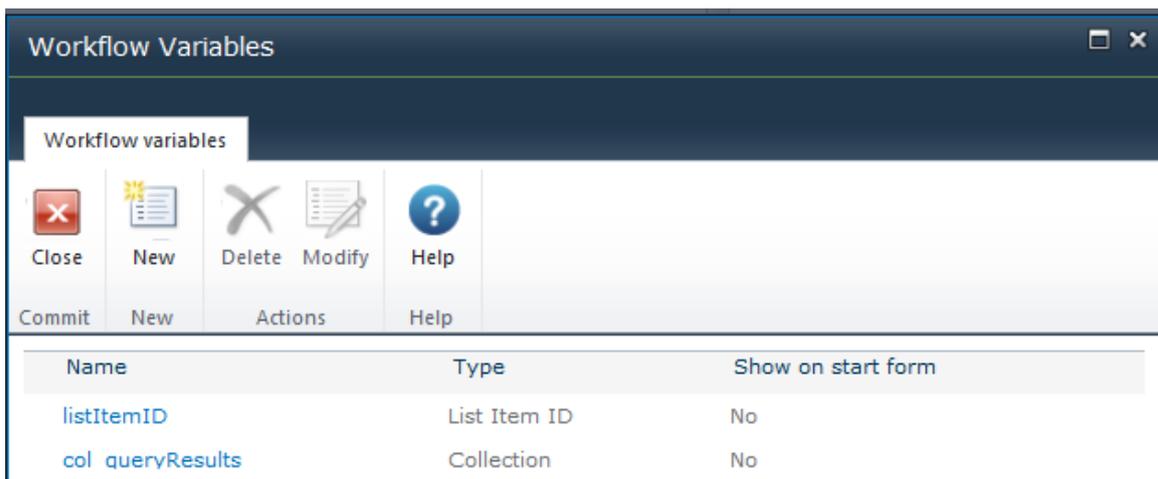
1. Select the **Workflow Settings** toolbar button then the **Create a Workflow in Nintex Workflow** drop-down selection.



2. Within the Workflow Designer, select the **Workflow Settings** drop-down then **Variables**.



3. Select **New**, and create a List Item ID and a collection variable as shown below.



- Next, add a **Query list**, **For each**, and **Call web service** action to the workflow designer. The finished workflow will look similar to the following design.



- Configure the **Query list** action to query the **wfCycle** list where **Run** is equal to **1** (which is equal to **Yes**) storing the ID field in the **col_queryResults** collection variable as shown below.

Configure Action - Query list □ ×

General

Save
 Cancel
Action
 Labels
 Common
 Variables
 Run Now
 Help

Commit
Settings
Variables
Run Now
Help

Editor mode
 Query builder CAML editor

List * wfCycle ▾

Recursive

Filter

Select all list items
 Select items only when the following is true:

Show the items when column

Run
▾

is equal to
▾

1

+ Add filter rule

Sort

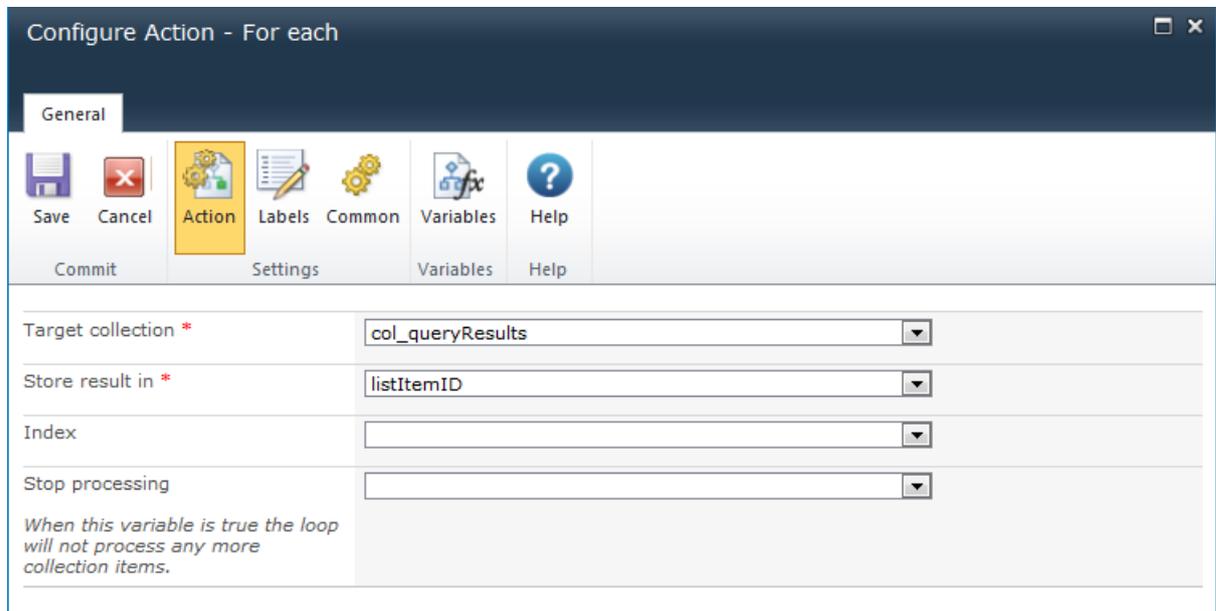
XML encode inserted tokens

Output options
 Include HTML formatting in rich text columns
 Specify item limit

Field
Please select ▾ Add

ID
col_queryResults ▾ ×

- Configure the **For each** action which iterates through the **col_queryResults** collection variable, storing each value in the **listItemID** List Item ID variable which is then used in the Web service call.



Configure Action - For each

General

Save Cancel Action Labels Common Variables Help

Commit Settings Variables Help

Target collection * col_queryResults

Store result in * listItemID

Index

Stop processing

When this variable is true the loop will not process any more collection items.

- Next, configure the **Call web service** action using the **StartWorkflowOnListItem** Web method and setting the Web service input fields as shown below.

The screenshot shows the 'Configure Action - Call web service' dialog box. The 'General' tab is selected. The 'Action' button is highlighted. The configuration fields are as follows:

- URL *: `Web URL/_vti_bin/NintexWorkflow/workflow.asmx`
- Username: `ntxsupport\Administrator`
- Password: `.....`
- Web method: `StartWorkflowOnListItem`
- Editor mode: SOAP builder SOAP editor
- Web service input:
 - itemId (int): `listItemID`
 - listName (string): `wfCycle`
 - workflowName (string): `wfCycle_log`
 - associationData (string):
- Encode inserted tokens

Note the following Web service input field descriptions.

itemId (int) = The listItemID workflow variable you set up above

listName (string) = The name of the List where the workflow resides

workflowName (string) = The name of the workflow that you would like to start

associationData (string) = This can be left blank

- Publish the workflow as **wfCycle_parent**.

That's it! The Web service will auto-start the workflow **wfCycle_log** if the SharePoint column **Run** is set to **Yes**.

Please see our *How to start a workflow using a web service* guide at http://connect.nintex.com/files/folders/tutorials_and_how_tos_nw2010/entry16833.aspx for additional details regarding the **Call web service** action and the above parameters.

Also, the above workflow actions are all described in our Nintex Workflow 2010 Online Help files located at <http://nintexdownload.com/helpfiles/nw2010/NintexWorkflow2010UserManual.html>.