

Tutorial: How to start a workflow with the Call Web Service action



www.nintex.com

support@nintex.com





_

Nintex Workflow 2010: How to start a workflow with the Call Web Service action

Table of Contents

ntroduction	3
Why use a Call web service action instead of a Start Workflow action?	3
The Call web service action	4
Using the StartWorkflow web service method	4
Using the StartWorkflowOnListItem web service method	6
Passing variable values between workflows (using start data)	8
Create a new workflow	8
Add a Log in History List workflow action	9
Add a Build String workflow action	10
Using the association data in the Call Web Service action	11
Run a workflow on all existing items in a list/library	12
Creating the workflow	12
Create two workflow variables	12
Add a Query List Workflow Action	13
Add a For Each workflow action	13
Add a Call web service workflow action	14
Run the workflow	16



www.nintex.com

support@nintex.com





Introduction

In some instances it may be necessary to increase performance by using a Call Web Service action to speak to another workflow. Extremely large workflows can be difficult to edit and may run slower than a smaller workflow designed for a specific process. In these cases, splitting the workflow into separate workflows and having the first workflow call the second and so on can increase performance. This can also be an advantage if there is an error; the workflow can be restarted from a certain point rather than from the beginning.

Why use a Call web service action instead of a Start Workflow action?

The Start Workflow action allows the user to start a workflow on the same list item only. This is useful in certain situations but more often than not starting workflows on different list items is required. Also, if the workflows are part of a larger process, passing variables between the workflows is essential. This can be achieved using the Call web service action and web service methods.

This tutorial will guide you through starting a workflow using the Call Web Service workflow action and the web service methods **StartWorkflow** and **StartWorkflowOnListItem**. Using these methods, workflows can be started on different lists and items.

There will also be examples of how to correctly use start data and the format required.



www.nintex.com

support@nintex.com





The Call web service action

The Call Web Service action allows the workflow to make a call to a SOAP web service method. This action can be used to call any web service. It is commonly used to call the SharePoint web services (<u>http://msdn.microsoft.com/en-us/library/cc752745.aspx</u>) to interact with list items in other sites, or call custom web services that perform processing external to the workflow.

For more detail about the Call web service action please refer to the following tutorial http://connect.nintex.com/files/folders/tutorials and how tos http://

Using the StartWorkflow web service method

This first section will demonstrate how to use the **StartWorkflow** web service within a Call Web Service workflow action. The **StartWorkflow** web service starts a workflow on an item in a document library, where the file URL of the document is known.

Note: This section demonstrates how to configure a Call web service action and is not part of a workflow.

- 1. Open a Call web service workflow action on the design canvas.
- 2. In the URL field, select the Insert Reference icon.
- 3. In the Common tab, select Web URL. Click Ok.
- 4. After the Web URL, type "_vti_bin/NintexWorkflow/Workflow.asmx" to complete the path to the web service.
- 5. Enter the appropriate credentials for the URL.

Configure Action - Call web	service	
General		
Save Cancel Action Labels Co	🔗 🚓 🕜	
Commit Settings	Variables Help	
Settings	valuates melp	
URL *	Web URL/_vti_bin/NintexWorkflow/Workflow.asmx	
	Username [Fam Admin Credential]	
Editor mode	SOAP builder SOAP editor	
Web method	Select 💌 Refresh	
Web service message		
	Encode inserted tokens	
XSL transform		
Store result in		
Result format	⊙ XML ⊚ Text	
+ Error handling		0





6. In the Web method field, click the Refresh button and select the StartWorkflow method.

Configure Action - Call web service					
General					
Save Cancel Commit Settings	Immon Variables Variables Help				
URL *	Web URL/_vti_bin/NintexWorkflow/Workflow.asmx				
	Username [Farm Admin Credential] Assword [Predefined]				
Editor mode	SOAP builder SOAP editor				
Web method	StartWorkflow Refresh				
Web service message					
fileUrl (string)					
workflowName (string)					
associationData (string)					
	Preview SOAP Encode inserted tokens				
Store result in					
Result format	⊙ XML ⊚ Text				
Error handling		0			

- 7. Enter the parameters for the Web service message.
 - **fileURL (string)** The URL of the document.
 - workflowName (string) The name of the workflow to start. This has been published on the target document library.
 - associationData (string) This parameter is used to pass start data to the workflow (values that can be used to set variables to certain values when the workflow starts).

Note: This field may be left blank as there is no start data.

*For an example on how to fill in the Web service message parameters, please see the following.

A test document library was created for this example, and the name of the published workflow is testworkflow01.



To start the workflow on the item shown above, configure the Web service message as:

• fileURL: <u>http://site/library/document.docx</u>





1

dent Software Vendor (ISV

Microsoft Partner

Nintex Workflow 2010: How to start a workflow with the Call Web Service action

workflowName: testworkflow01

Configure Action - Call web service						
General						
Save Cancel Action Labels Cor	Import Import Variables Help					
Commit Settings	Variables Help					
URL *	Web URL/_vti_bin/NintexWorkflow/Workflow.asmx					
	Username [Farm Admin Credential]					
Editor mode	SOAP builder					
Web method	StartWorkflow 🕞 Refresh					
Web service message						
fileUrl (string)	nttp://nspui.nintexsupport.com/testaoc%					
workflowName (string)	testworkflow01					
associationData (string)						

8. Click Save.

Using the StartWorkflowOnListItem web service method

The StartWorkflowOnListItem web service method should be used if a workflow needs to be started on a list item directly or the item URL is not known.

This Call Web Service action can then be added to any other workflow in your SharePoint site to start the selected workflow; in my example I will use a workflow titled **testworkflow01**.

Begin configuring the Call web service action like the previous section and instead select StartWorkflowOnListItem as the web service method, or if you can't remember, I've included the steps below.

- 1. In the URL field, select the Insert Reference icon.
- 2. In the **Common** tab, select **Web URL.** Click **Ok**.
- 3. After the Web URL, type "_vti_bin/NintexWorkflow/Workflow.asmx" to complete the path to the web service.
- 4. Enter the appropriate credentials for the URL entered.



www.nintex.com

L

support@nintex.com



5. In the Web method field, click Refresh and select the StartWorkflowOnListItem method.

Configure Action - Call web service						
General						
Save Cancel	Action Labels Cor	mmon Variables	? Help			
Commit	Settings	Variables	Help			
URL *		Web URL/_vti_l	bin/Ninte>	Workflow/Work	flow.asmx	
		Username [Fa Password [[Pre		Credential]	 	
Editor mode		SOAP builder				
Web method		StartWorkflowOnListItem				
Web service I	message					
itemId (int)						
listName (string	3)					52
workflowName (string)						
associationData	a (string)					<u>60</u>
				W SOAP		
		Encode inse	rted toke	ns		

- 6. Complete the following Web service message parameters.
 - itemID (int) Contains the ID of the item the workflow will start on.
 - listName (string) The name of the list where the workflow will start.
 - workflowName (string) The name of the workflow to start. This is published on the target list or document library.
 - associationData (string) This parameter is used to pass start data to the workflow (values that can be used to set variables to certain values when the workflow starts).
 Note: This field may be left blank as there is no start data.

*For an example on how to fill in the Web Service message parameters, see the following.

The workflow I am using is called Testworkflow01 and the list is in the image below.

NSP Home testdoc library All Documents -								
NSP Home	Shawna	Peter	Vadim Dee	Vadim2	German	French	Site One Workflow	
Podcast Di	rectory	🔲 ID	Тур	e	Name		Modifi	ed
Pictures		1	W)	test I	NEW	5/04/2	2011 11:21 AM
Pictures test								





I want to start a workflow on this item so I would configure the Web service message as:

- itemID: 1
- listName: testdoc library
- workflowName: testworkflow01

Configure Action - Call wel	b service				
General					
Save Cancel Commit Settings	Image: Common Variables Image: Common Variables Variables Help				
URL *	Web URL/_vti_bin/NintexWorkflow/Workflow.asmx				
	Username [Farm Admin Credential] Assword [Predefined]				
Editor mode	SOAP builder SOAP editor SOA				
Web method	StartWorkflowOnListItem				
Web service message					
itemId (int)	1				
listName (string)	testdoc library				
workflowName (string)					
associationData (string)					
	Preview SOAP				
	Encode inserted tokens				

7. Click Save.

Passing variable values between workflows (using start data)

We now know how to configure the Call Web Service workflow action with two different web methods.

Another requirement may be to pass certain workflow variables between workflows when they are part of a larger process.

To do this, we must populate the target workflow with start data. These start data variables can then automatically be populated when the workflow is started.

The first step is to create a simple workflow with start data configured.

Create a new workflow

1. In the Ribbon, select Workflow Settings and then Variables.

www.nintex.com

2. Click New.



support@nintex.com

Microsoft Partner

8 | 16

© 2011 Nintex USA LLC, All rights reserved. Errors and omissions excepted.



- 3. In the Name field, type "Texta" and select Single line of text as the type.
- 4. Check the Show on start form box and click Save.
- 5. Create a second variable called 'Number1' and select Number as the variable type.
- 6. Check the Show on start form box and click Save.



Add a Log in History List workflow action

A **Log** in **History List** workflow action can be added to the workflow to log the values of these variables and test whether they were correctly set.

- 1. Click and drag a Log in History List action to the design canvas.
- 2. Double-click on the action to open the configuration dialog.
- 3. In the text box, type "Text =" and then click on the Insert Reference icon.
- 4. In the Workflow Variables list, select Texta and click Ok.
- 5. In the text box, type "Number=" and click on the Insert Reference icon.
- 6. In the Workflow Variables list, select Number1 and click Ok.

Configure A	ction - Log in history	' list		
Save Cancel	Action Settings	Variables	? Help	
	(flowVariable:texta} VorkflowVariable:Numb(er1}		Insert Reference अ

7. Click Save.





Add a Build String workflow action

To test starting a workflow with these variables, automatically set when the workflow starts either of the web service methods listed in this tutorial can be used.

In this example, the **StartWorkflowOnListItem** method will be used.

To begin, a dynamic string action must be built to create the associationData XML required in the Call Web Service action. We will use this action to create an association data variable, adata.

- 1. Select and drag a **Build String** workflow action onto the design canvas.
- 2. In the **Build String** textbox, enter the following association data. Refer to the text below for format.

The format for the association data parameter is as follows:

<Data> <StartDataItem1>value1</StartDataItem1> <StartDataItem2>value2</StartDataItem2> <StartDataItem3>value3</StartDataItem3> </Data>

The workflow we are attempting to start has two Start Data items texta and Number1.

Therefore, the association data will be as follows:

<Data> <texta>testvalue1</texta> <Number1>12345</Number1> </Data>

This will set the variable 'texta' to testvalue1 and 'number1' to 12345.

- 3. Create the association data variable.
- 4. Click Variables in the Ribbon and select New.
- 5. In the Name, type "adata" and select Single line of text. Click Save.
- 6. In the Store result in field, select adata.



www.nintex.com

support@nintex.com



Configure Action - Build	string	□ ×
General		
Save Cancel Action Labels	Image: Common Variables	
Commit Setting	s Variables Help	
Build string	<data></data>	Insert Reference 🖉
	<texta>testvalue1</texta> <number1>12345</number1> <data></data>	
-		
Parse for tokens twice		
Store result in *	adata	-

7. Click Save.

The two start data variables, texta and number1 have now been combined into the one association data variable.

Using the association data in the Call Web Service action

The association data can now be added to the Call Web Service action.

- 1. In the associationData (string) field, select the Insert Reference icon. In the Workflow Variables section, select adata.
- 2. Check the Encode inserted tokens box.
- 3. Click Save.

General					
Save Cancel Commit	Image: Symplex of the symplex of t				
comme Settings	variables (http://www.aliables.com				
URL *	Web_URL/_vti_bin/NintexWorkflow/Workflow.asmx				
	Username [Farm Admin Credential] Assword [Predefined]				
Editor mode	SOAP builder SOAP editor SOAP editor				
Web method	StartWorkflowOnListItem				
Web service message					
itemId (int)	1				
listName (string)	testdoc library				
workflowName (string)	testworkflow02				
associationData (string)	adata				
	Preview SOAP				
	Encode inserted tokens				





Run a workflow on all existing items in a list/library

There may be a case where a common process workflow will need to be run on all existing items in a list/library, not only a single item. To achieve this, we must first query all the items within a list and then loop through this result. Within the loop, the current item will be passed to the **StartWorkflowOnListItemID** web service method. Use the 'StartWorkflowOnListItemID' method to start a workflow on each item within a list or library. This will in turn start the workflow on that item.

This type of workflow can be executed once on any item within the list and will then trigger the specified workflow to run on all list items.

Creating the workflow

- 1. Select the list or library to run the workflow on.
- 2. In the Ribbon, select the List tab and click Workflow Settings. Click on Create a workflow in Nintex Workflow.
- 3. Select a Blank canvas.

Create two workflow variables

Two variables are required for the example.

ItemIDcollection

A collection variable used to store the item ID collection from the initial query.

CurrentItemID

This is used within the For Each action to store the current ID being referenced within the loop, the value will also be passed to the Call Web Service action.

- 1. Click Workflow Settings in the Ribbon and select Variables. Click on New.
- 2. In the Name type "ItemIDcollection" and select Collection as the type. Click Save.
- 3. Select **New** and in the **Name** field, type "currentItemID". Select **List Item ID** as the type.

Workf	low varia	bles			
×	×	\mathbf{X}	?		
Close	New	Delete Modify	Help		
Commit	New	Actions	Help		
Nam	ne		Тур	e	Show on start form
ItemIDcollection		Collection		No	
currentItemID		List Item ID		No	





Add a Query List Workflow Action

The first step is to query all the items in the current list/library and store the resulting IDs within a collection variable. We can do this using a **Query List** action.

The **Query List** action will be set to return each item ID in the list (regardless of the values they contain) to be used within a For Each loop. The result is stored in a collection variable. This variable allows for more than a single value to be returned.

- 1. In the List field, use the drop-down to select the List to query the items. For this example, we will select testlist1.
- 2. In the **Recursive** field, check the box.
- 3. In Field, select ID and click Add.
- 4. In the drop-down list, select the **ItemIDcollection** variable.

Configure Action - Query list						
General						
Save Cancel A	tion Labels Com	Import Variables				
Commit	Settings	Variables Help				
Editor mode		Query builder O CAML editor				
List *		testlist1				
Field *		ID				
Recursive						
🖻 Filter						
		 Select all list items Select items only when the following is true: 				
+ Sort						
XML encode insert	ted tokens					
Store results in *		ItemIDcollection				

Note: In the **Filter** section, you can select to start the workflow on items which meet a specific criteria; for example, only items with a title field which contains the value 'red'.

Add a For Each workflow action

The **For Each** action will loop through each of the ID values returned, once a collection variable has been populated.

The For Each loop will also contain the call web service action that will be starting the target workflow.





- 1. Select and drag a **For Each** action onto the design canvas.
- 2. In the **Target Collection** field, select the **ItemIDcollection** variable. This is the variable which contains the result of the query.
- 3. In the Store result in field, select the currentItemID.
- 4. Click Save.

	ction - For eac	h						
General								
Save Cancel	Action Labels C	ommon.	Variables	? Help				
Commit	Settings		Variables	Help				
Target collection *		Iten	ItemIDcollection					•
Store result in *		curr	currentItemID					•
Index								•
Stop processing								•
When this variable is true the								

Add a Call web service workflow action

The configuration of the web service action is the same as explained in the previous section "Nintex 'StartWorkflowOnListItem' web service method example" except that the ID will be the 'currentID' variable, which is based on the result of the 'for each' loop.

- 1. Select and drag a **Call web service** action on to the design canvas within the For Each workflow action.
- 2. In the URL field, click on the Insert Reference and select WebURL.
- 3. After WebURL, type "_vti_bin/NintexWorkflow/Workflow.asmx".
- 4. Enter the credentials for the URL.
- 5. In the **Web method** field, select **Refresh.** Select the **StartWorkflowOnListItem** method.



www.nintex.com

support@nintex.com





- 6. Configure the Web service message.
- 7. In the itemID (int) field, click on the Insert Reference icon and select currentItemID.
- 8. In the listName (string) field, click on the Insert Reference icon and select List Name.
- 9. In the **workflowName (string)** field, enter the name of the workflow that be starting on each item.

Configure Action - Call we) service							
General								
Save Cancel Action Labels C	Image: Wariables Image: Wariables							
Commit Settings	Variables Help							
URL *	Web URL/_vti_bin/NintexWorkflow/Workflow.asmx	<u>81</u>						
	Username ntxsupport\administrator							
Editor mode	SOAP builder SOAP editor							
Web method	StartWorkflowOnListItem	Refresh						
Web service message								
itemId (int)	currentItemID	<u>1</u>						
listName (string)	List Name	1						
workflowName (string)	simpleworkflow	<u>9</u>						
associationData (string)		<u>90</u>						
	Preview SOAP	_						
	Encode inserted tokens							
+ XSL transform								
Store result in	•							
Result format	© XML ⊚ Text							
+ Error handling								



www.nintex.com

support@nintex.com 

© 2011 Nintex USA LLC, All rights reserved. Errors and omissions excepted.



The completed workflow will consist of the following design:



Run the workflow

Start the workflow manually on any item within the target list. This will cause the specified workflow to run on each item based on the query result.

