Tutorial: How to use the Collection variable

Table of Contents

Introduction

This short tutorial demonstrates the use of the collection variable by creating two workflows that use a collection variable.

The Collection Variable

The collection variable is a collection of values stored in a single workflow variable. This variable type can be used to store any other type of variable data. If you add a number variable, it will contain a collection of numbers, or if the collection is built using a regular expression, it will store a collection of text values. However, a collection can only contain one type of variable data.

There are two workflow actions designed to work with a collection variable.

- The **For Each** loop action will iterate through each value in the collection and store it to another variable so the value is accessible by other workflow actions. Use the **For Each** action when each item in the collection must be processed.
- The **Collection Operation** action is used to perform other operations on a collection, such as adding or removing individual items, counting the number of items in a collection and retrieving an item at a specified position.

Create a SharePoint List

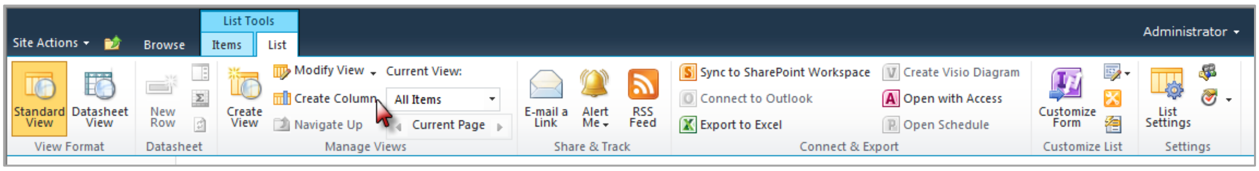
Let's create a Tasks list for this tutorial.

1. In the Quick Launch menu, select **Lists** and click on **Create.**
2. In the **Filter by** section, select **List** and click on **Tasks.**
3. Enter a name for the list and click **Create.**
4. Finally create an item within the list. Enter any values you like.

Create New Columns

The first step is to create columns for our SharePoint Task list.

1. In the Nintex Workflow 2010 Ribbon, click on **Create Column**.

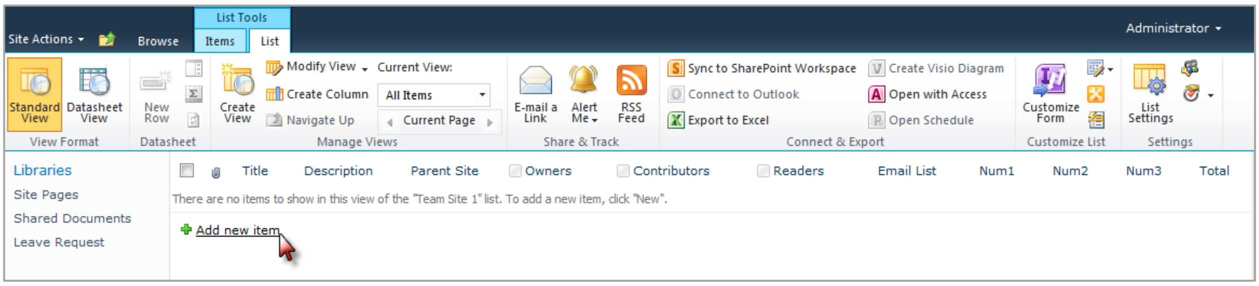


2. In the **Column name,** type "Email List" and select **Single line of text** as the type of information.
3. Repeat steps 1 and 2 for the following information:
   a) Num 1
      i. In the **Column name** field, type "Num1".
      ii. Select **Number** as the type of information.
   b) Num 2
      i. In the **Column name** field, type "Num2".
      ii. Select **Number** as the type of information.
   c) Num 3
      i. In the **Column name** field, type "Num3".
      ii. Select **Number** as the type of information.
   d) Total
      i. In the **Column name** field, type "Total".
      ii. Select **Number** as the type of information.

Enter a New Item

For this tutorial, we will enter some sample item data.

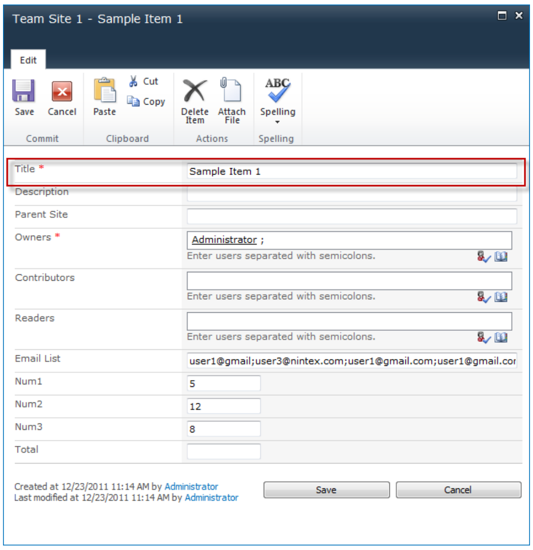
1. Click on **Add new item**.



There are no items to show in this view of the "Team Site 1" list. To add a new item, click "New".

Add new item

2. In the **Title** field, type "Sample Item 1".



Title *	Sample Item 1
Description	
Parent Site	
Owners *	Administrator ; Enter users separated with semicolons.
Contributors	Enter users separated with semicolons.
Readers	Enter users separated with semicolons.
Email List	user1@gmail;user3@nintex.com;user1@gmail.com;user1@gmail.cor
Num1	5
Num2	12
Num3	8
Total	

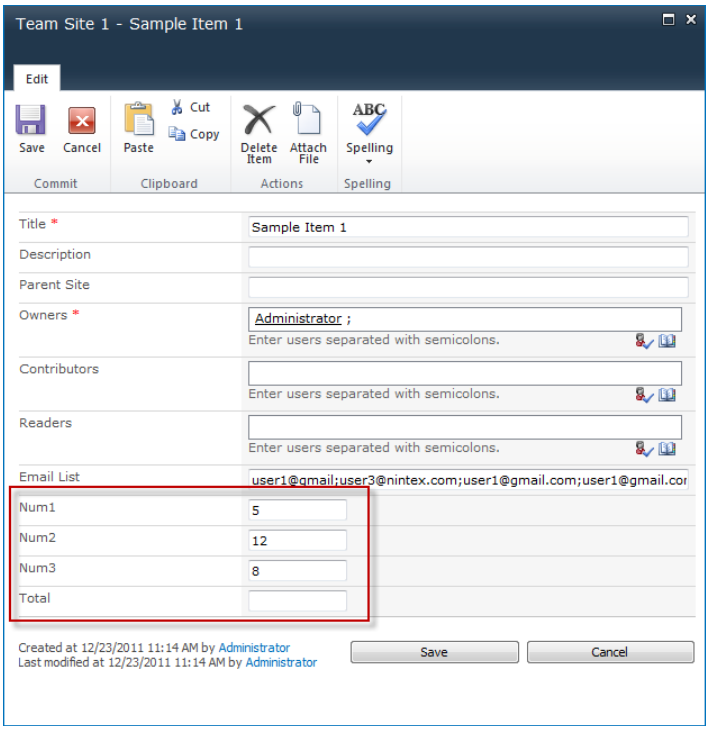
Created at 12/23/2011 11:14 AM by Administrator
Last modified at 12/23/2011 11:14 AM by Administrator

3. In the **Owners** field, add an owner. In this case, we selected **Administrator**.
4. In the **Email List,** enter the following emails**.** Separate each email with a semi-colon.
   a) user1@gmail.com
   b) user3@nintex.com
   c) user1@gmail.com
   d) user1@gmail.com
   e) user2@nintex.com
   f) user2@gmail.com
   g) user3@nintex.com

Note: As you can see, some of the email addresses repeat themselves. This is necessary to demonstrate the use of a collection that will not send a message to duplicate emails.

5. In the **Num1** field, type "5".
6. In the **Num2** field, type "12".

7. In the **Num3** field, type “8”.



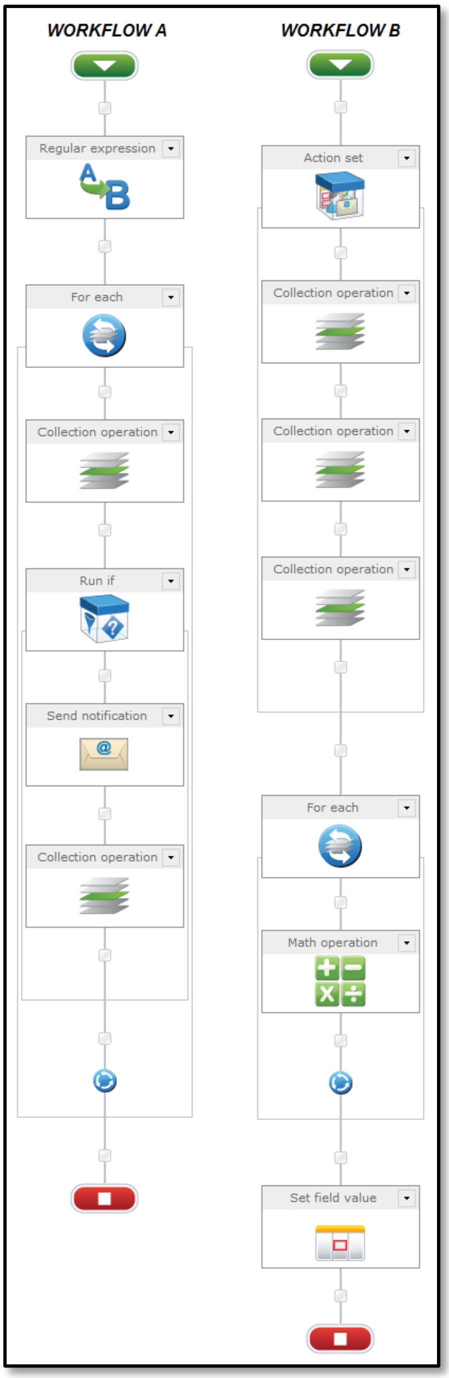
Title *	Sample Item 1
Description	
Parent Site	
Owners *	Administrator ; Enter users separated with semicolons.
Contributors	Enter users separated with semicolons.
Readers	Enter users separated with semicolons.
Email List	user1@gmail;user3@nintex.com;user1@gmail.com;user1@gmail.cor
Num1	5
Num2	12
Num3	8
Total	

Created at 12/23/2011 11:14 AM by Administrator
Last modified at 12/23/2011 11:14 AM by Administrator

8. Click **Save.**

Workflow Design

Now that we have a SharePoint list and columns sorted, we can begin to design the workflows. Two workflows will be created and should look as shown.



Creating the Workflows

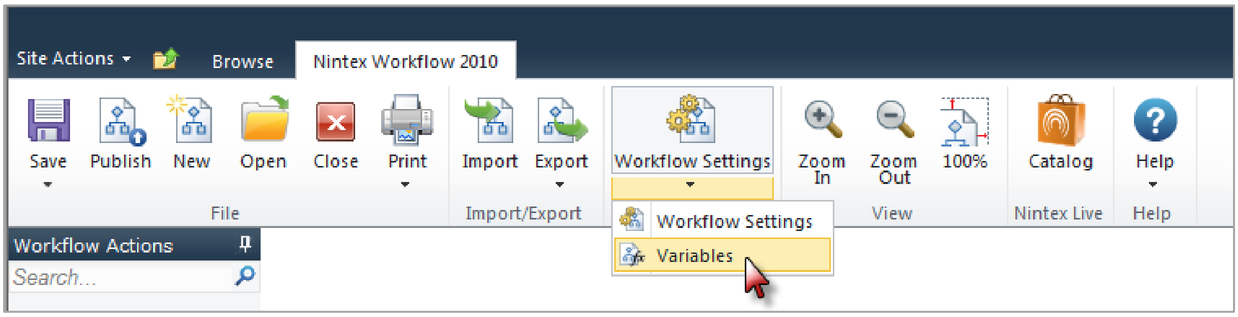
The first step is to create Workflow A. The first workflow will read the email list field and create a collection variable based on each of the emails. Only one email will be sent regardless of how many times the email appears in the list. A new collection will be created which has no duplicate email entries.

Create Workflow A

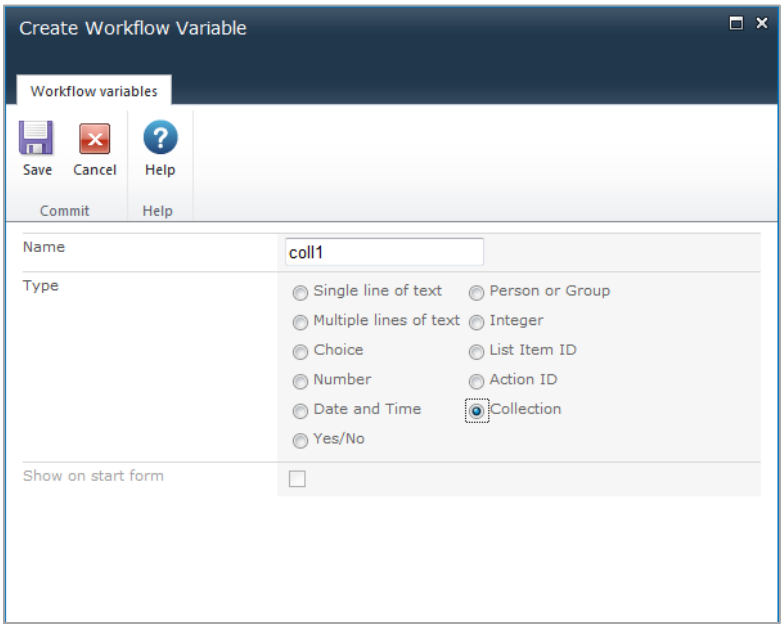
1. In the Ribbon, click on **Workflow Settings** and select **Create a Workflow in Nintex Workflow.**
2. Select the **Blank** template and click **Create.**

Create the Workflow Variables

1. Select the **Workflow Settings** drop-down and click on **Variables.**



2. In the **Workflow Variables** dialog, click **New**.
3. In the **Name** box, type “coll1” and select **Collection** as the **Type.**



Name: coll1

Type:
○ Single line of text ○ Person or Group
○ Multiple lines of text ○ Integer
○ Choice ○ List Item ID
○ Number ○ Action ID
○ Date and Time ● Collection
○ Yes/No

Show on start form □

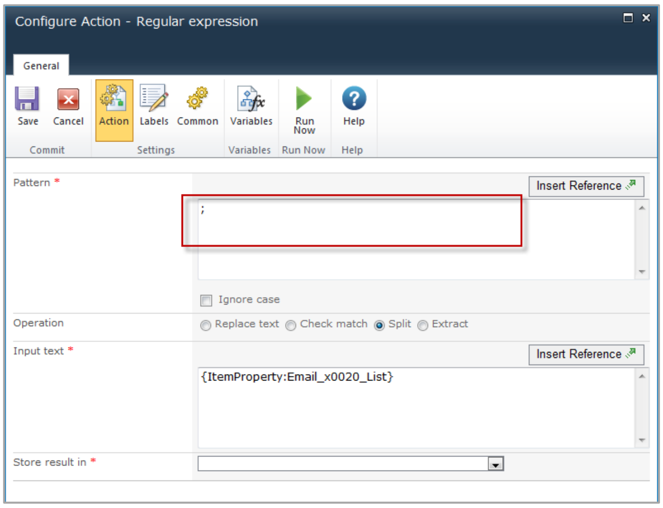
4. Repeat steps 2 and 3 for the following workflow variables.
   a) In the **Name** box, type “coll2” and select **Collection.**
   b) In the **Name** box, type “currcoll1” and select **Single line of text.**

c) In the **Name** box, type “exists” and select **Yes/No.**

Adding the Regular Expression workflow action

The **Regular expression** action will create a collection that consists of the emails in the email list field. Each email address is stored as an item within the collection.

1. Select and place a **Regular Expression** workflow action onto the design canvas.
2. In the **Pattern** box, type a semi-colon (;).


Configure Action - Regular expression

Pattern * ;

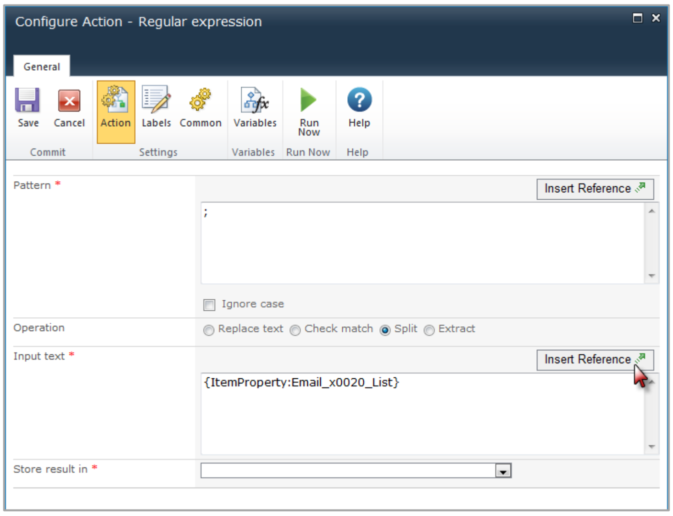
☐ Ignore case

Operation ○ Replace text ○ Check match ● Split ○ Extract

Input text * {ItemProperty:Email_x0020_List}

Store result in *

3. In the **Operation** field, click on **Split**.
4. In the **Input text** field, select **Insert Reference**. Click on the **Item Properties** tab and select **Email List.** Click **OK.**


Configure Action - Regular expression

Pattern * ;

☐ Ignore case

Operation ○ Replace text ○ Check match ● Split ○ Extract

Input text * {ItemProperty:Email_x0020_List}

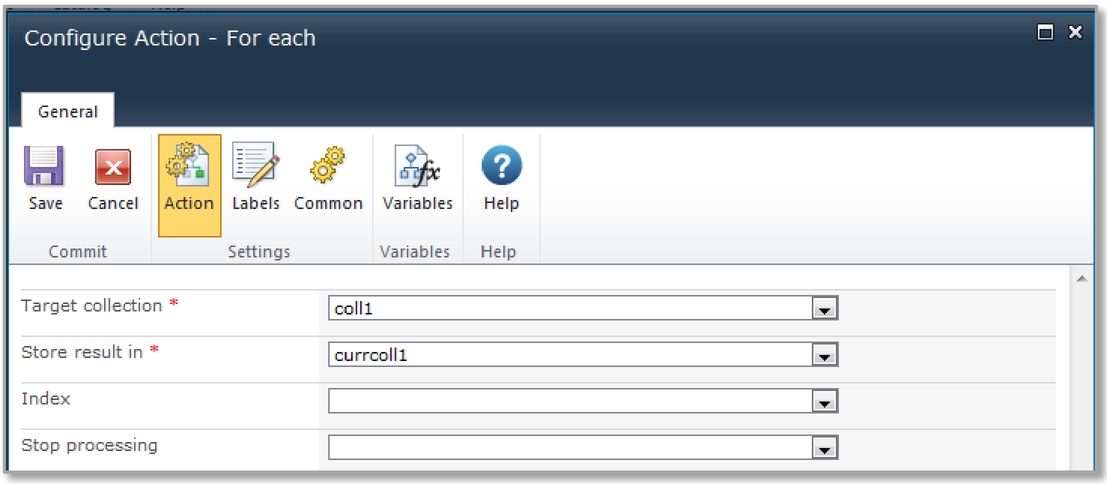
Store result in *

Adding a For Each workflow action

The next step is to add a **For Each** action loop, which will go through each item in the collection and store the current item into the variable **currcoll1.** This can then be used to send an email notification.

Note: The **Index** parameter is optional and can be left blank. If a variable is selected, the position of the current item will be stored into the selected variable.

1. In the **Target collection** field, select **coll1**.
2. In the **Store result in** field, select **currcoll1**. Click **Save.**



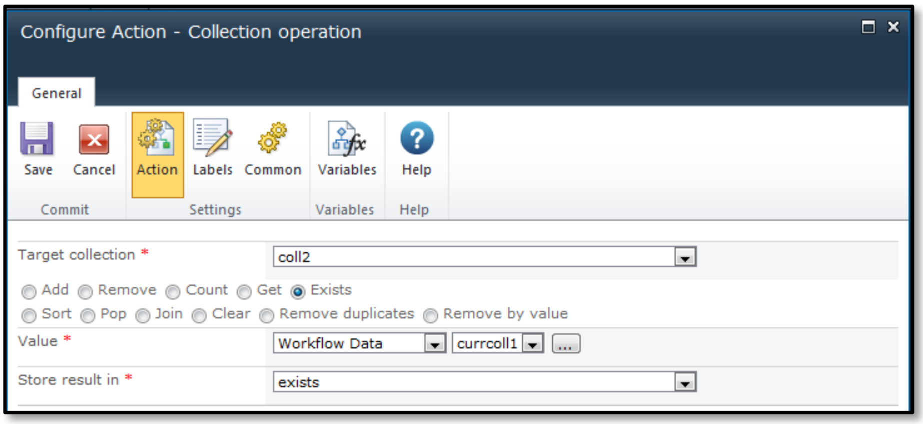
Configure Action - For each

Target collection *	coll1
Store result in *	currcoll1
Index	
Stop processing	

Adding a Collection Operation workflow action to the For Each action

Place a **Collection Operation** action within the For Each workflow action. This will ensure that duplicate email notifications are not sent and will check if the current item has already been added to the second collection. If it has, we can tell it to not add it and not to send a notification email. Configure the Collection Operation as follows.

1. In the **Target Collection** field, select **coll2**.
2. Click on **Exists.**
3. In the **Value** field, select **Workflow Data** and then select **currcoll1** in the drop-down menu next to **Workflow Data.**
4. In the **Store result in** field, select **exists**.



Configure Action - Collection operation

Target collection * coll2

○ Add ○ Remove ○ Count ○ Get ● Exists
○ Sort ○ Pop ○ Join ○ Clear ○ Remove duplicates ○ Remove by value

Value * Workflow Data | currcoll1 | ...

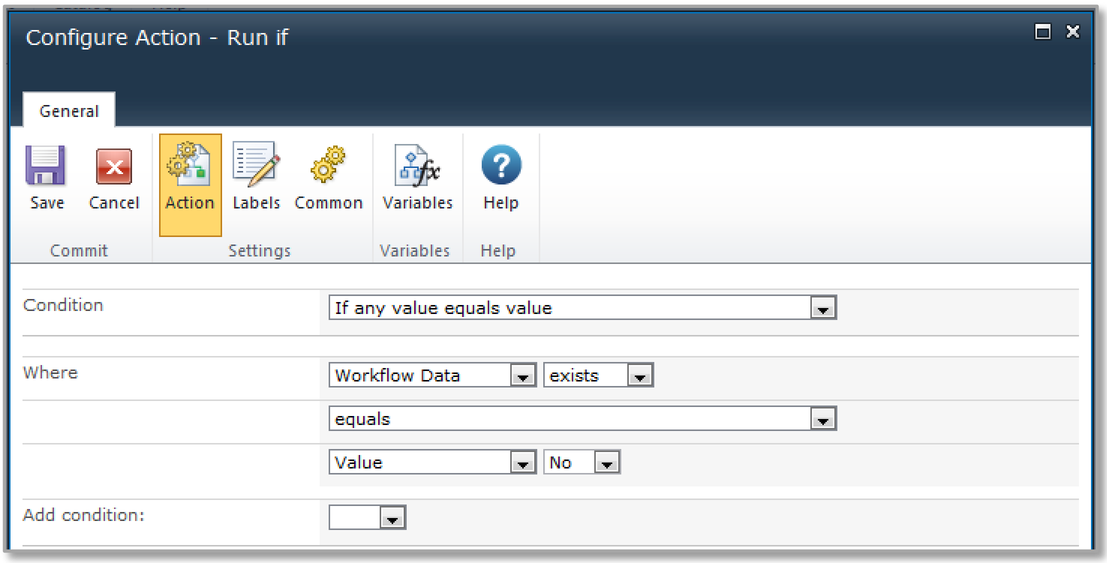
Store result in * exists

Adding a Run If workflow action to the For Each action

A Run if action is used to make sure duplicate emails are not processed. This action needs to be placed within the For Each workflow action. The Run if action will check the result of the collection operation action.

Note: The actions encapsulated in the **Run If** action only execute if the current email is not already in the second collection.

1. In the **Condition** field, select **if any value equals value**.
2. In the **Where** field, select **Workflow Data; exists; Value; No**.



Condition: If any value equals value

Where: Workflow Data | exists

equals

Value | No

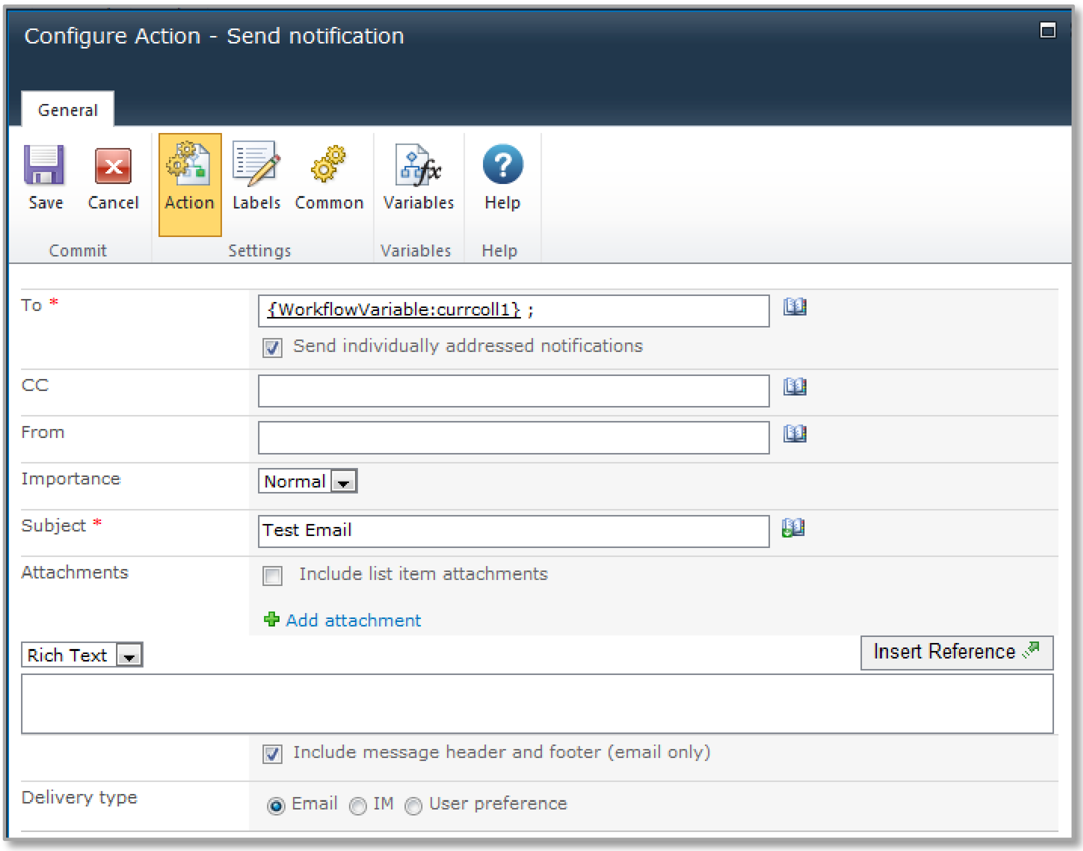
Add condition:

Adding a Send Notification action within the Run If workflow action

The next action will send an email to the current email in the collection. To do this, add a **Send Notification** action within the **Run If** action. Configure the action as show below.

As like all the previous actions, we need to configure the Send notification action so an email is sent.

1. In the **To** field, click on the **Insert Reference** icon.
2. Click on **Lookup** and scroll down to **Workflow Variables.**
3. Select **currcoll1,** click **Add** and then **Ok.**
4. In the **Subject** field, type in the subject of your email. For this example, we will enter "Test email".
5. Click **Save**.



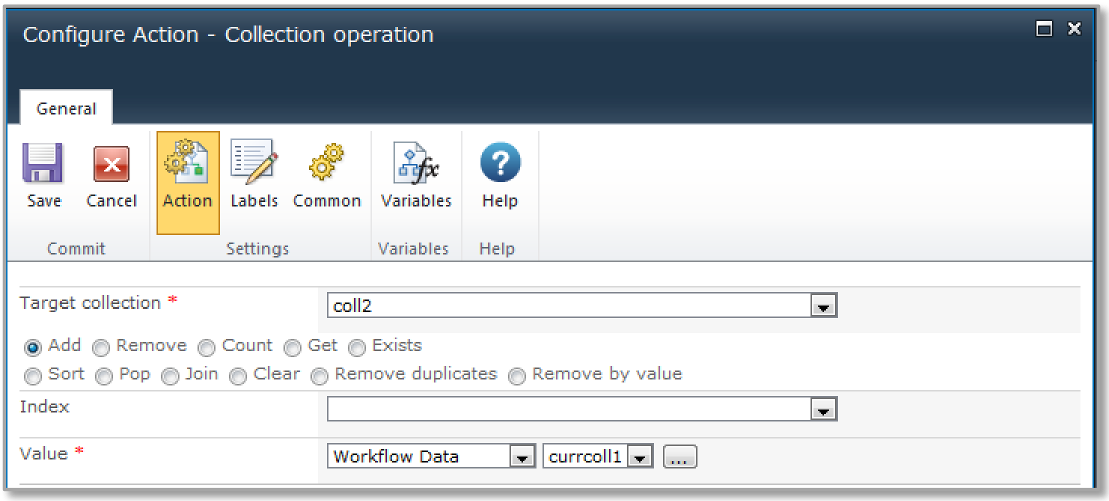
To *	{WorkflowVariable:currcoll1} ; ☑ Send individually addressed notifications
CC	
From	
Importance	Normal
Subject *	Test Email
Attachments	☐ Include list item attachments Add attachment
Rich Text	
	☑ Include message header and footer (email only)
Delivery type	● Email ○ IM ○ User preference

Add a Collection Operation action to the Run If workflow action

The final step for Workflow A is to add any non-duplicate emails to another collection to ensure emails are only sent once. This collection contains a list of all the emails sent. The **Collection Operation** action can do this for us.

1. Select and drag a **Collection Operation** workflow action within the **Run If** action.
2. In the **Target Collection** field, select **coll2** in the drop-down menu.

3. In the **Value** field, select **Workflow Data: currcoll1.**
4. Click **Save.**



Target collection * coll2

● Add ○ Remove ○ Count ○ Get ○ Exists
○ Sort ○ Pop ○ Join ○ Clear ○ Remove duplicates ○ Remove by value

Index

Value * Workflow Data currcoll1 ...

This is the end of Workflow A, now we begin on workflow B.

Creating Workflow B

Workflow B generates a collection of numbers based on the three number fields in the list item: **Num1**, **Num2** and **Num3**. The collection is then cycled through and each item is added to another number variable stored in the list field **Total**. Essentially, we want to add the three fields together using a **Collection Variable** action, **Math operation** action and **For Each** action loop.

To create this workflow, follow the steps outlined below.

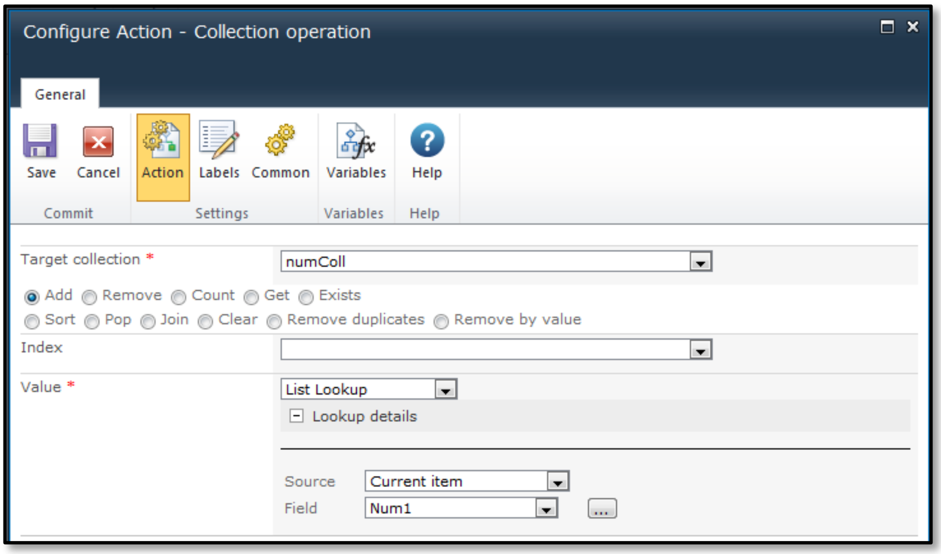
Create the Workflow Variables

1. Select the **Workflow Settings** drop-down and click on **Variables.**
2. In the **Workflow Variables** dialog, click **New.**
3. In the **Name** field, type “currNum” and select **Number** as the type.
4. Repeat steps 2 and 3 for the following information.
   a) **Name**: numColl, **Type:** Collection
   b) **Name:** total, **Type:** Number

Configure the Collection Operation Workflow Actions

We're going to place three **Collection Operation** actions within an **Action Set** workflow action.

1. To begin, select and drag an **Action Set** workflow action onto the design canvas.
2. Now do the same thing for three **Collection Operation** workflow actions and add onto the design canvas.
3. Double-click on the first **Collection Operation** action to open the workflow dialog.
4. In the **Target Collection** field, select **numColl** from the drop-down menu.
5. In the **Value** field, select **List Lookup.**
6. In the **Source** field, select **Current Item** and **Num1.**


Configure Action - Collection operation

Target collection * numColl

● Add ○ Remove ○ Count ○ Get ○ Exists
○ Sort ○ Pop ○ Join ○ Clear ○ Remove duplicates ○ Remove by value

Index

Value * List Lookup

⊟ Lookup details

Source Current item

Field Num1

Collections Operation action - number 2

7. Double -click on the second **Collection Operation** workflow action to open the dialog.
8. Repeat steps 3-4.
9. In the **Source** field, select **Current item** and **Num2.**
10. Click **Save.**

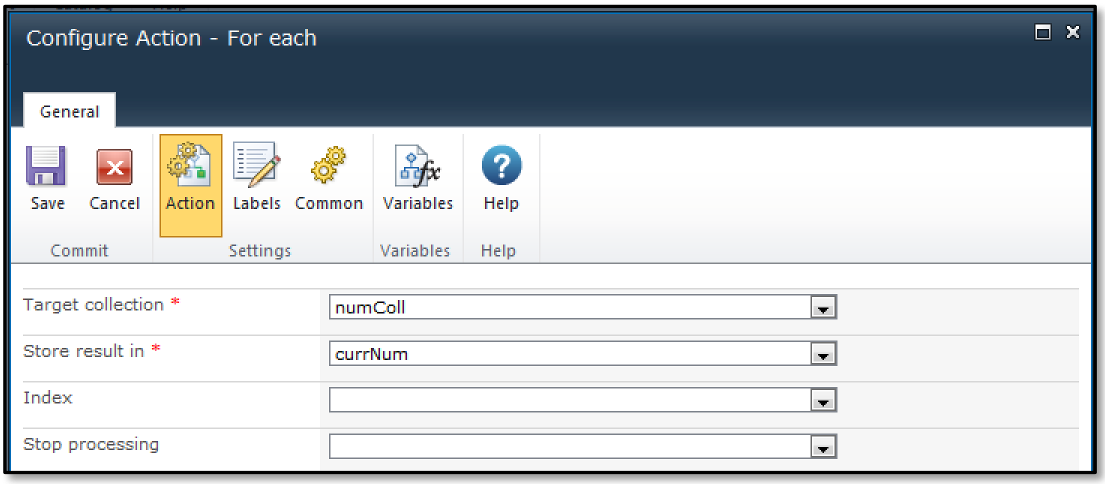
The third and final Collection Operation action

11. Double -click on the third **Collection Operation** action.
12. Repeat steps 3-4.
13. In the **Source** field, select **Current item** and **Num3.**
14. Click **Save.**

Configure a For Each Workflow Action

The next step is to add a **For Each** action loop to iterate though each number in the collection and store the current item into the workflow variable **currNum** created earlier.

1. In the **Target Collection** field, select **numColl** in the drop-down menu.
2. In the **Store result in** field, select **currNum** in the drop-down menu.
3. Click **Save.**

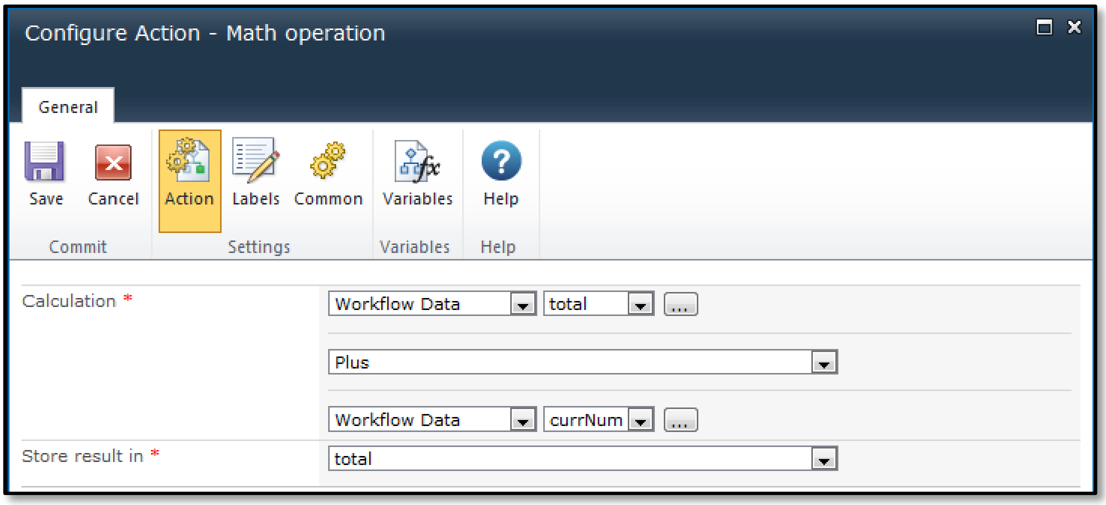


Target collection *	numColl
Store result in *	currNum
Index	
Stop processing	

Configure a Math Operation Workflow Action

Now we can add a **Math Operation** action within the **For Each** action to add the current number to the total and store it in a workflow variable.

1. In the **Calculation** field, select **Workflow Data: Total: Plus: Workflow Data: currNum.**
2. In the **Store result in** field, select **total.**
3. Click **Save.**

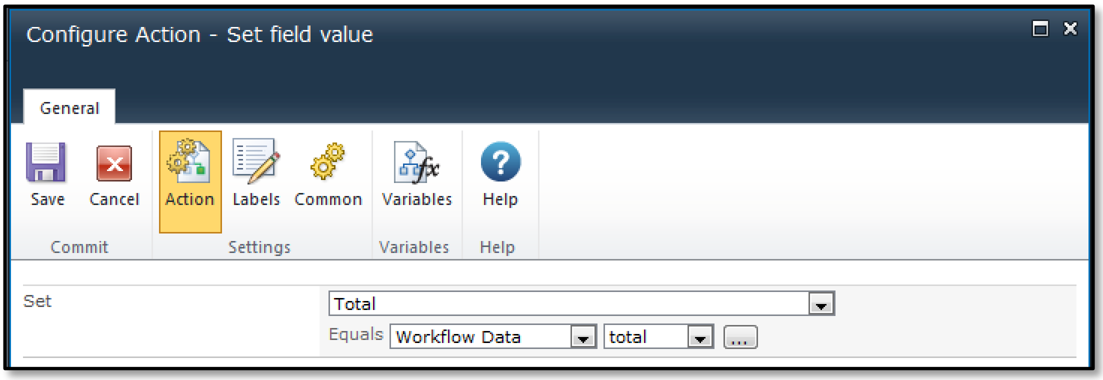


Calculation *	Workflow Data · total ... Plus Workflow Data · currNum ...
Store result in *	total

Configure a Set Field Value workflow action

Last of all, the **Set Field Value** action is used to set the **Total** field in the list item, to the calculated total of all the numbers in the collection.

1. In the **Set** field, select **Total.**
2. In the **Equals** field, select **Workflow Data; total.**



Set: Total

Equals: Workflow Data | total

The workflow is now complete.

This tutorial has demonstrated two workflow actions designed to work with the collection variable type and you can now start using the collection variable for workflows you are creating.