SILVERLIGHT WORKFLOW DESIGNER: USING QUERY STRING PARAMETERS TO INTEGRATE THE DESIGNER WITH OTHER APPLICATIONS

HOW TO HOST AND USE THE SILVERLIGHT-BASED WORKFLOW DESIGNER IN OTHER APPLICATIONS BY LEVERAGING QUERY STRING PARAMETERS. INCLUDES A WALKTHROUGH OF THE CONCEPT BY EXPOSING AND USING THE WORKFLOW DESIGNER IN BOX.

Originally published September 30th, 2015. Updated April 11th, 2016

Contents

INTRODUCTION

Several enhancements to the Silverlight-based Workflow Designer (as of K2 4.6.11 and later) allow you to integrate the designer into other line of business (LOB) systems via query string parameters. These enhancements allow you to design K2 workflows and start workflows based on events and artifacts from LOB systems for which K2 does not ship OOTB integration.

This document is divided into two parts: a description of the available query string parameters, and a walkthrough that demonstrates the concept by integrating the workflow designer into Box and starting a workflow from within Box.

WORKFLOW DESIGNER QUERY STRING PARAMETERS

The base URL to load the workflow designer is as follows (substitute the appropriate servername for your installation of K2):
https://{servername}/Designer/Workflow/WebDesigner.aspx

This URL allows you to call in to the workflow designer and pass in parameters, for example https://k2.denallix.com/Designer/Workflow/WebDesigner.aspx?*categoryid=42020&* workflowname=MyWorkflow. These options, in turn, allow you to instantiate the Silverlight workflow designer with various pre-configured values.

The table below describes the available parameters. Note that you may need to query tables in the K2 database to determine the appropriate values.

Parameter	Required	Description	Example & Notes
categoryid	Yes	Represents the category to which the workflow is deployed. If you only supply this ID, the workflow designer defaults to a SmartForms-based workflow and requires a start form.	*categoryid=42020* To retrieve the ID of a category, query the [Category].[Category] table.
id		The ID of the workflow to load when the workflow designer loads. This is used if you want to edit an existing workflow.	*id=10006* To retrieve the ID of a workflow, look in the [Designer].[ProcessXML] table.
workflowname	Yes	The name of the workflow to create.	*workflowname=MyWorkflow*

Parameter	Required	Description	Example & Notes
objectid		The GUID of the SmartForm used to start the workflow.	*objectid=6bea57cc-d45b-4991-9172-59301eecc2ff* You can determine the form's GUID by looking at the GUID query string parameter when editing the form. Note: If you use this parameter the workflow is created in the same category as the SmartForm, even if you specify a different category using the **categoryid** parameter.
exiturl		The URL for the page the workflow designer redirects to after clicking **Exit** on the **File** menu or, on the Workflow Deployment dialog, the **Close & Exit** button. If you do not specify an **exiturl**, these UI options will not be displayed.	*exiturl=https://k2.denallix.com/runtime/runtime/workflows+deployed/* The parameters **id**, **workflowname** and **startformid**, along with other parameters specified, will be appended to the **exiturl** URL.
namereadonly*		When this parameter is added to the URL and set to **True,** the user cannot rename the workflow from within the workflow designer. The name of the workflow is set by the **WorkflowName** parameter.	*namereadonly=True*
formrequired*		When set to **False**, the user can build a workflow without specifying a Start Form on the **Workflow Settings** page	*formrequired=False* If a Start Form is not specified, the Item Reference and Folio are blank but can be set manually.
datafields*		Adds string-type data fields to the workflow and default values for those datafields if specified.	*datafields=Field1=Value1;Field2;Field3=Value3* Data Field names are separated by a semicolon (;), are always created as String type values, and the default value is optional
templatename*		The name of the template used as a basis for the new workflow.	*templatename=MyTemplate1*

* Parameter is available in K2 4.6.11 and later

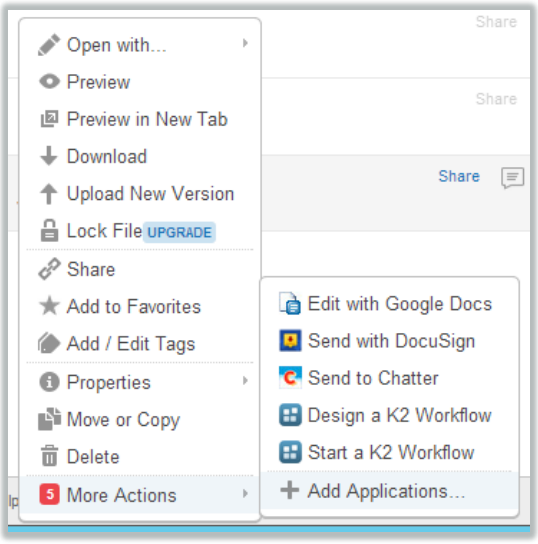
EXAMPLE: INTEGRATING THE SILVERLIGHT WORKFLOW DESINGER WITH BOX

This section describes how to use the query string parameters to create workflow integration with Box.com, the file sharing and collaboration site. While the information in this section is specific to how Box integration works, the principles are the same for any other integration you may want to implement. The main components of this integration include:

1. Creating an App in Box which will expose links from a Box menu, to allow the user to design a K2 workflow using the Silverlight workflow designer or to initiate a new workflow instance. See Creating an App on Box and Starting a K2 Workflow from Box
2. Passing parameters from Box to K2, which can be used to pass context and variables into the workflow designer or when launching a new instance of a workflow. See Callback Configuration and Callback Parameters
3. Creating a SmartBox SmartObject to track the workflows designed in Box, and saving data in the SmartObject for later use in a SmartForm to allow the user to select from a list of workflows. See Tracking the Box Workflows
4. Creating a SmartForm that lists the Box-based workflows that can be started from Box, which will be shown to the user when they start a new workflow instance from a Box item. See Tracking the Box Workflows

Note: For components 3 and 4, a K2 deployment package containing the SmartObjects, Views and Forms is provided in this document. See K2 Package.

Box allows you to create applications to extend the functionality of the Box site. In this example we will create an application that adds a **Design a K2 Workflow** option and a **Start a K2 Workflow** option to the right-click menu of files and folders in Box:



Selecting **Design a K2 Workflow** launches the K2 Designer to start building a new workflow definition. Selecting **Start a K2 Workflow** will show a SmartForm where the user can select from a list of available workflows and then starts an instance of the selected workflow, passing in context of the original item that was right-clicked.

These menu options are both Web App Integrations for the same Box App, in this case called **RESTtestONE**:

CREATING AN APP ON BOX

To create an App on Box, login to Box and browse to the following URL: https://app.box.com/developers/services/

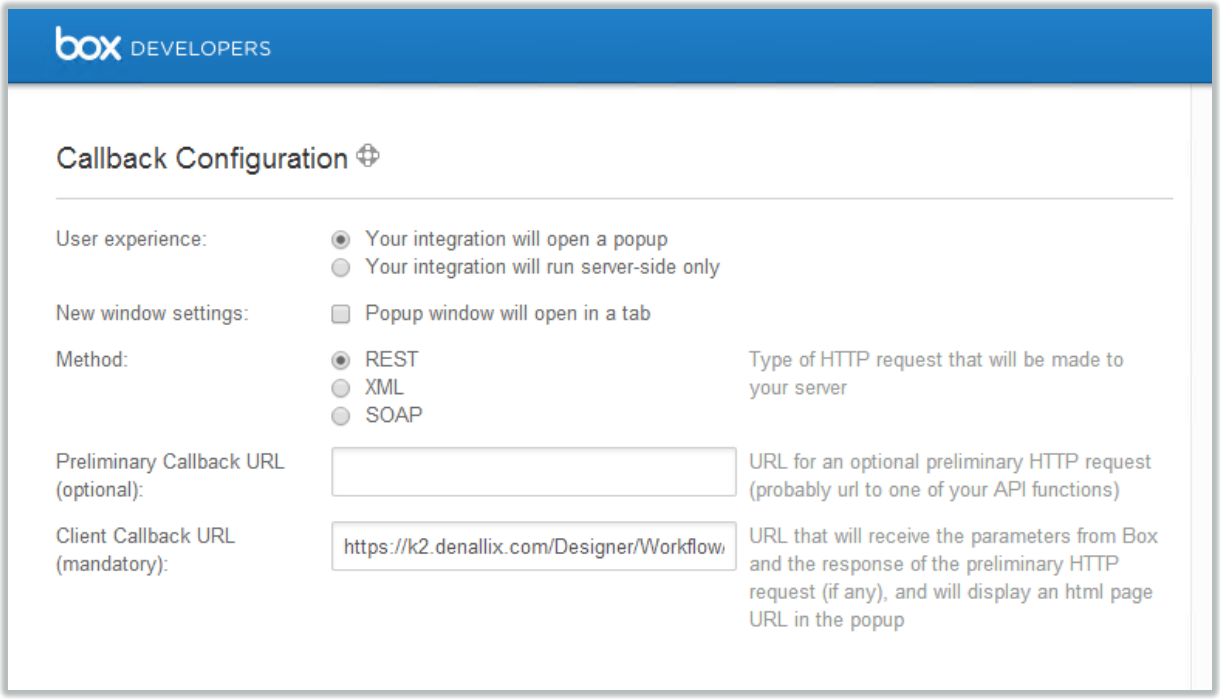
Create a new App and give it a name and a description.

Important Considerations about Box integration:

1. Once you create an app on Box there is no way to delete it. Try to limit the number of test apps you create.
2. If you want to enable the right-click menu for a ***folder*** in Box you must open a support ticket with Box.
3. If you want to call workflows automatically when a document is added to Box you must use **Webhooks**. Using Webhooks is not covered in this document, but the same principles will apply as when you use Web App Integration to manually start workflows.

Callback Configuration

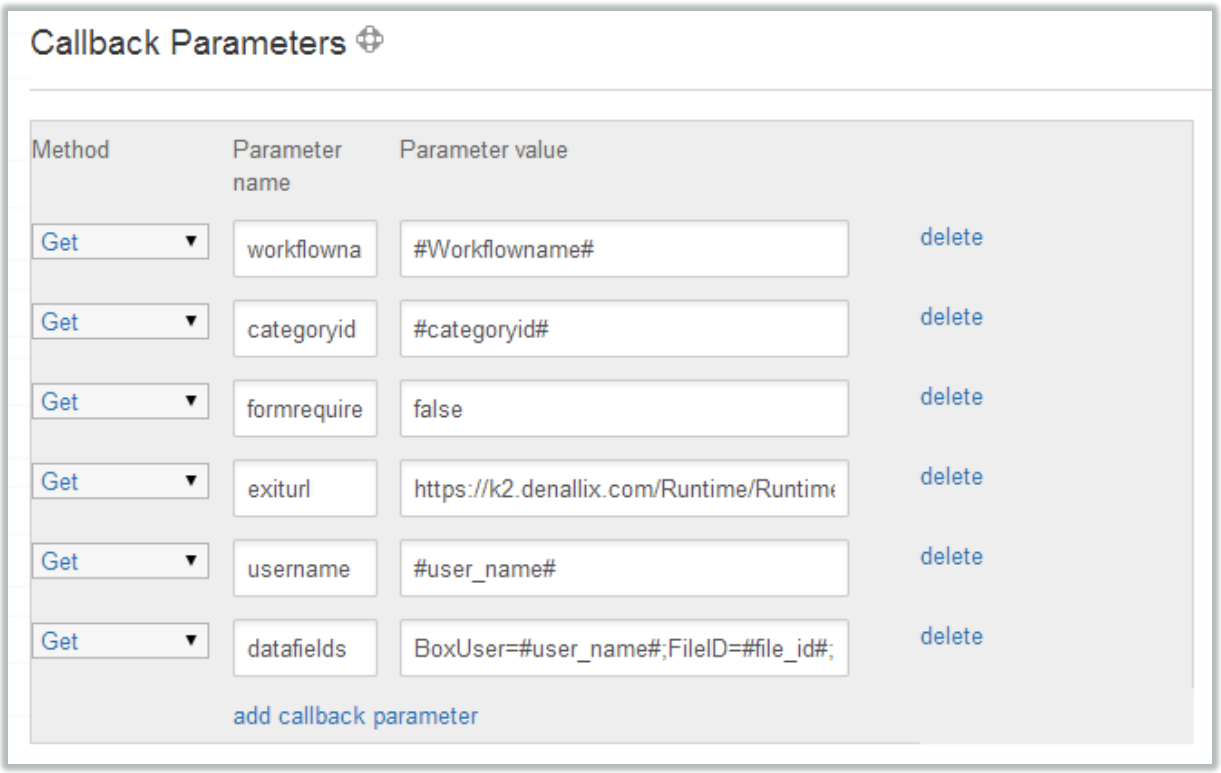
To create a right-click menu item you must add a **Web App Integration**. The Web App Integration requires a **Client Callback URL**, which is the URL to your K2 designer. For example, when using the Denallix core your URL is https://k2.denallix.com/Designer/Workflow/WebDesigner.aspx



Callback Parameters

Adding parameters to the client callback URL is necessary to control how the workflow designer loads, with what options and data from Box.com in the form of **Callback Parameters**.

The parameters that you specify are the ones you need for the design of your workflow as well as the ones you need to track information about workflows designed for Box. As an example, the following example parameters are included in the **Design a K2 Workflow** integration.

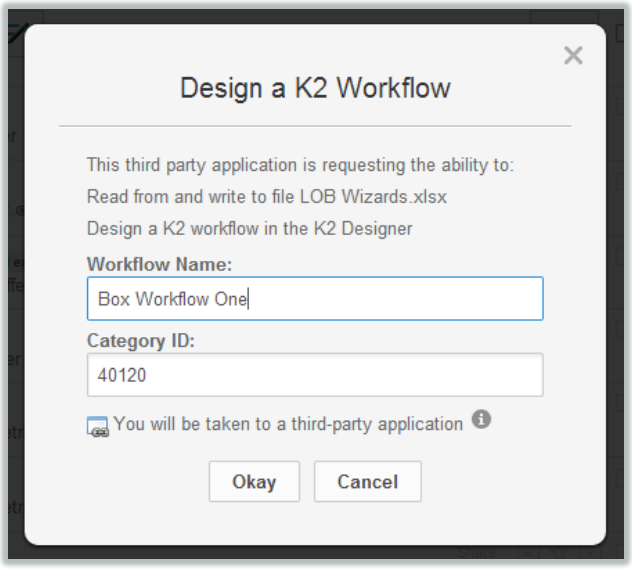


The parameter values for these parameters are taken either from the initial prompt shown in Box when you click **Design a K2 Workflow,** or the parameters refer to variables available in in Box (e.g. user_name, file_id, etc.). In the example, #Workflowname# and #categoryid# are shown on the UI prompt. The values that you specify on that prompt are propagated to the dropdown you see when specifying the value of a parameter.

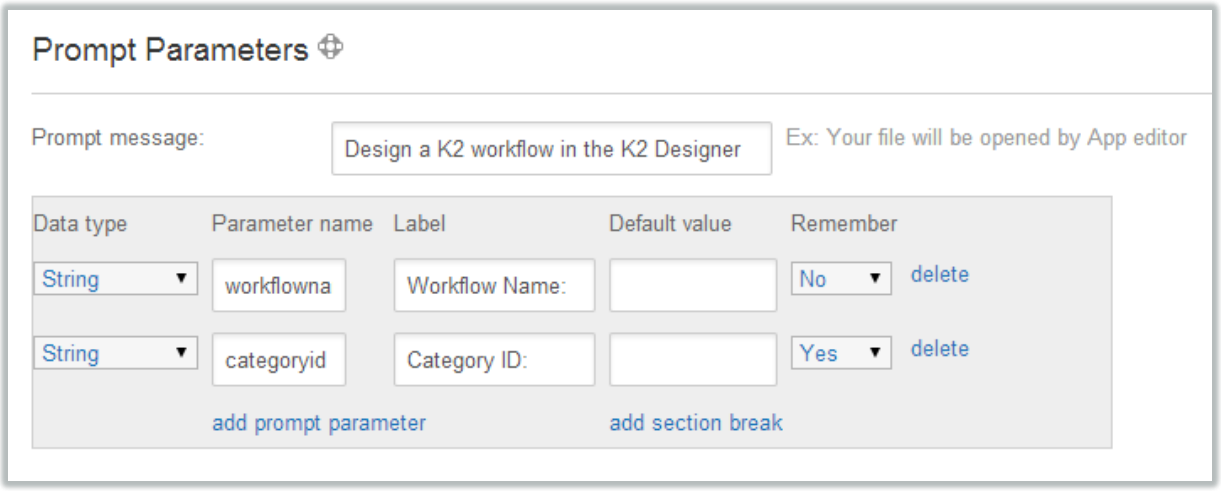
Notice the **exiturl** in the parameters: if you do not specify this you will not see the **Exit** menu item or the **Close & Exit** button in the workflow designer. In this case, the value of **exiturl** is a URL to a SmartForm that displays a confirmation to the user that their workflow deployed and then gives them a link to return to Box.

DESIGNING THE K2 WORKFLOW

When you click **Design a K2 Workflow** in Box, the initial prompt, shown below, asks you for a **Workflow Name** and **Category ID**.



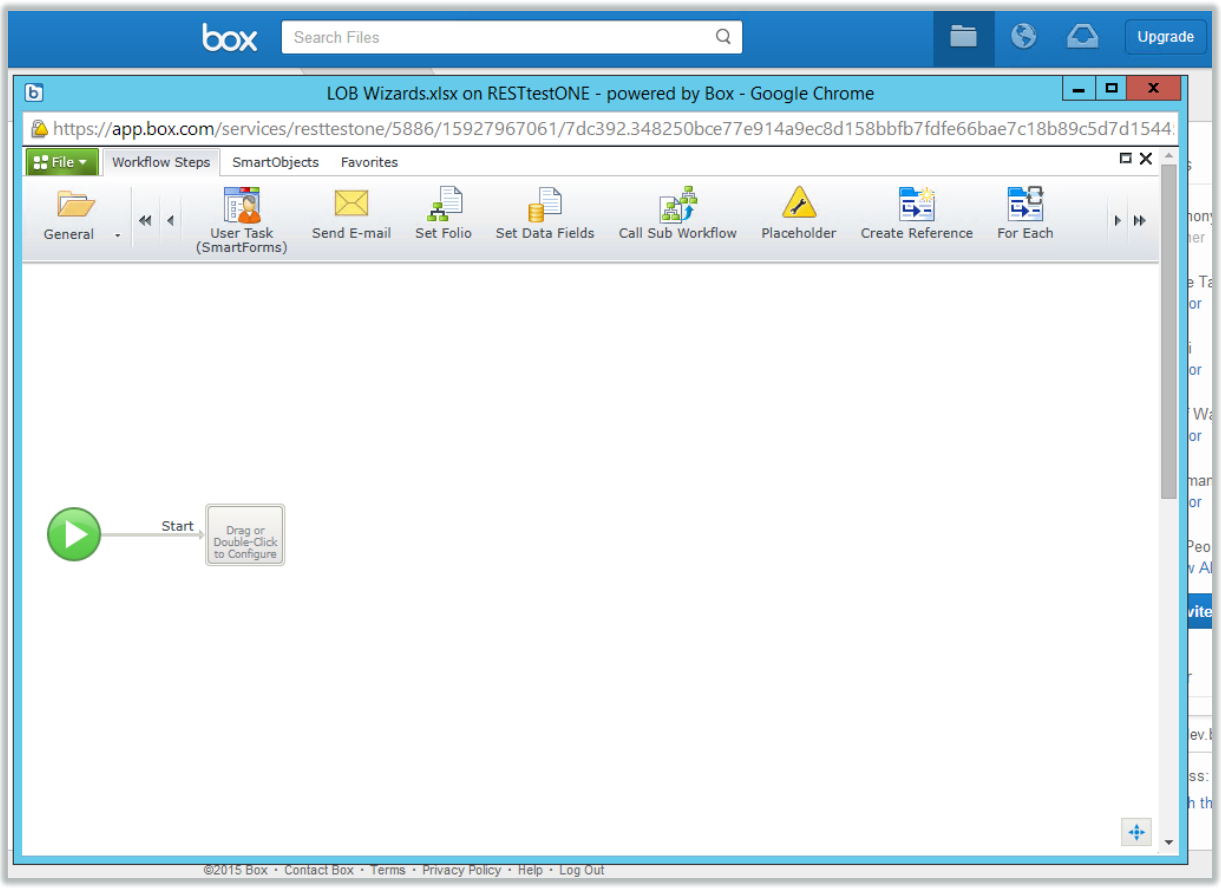
These two items appear on the **Prompt Parameters** section of the Web App Integration.



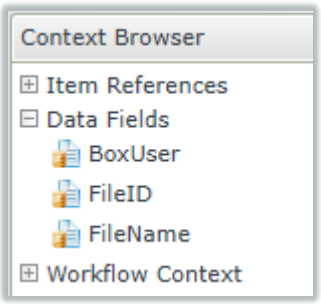
After clicking **Okay** to the prompt, Box appends the **workflowname** and **categoryid** Parameters to the **Client Callback URL** and sends the request to K2. In turn, K2 opens the workflow designer in a new pop-up window.

When the workflow designer is displayed in the pop-up browser window, as shown below, you notice that the URL is not a direct correlation to what you've configured for the callback configuration and parameters. However, you are able to see some parameters if you scroll to the end of the URL. Keep in mind, though, that you won't see all of them. This may be different with other LOB systems, but in this case Box sends through the following URL to K2:

```
https://app.box.com/services/resttestone/5886/15927967061/7dc392.348250bce77e914a9ec8d158bbfb7fdfe66bae7c18b89c5d7d154459f7033d92/LOB_Wizards?name[]=workflowname&value[]=Box%20Workflow%201&name[]=categoryid&value[]=40120
```

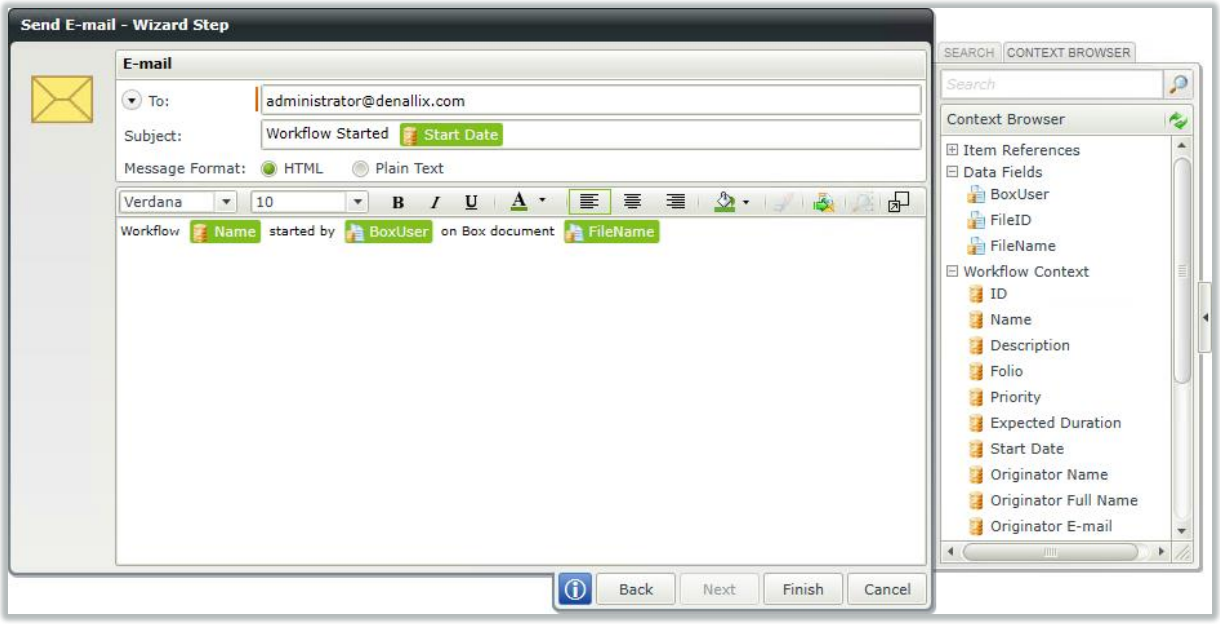


To test that some of parameters are being sent through correctly, you can look at the **Context Browser** in the workflow designer for your data fields. In this case, the Data Fields **BoxUser, FileID** and **FileName** are present so you know that the **datafields** parameter was sent properly.

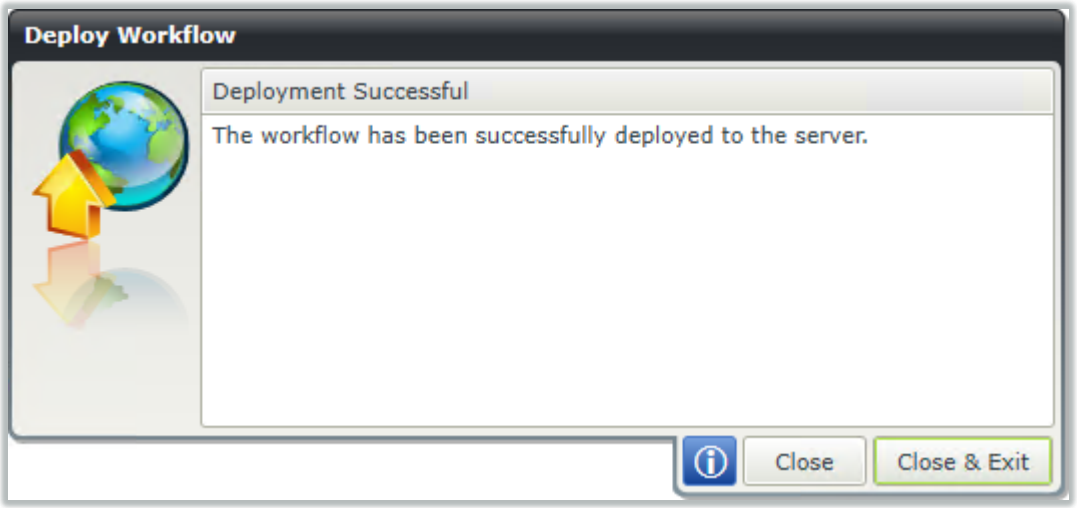


For testing purposes, add a **Send E-mail** event that uses the parameters. Remember that you've configured the workflow to have these data fields and you must pass the same parameters later on and use them when starting the workflow. The key thing to keep in mind is that you must name the parameters exactly the same for each Web App Integration

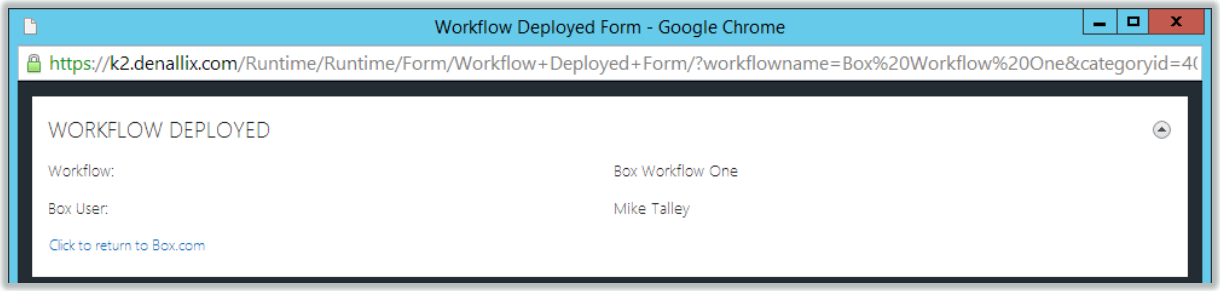
(**Design a K2 Workflow** and **Start a K2 Workflow**).



When you deploy the workflow you can click the **Close & Exit** button. This is available because you've specified an **exiturl** parameter.



When you exit the workflow designer you are redirected to a SmartForm that confirms the workflow has been deployed. Remember you are free to build your own SmartForm to display whatever information you choose as the **exiturl**. For testing purposes, you may want to use Form Rules to show the values of the parameters passed in the **exiturl** query string.

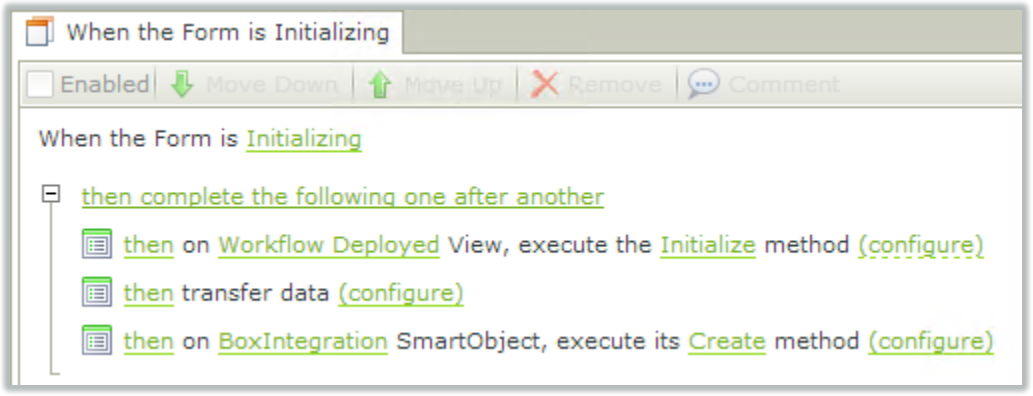


TRACKING THE BOX WORKFLOWS

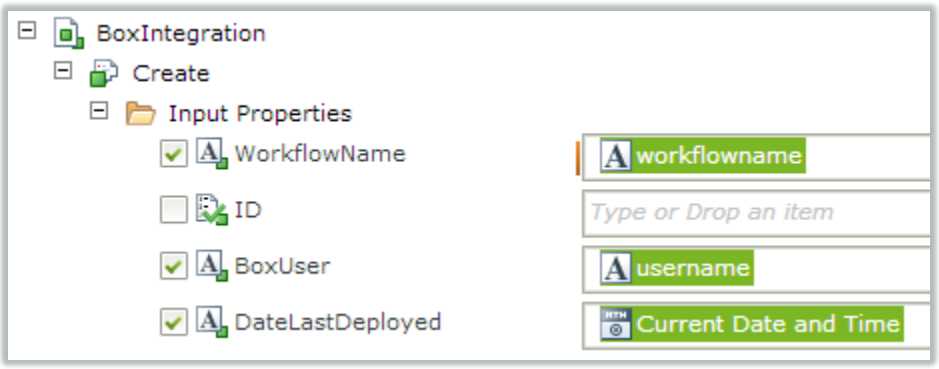
The **workflowname, categoryid** and **username** parameters are passed along to the **Workflow Deployed** form via the **exiturl**. This form is configured with matching parameters and use a Data Transfer rule to populate the data labels. This form also calls the **Create** method on the **BoxIntegration** SmartObject during the initialization of the form to track Box workflows, writing the workflow name, user name and date created information to the SmartObject.

BoxIntegration is a SmartBox SmartObject that contains the following structure:

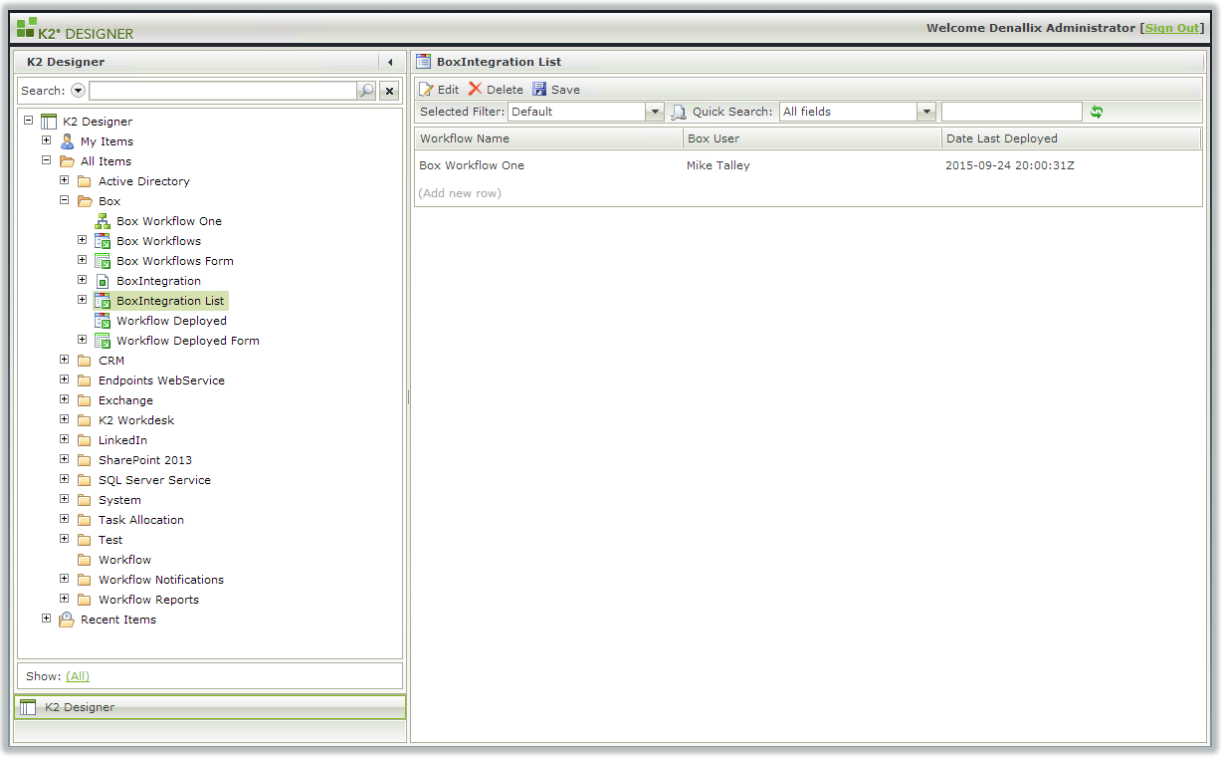
In the rule **When the Form is Initializing**, data from the Parameters from the **exiturl** are transferred to data labels on the form.



Following the data transfer, the **Create** method of the **BoxIntegration** SmartObject is used to create a new record for the workflow.



You can confirm that your tracking SmartObject contains data by generating a list view for the **BoxIntegration** SmartObject, and then running the view.



The **List** method of the **BoxIntegration** SmartObject is used on the **Start a K2 Workflow** form to allow the user to select which workflow they want to manually start on a Box file or folder.

K2 PACKAGE

Download the SmartObject, Forms and Views for this solution by clicking on the **Box Querystring Demo.kspx** package attached to this article. Note that the Box app configuration is separate from this package and the **Box Workflow One** workflow is not included in the package. The service instance for the Endpoint Web Service is called **K2WS** and is configured for the Denallix 6.1 core.

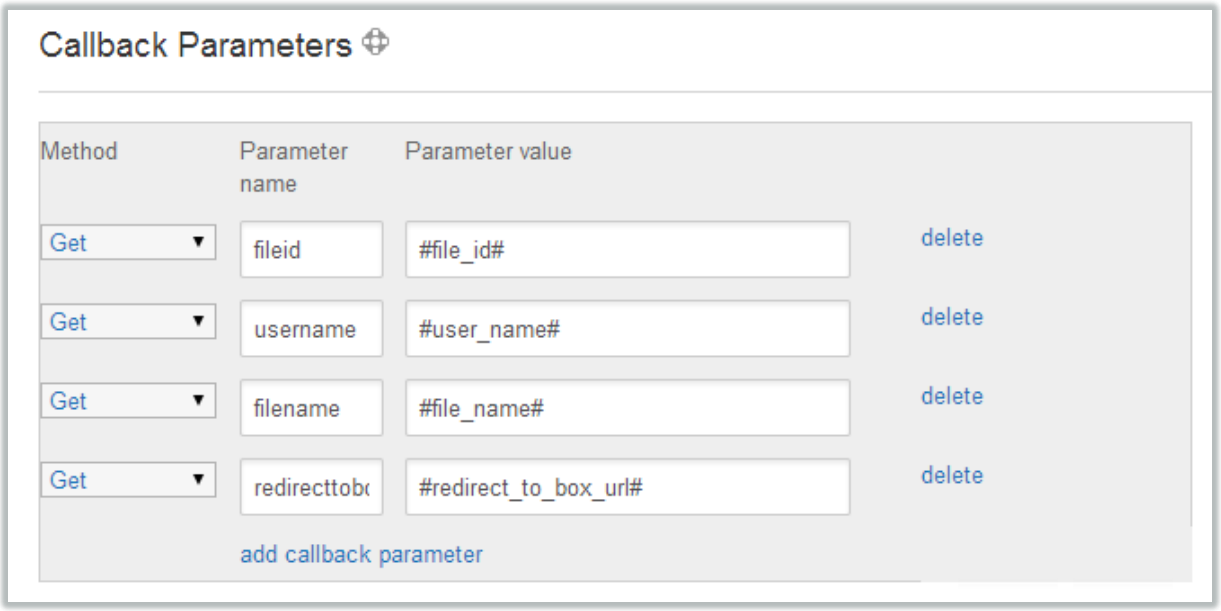
Box Querystring Demo.kspx

STARTING A K2 WORKFLOW FROM BOX

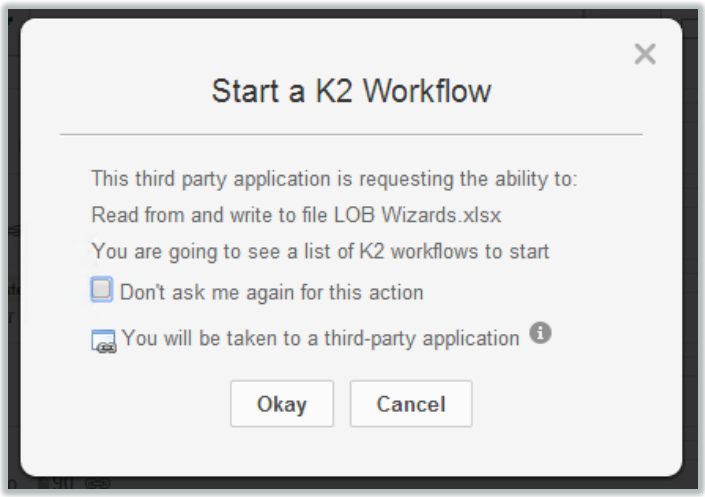
Your second Web App Integration, Start a K2 Workflow, is where you pass the parameters from Box to a SmartForm that lists the available Box workflows and lets the user select the workflow they want to start.

The Client Callback URL for this integration is a direct link to the form that contains the view. In this example the URL is https://k2.denallix.com/Runtime/Runtime/Form/Box+Workflows+Form/

The parameters passed to this form are configured as **Callback Parameters** as shown below. Note the same names are used, but not every parameter needed for the **Design a K2 Workflow** integration is necessary to start a workflow.

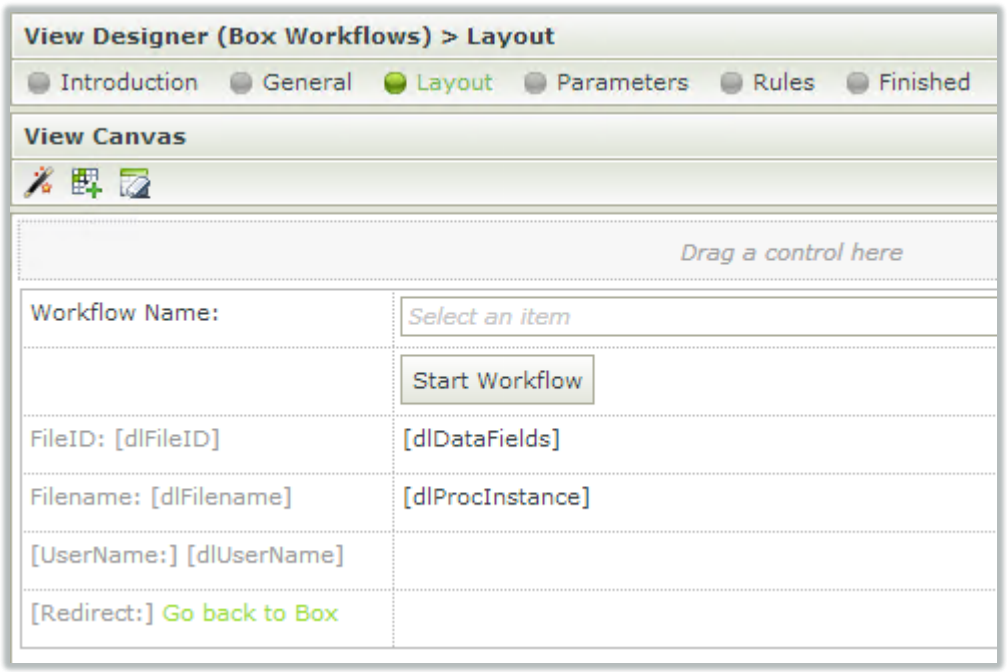


There are no **Prompt Parameters** for this second Web App Integration, but you still see the prompt from Box before being redirected to the SmartForm.

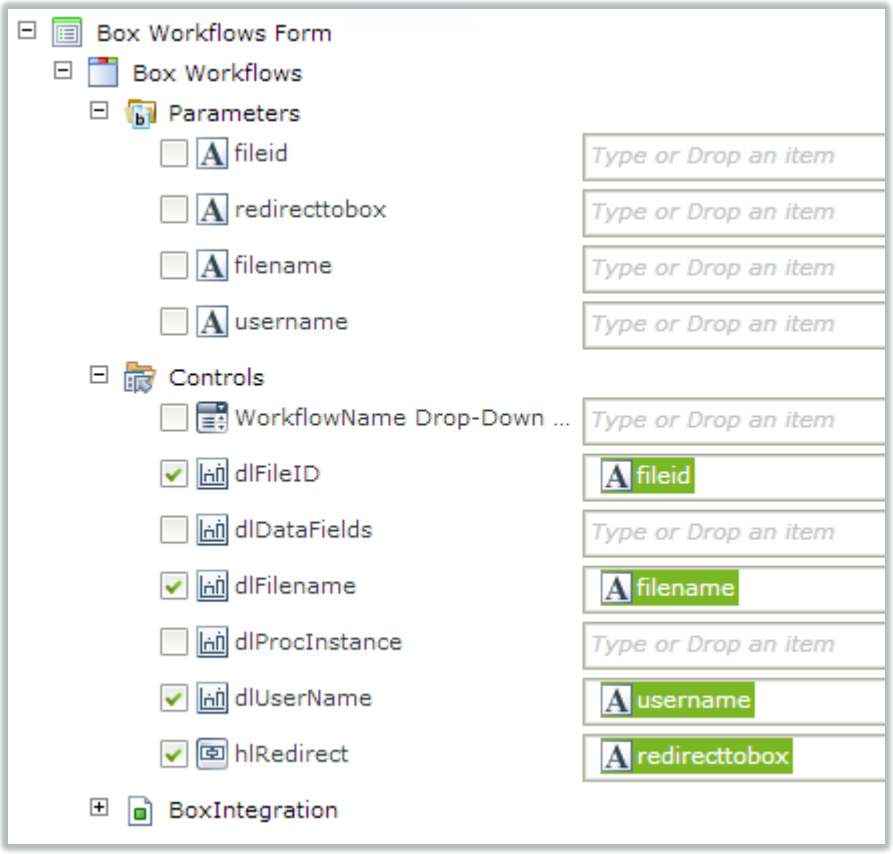


After clicking **Okay** you are redirected to the Form. This form takes in the Parameters from Box, queries the **BoxIntegration** SmartObject, and allows you to select from the available list of workflows created for Box files and folders.

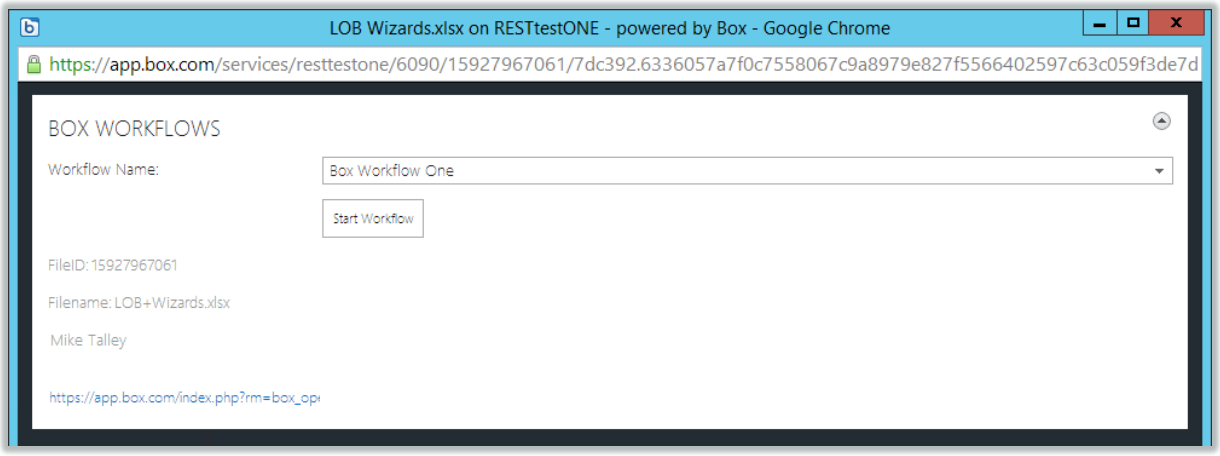
The View that the form uses contains a dropdown bound to the list method of the **BoxIntegration** SmartObject, and lists the available workflows. The View also contains some data labels to which the Parameter values are mapped.



In the rule **When the Form is Initializing** the Parameter data from the **exiturl** is mapped to the labels.

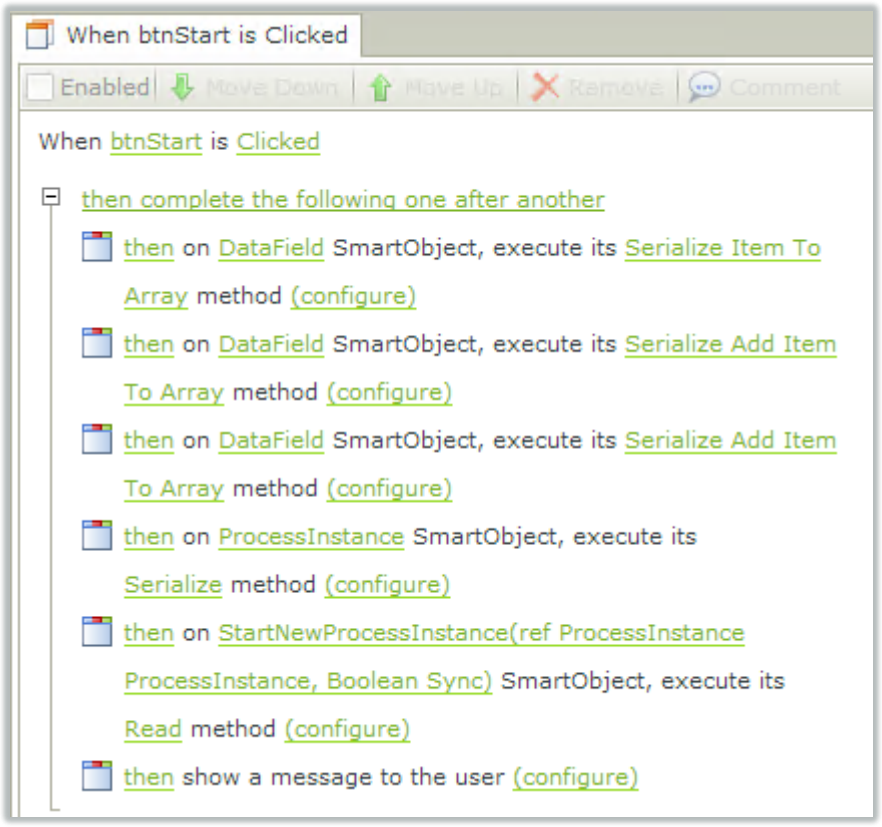


When the form is opened, data from the Parameters and the **BoxIntegration** SmartObject are presented.

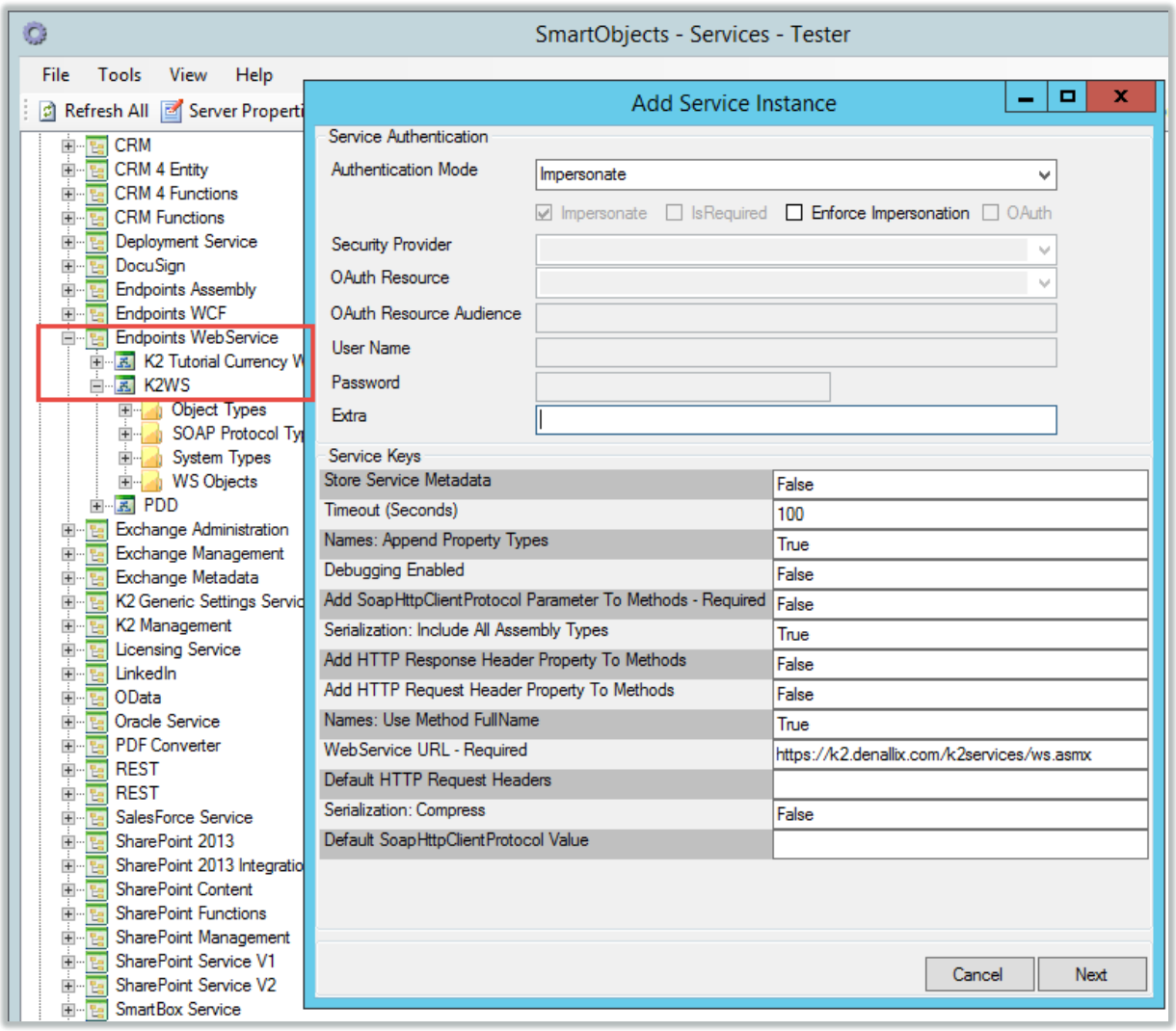


Starting a workflow is a matter of selecting the workflow from the dropdown, in this case **Box Workflow One**, and clicking the **Start Workflow** button.

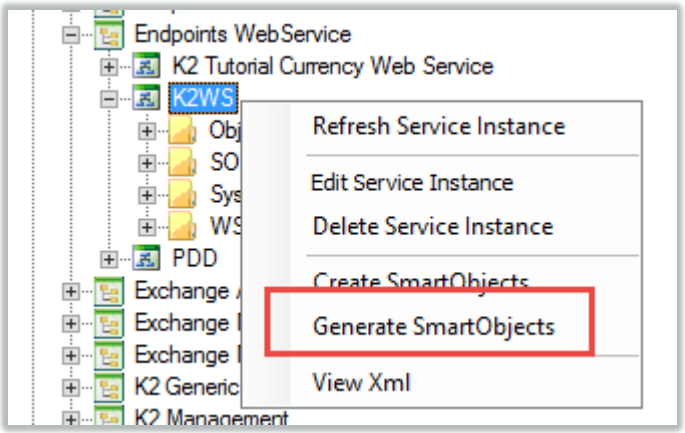
However, there's a rule behind the **Start Workflow** button in the View that you must configure first.



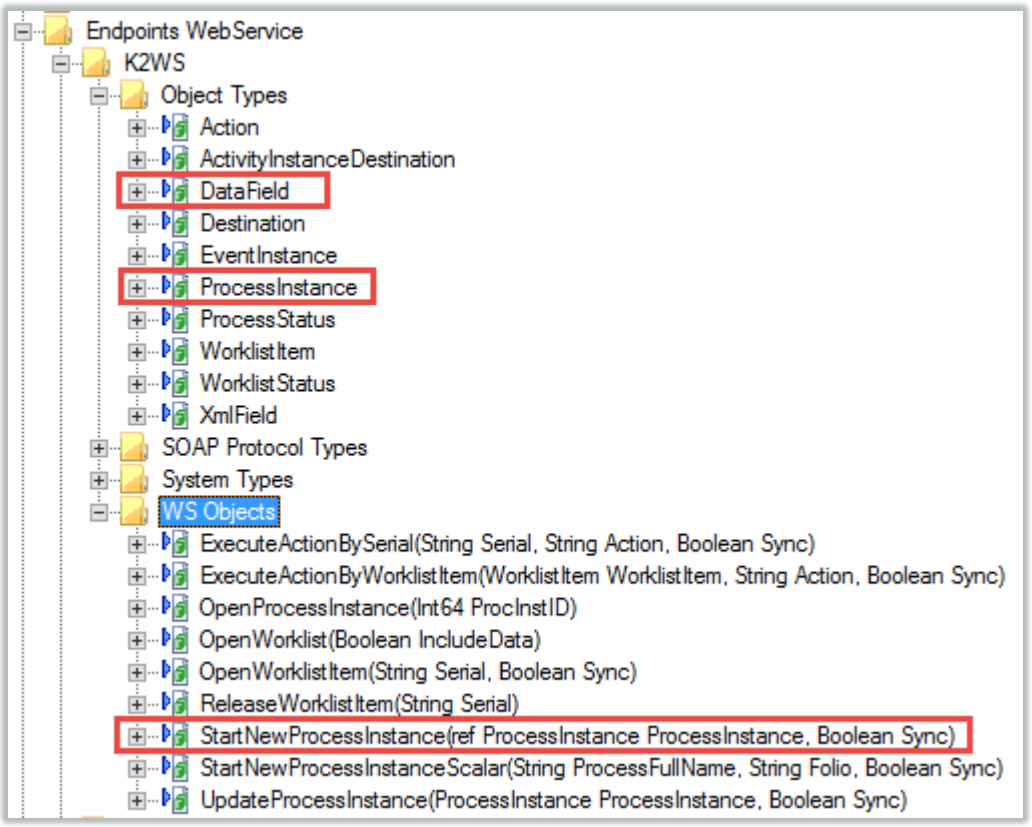
This rule, in turn, uses SmartObjects created by using the **Endpoints Web Service** service and registering a service instance on the K2 Services endpoint, https://k2.denallix.com/k2services/ws.asmx



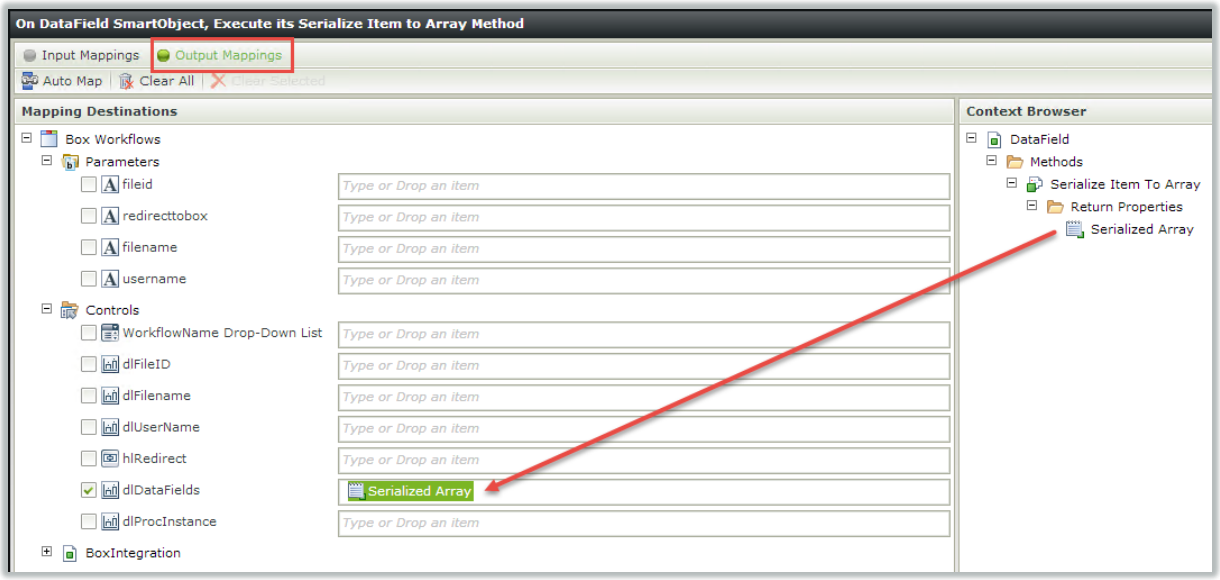
Once you generate the service instance, right-click the service instance and choose **Generate SmartObjects** to generate SmartObjects for this service. Once generated you can use these SmartObjects in your view rule.



The SmartObjects used in the view rule are **DataFields**, **ProcessInstance** and **StartNewProcessInstance**.

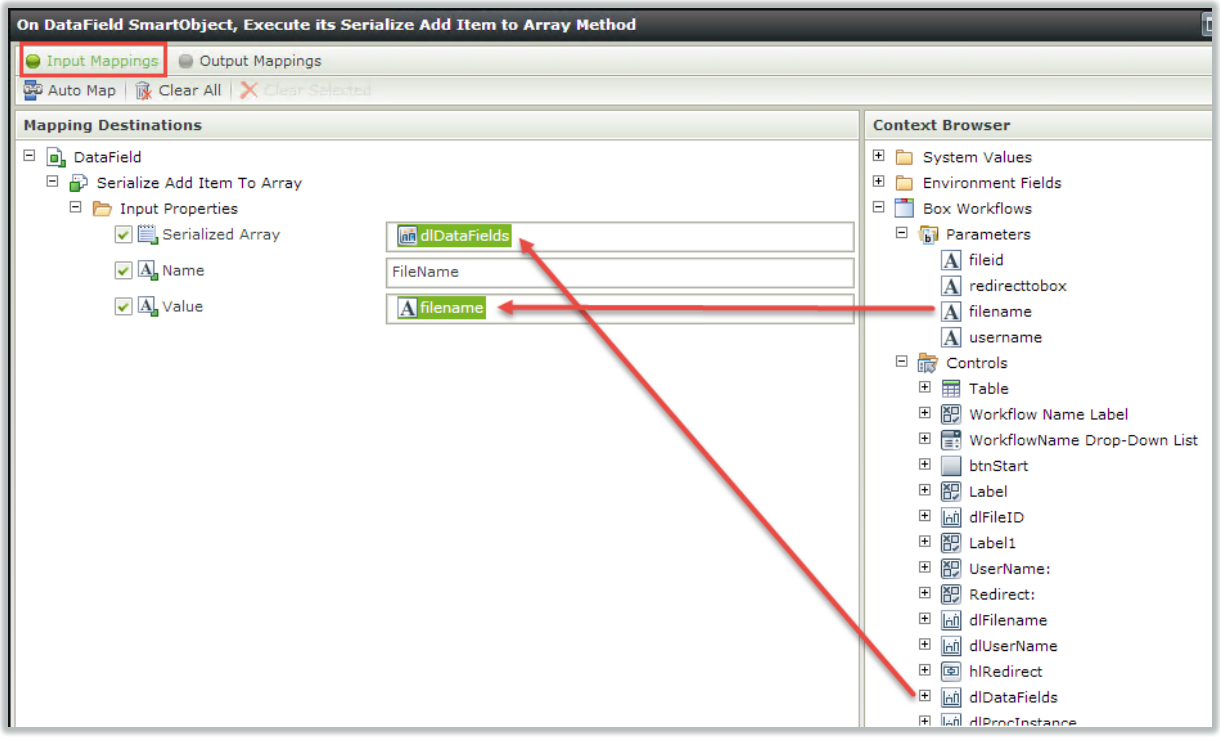


Back to the rule on the view, when you click the **Start Workflow** button (**btnStart**), a few things happen to prepare the workflow to be started. It starts with taking the first parameter from Box and, along with the Data Field named **BoxUser**, serializes it using the **Serialize Item to Array** method of the **DataField** SmartObject, storing the result in the **dlDataFields** data label on the form.

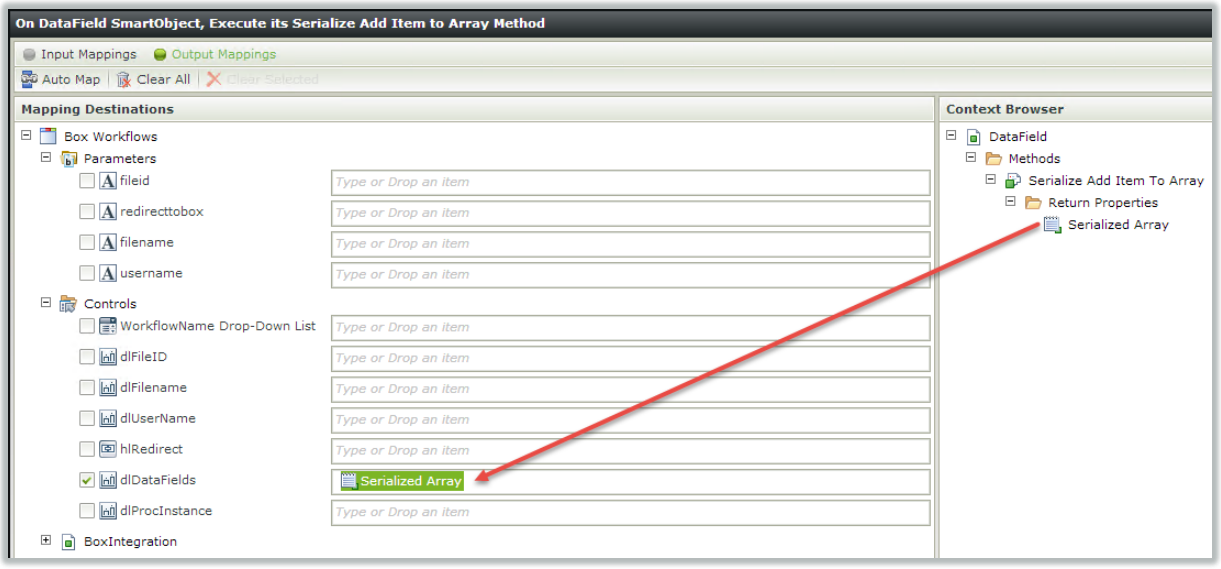


This creates the first item in the **DataFields** serialized array that is required to start the workflow instance so that the data fields are populated with the parameters from Box.

The second data field, **FileName**, is serialized to the DataFields array using the **Serialize Add Item to Array** method of the **DataField** SmartObject. The reason why you use a different method is because the array is already created and stored in the **dlDataFields** data label by the first parameter mapping above.

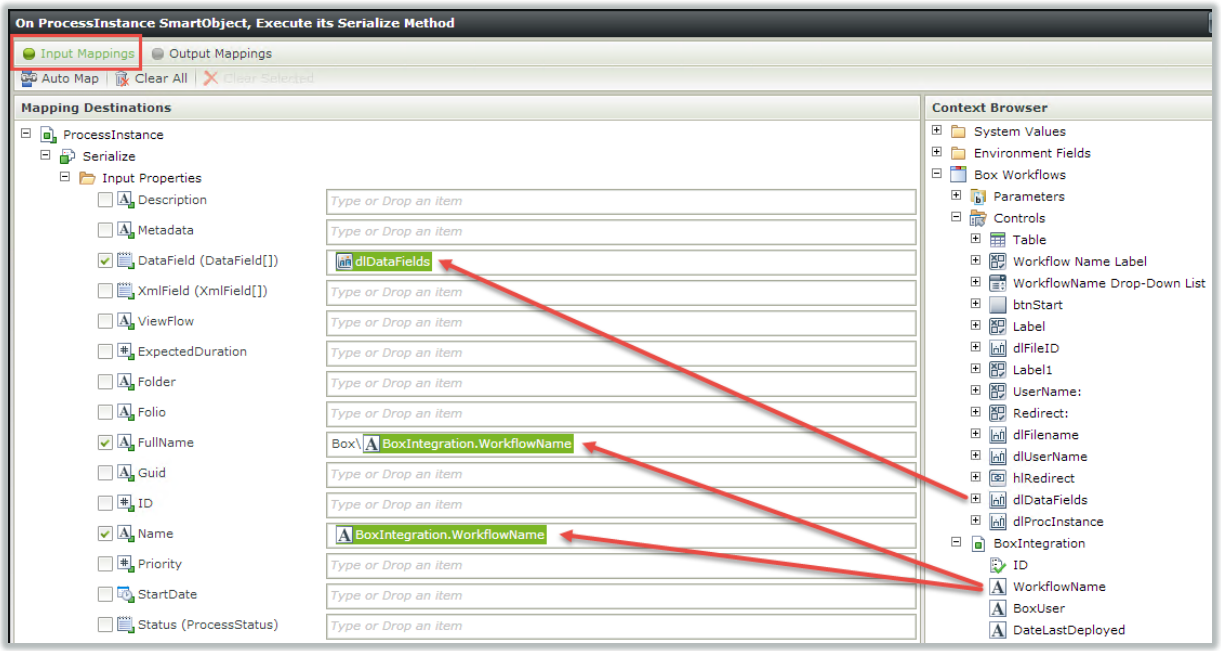


You then map the result back to the **dlDataFields** data label.

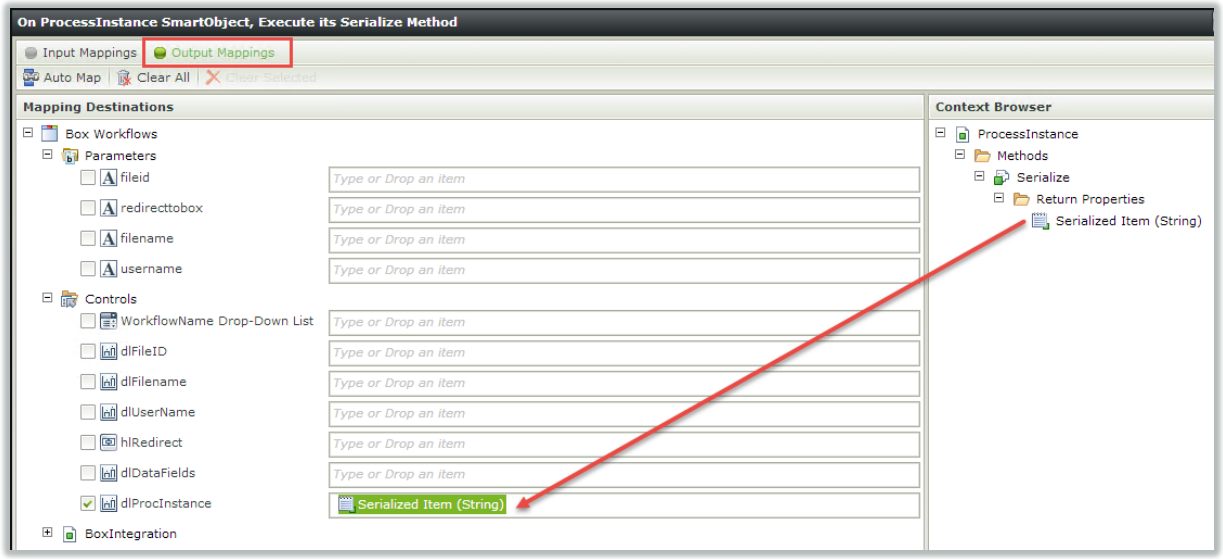


Do the same thing for the next parameter, mapping the **filename** parameter to the **FileName** data field.

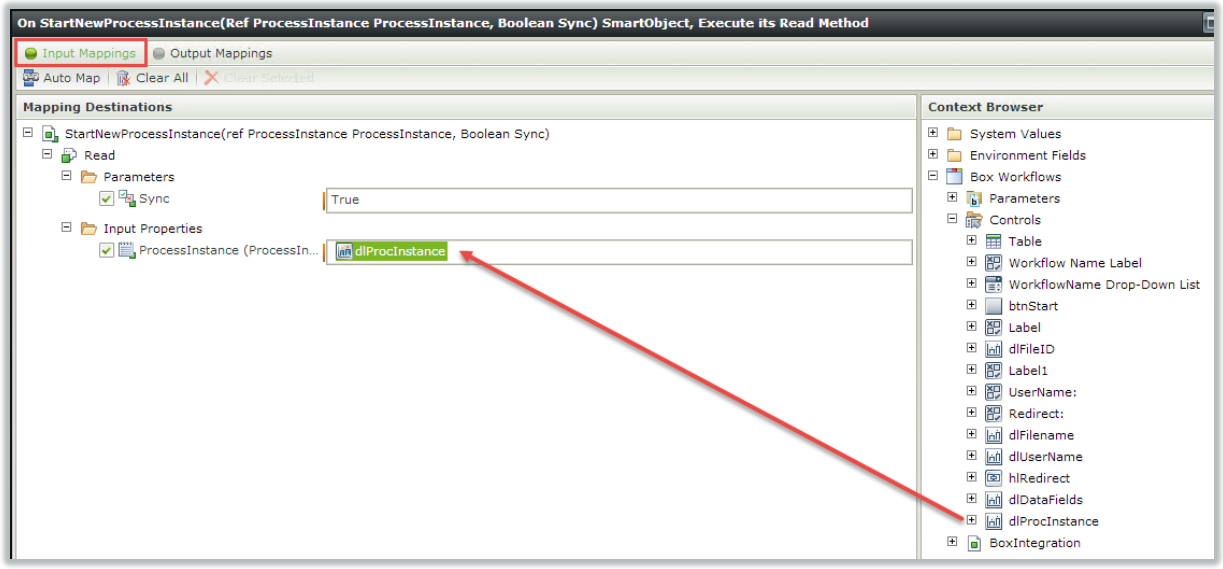
Once finished with the data fields you use the **Serialize** method of the **ProcessInstance** SmartObject to create a serialized representation of the workflow, using the **dlDataFields** serialized string and the **workflow name** as inputs. Note that you need to specify both the workflow **FullName** and the **Name** as inputs, with the **FullName** including the category name where the workflow is deployed. Note: If you wanted to specify values for other fields in the workflow, such as **Folio**, you would specify them here.



The result of the **Serialize** method is stored in the **dlProcInstance** data label on the view.

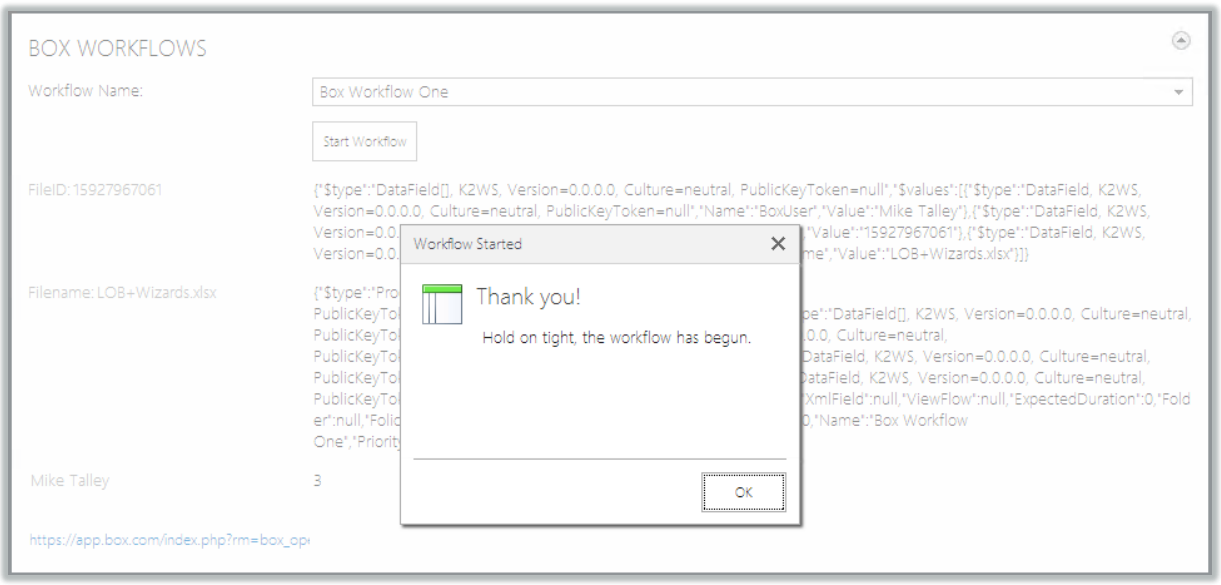


The last step is to call the **Read** method of the **StartNewProcessInstance** SmartObject, passing in the serialized workflow instance from the **dlProcInstance** data label. You must also pass **True** or **False** as the **Sync** parameter, depending on whether you want to wait for the start workflow call to reach a wait state in the workflow (e.g. a client event), or just return as soon as the workflow is started.

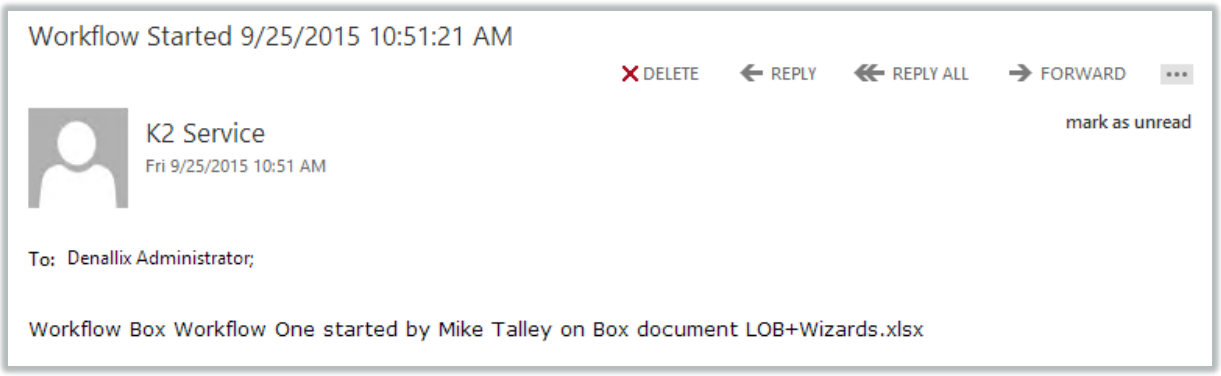


The final action in the rule simply shows a confirmation dialog that the workflow started. You see in the background that the form contains the serialization fields that would normally be

hidden, but for debugging purposes are left visible.



This simple workflow sends an e-mail to confirm that the context came through from Box.



Once this is done, you have confirmed that the workflow integration from Box to K2 is working. Feel free to continue editing the workflow as desired.