K2® LEARNING

K2 blackpearl Core

Hands-On Exercises Guide

Version: Version 3 Revision 1

Date: October 2016

"Terms Of Use" on page 332

Table of Contents

Table of Contents 2

100.BHX: Introduction to K2 Applications with K2 Designer 4

Part 1: K2 Application Basics 5

EXERCISE 1: Leave Application (Basic Version) 6

Part 2: Diving Deeper into Data, Forms and Workflows 56

EXERCISE 2: SmartObjects 57

EXERCISE 3: SmartForms 75

EXERCISE 4: Workflows 83

100.CWL: Reporting in K2 114

Part 1: Standard Reports in K2 115

EXERCISE 1: Standard K2 Reports 116

Part 2: Custom reporting in K2 137

EXERCISE 2: Building custom reports in K2 Workspace (Optional) 138

EXERCISE 3: Building Custom Reports with Microsoft Excel (Optional) 148

100.DND: Basic K2 Administration 160

EXERCISE 1: Deploying a K2 Package 161

Download and Deploy a K2 Package 161

EXERCISE 2: Managing processes and process instances 165

Managing Processes and Process Instances 165

EXERCISE 3: Managing Task Lists 176

Managing Task Lists 176

EXERCISE 4: Managing security and setting permissions 193

Managing Permissions 193

Administering K2 - Summary and Challenge Steps 199

100.EDI: Building Workflows and SmartObjects with K2 Studio 202

Part 1: Workflows in K2 Studio 203

EXERCISE 1: Building basic Workflows with K2 Studio 204

EXERCISE 2: Intermediate Workflow concepts with K2 Studio 254

Part 2: SmartObjects in K2 Studio 308

EXERCISE 3: Building a composite SmartObject with K2 Studio 309

Terms Of Use 332

ACCEPTANCE OF TERMS 332

COPYRIGHT 332

THIRD-PARTY INTEGRATION 332

NOTICE REGARDING CONTENT OF THIS DOCUMENTATION 332

100.BHX: Introduction to K2 Applications with K2 Designer

100.BHX
Version:1 Revision:1
Duration: 3 hours

K2 LEARNING

Introduction to K2 Applications with K2 Designer

- Using K2 Designer to build simple applications
- How Data, Forms, and Workflows are used to build an application
- Integrating with external systems with SmartObjects
- Integrating with external systems with Workflow wizards
- Using SmartObjects in Forms and Workflows
- Workflow concepts: escalations, task slots, workflow patterns

Video

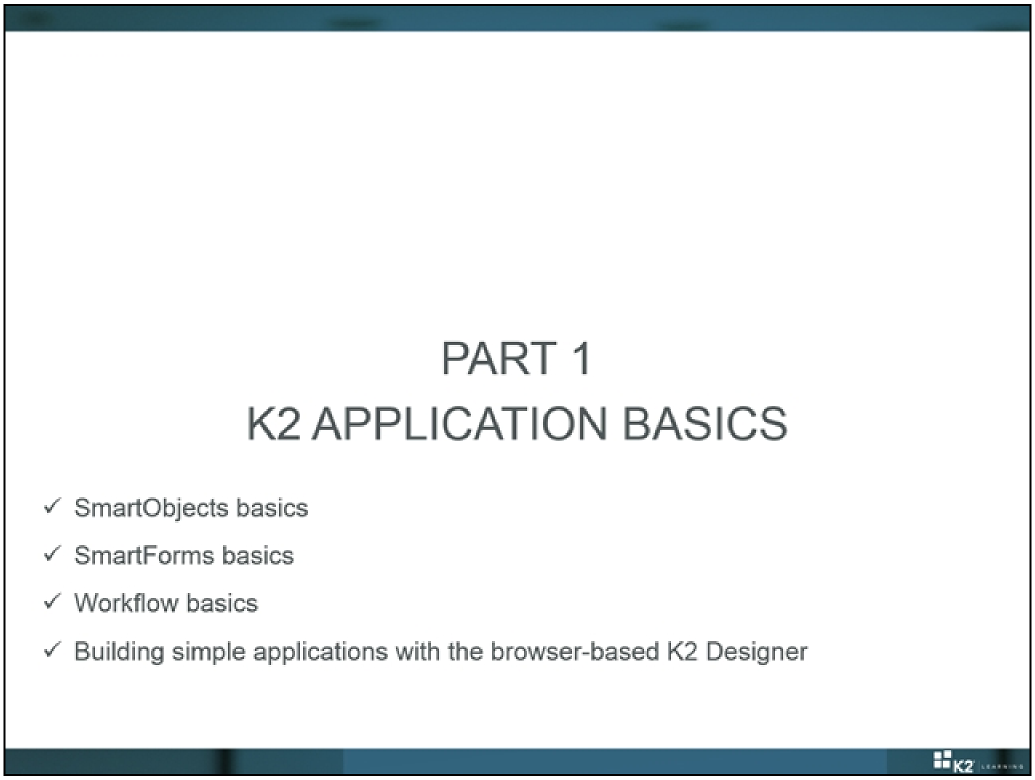
The *100.BHX: Introduction to K2 Applications with K2 Designer* training module explains how to build K2 applications in terms of Data, Forms and Workflows. In this module, we will use K2 Designer to build an application with Data (SmartObjects), Forms (SmartForms) and Workflow.

This module covers the following concepts:

- Using K2 Designer to build simple applications
- How Data, Forms, and Workflows are used to build an application
- Integrating with external systems with SmartObjects
- Integrating with external systems with Workflow wizards
- Using SmartObjects in Forms and Workflows
- Workflow concepts: escalations, task slots, workflow patterns

Note
Although this module focuses on the web-based K2 Designer which may or may not be used by all organizations, we will be explaining some fundamental concepts about the elements that make up an application, and the knowledge gained in this module will set the scene for other learning modules that delve further into Workflow and SmartObjects. Therefore, even if you do not intend to use K2 Designer in your organization, we recommend that you complete this learning module to build up your initial knowledge of K2 applications.

Part 1: K2 Application Basics



PART 1

K2 APPLICATION BASICS

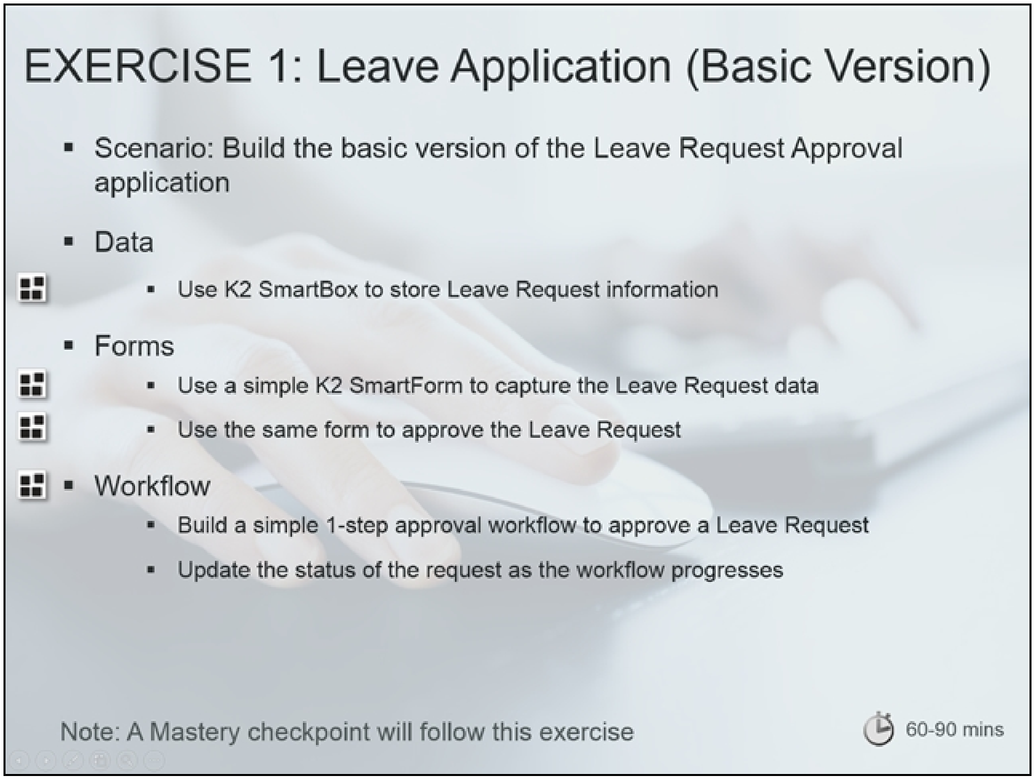
- ✓ SmartObjects basics
- ✓ SmartForms basics
- ✓ Workflow basics
- ✓ Building simple applications with the browser-based K2 Designer

K2 LEARNING

▶ Video

In Part 1 we will look at the basics of K2 applications that combine Data (SmartObjects), Forms (SmartForms) and Workflows. We will be using the browser-based K2 Designer for this module. At the end of Part 1, we will build a simple Leave Request Approval application from start to end. In Part 2, we will extend upon this basic version of the application with more advanced Data, Forms and Workflows.

EXERCISE 1: Leave Application (Basic Version)



EXERCISE 1: Leave Application (Basic Version)

- Scenario: Build the basic version of the Leave Request Approval application
- Data
  - Use K2 SmartBox to store Leave Request information
- Forms
  - Use a simple K2 SmartForm to capture the Leave Request data
  - Use the same form to approve the Leave Request
- Workflow
  - Build a simple 1-step approval workflow to approve a Leave Request
  - Update the status of the request as the workflow progresses

Note: A Mastery checkpoint will follow this exercise

60-90 mins

Video

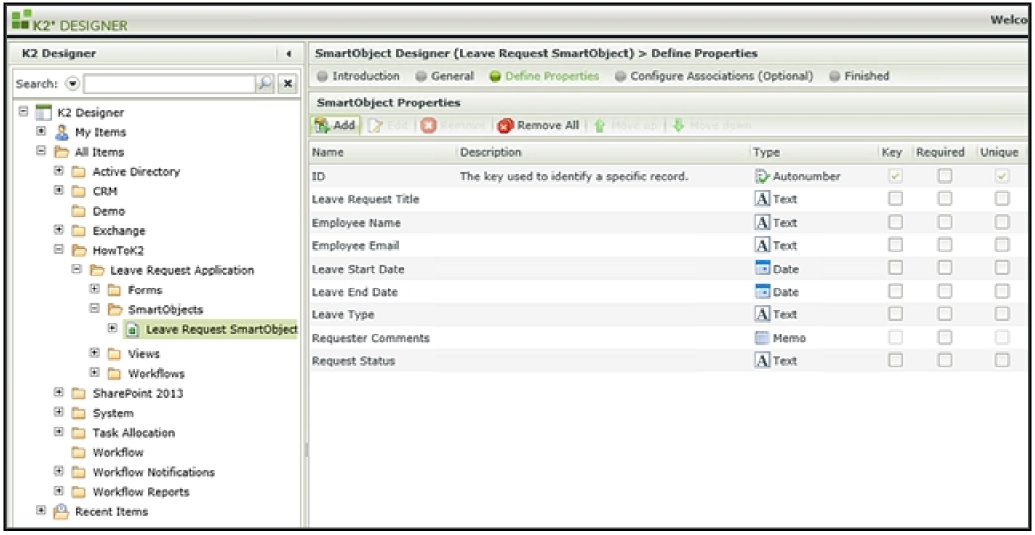
Now that you are familiar with the basics of SmartObjects, SmartForms and workflows, let's build an application that combines all these elements.

In this exercise, you will build a simple "Leave Request Approval" application start to end, including Data, Forms and Workflow. We will be building and testing the application as one exercise; although this is a long exercise, it will give you a good idea of the steps necessary to implement a simple K2 application from start to end.

The application components are described briefly below:

For the Data component, we will use a single SmartBox SmartObject to store the leave request data entered by the user.

The Leave Request SmartObject

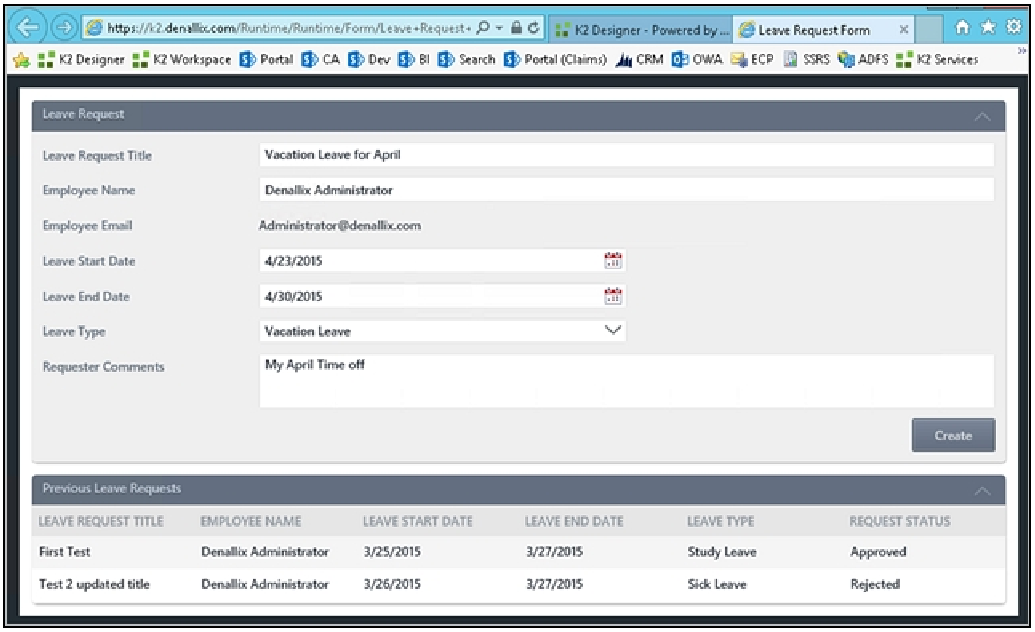


SmartObject Designer (Leave Request SmartObject) > Define Properties

SmartObject Properties

Name	Description	Type	Key	Required	Unique
ID	The key used to identify a specific record.	Autonumber	☑	□	☑
Leave Request Title		Text	□	□	□
Employee Name		Text	□	□	□
Employee Email		Text	□	□	□
Leave Start Date		Date	□	□	□
Leave End Date		Date	□	□	□
Leave Type		Text	□	□	□
Requester Comments		Memo	□	□	□
Request Status		Text	□	□	□

For the Forms component, we will use basic SmartForms Views and Forms as the user interfaces for this application, to allow users to enter and review the leave request data. For this exercise, we will only do basic configuration of the Views and Forms; the main purpose is just to get you familiar with how Forms fit into an application.



Leave Request

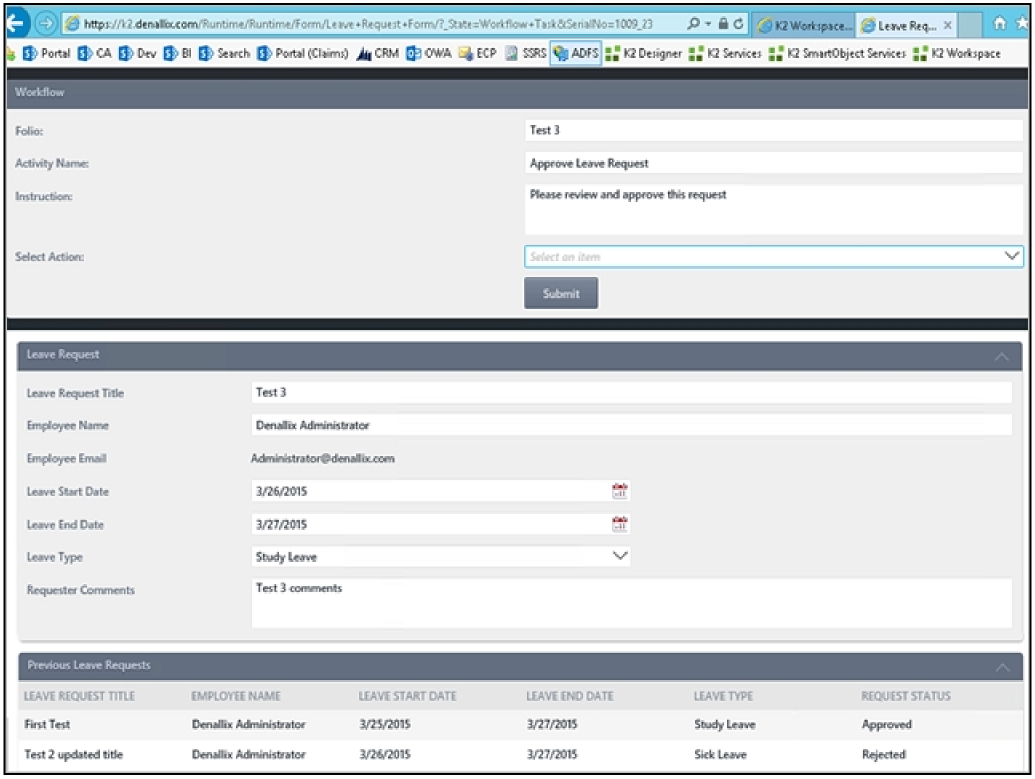
Leave Request Title	Vacation Leave for April
Employee Name	Denallix Administrator
Employee Email	Administrator@denallix.com
Leave Start Date	4/23/2015
Leave End Date	4/30/2015
Leave Type	Vacation Leave
Requester Comments	My April Time off

Previous Leave Requests

LEAVE REQUEST TITLE	EMPLOYEE NAME	LEAVE START DATE	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
First Test	Denallix Administrator	3/25/2015	3/27/2015	Study Leave	Approved
Test 2 updated title	Denallix Administrator	3/26/2015	3/27/2015	Sick Leave	Rejected

On the Approval form, we will let K2 "inject" a control to allow the approver to Approve or Reject the leave request.

The approval form



Workflow

Folio:	Test 3
Activity Name:	Approve Leave Request
Instruction:	Please review and approve this request
Select Action:	Select an item

Leave Request

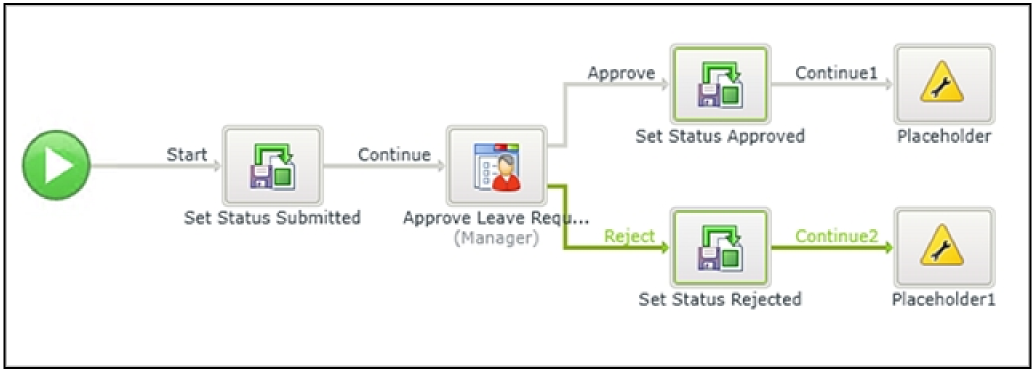
Leave Request Title	Test 3
Employee Name	Denallix Administrator
Employee Email	Administrator@denallix.com
Leave Start Date	3/26/2015
Leave End Date	3/27/2015
Leave Type	Study Leave
Requester Comments	Test 3 comments

Previous Leave Requests

LEAVE REQUEST TITLE	EMPLOYEE NAME	LEAVE START DATE	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
First Test	Denallix Administrator	3/25/2015	3/27/2015	Study Leave	Approved
Test 2 updated title	Denallix Administrator	3/26/2015	3/27/2015	Sick Leave	Rejected

The workflow will be a very simple, single-step approval style workflow. In addition to the User Task where the approver will approve the request, we will add server events that will update the status of the request as the workflow progresses.

The Leave Request Approval workflow



When you are ready, continue on to " Leave Request Approval (V1 or "Basic" version)" below to review the design of the application, and then you can start building the application as described in "Building the Leave Request Approval Application (V1 or "Basic" Version)" on page 12.

Leave Request Approval (V1 or "Basic" version)

This tutorial explains how to build a simple Leave Request Approval application, using K2 Designer to build an application with Data (SmartObjects), Forms (SmartForms) and Workflow elements. The tutorial is intended for users new to K2, or with little K2 experience.

This tutorial can be completed in any environment that has K2 version 4.6.9 or later installed. K2-delivered training events normally include access to a Virtual Server environment provided by K2 which you will use for the exercises. The screenshots and users used in the tutorial reflect this K2-provided virtual environment. You can, however, complete this tutorial in any other environment as long as the necessary K2 components are installed and operational. You must also have the necessary rights and permissions to create the K2 artifacts in the target environment.

Some tutorials require that you have internet access to be able to connect to an external SQL database used for retrieving data. Those tutorials contain expanded information on internet connection requirements.

Leave Request Approval Application Overview

K2 applications consist of four main components: **Data**, **Forms**, **Workflows** and **Reports**. This tutorial is intended to give you an entry-level understanding of the Data, Forms and Workflow components and how these work together to build a K2 application. For this particular application, we will use a SmartBox-based SmartObject as the Data component, a K2 smartforms-based Form as the Form component and a simple, single-approval workflow. All the components in this application will be designed using K2's browser-based K2 Designer tool.

Understanding the Main Components of K2

The following is an overview of each component and how they integrate with each other to form a complete K2 Application.

Data represents information that could reside in a variety of sources including SharePoint lists, Active Directory, SQL Server, CRM, SAP and other systems. K2 interacts with these data sources primarily through a technology called **SmartObjects**. SmartObjects are the connections made to the data sources and the data source properties and methods that are returned through those connections.

Forms represent the User Interfaces that people use to capture and view information and potentially start a workflow or complete tasks during a workflow. For this Application, we will use a technology called **SmartForms**, which allow you to easily create User Interfaces incorporating SmartObjects, Controls, Rules and Workflow integration. (With K2, you can also use other Forms technologies such as web-based forms and InfoPath, among others. You are not limited to only SmartForms.)

Workflows provide the mechanism for moving data along in a logical sequence of events and tasks.

Reports are automatically available and generally use K2 Workspace as the Report and Administer tool for K2 Applications. You can also create dashboard-style Reports using reporting controls found in K2 Designer. (This application will not feature any custom reporting components.)

Designing K2 Applications

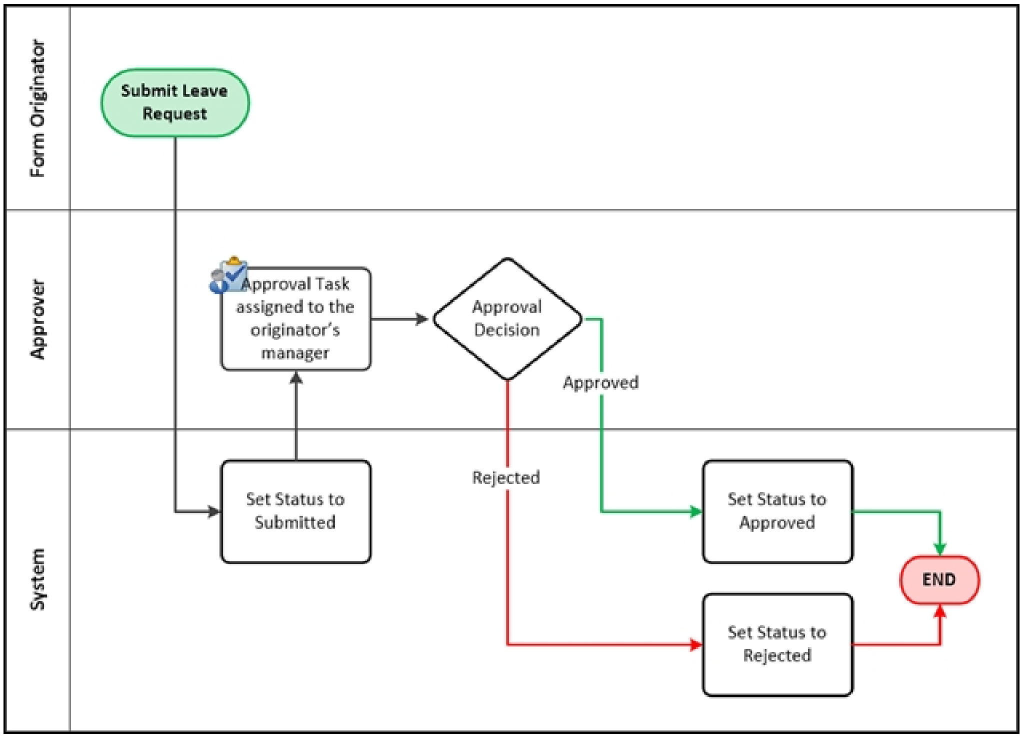
Before starting to build an application, it is important to perform some preliminary design to decide exactly *what* we need our application to accomplish. If your application has a workflow component, a good place to start is to create a flowchart to determine what tasks need to take place and who (or what) will be performing them. Following that, design the Forms (User Interfaces) to determine what data is needed for the users to complete their work. Finally, design the Data elements for the application based on what you discovered during the Workflow and Forms design stages.

Leave Request Workflow

The key to a successful workflow is to completely understand the steps of the process. What are the steps necessary to accomplish our workflow goal? Who (or what) will be performing these steps and what happens during the step?

A very good approach for this is to map the workflow as a flowchart on paper or using any number of software tools. The flowchart below was created with Visio, but you can use any method, even paper and pencil. What is important is to determine the steps in the workflow, how those steps are connected and what happens during those steps.

Below is an example of a workflow design for the basic version of the Leave Request Approval application. This flowchart represents a swim-lane format, where we have separated the User Tasks and System Tasks into separate lanes.



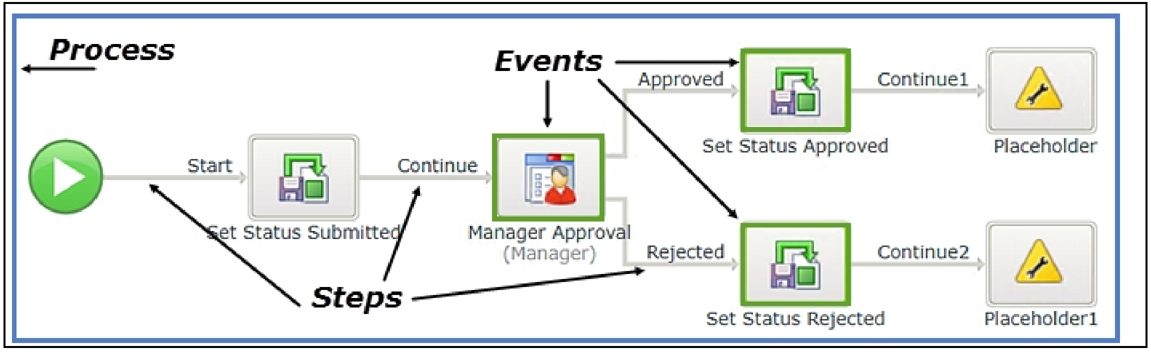
There are two swim-lanes for User Tasks. The first is the Form Originator, or the person submitting the form for the first time. The second swim-lane represents the Approver lane, in our case this will be the form originator's manager. The third lane represents System Tasks, or tasks that K2 will manage behind-the-scenes.

Notice that along the way, the workflow may move from one lane to another lane. K2 seamlessly interacts with both human and system components.

K2 Workflow concepts: Process, Workflow, Steps and Events

When working with K2 workflows, a basic understanding of some terms and definitions will be valuable. A **Process** or **Workflow** is a container for all of the steps that make up a workflow. These **Steps** contain **Events** that define the actual work to be performed, which in turn might be performed by the K2 Server (System steps) or a human (Client steps).

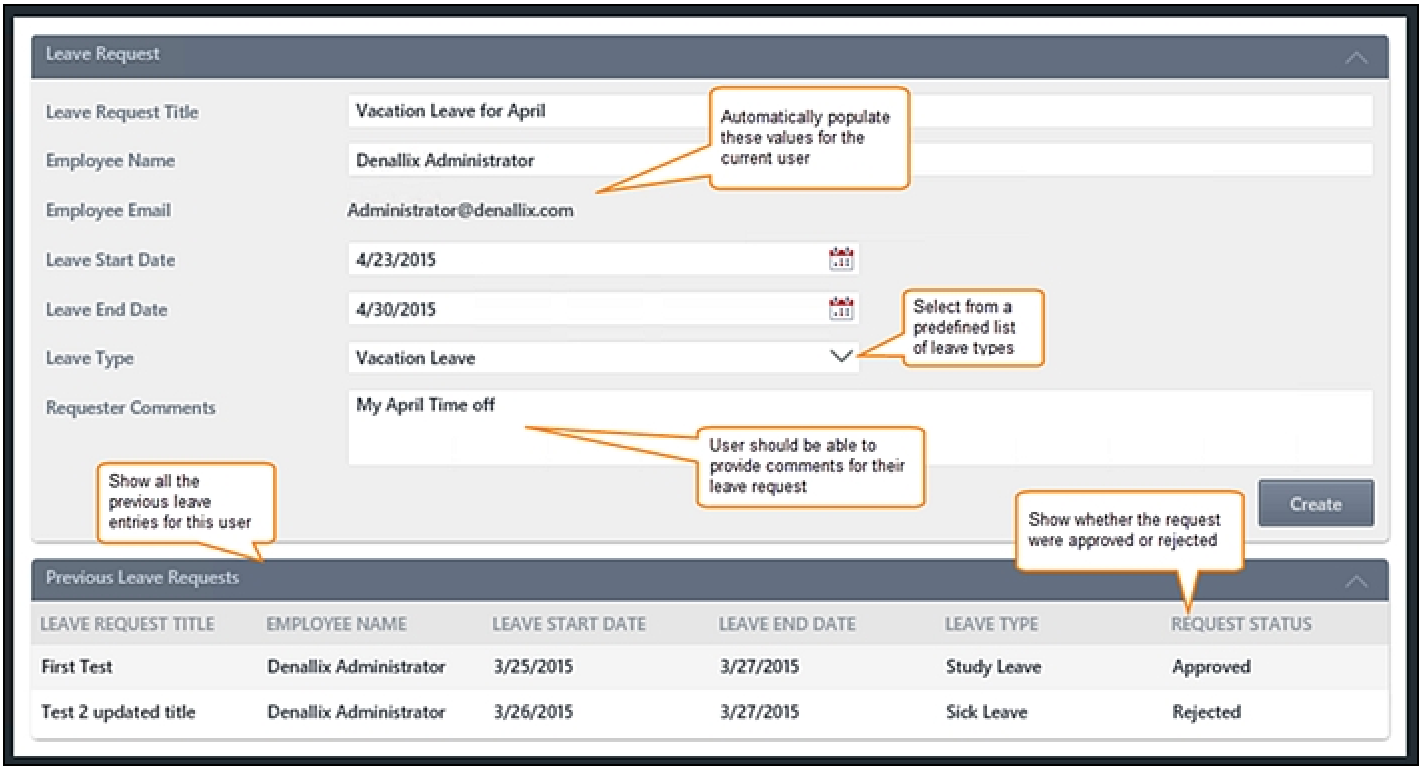
The diagram below shows the eventual implementation of the workflow design, along with the specific K2 terms highlighted.



Leave Request Forms

Once you have the workflow design, the next step is to design the Forms ("user interfaces") that will be used in each step where users will interact with the workflow. In this application, there are two user steps: the first is when the workflow is Started, the second is when the manager needs to approve the leave request.

As with designing the workflow, you can use any tool you like to design the Forms, even pen and paper. The most important part is to determine what values/fields should show on the Form to allow users to complete their tasks, and whether there are any particular validation (e.g. what fields may be required) and behavioral (e.g. what happens when the user clicks a button) requirements. The image below is a sample of what a screen design might look like.



Leave Request

Leave Request Title	Vacation Leave for April
Employee Name	Denallix Administrator
Employee Email	Administrator@denallix.com
Leave Start Date	4/23/2015
Leave End Date	4/30/2015
Leave Type	Vacation Leave
Requester Comments	My April Time off

Automatically populate these values for the current user

Select from a predefined list of leave types

User should be able to provide comments for their leave request

Show all the previous leave entries for this user

Show whether the request were approved or rejected

Previous Leave Requests

LEAVE REQUEST TITLE	EMPLOYEE NAME	LEAVE START DATE	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
First Test	Denallix Administrator	3/25/2015	3/27/2015	Study Leave	Approved
Test 2 updated title	Denallix Administrator	3/26/2015	3/27/2015	Sick Leave	Rejected

SmartForms Concepts: Forms, Views, Controls, Rules and States

For this simple application we will use two K2 smartforms. SmartForms are browser-based forms, allowing the user to access and submit the form without any additional client-side software installed on their computer. You don't HAVE to use SmartForms for K2 applications: you are free to use a number of different technologies. SmartForms however, provide easy and seamless integration between Data, Forms and Workflows and are the recommended approach for building applications quickly. To better understand how SmartForms are used, it is important to understand a few basic concepts.

Forms

A form is the main container for all of the form related elements. Typically, a form will contain one or more views and controls. Forms can also have rules applied to them. A form is the "web page" that is exposed to users.

Views

Views are parts of a form broken down into logical sections. There are two types of views: **Item Views** and **List Views**. Item Views contain the content from *one* record. List Views contain the content from *multiple* records, often displayed as rows of data.

Controls

Controls are all of the bits and pieces that make up views and forms. They include form fields, labels, (submit) buttons, images, etc.

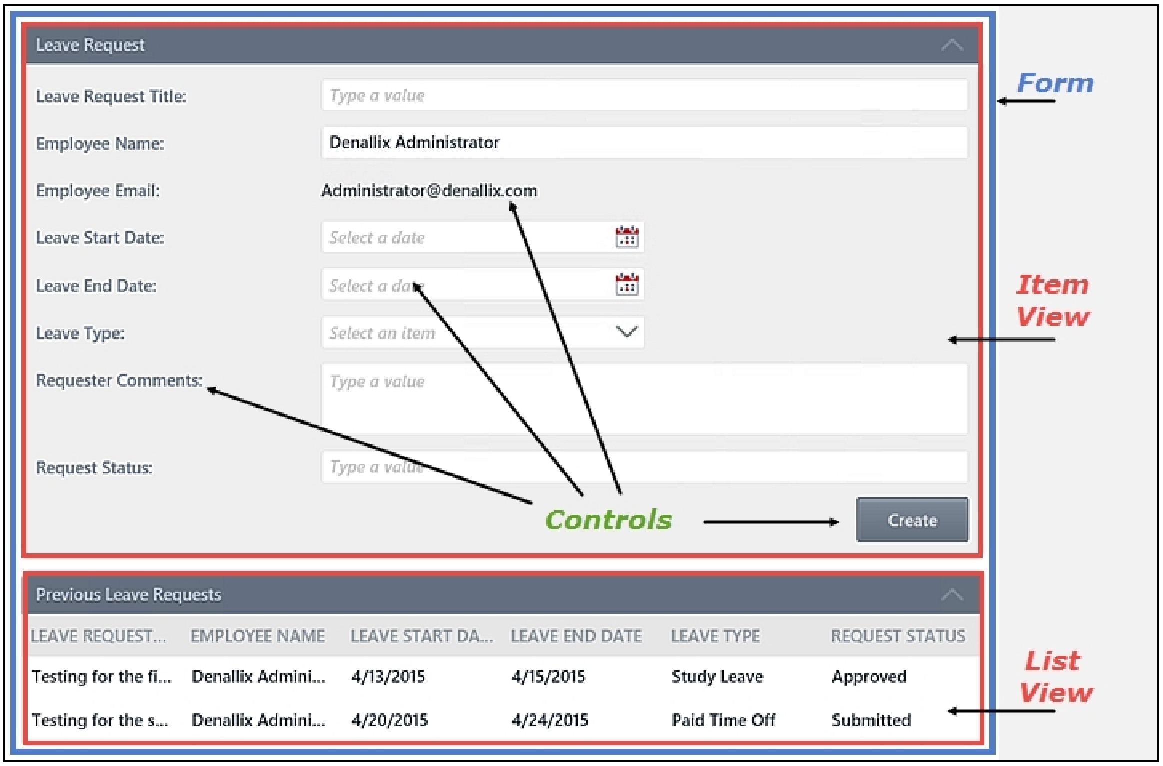
Rules

Rules contain **Events**, **Conditions** and **Actions**. An event is *when* something occurs (such as a button clicked). A condition checks to see *if some criteria has been met* (such as required form fields) and an action says *do something* if the event and conditions are met (such as start the workflow). Rules provide dynamic functionality to your form.

States

A State is just a value that determines how the Form should behave. For example, a Form may have a Read-Only state where rules will disable the Form for any input. Or it may have a "Workflow Task" state which indicates that the Form is being used to complete a workflow task and therefore needs some workflow integration actions.

The image below explains some of these SmartForms concepts in terms of the Leave Request form:



Leave Request

Leave Request Title:	Type a value
Employee Name:	Denallix Administrator
Employee Email:	Administrator@denallix.com
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Select an item
Requester Comments:	Type a value
Request Status:	Type a value

Previous Leave Requests

LEAVE REQUEST...	EMPLOYEE NAME	LEAVE START DA...	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
Testing for the fi...	Denallix Admini...	4/13/2015	4/15/2015	Study Leave	Approved
Testing for the s...	Denallix Admini...	4/20/2015	4/24/2015	Paid Time Off	Submitted

Leave Request Data

Once we have our Workflow and Forms laid out and know what data we need to collect, we can plan our data sources. Some questions to answer during this design stage are whether there is an existing system that can store the data, or whether we need to create a new data store. Are there form fields that can be populated from data sources already in existence, for example drop-down lists or search boxes? In this particular application we will assume there is no existing data source that can store Leave Request details, so we will be creating our own data storage area from scratch. To keep things simple, we will leverage K2's built-in storage area called "SmartBox" so that we do not have to create the underlying database ourselves.

When designing the Data sources, try to identify the logical "objects" in the application, and then define the properties for these objects. In this sample application there is only one logical "object", the Leave Request itself. We used a table like this to describe the properties of this Leave Request object:

Leave Request Data Design

Name	Data Type	Notes
Leave Request Title	Text	Allows the user to enter a unique title for their leave request
Employee Name	Text	The name of the employee who submitted the leave request
Employee Email	Text	The email of the employee who submitted the leave request
Leave Start Date	Date	The date on which the leave is due to start
Leave End Date	Date	The date on which the leave is due to end
Leave Type	Text	The type of leave. To keep the data clean, we want users to select from a predefined list of leave types.
Requester Comments	Long Text	Allow the requester to add comments about their leave request
Request Status	Text	The workflow will update the status of the leave request as the request progresses (e.g. Submitted-Approved-Rejected)

Building the application

Now that you have designs for each of the elements of the application, you can start building them. In most cases, building the application goes in the reverse order from designing it: start with the Data elements, then do the Forms elements and finally the Workflow elements.

When you are ready to start building the Leave Request Approval application, continue on to the Building the Leave Request Application (V1 or "Basic" Version) section.

Building the Leave Request Approval Application (V1 or "Basic" Version)

This document contains the step-by-step instructions to build Version 1 (or the "basic" version) of the Leave Request Approval Application. If required, please refer to the Application Design topic for an overview of the Application.

For clarity, this tutorial is divided into four parts: in Part 1, you are introduced to the data component by creating a SmartBox SmartObject to store your submitted form content. In Part 2, you concentrate on the forms component by creating the views and form necessary for your Application. In Part 3, you focus on the workflow component by building a simple approval workflow and then editing the forms for the additional workflow integration tweaks. In Part 4, you will test your application.

Part 1: Data

Most applications require some kind of Data storage. In K2, this data storage usually refers to K2 SmartObjects. Part 1 of this tutorial will introduce you to K2 SmartObjects by creating a SmartBox SmartObject. SmartBox is K2-provided storage where K2 creates a dedicated table in the K2 database on SQL Server. For this scenario you will create a Leave Request SmartObject which will store the leave request information entered by the user.

This part should take around 15-20 minutes to complete.

> **Note**
> If you are using a virtual environment provided by K2, you may log into your environment using the following credentials:
> User Name: Denallix\Administrator
> Password: K2pass!

Step 1: Prepare K2 Designer by adding new categories

The first step for creating your Leave Request Approval Application is to add categories in K2 Designer that will house your forms, views and workflow. Categories can be thought of as folders. While this step is not required, it is a good practice to keep your K2 artifacts organized, especially as you begin to build larger applications.

Note
If you are using a virtual environment provided by K2, you may log into your environment using the following credentials:
User Name: Denallix\Administrator
Password: K2pass!

Step 1 Tasks

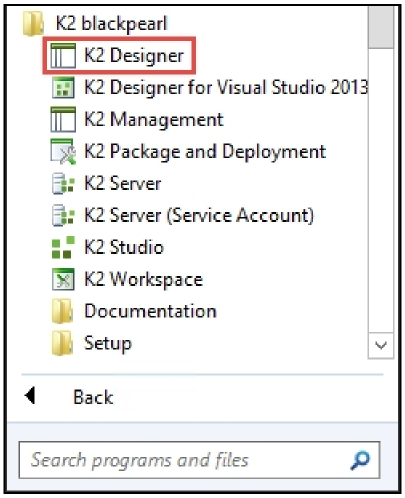
1. In K2 Designer, create a new category under **All Items** and name it
   K2 Learning

   Note
   If you are using an environment that is shared with other users, you may want to add your name to the folder name, so that your application does not conflict with another user's applications.

2. Create a new category under **K2 Learning** and name it
   Leave Request
3. Create three new categories under **Leave Request** and name them
   Forms
   SmartObjects
   Views

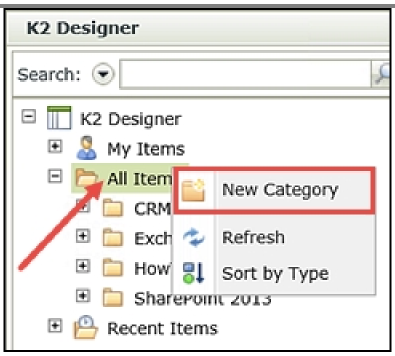
Step 1 Walkthrough

a. Launch K2 Designer (**Start** > **All Programs** > **K2 blackpearl** > **K2 Designer**).
(If the shortcut is not on your desktop or if you are not using a K2-provided virtual machine for this tutorial, check with your K2 administrator what the web address (URL) is for your K2 Designer environment, then open the site using a browser like Internet Explorer, Chrome or Firefox.)

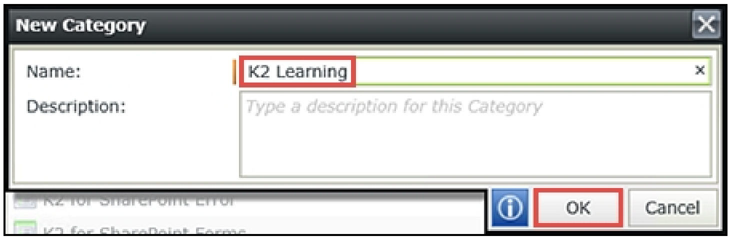

K2 blackpearl
K2 Designer
K2 Designer for Visual Studio 2013
K2 Management
K2 Package and Deployment
K2 Server
K2 Server (Service Account)
K2 Studio
K2 Workspace
Documentation
Setup
Back
Search programs and files

Note
On the left side of the screen is the Category Browser. This is where your K2 artifacts (forms, views, workflows, etc.) will be stored. The **All Items** category is the primary category for storing applications that you will build. First you are going to create a main category for your applications called *K2 Learning*. As you work through K2 tutorials, you can create categories under the K2 Learning category to keep your files organized. Once again, note that creating categories is not a requirement to building K2 applications. It does however, keep your K2 artifacts organized as you work through the tutorials.

b. Right-click All Items and select **New Category**.



c. Name the new category
K2 Learning
then click **OK**.



> **Note**
> If you are using an environment that is shared with other users, you may want to add your name to the folder name, so that your application does not conflict with another user's applications.

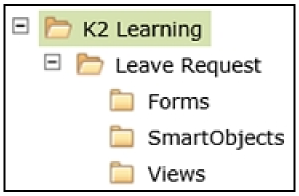
Now you will create the categories specific to your Leave Request Basic application. Using the same steps as above, you'll add folders that will store the Forms, Views and SmartObjects for your application.

d. Right-click the **K2 Learning** category and select **New Category**. Name the new category
Leave Request
then click **OK**.

e. Right-click the **Leave Request** category and select **New Category**. Name the new category
Forms
then click **OK**. This category will house the forms for your application.

f. Repeat the step above for two more categories and name them
SmartObjects
Views
These categories will house your views and SmartObject.

Your Category Browser should look like the image below.



STEP 1 REVIEW

In this step, you prepared the K2 Designer environment by creating categories (or folders) for your application (Leave Request) and application artifacts (Forms, SmartObjects, Views). You created a main category (K2 Learning) that you can use for future tutorials and for practice applications.

Step 2: Create a SmartBox SmartObject to store submitted leave request records

In this step you will create the single SmartObject required for the basic version of your application. This will be a SmartBox SmartObject and you will manually define the properties ("columns" or "fields") to store the Leave Request details. K2 provides default methods (Create, Get List, Save) for creating the new record when a Leave Request is submitted, listing previous Leave Requests and updating the record's status property. You will use these methods in later steps when configuring the Forms for the application.

Step 2 Tasks

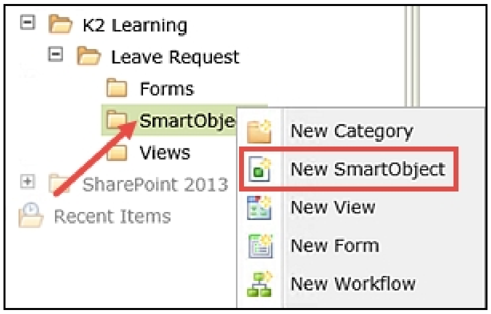
1. Create a New SmartBox SmartObject under the SmartObjects category and name it
   Leave Request SmartObject
   and allow it to be used in workflows.
2. Add the following properties with their corresponding data types to the SmartObject.

Name	Type	Notes
ID	Autonumber	This field should be added already by default
Leave Request Title	Text	
Employee Name	Text	
Employee Email	Text	
Leave Start Date	Date	
Leave End Date	Date	
Leave Type	Text	
Requester Comments	Memo	
Request Status	Text	This field will be updated by the workflow as the leave request is submitted and then approved or rejected

3. Finish the wizard to publish the SmartObject.

Step 2 Walkthrough

a. Right-click the **SmartObjects** category and select **New SmartObject**.

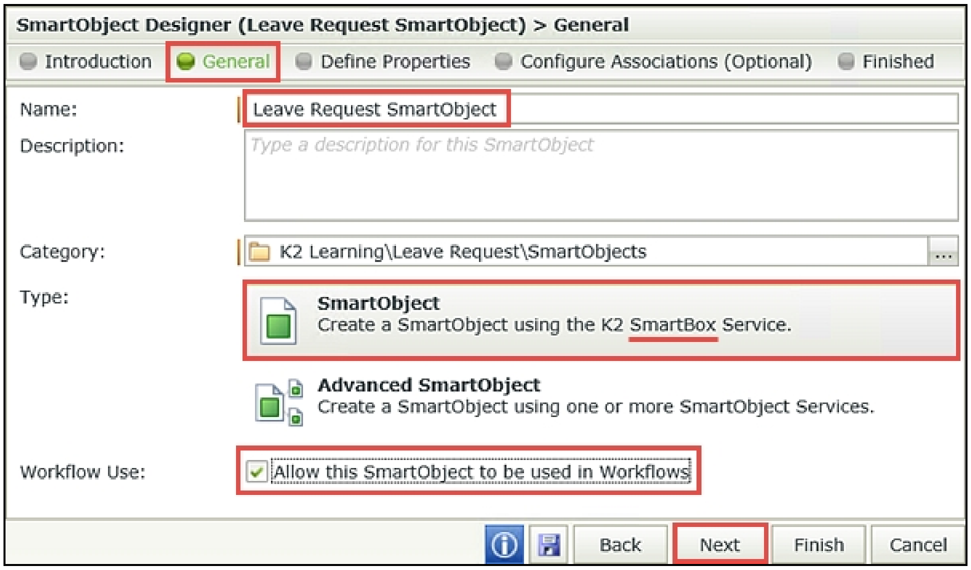


b. Navigate to the **General** settings screen. (If you are on the SmartObjects landing screen, click **Next** in the lower right corner of the screen.) On the General settings screen, name the SmartObject
Leave Request SmartObject
and confirm the SmartObject Type is set to the default **SmartBox**. CHECK the option to **Allow this SmartObject to be used in Workflows**. The Category setting is fine as it is. Click **Next** when ready.

Tip
You don't need to adjust the category because you started from the SmartObjects category to create this SmartObject. You can create SmartObjects in a number of ways, then move them into the SmartObjects category from this General settings screen.

Note
When you select 'Allow this SmartObject to be used in Workflows', you are telling K2 to make the SmartObject properties and methods available as options in the K2 Workflow Designer. As you build your own applications, you may find that your SmartObjects do not appear as options in the K2 Workflow Designer. Simply edit the SmartObject in question from K2 Designer, and from this General settings screen, check the 'Allow'

option, then click Finish to save the SmartObject edit. In Part 3 of this tutorial, you will observe the SmartObject options as you work through building your Leave Request Workflow.



SmartObject Designer (Leave Request SmartObject) > General

Name: Leave Request SmartObject

Description: Type a description for this SmartObject

Category: K2 Learning\Leave Request\SmartObjects

Type:

SmartObject
Create a SmartObject using the K2 SmartBox Service.

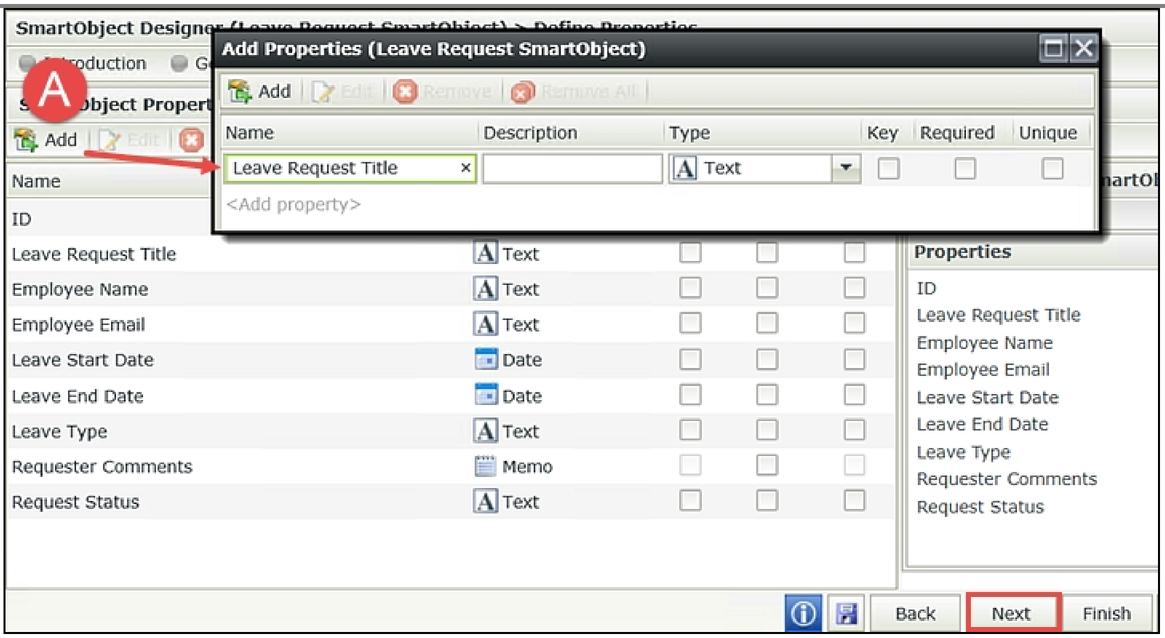
Advanced SmartObject
Create a SmartObject using one or more SmartObject Services.

Workflow Use: ☑ Allow this SmartObject to be used in Workflows

You will now be directed to the Define Properties screen. This is where you will add the properties (or columns/fields) that you need for your application to your SmartBox SmartObject. Notice that K2 has automatically added the ID property, which is the key identifier for each SmartObject record. In the next few steps, you will be adding the remaining properties that you need for your application to your Leave Request SmartObject.

c. Begin by clicking **Add**. (Refer to **A** in the image below the table.) Enter the property Name and data Type using the table entries below as a guide. Click the **<Add property>** link to continue adding the properties until you have entered all eight. Click **OK**, then **Next** to continue.

Name	Type	Notes
Leave Request Title	Text	
Employee Name	Text	
Employee Email	Text	
Leave Start Date	Date	
Leave End Date	Date	
Leave Type	Text	
Requester Comments	Memo	
Request Status	Text	This field will be updated by the workflow as the leave request is submitted and then approved or rejected


SmartObject Designer (Leave Request SmartObject) > Define Properties
Add Properties (Leave Request SmartObject)
A
Add
Edit
Remove
Remove All
Back
Next
Finish

Name	Description	Type	Key	Required	Unique
Leave Request Title		Text	□	□	□
<Add property>					

Name	Type			
ID				
Leave Request Title	Text	□	□	□
Employee Name	Text	□	□	□
Employee Email	Text	□	□	□
Leave Start Date	Date	□	□	□
Leave End Date	Date	□	□	□
Leave Type	Text	□	□	□
Requester Comments	Memo	□	□	□
Request Status	Text	□	□	□

Properties

- ID
- Leave Request Title
- Employee Name
- Employee Email
- Leave Start Date
- Leave End Date
- Leave Type
- Requester Comments
- Request Status

d. On the Configure Associations screen, click **Finish** to publish the new SmartObject to the K2 server.

STEP 2 REVIEW

You have just created your first K2 SmartBox SmartObject! Taking a closer look at the SmartObject properties, you might recognize them as the future form fields you will use for your Leave Request Form (with the exception of the ID property, which is hidden). When a user submits a Leave Request, the form content will be saved to this SmartBox SmartObject. As the workflow progresses, the Request Status property will be updated with the appropriate status value.

Part 2: Forms

For your Leave Request Approval application you will be using K2 smartforms as the user interfaces to capture the request details and to approve the request. With your Data objects created in Part 1, you are ready to start on the Forms component of your application. The first step is to create an Item View.

Step 3: Create an item view for employee details

Now that you have your SmartBox SmartObject created, you can select to design a view based off of that SmartObject. K2 will create the view and assign the SmartObject properties as controls in the view, which you can then move around and edit as needed. The first view you will create is an Item View, or a view that contains the content for a single record. This view will become your Leave Request entry form and when submitted, will create one record in the Leave Request SmartObject.

Step 3 Tasks

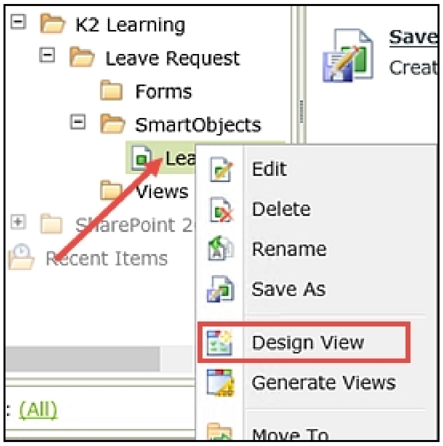
1. Design a new View based on the **Leave Request SmartObject** and name it
   Leave Request Item View
   then move it to the **Views** category.
2. **Create Labels and Controls** and include all of the fields except the ID. Make the **Employee Email** field Display Only. Confirm/set 2 columns for the layout table. Change the Label position to **Left** and add the **Colon** suffix. CHECK the option for the **Create** button (only).
3. Change the width of the first column so that it is in closer alignment to the view labels.
4. Change the **Width** of the Leave Start Date and Leave End Date fields to
   50%
5. Change the Leave Type Text Box control to a **Drop-Down List** and manually add the following entries (for both Value and Display):
   Paid Time Off
   Study Leave

Family Responsibility
then change the **Width** of the Leave Type Drop-Down List to
50%

6. Move the **Create** button to the second cell and align it to the right.

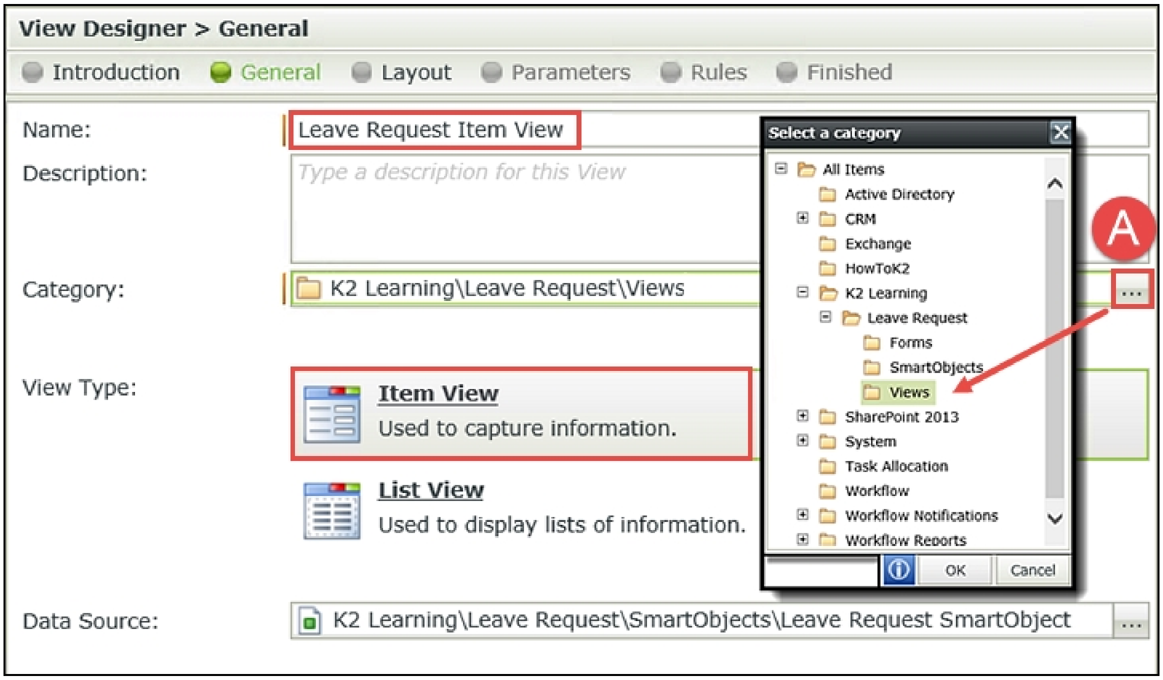
Step 3 Walkthrough

a. Right-click the **Leave Request** SmartObject (located in the SmartObjects category) and select **Design View**. In this step, you are designing a view directly from the Leave Request SmartObject. You don't *have* to create views in this manner, but you can save several steps by letting K2 do all of the basic work. K2 will create the view and automatically add the controls (which are bound to the SmartObject properties you added in the previous step).



b. If on the Views Introduction screen, click **Next**.

c. On the **General** settings screen, name the new view
Leave Request Item View
then change the category to the Views category. (**A** below) Confirm **Item View** is selected and click **Next**. You do not need to make any edits to the Data Source, because you designed this view directly from the Leave Request SmartObject (the data source) itself.


View Designer > General

Name: Leave Request Item View

Description: Type a description for this View

Category: K2 Learning\Leave Request\Views

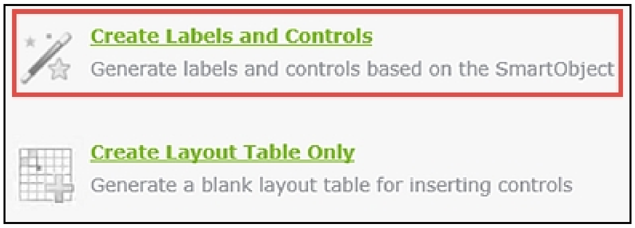
View Type: **Item View** Used to capture information. / **List View** Used to display lists of information.

Data Source: K2 Learning\Leave Request\SmartObjects\Leave Request SmartObject

d. On the Layout design screen, click **Create Labels and Controls**.

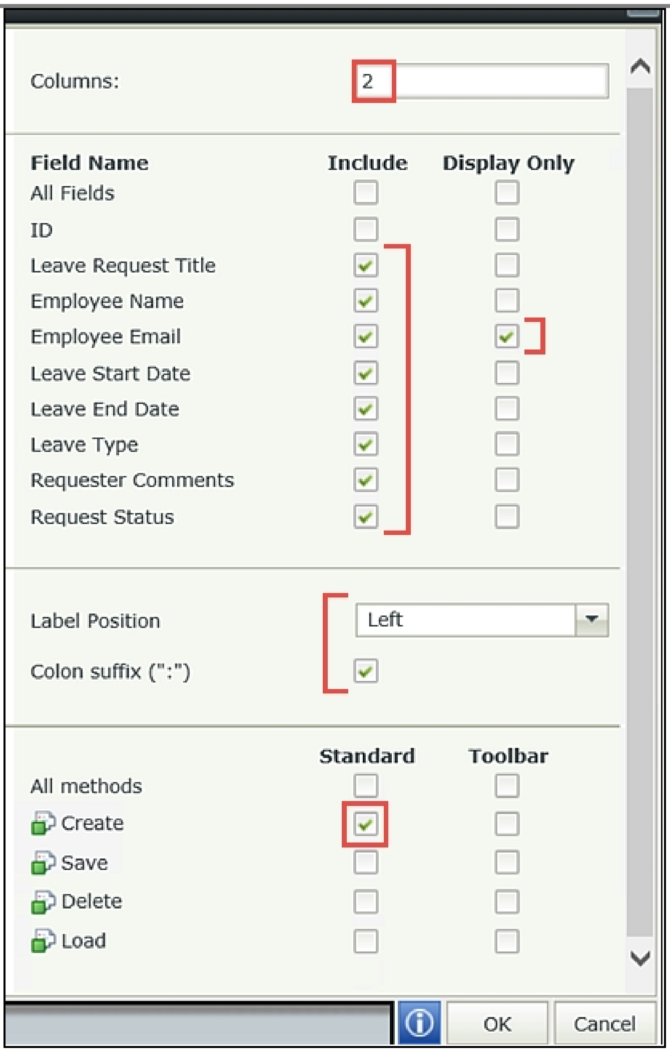
Note

When you select **Create Labels and Controls**, K2 will give us an option to select which fields (SmartObject properties) you want on your view, plus a few other settings for the layout table. (The **Create Layout Table Only** option creates a blank table where you can drag and drop controls yourself. For this exercise you will use the **Create Labels and Controls** option.)


Create Labels and Controls
Generate labels and controls based on the SmartObject
Create Layout Table Only
Generate a blank layout table for inserting controls

e. On the **Labels and Controls** settings screen, make the following configurations:
   - Keep the number of columns set to **2**
     - The first column will house your control labels and the second column will house the controls
   - **Include** all of the fields, except the ID
     - This is a hidden field that will contain the SmartBox SmartObject record ID. You do not need it to be a part of your user interface.
   - Select the **Display Only** option for the Employee Email
     - We don't want the user to be able to change the email value.
   - Change the Label Position to **Left**
   - CHECK the box to add the **Colon suffix**
   - CHECK the box to add the **Standard Create** Button
     - Once again, you are allowing K2 to perform some of your legwork. You could manually add a button control to your view and configure it to create a new record in the SmartBox SmartObject. You will have K2 generate the Create button automatically and add the necessary rules that will create the record for us.

   Click **OK** when ready.



Columns: 2

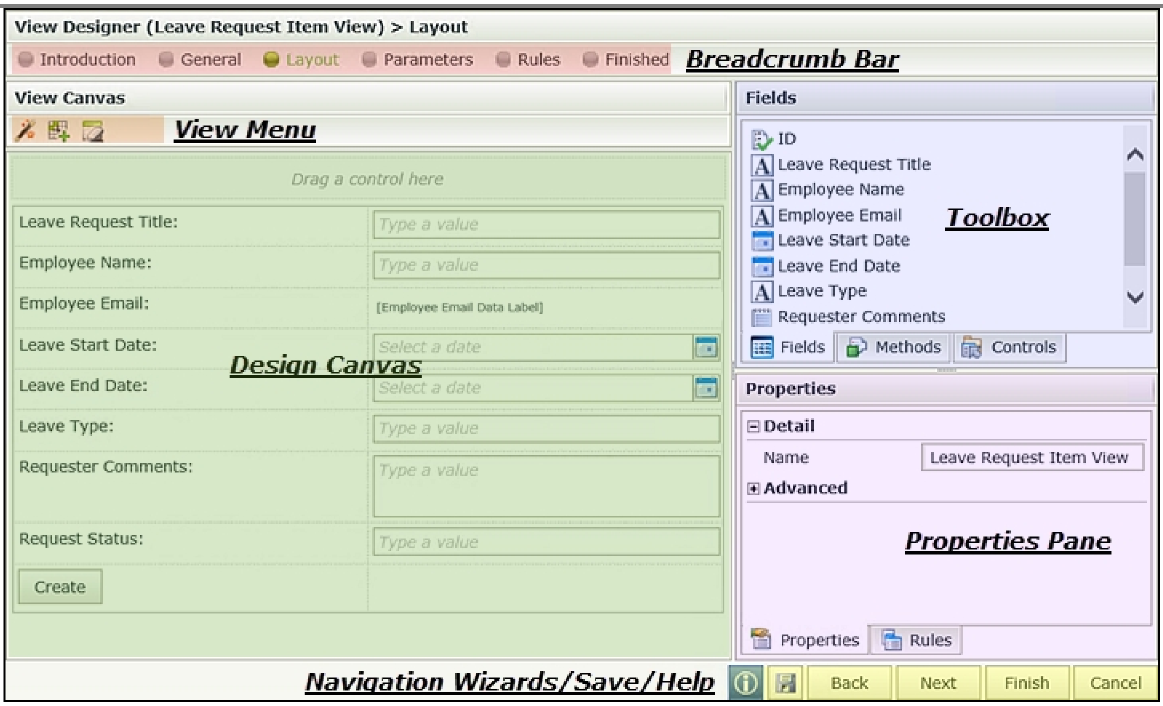
Field Name	Include	Display Only
All Fields	☐	☐
ID	☐	☐
Leave Request Title	☑	☐
Employee Name	☑	☐
Employee Email	☑	☑
Leave Start Date	☑	☐
Leave End Date	☑	☐
Leave Type	☑	☐
Requester Comments	☑	☐
Request Status	☑	☐

Label Position: Left

Colon suffix (":") ☑

	Standard	Toolbar
All methods	☐	☐
Create	☑	☐
Save	☐	☐
Delete	☐	☐
Load	☐	☐

Your new View should look similar to the image below. You've highlighted the sections you will use the most (Breadcrumb Bar, View Menu, Design Canvas, Toolbox, Properties Pane and Navigation Wizards/Save/Help) to help you with terminology if you are new to K2 Designer. As you move forward with the steps and exercises, you will become more familiar with the Designer layout and terms.

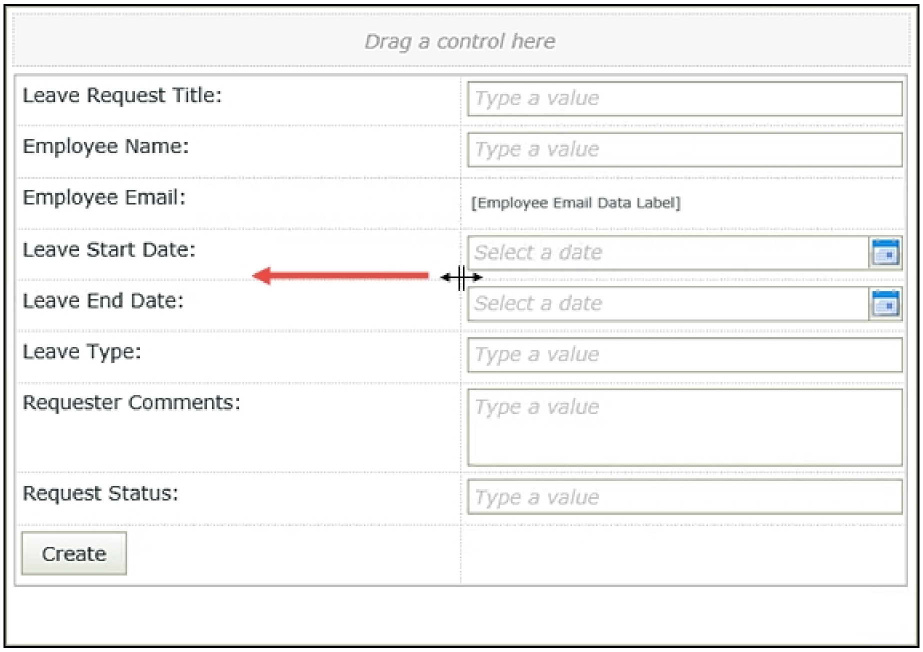


Name: Leave Request Item View

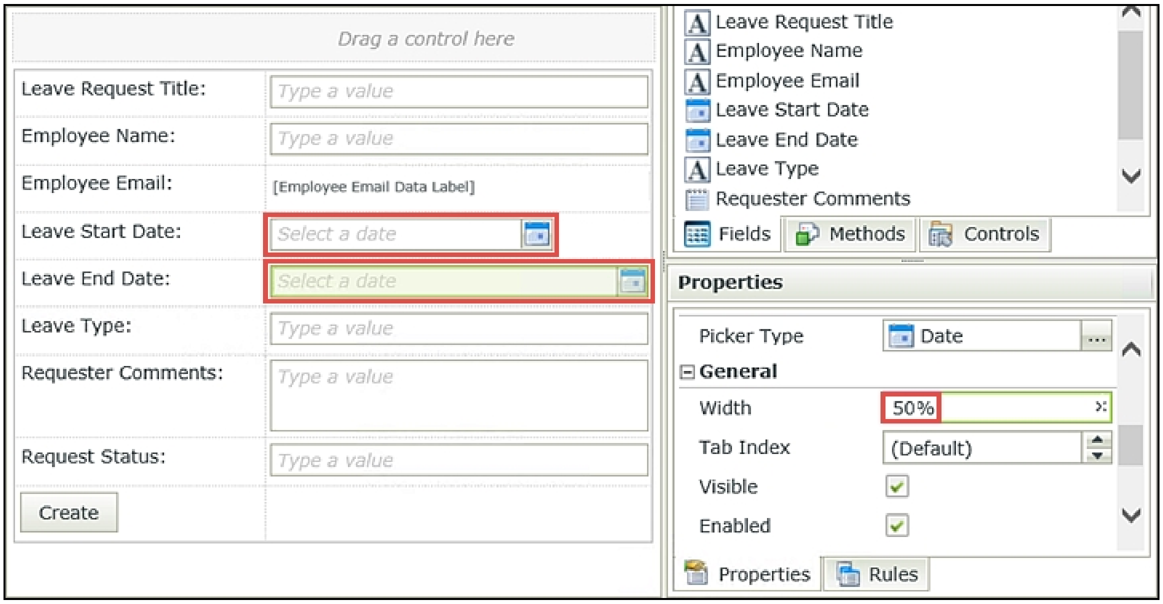
Now that you have the basic view designed, you'll add a few customizations that will make the view a little more user friendly. 99% of view and form design are preference and you will learn what works for your users and what doesn't as you build more applications. First, you'll adjust the width of the label column and second you'll adjust the width of the two date fields.

Note
Notice the **Employee Email Data Label**. Where a label is a static control (it never changes), a data label is a control that cannot be changed by the user, but is dynamic in nature. The Employee Email value will change to reflect the user currently logged in. (This is actually a rule you will set in a later step.) Data labels can also be hidden to perform some type of function within the form or view.

f. Hover over the **center column line** until you see the double lines/arrows. Click+drag the column line so that it is closer to your labels. In other words, reduce the width of the Labels column.



g. Highlight the **Leave Start Date Calendar** box by clicking it. In the **Properties** Pane, change the **Width** to
50%
then repeat this same step for the **Leave End Date Calendar** box.



Drag a control here

Leave Request Title:	Type a value
Employee Name:	Type a value
Employee Email:	[Employee Email Data Label]
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Type a value
Requester Comments:	Type a value
Request Status:	Type a value
Create	

Leave Request Title
Employee Name
Employee Email
Leave Start Date
Leave End Date
Leave Type
Requester Comments

Fields | Methods | Controls

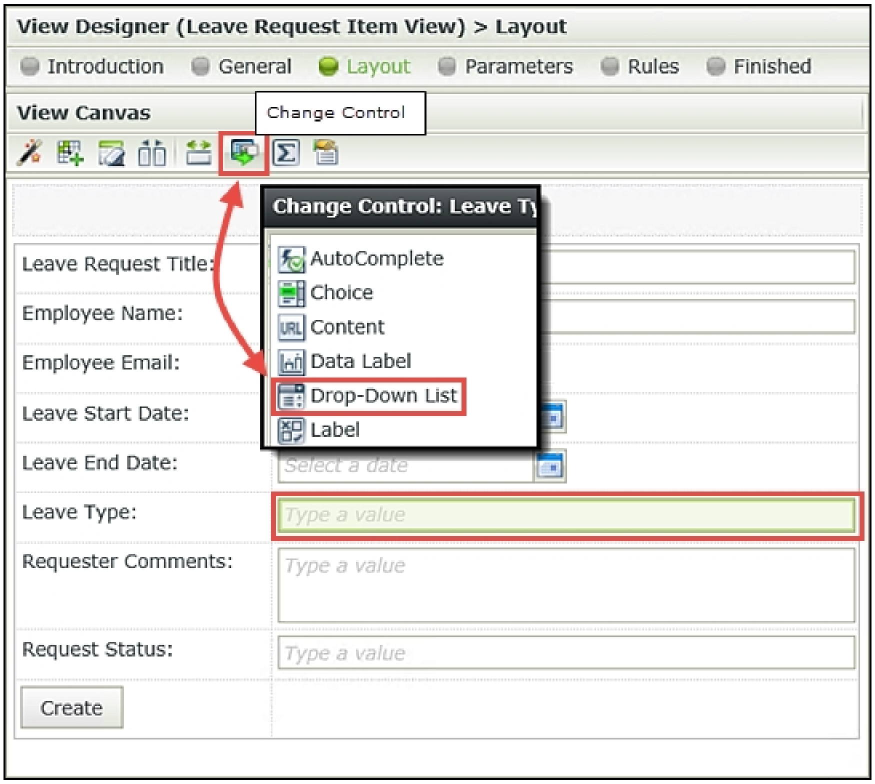
Properties

Picker Type	Date
⊟ **General**	
Width	50%
Tab Index	(Default)
Visible	☑
Enabled	☑

Properties | Rules

Next, you'll change the Leave Type text box to a drop-down list and add some choices for your users to select. This will keep the Leave Types consistent as they are saved to the SmartBox.

h. Highlight the **Leave Type Text** box. Select the **Change Control** icon found in the View Menu. Change the control to a **Drop-Down List**.



View Designer (Leave Request Item View) > Layout

Introduction | General | Layout | Parameters | Rules | Finished

View Canvas

Change Control

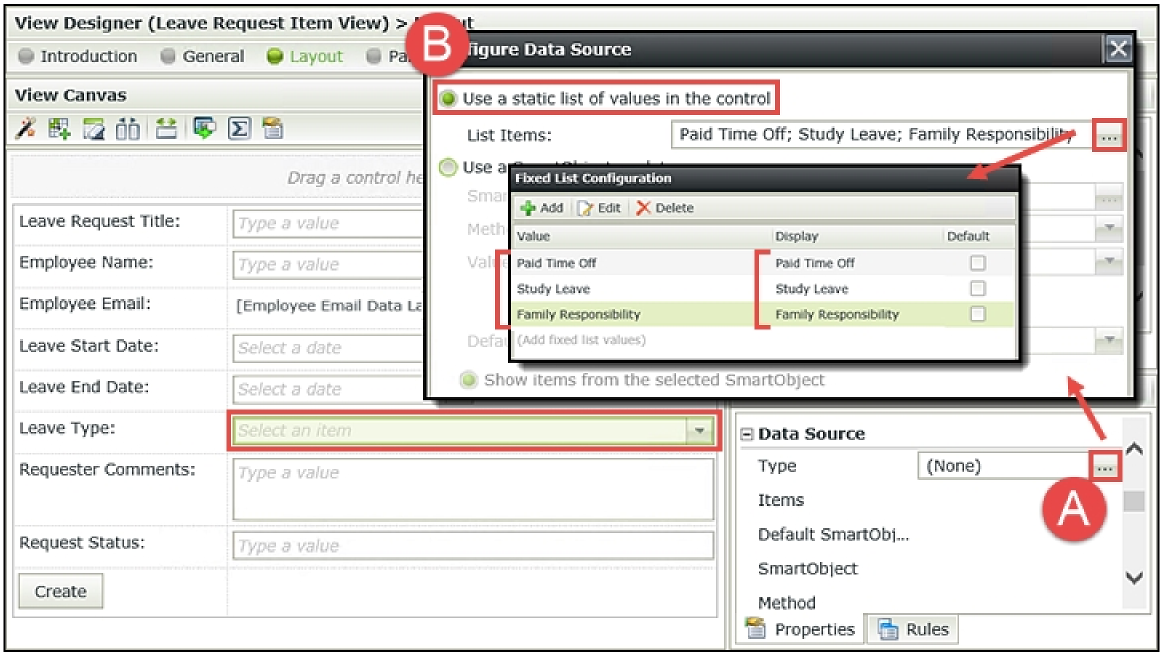
Change Control: Leave Ty

- AutoComplete
- Choice
- Content
- Data Label
- Drop-Down List
- Label

Leave Request Title:	
Employee Name:	
Employee Email:	
Leave Start Date:	
Leave End Date:	Select a date
Leave Type:	Type a value
Requester Comments:	Type a value
Request Status:	Type a value
Create	

i. With the Leave Type control still highlighted, open the Data Source **Type** editor (**A** below) found in the Properties Pane. Select the option to **Use a static list of values in the control**. (**B** below) Open the list editor and **Add** the following options for *both* the Value *and* the Display, then click **OK** when you are done.

- *Paid Time Off*
- *Study Leave*
- *Family Responsibility*



View Designer (Leave Request Item View) > Layout

Introduction | General | Layout | Pa...

View Canvas

Drag a control here

Leave Request Title:	Type a value
Employee Name:	Type a value
Employee Email:	[Employee Email Data La...
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Select an item
Requester Comments:	Type a value
Request Status:	Type a value
Create	

Configure Data Source

● Use a static list of values in the control

List Items: Paid Time Off; Study Leave; Family Responsibility

○ Use a SmartObject...

○ Show items from the selected SmartObject

Fixed List Configuration

Add | Edit | Delete

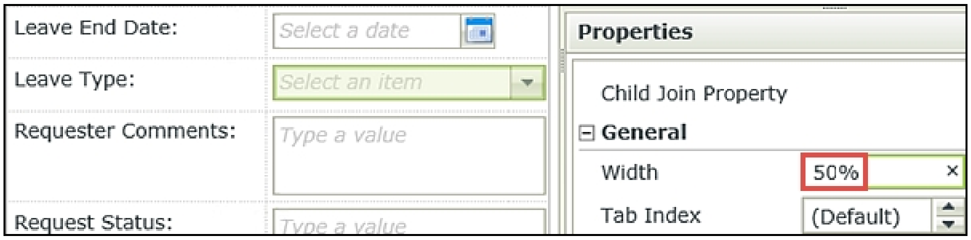
Value	Display	Default
Paid Time Off	Paid Time Off	□
Study Leave	Study Leave	□
Family Responsibility	Family Responsibility	□
(Add fixed list values)		

Data Source

Type	(None)
Items	
Default SmartObj...	
SmartObject	
Method	

Properties | Rules

j. Change the **Width** property of the Leave Type drop-down control to *50%*
(Remember to click and highlight the drop-down control so that you see its properties represented in the Properties pane.)



Leave End Date:	Select a date
Leave Type:	Select an item
Requester Comments:	Type a value
Request Status:	Type a value

Properties

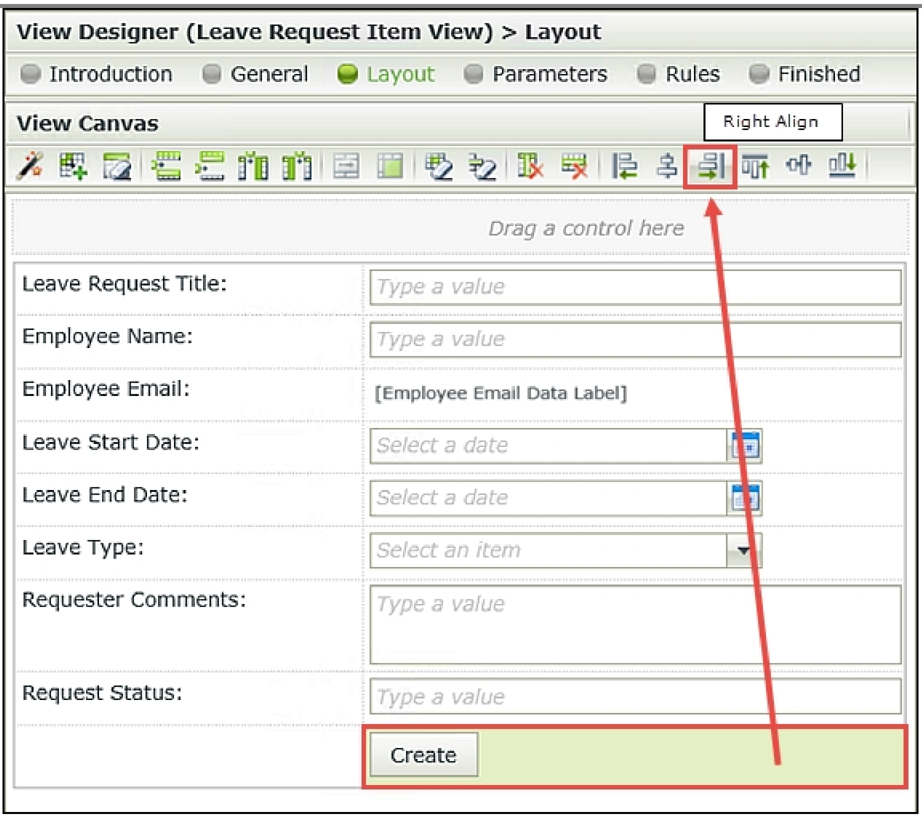
Child Join Property

General

Width	50%
Tab Index	(Default)

You have one last adjustment to make to your view layout. You'll move the **Create** button to the right side of the view, which will make the view flow smoother for your users.

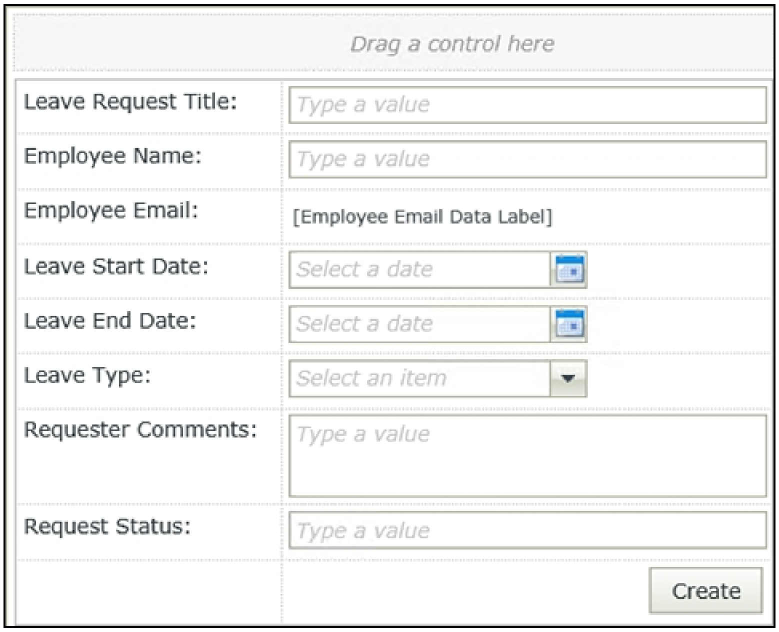
k. Click and drag the **Create** button into the second cell. Click within the cell itself to highlight it, then click the **Right-Align** icon found in the View Menu.



View Designer (Leave Request Item View) > Layout

Leave Request Title:	Type a value
Employee Name:	Type a value
Employee Email:	[Employee Email Data Label]
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Select an item
Requester Comments:	Type a value
Request Status:	Type a value
	Create

Your view should now look like the image below.



Leave Request Title:	Type a value
Employee Name:	Type a value
Employee Email:	[Employee Email Data Label]
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Select an item
Requester Comments:	Type a value
Request Status:	Type a value
	Create

I. Click **Finish** in the Navigation Wizard (lower right corner of the screen) to save and exit from this view.

STEP 3 REVIEW

In this step, you created an Item View based on the Leave Request SmartObject. By designing the view directly from the SmartObject, K2 automatically created the view fields which you were able to include and /or display. You selected the Create button for your View. K2 creates all of the necessary rules and actions so that when the button is clicked, a new SmartBox SmartObject record is created in the associated Leave Request SmartObject. You also applied some minor formatting and you manually entered the values for your Leave Type drop-down list.

Step 4: Create a list view to display previous leave requests

The next view you are going to create will be a List View. List views contain the content from multiple records. The purpose of this view is to output all of the previous leave requests from the current user. This will be especially helpful to the approving manager, to assess how much leave this employee has requested, or for the requesting user to see the status of their prior leave requests.

Step 4 Tasks

1. Design a new **List View** based on the Leave Request SmartObject and name it
   Leave Request List View
   then move it to the **Views** category. Change the View Type to **List View** and disable the **Call this method when the form loads** option.
2. Add the following fields:
   - Leave Request Title
   - Employee Name
   - Leave Start Date
   - Leave End Date
   - Leave Type
   - Request Status

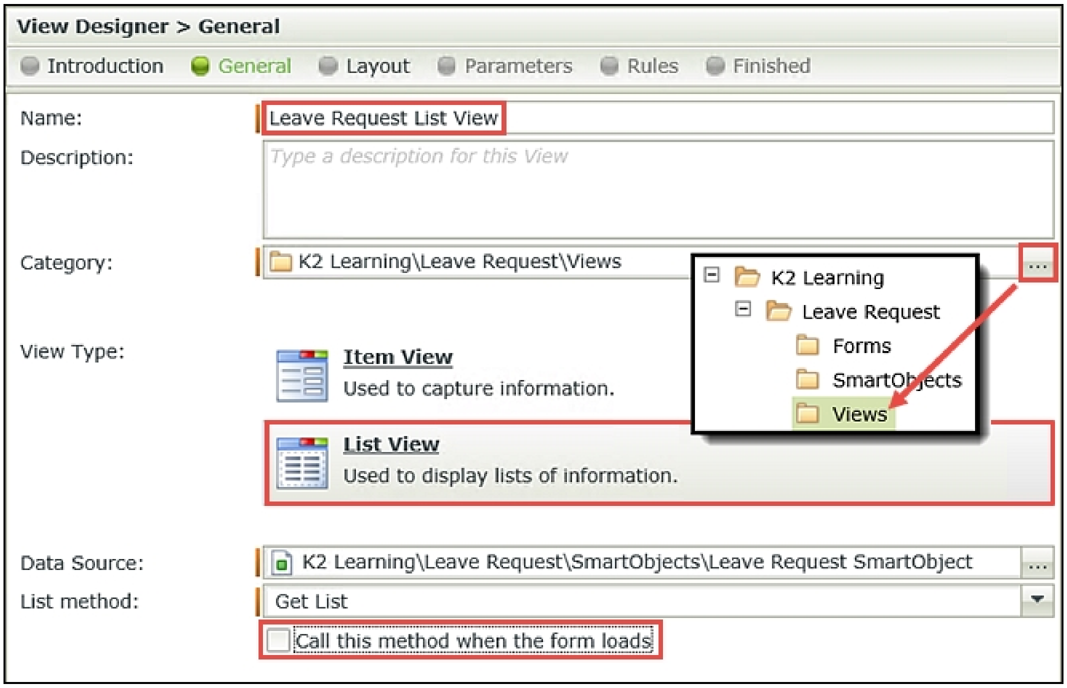
Step 4 Walkthrough

a. Right-click the **Leave Request** SmartObject and select **Design View**. If on the View home screen, click **Next**.

b. Name the new view
   Leave Request List View
   then move it to the **Views** category.

c. Change the View Type to **List View**. UNCHECK the option to **Call this method when the form loads**. Click **Next** when ready.

> Note
> *Call this method when the form loads*. Because you have designed this list view based off of a SmartObject, K2 by default will retrieve all content from the SmartObject when the view is loaded. This is not what you want. You want only the records that pertain to the current user, not *all* of the records. That is why you turn off this option. In a later step, you will configure a rule to retrieve only the records for the current user.



Name: Leave Request List View

Description: Type a description for this View

Category: K2 Learning\Leave Request\Views

View Type:

Item View Used to capture information.

List View Used to display lists of information.

Data Source: K2 Learning\Leave Request\SmartObjects\Leave Request SmartObject

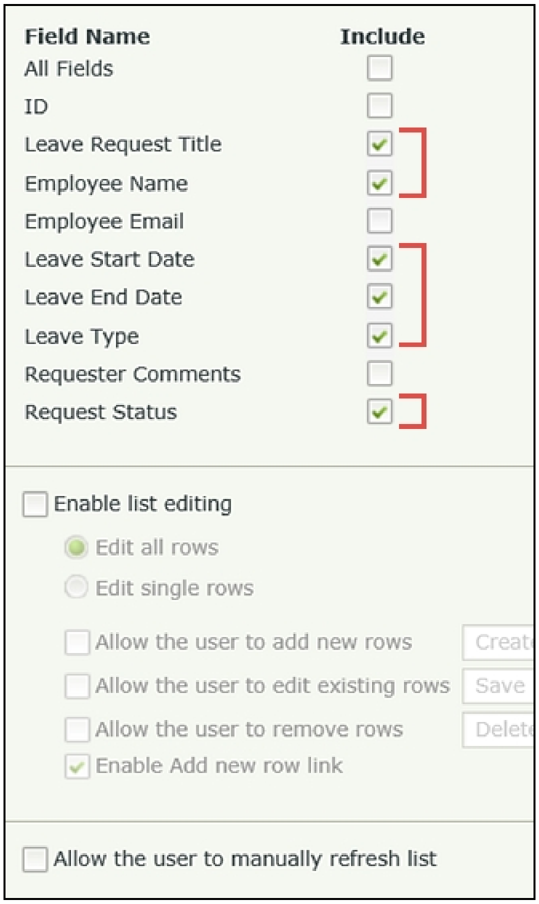
List method: Get List

□ Call this method when the form loads

d. Select the **Create Labels and Controls** option. On the layout screen, include the following fields:

- Leave Request Title
- Employee Name
- Leave Start Date
- Leave End Date
- Leave Type
- Request Status

There are no other changes to the layout editor, so click **OK** when ready. Click **Finish** (Navigation Wizard - lower right corner) to save and exit from this view. Here you are only displaying the current user's records. You don't want the user to make any changes, so you keep the 'Enable list editing' option disabled.



Field Name	Include
All Fields	□
ID	□
Leave Request Title	☑
Employee Name	☑
Employee Email	□
Leave Start Date	☑
Leave End Date	☑
Leave Type	☑
Requester Comments	□
Request Status	☑

□ Enable list editing

● Edit all rows

○ Edit single rows

□ Allow the user to add new rows Creat

□ Allow the user to edit existing rows Save

□ Allow the user to remove rows Delete

☑ Enable Add new row link

□ Allow the user to manually refresh list

STEP 4 REVIEW

In this step you created a basic List View based off of the Leave Request SmartObject. You will use this view to retrieve previous leave request records for the current user. You have disable the list editing functionality (by keeping the default) so that the user cannot change any records that have already been submitted.

Step 5: Create the Leave Request form and add the item view and list view

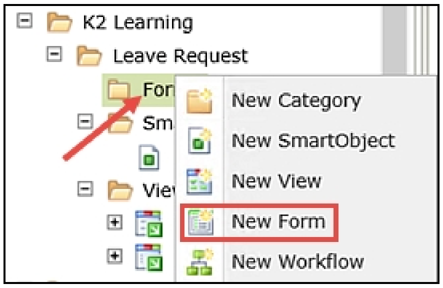
Recall that a form is the main container for views and additional controls. In this next step, you will create a Form and add the two views created in the previous steps. You will then apply a new theme to change the form's look and feel. Oftentimes, forms will contain multiple views. And views can be used in multiple forms. For example, you might create a basic employee details view (Name, Phone, Email, etc.) and use it in multiple forms. This allows you to reuse views instead of creating multiple views containing identical information.

Step 5 Tasks

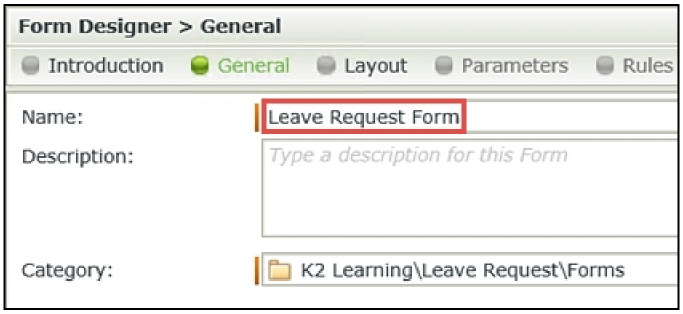
1. Create a **New Form** from the **Forms** category.
2. Add the **Leave Request Item View** and change the view title to *Leave Request*
3. Add the **Leave Request List View** to just below the Item View and change the view title to *Previous Leave Requests*
4. Change the default form Theme to the **Lithium** theme.

Step 5 Walkthrough

a. Right-click the **Forms** category and select **New Form**. If on the Forms home screen, click **Next**.



b. Name the new form *Leave Request Form* then click **Next**. Because you created the form directly from the Forms category, it will created withing that category, so you don't need to move it.


Form Designer > General

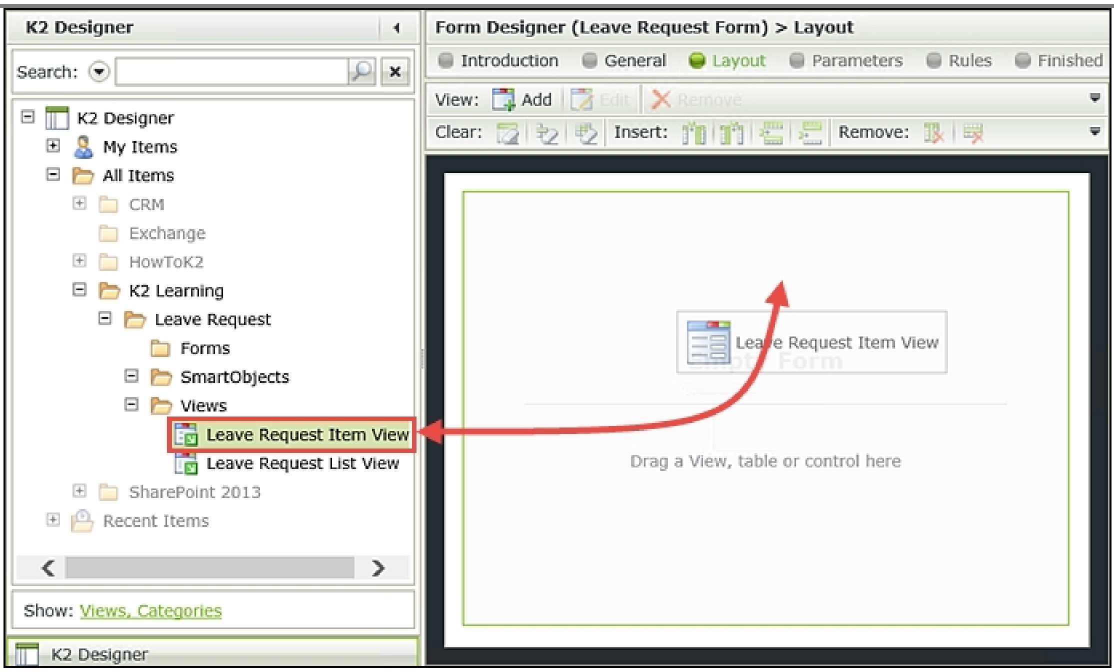
Name: Leave Request Form

Description: Type a description for this Form

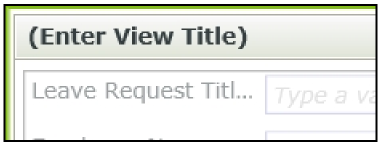
Category: K2 Learning\Leave Request\Forms

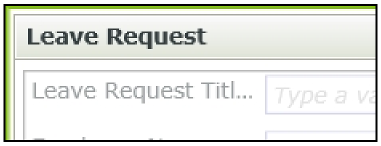
You should now be on the Form Designer screen. The Form Designer has a design canvas and many of the same toolbars and panes as the View Designer.

c. Drag the **Leave Request Item View** onto the form design canvas.

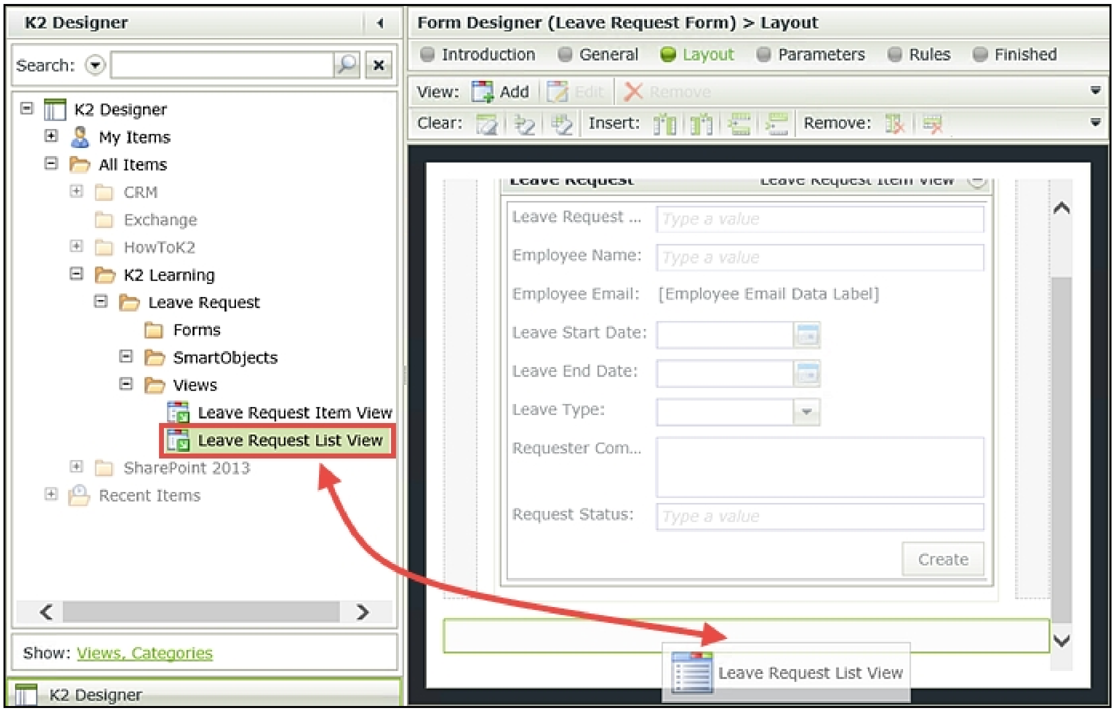


d. Double-click the **(Enter View Title)** view title and change it to *Leave Request*

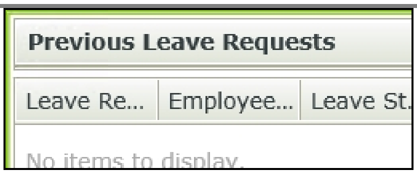




e. Now you'll add the List View to just below the Item View. Drag the **Leave Request List View** to just under the Item View. You will notice that K2 automatically adds a new placeholder as you drag the view onto the canvas. Drop the view into the placeholder.



f. Double-click the **(Enter View Title)** view title and change it to *Previous Leave Requests*
(Remember that this view will contain all of the prior leave requests for the current user.)



Previous Leave Requests

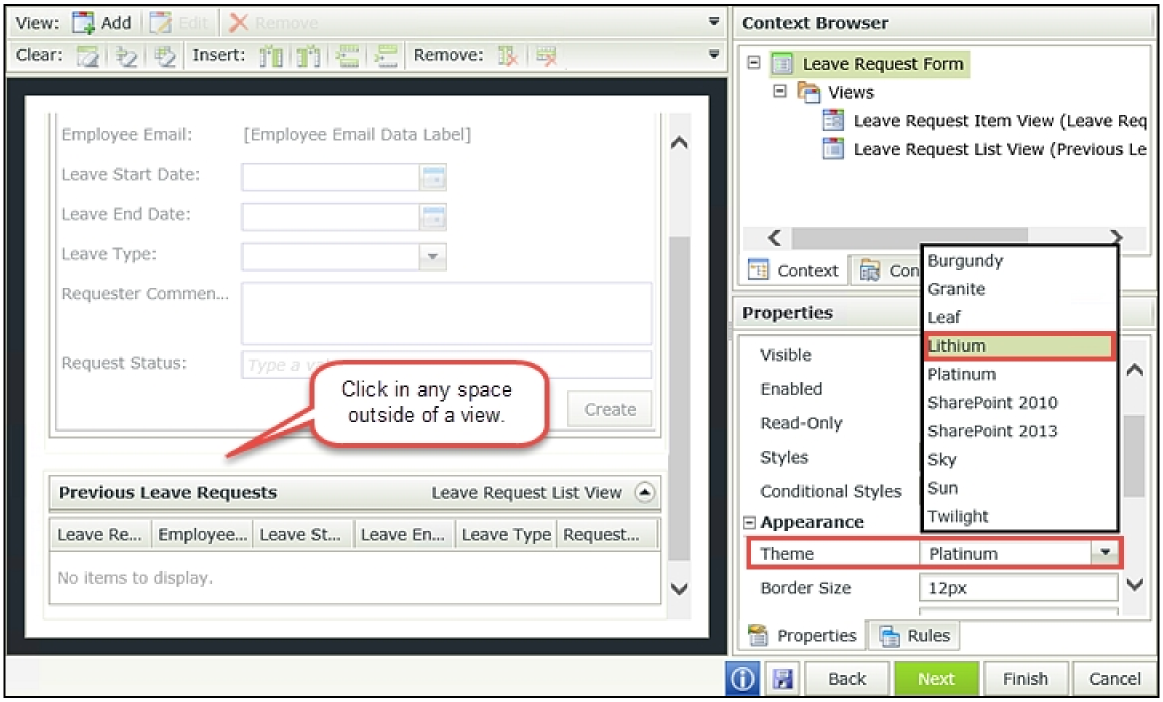
Leave Re...	Employee...	Leave St.
No items to display.		

The last step in creating your form will be to customize the look and feel by changing the default theme to the Lithium theme.

Note
K2 has a number of predefined themes that you can apply to your forms, or you can create your own as you advance in your K2 knowledge. Lithium is K2's latest theme and is compatible with mobile devices, so it is a good option if your users will be using devices like tablets and phones to access SmartForms. It is possible to extend the standard list of themes with custom themes, for example if your organization has a particular color scheme.

g. First, you want to access the Properties Pane for the form. You need to make sure that neither of the views is already highlighted. **Click** anywhere outside of the views. In the Properties Pane, navigate to the **Theme** property found under the Appearance heading. Select **Lithium**.

(If you have trouble seeing the Theme property, it's quite possible that a view is currently selected.)



Click in any space outside of a view.

Employee Email: [Employee Email Data Label]
Leave Start Date:
Leave End Date:
Leave Type:
Requester Commen...
Request Status: Type a v

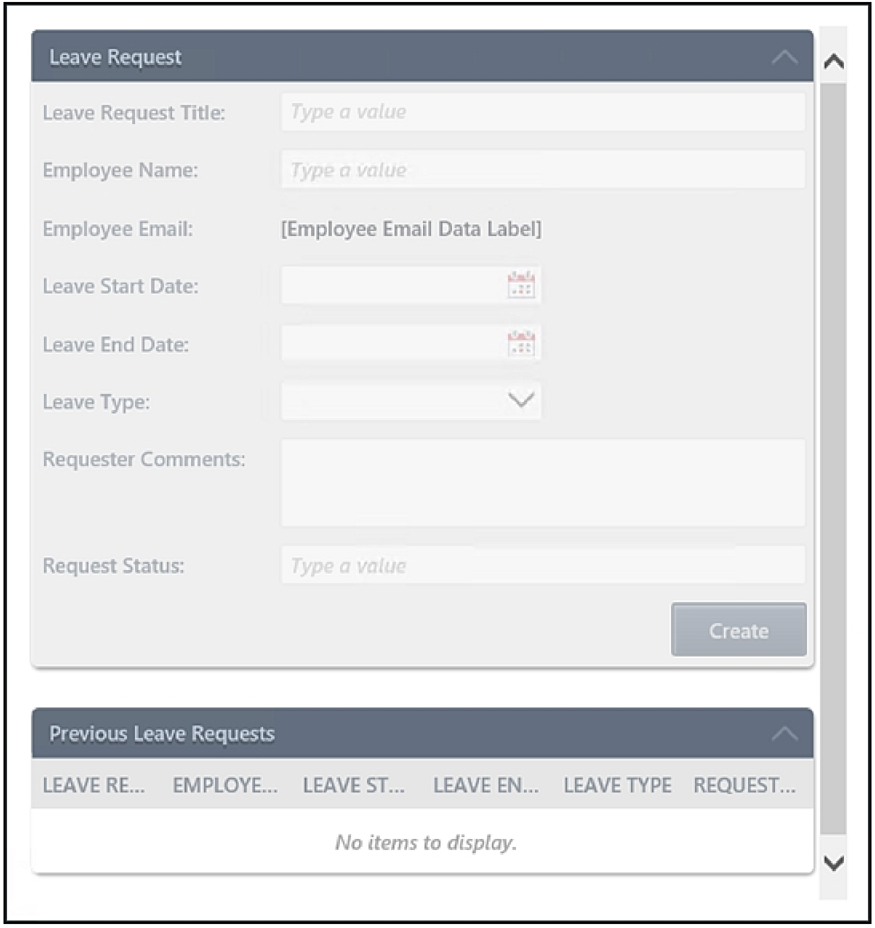
Previous Leave Requests — Leave Request List View

Leave Re...	Employee...	Leave St...	Leave En...	Leave Type	Request...
No items to display.					

Context Browser
- Leave Request Form
  - Views
    - Leave Request Item View (Leave Req
    - Leave Request List View (Previous Le

Properties
- Visible
- Enabled
- Read-Only
- Styles
- Conditional Styles
- Appearance
  - Theme: Platinum
  - Border Size: 12px

Once the Lithium theme is applied, the form style should change to something like the image below.



Leave Request

Leave Request Title: Type a value

Employee Name: Type a value

Employee Email: [Employee Email Data Label]

Leave Start Date:

Leave End Date:

Leave Type:

Requester Comments:

Request Status: Type a value

Create

Previous Leave Requests

LEAVE RE...	EMPLOYE...	LEAVE ST...	LEAVE EN...	LEAVE TYPE	REQUEST...
No items to display.					

h. Click **Finish** to save and exit the Form Designer. In the Category Browser, right-click the Leave Request Form and select **Check In**. If you get a message about associated views, click **OK** to check in the Views associated with this Form.
Views and forms are not available to your users until you have checked them in. This essentially "publishes" the artifacts to the K2 server. If you make a change to a view or form, you must check it out then check it back in, before those changes are evident to your users.

STEP 5 REVIEW

In this step, you created a new form and then added your Item View and List View to the form canvas. You gave your form a complete make-over by simply changing the theme.

Part 3: Workflow

In Part 3 you will create the Leave Request Workflow, which will incorporate the Data and Forms components created in Parts 1 and 2. The workflow will contain a User Task (the manager making a decision on the request) and System Tasks (updating the status property in the Leave Request SmartObject). Recall that User Tasks are tasks performed by a human, such as making a decision. A System task is performed by K2, such as sending an email or updating a list of some kind.

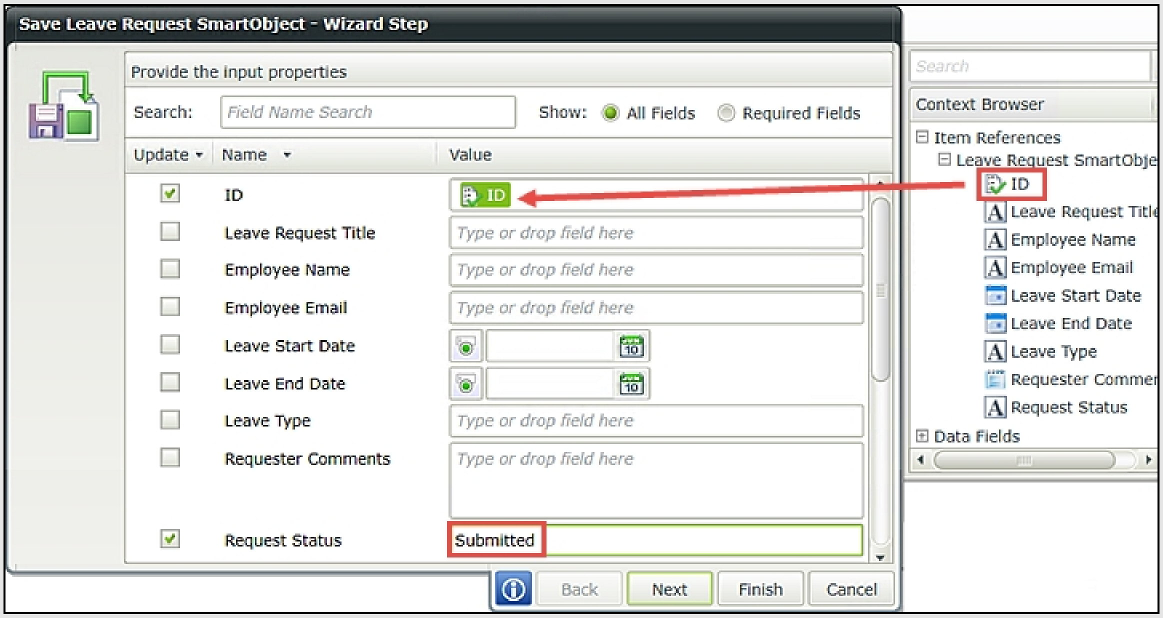
Step 6: Create a new workflow, assign permissions and add a system task to update the request status property

In this step, you will create a new workflow in K2 Workflow Designer. You will associate the workflow with the Leave Request Form and assign workflow rights to all domain users.

Step 6 Tasks

1. Create a New Category under Leave Request and name it
Workflows

2. From the Workflows category, create a **New Workflow**. On the Workflow setting screen, make the following adjustments:
   - **Name**: *Leave Request Workflow*
   - **Form**: *Leave Request Form*
   - **Folio**: Use the *Leave Request Title Text Box* control found in the Context Browser (**Leave Request Item View** > **Controls**).
   - **Item Reference**: UNCHECK the Create Item Reference box, then CHECK it. Confirm that the **Leave Request SmartObject** has been added as the value for the Create Item Reference > Item Reference field.
3. Confirm the **start rule** is called when the Create Button is clicked.
4. Assign **Start** and **View** Workflow Rights to all domain users.
5. Add a **Save Leave Request** SmartObject event to the first event box. For the **Request Status** field, enter
   Submitted
   then use the Leave Request SmartObject **ID** as the input properties' **ID** value.



Save Leave Request SmartObject - Wizard Step

Provide the input properties

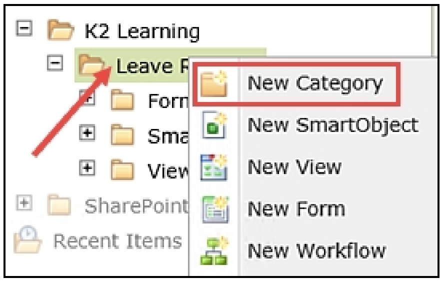
Search: Field Name Search   Show: ● All Fields ○ Required Fields

Update	Name	Value
☑	ID	ID
□	Leave Request Title	Type or drop field here
□	Employee Name	Type or drop field here
□	Employee Email	Type or drop field here
□	Leave Start Date	
□	Leave End Date	
□	Leave Type	Type or drop field here
□	Requester Comments	Type or drop field here
☑	Request Status	Submitted

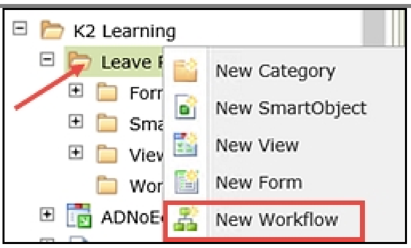
6. Change the Save step title to
   Set Status Submitted

Step 6 Walkthrough

a. Create a **New Category** under Leave Request and name it
   Workflows

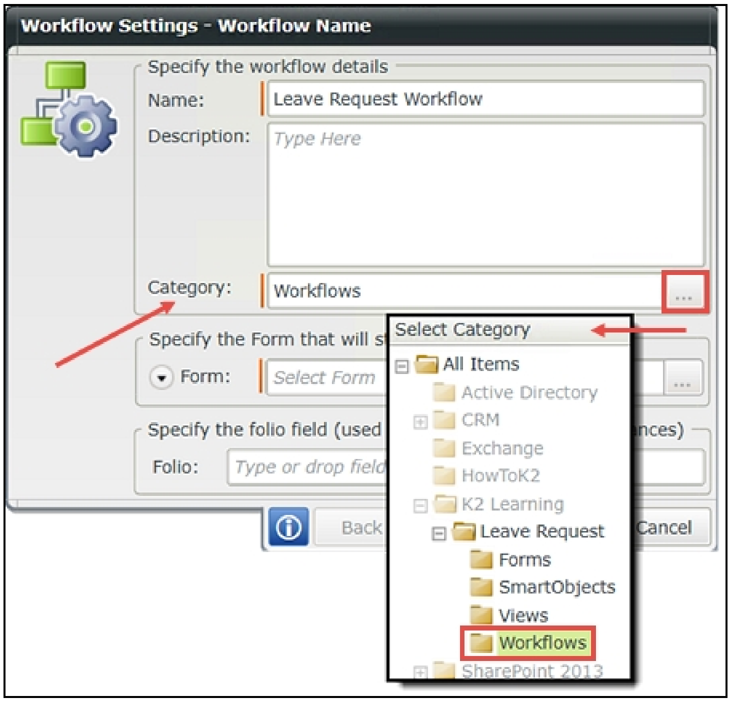


b. Right-click the **Leave Request** category and select **New Workflow**.



c. The Workflow Settings screen opens. **Name** the workflow *Leave Request Workflow*

d. Click the ellipses to the right of the **Category** field, then change the category to **Workflows**.



Workflow Settings - Workflow Name

Specify the workflow details

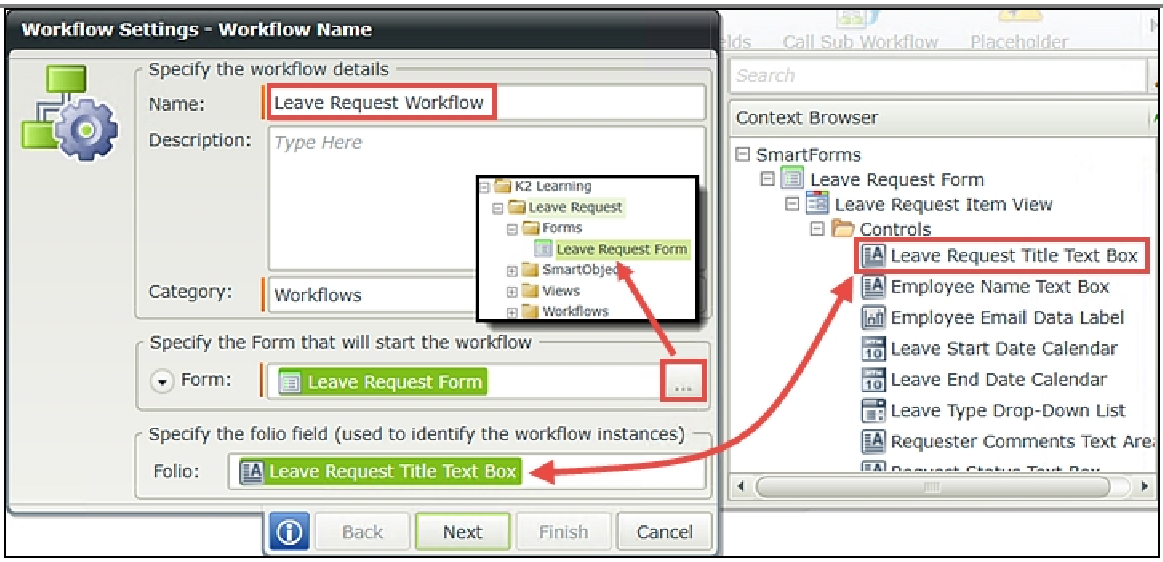
Name: Leave Request Workflow

Description: Type Here

Category: Workflows

e. Click the ellipses to the right of the **Form** field. Navigate to and select the **Leave Request Form**. In this step, you are associating the Leave Request Workflow with the Leave Request Form. This will in turn, expose all of the form controls (actually the SmartBox SmartObject properties) to the workflow designer. In later steps, you will update the Request Status property using the SmartObject property in the workflow.

f. For the **Folio**, expand the SmartForms tree in the Context Browser and drag the **Leave Request Title Text Box** control into the Folio field. Use the image below as a guide.



Specify the workflow details

Name: Leave Request Workflow

Description: Type Here

Category: Workflows

Specify the Form that will start the workflow

Form: Leave Request Form

Specify the folio field (used to identify the workflow instances)

Folio: Leave Request Title Text Box

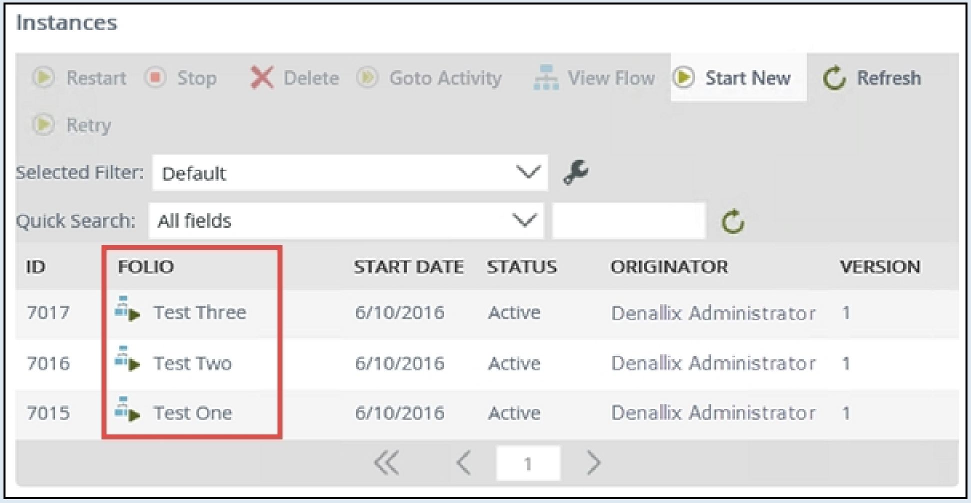
Note

What is the Folio? The folio is a text field that is commonly used to distinguish one process instance from another process instance (of the same workflow). The folio is not required, nor does it have to be unique. Since all processes, or workflows, will have the same process name, using a unique folio value will aid in identifying the individual process instances of the workflow. This is especially useful when viewing workflow reports having many process instances.

To make the folio unique, you can use a combination of fields, such as a customer name combined with an order number. Setting the folio value typically comes from properties found in the context browser. The property values are variables that are replaced at runtime by actual content specific to the process instance.

You can also use the folio as a variable throughout your workflow. For example, if you assign a customer name as the folio, you can use the folio to customize emails so that the customer is referenced by their name.

Three instances of the same workflow showing unique folio values

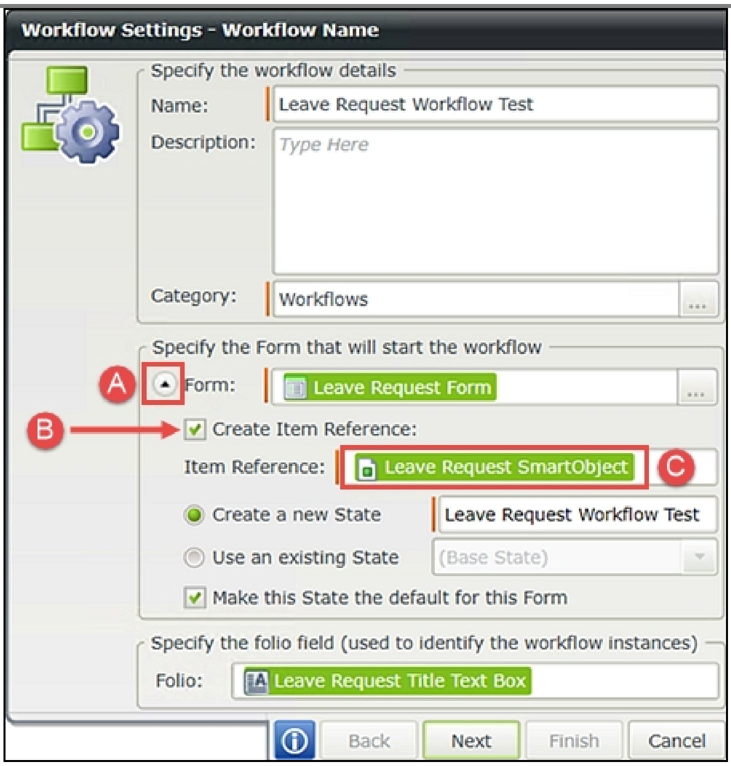


Selected Filter: Default

Quick Search: All fields

ID	FOLIO	START DATE	STATUS	ORIGINATOR	VERSION
7017	Test Three	6/10/2016	Active	Denallix Administrator	1
7016	Test Two	6/10/2016	Active	Denallix Administrator	1
7015	Test One	6/10/2016	Active	Denallix Administrator	1

g. Click the black arrow just to the left of the **Form** label to expose the Item Reference settings. UNCHECK the option to **Create Item Reference**. Now, CHECK the option once again. Confirm that the **Leave Request SmartObject** has been added as the Item Reference.



Workflow Settings - Workflow Name

Specify the workflow details

Name: Leave Request Workflow Test

Description: Type Here

Category: Workflows

Specify the Form that will start the workflow

Form: Leave Request Form

☑ Create Item Reference:

Item Reference: Leave Request SmartObject

● Create a new State: Leave Request Workflow Test

○ Use an existing State: (Base State)

☑ Make this State the default for this Form

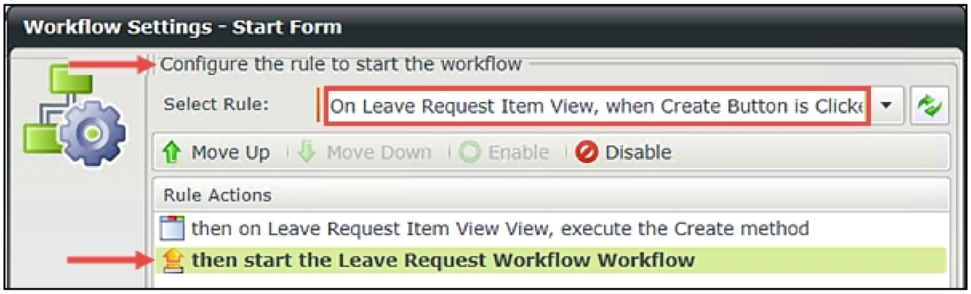
Specify the folio field (used to identify the workflow instances)

Folio: Leave Request Title Text Box

At this time, you will not make any changes to the State options, so click **Next** when ready.

> Note
>
> **Item References**
>
> K2 keeps track of items that relate to the current process instance, known as Item References. For example, when a workflow is started from a SharePoint list or library, or from a SmartObject, K2 will store a *pointer* to the current record in the Item References properties so that those values can be used later on in the workflow or rules. For now, just think of Item References as a shortcut to the current record values from the SmartObject, or data source.

h. You should now be on the **Configure the rule to start the workflow** screen. Here, you want to tell K2 to start the workflow when the Create button has been clicked. Confirm that the **Select Rule** drop-down contains the value **On Leave Request Item View, when the Create Button is Clicked**. Click **Finish**.



Workflow Settings - Start Form

Configure the rule to start the workflow

Select Rule: On Leave Request Item View, when Create Button is Clicke

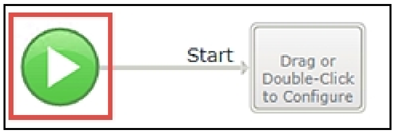
Rule Actions

then on Leave Request Item View View, execute the Create method

then start the Leave Request Workflow Workflow

At this point, you have configured the basic settings for your Leave Request Workflow. You still need to assign permissions however, and to do that, you need to access the advanced workflow settings.

i. Double-click the **Start** button to open the advanced workflow settings wizard.

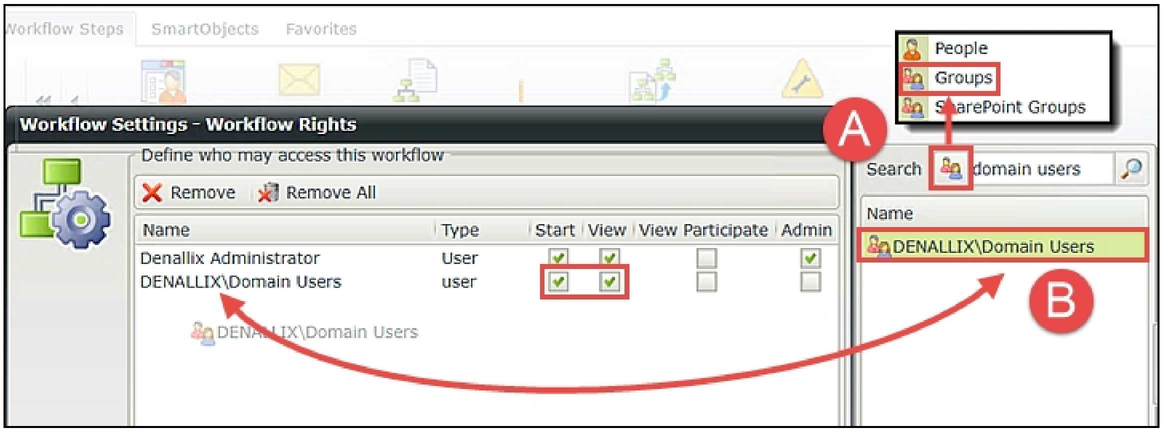


j. This is the same wizard you just worked through, however the advanced version contains the permissions settings. Click **Next** until you reach the **Workflow Rights** screen (five times).

k. Begin by changing the search scope to **Groups**. (**A** below) Enter
domain users
into the search box and click the search icon or hit enter. You should now see the Domain Users group in the search results pane. (**B** below) Click and drag **Domain Users** into the permissions pane. CHECK the **Start** and **View** boxes.

> **Caution**
> If you do not assign permissions, your users will see an error when they submit the form. The Start setting allows users to start the workflow and the View setting allows users to view the workflow reporting information.

Click **Finish**



Search: domain users

Name
DENALLIX\Domain Users

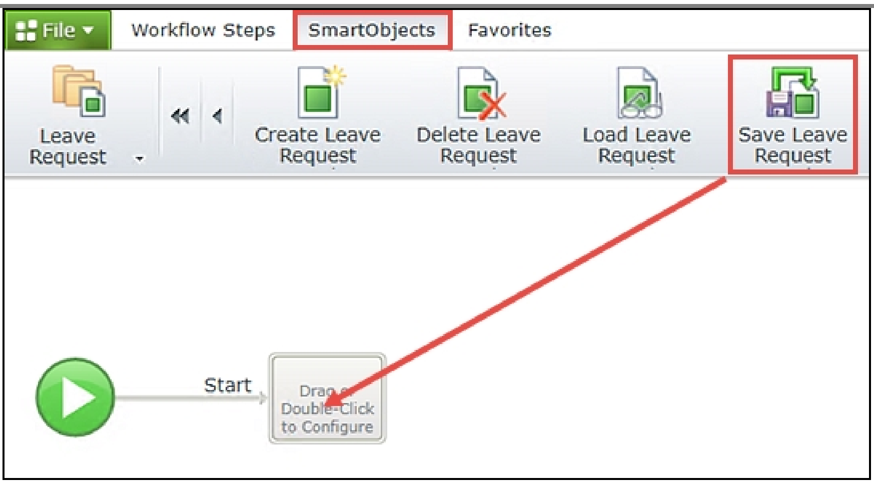
Name	Type	Start	View	View Participate	Admin
Denallix Administrator	User	☑	☑	□	☑
DENALLIX\Domain Users	user	☑	☑	□	□

Now that you have your workflow settings complete, you can move on to adding steps to the workflow. recall that the workflow steps are basically the lines that connect one event to another. The first event you are going to add will be a System Event using the SmartObject Event wizard. (System events are tasks that K2 performs, rather than a user. Here, you will set the Request Status property to indicate the Leave Request has been submitted.

l. Select the **SmartObjects** tab in the workflow ribbon. Drag the **Save Leave Request** event into the event box. ("Save" is the equivalent of "Update".)

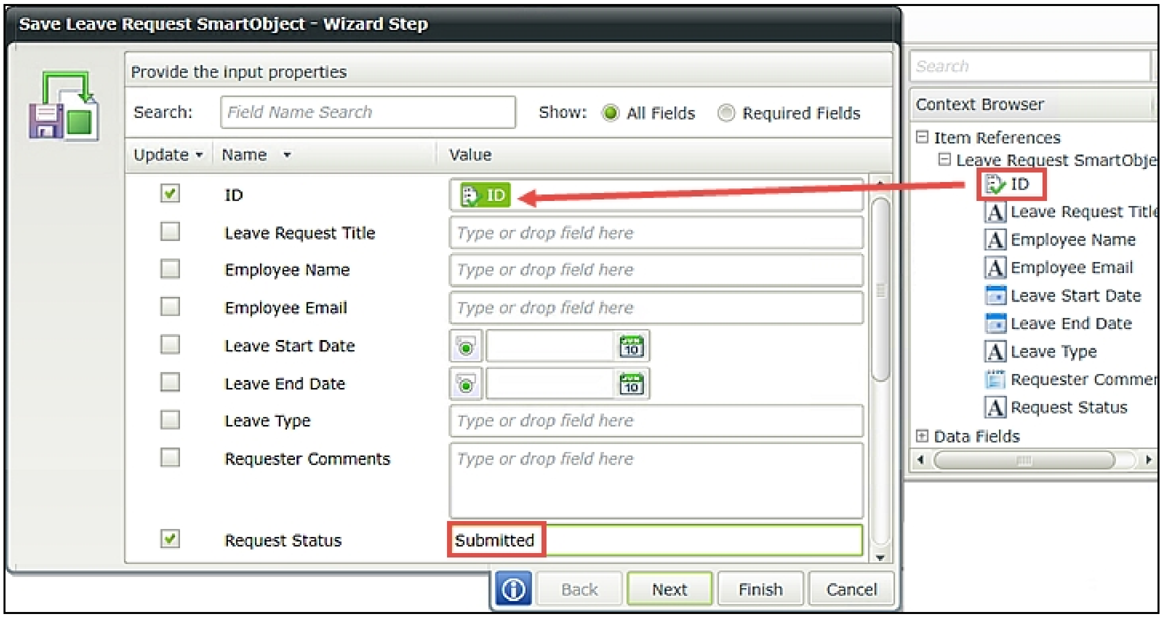
> **Note**
> If you do not see the Leave Request SmartObject methods in the ribbon toolbar, it's possible you have more than one SmartObject created. (Perhaps this is not the first application being built.) In this case, click the black drop-down arrow next to the SmartObject category name (this will be the folder icon at the left of the ribbon toolbar) and select the Leave Request SmartObject. You should now see the correct methods for this application.

m. Recall that when you created the Leave Request SmartObject, you *allowed* the SmartObject to be used in workflows. This is where that setting comes into play. Had you not checked the 'Allow' option, none of the Leave Request SmartObject methods would appear as they do below. (The SmartObject events displayed in the image below are 'Create', 'Delete', 'Load' and 'Save'.)



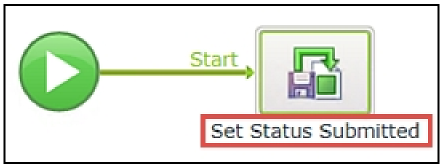
n. The Save wizard opens. The first configuration you need to make is to tell K2 *which* record you want to update. You do this by assigning the SmartObject ID for the current record into the Save event. In the Context Browser, expand the **Item References** node, then Leave Request SmartObject. Drag the SmartObject **ID** into the ID input property. The ID as the input property tells K2 to update the record that is part of the current workflow, or the current record.

o. Now you'll set the status property. Scroll down until you see **Request Status**. Enter
Submitted
into the value field. Click **Next**, then **Finish** to complete the Save event configuration. You have just instructed K2 to update the current SmartBox SmartObject record, Request Status value.



Update	Name	Value
☑	ID	ID
□	Leave Request Title	Type or drop field here
□	Employee Name	Type or drop field here
□	Employee Email	Type or drop field here
□	Leave Start Date	
□	Leave End Date	
□	Leave Type	Type or drop field here
□	Requester Comments	Type or drop field here
☑	Request Status	Submitted

p. To keep your workspace organized, you'll rename the Save event so that you know exactly what that event does. Double-click the **Save Leave Request** title and change it to
Set Status Submitted



STEP 6 REVIEW

In this step you created the Leave Request Workflow and configured it to use the Leave Request Form. You assigned Workflow Rights, or permissions, to all domain users so that they can start the workflow and view workflow progress. You added a SmartObject Save event, which will update the status property to indicate the form has been submitted. You will update the status property again as the workflow progresses by adding additional SmartObject Save events, indicating either Approved or Rejected.

Step 7: Add the Manager Approval user task

In this step, you will add a User Task which will be the approval decision assigned to the workflow originator's manager. After submitting the form, the form originator's manager will receive a task notification email indicating their decision is now required to continue the workflow. The manager can select either Approved or Rejected.

Step 7 Tasks

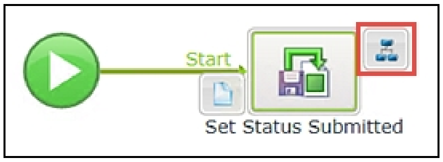
1. Add an outcome to the Set Status Submitted event. Name the outcome
   Continue
2. Add a **User Task (SmartForms)** to the new event box and name it
   Manager Approval
   then add two actions and name them
   Approved
   Rejected
3. Add the **Manager** as the task participant, then check the option to notify them when they receive a work item. You will not customize the email notification for this application.

   Note
   If you are building this application within your own environment, you will want to assign the task participant to a user where you have access to their Outlook. You might consider assigning this task to yourself for access/testing purposes. If you are building this application on a K2-provided VM, you will have access to the manager's Outlook.

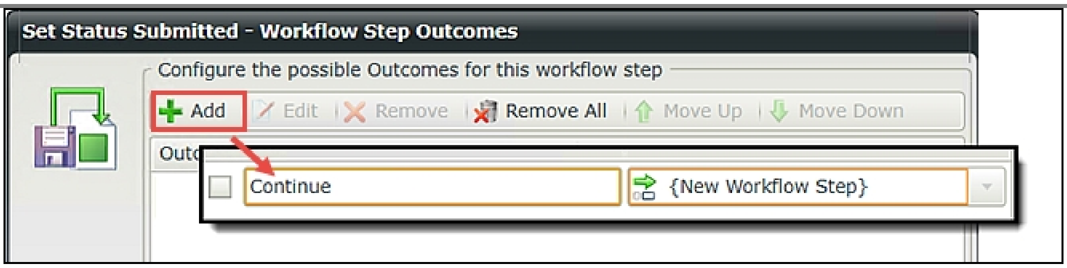
4. Save your work.

Step 7 Walkthrough

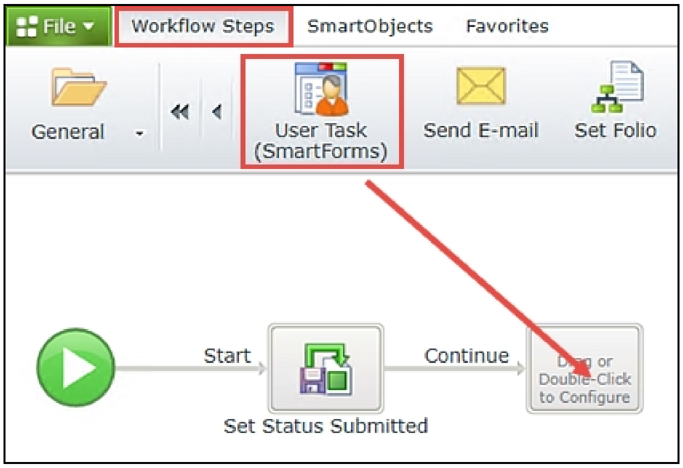
a. Hover over the **Set Status Submitted** event until you can see the **Workflow Step Outcome icon**. (The icon looks like a blue mini-flowchart. If you don't see the icon when hovering, click once on the workflow canvas to refocus your mouse to the canvas. Then try hovering again.) Click the **Outcome** icon to opens its wizard.



b. On the configuration screen, click **Add**. Replace New Outcome with
   Continue
   then click **OK**. Here, you are simply giving your step, or outcome line, a label so that it is easy for us to tell exactly what that step does. In this case, it is very straightforward. However, as you build more complex workflows, you may be adding conditions and more advanced options to the step. Giving the label a descriptive name will assist you (and others working with your workflow design), insight as to what the step is doing or the logic behind the step.

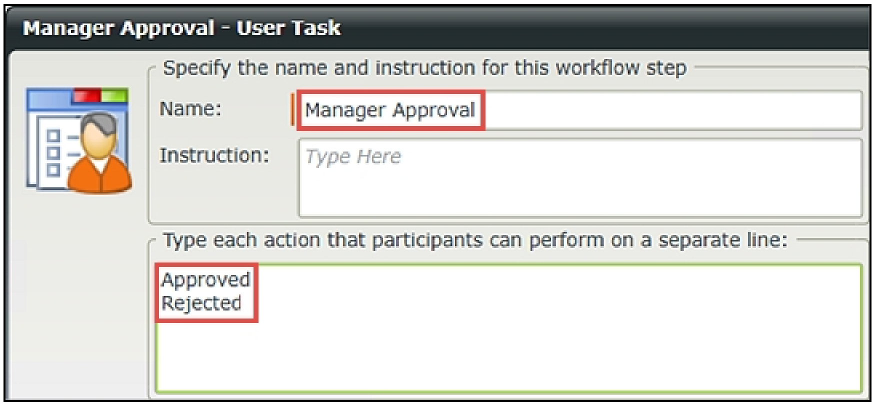

Set Status Submitted - Workflow Step Outcomes
Configure the possible Outcomes for this workflow step
Add
Edit
Remove
Remove All
Move Up
Move Down
Outc
Continue
{New Workflow Step}

c. Notice that K2 has added a new event box. Switch back to the **Workflow Steps** tab, then drag a **User Task (SmartForms)** event into the new event box.



d. The User Task wizard opens. Name the task
Manager Approval

e. On this screen you can will also configure the actions. Actions are the decision options that the user assigned the task can make. You can have many actions, but you need at least one. For your workflow, you will have two. Enter the following actions (on separate lines):
Approved
Rejected
then click **Next** when ready.

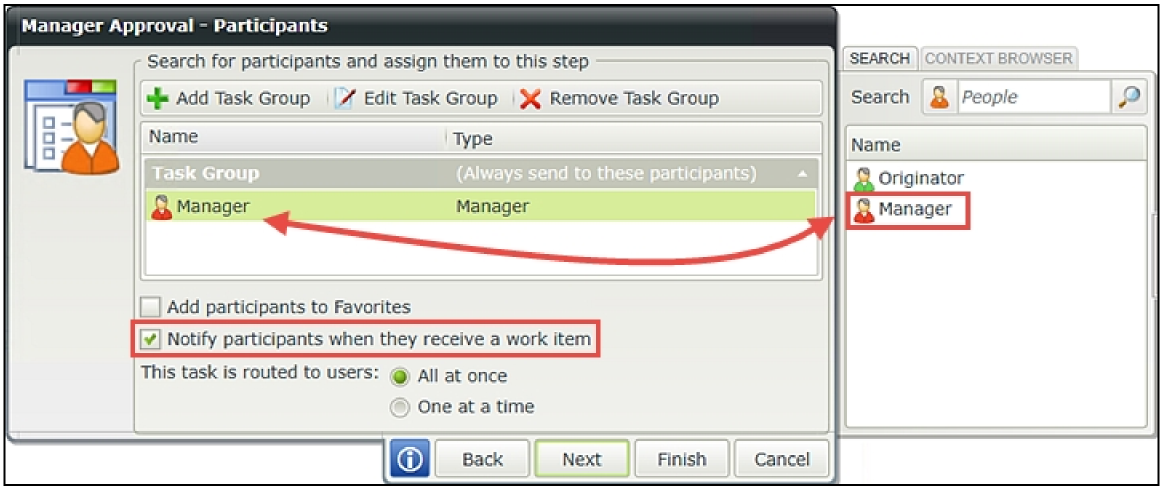

Manager Approval - User Task
Specify the name and instruction for this workflow step
Name:
Manager Approval
Instruction:
Type Here
Type each action that participants can perform on a separate line:
Approved
Rejected

f. The User Form setting is fine, so click **Next**. The work item rule is fine, so click **Next**.

g. On the Participants screen, drag the **Manager** name into the Task Group window. CHECK the box to **Notify participants when they receive a work item**. Here, you are assigning this task to the form originator's manager. (You also have the option to search for participants in the same manner as you did for the workflow permissions.) If you choose to *not* notify the participant, they will have to rely on manually checking their worklists to see if they have any tasks assigned to them.

Note
If you are building this application within your own environment, you will want to assign the task participant to

a user where you have access to their Outlook. You might consider assigning this task to yourself for access/testing purposes. If you are building this application on a K2-provided VM, you will have access to the manager's Outlook.



Manager Approval - Participants

Search for participants and assign them to this step

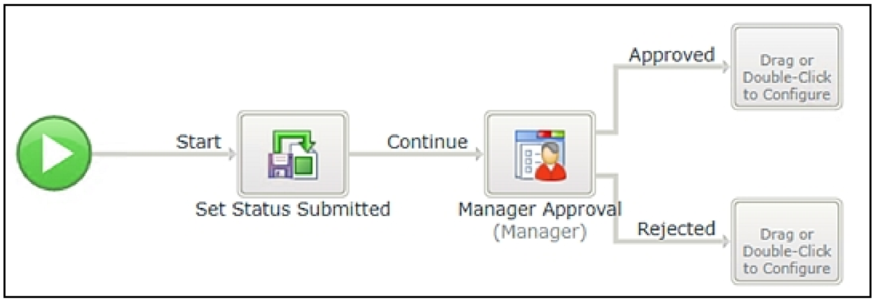
Name	Type
Task Group	(Always send to these participants)
Manager	Manager

☐ Add participants to Favorites

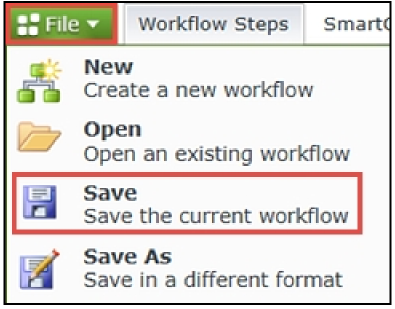
☑ Notify participants when they receive a work item

This task is routed to users: ● All at once ○ One at a time

You will not customize the task notification email for this application, so click **Finish**. Your screen should look like the image below. Notice that K2 has added two additional steps and event boxes for the actions you just configured.



h. **Save** your work. (File > Save)



File
Workflow Steps
SmartO
New
Create a new workflow
Open
Open an existing workflow
Save
Save the current workflow
Save As
Save in a different format

STEP 7 REVIEW

In Step 7, you added a User Task and assigned it to the form originator's manager. You added two actions (Approved and Rejected). K2 automatically generated outcome lines for each of the actions you added. You can have many actions, but you need to have at least one. By checking the option to notify the participant of a worklist item, the manager will receive an email indicating they have a task, with a link to open the form and action the task.

Step 8: Complete the workflow by updating the request status property for both outcomes

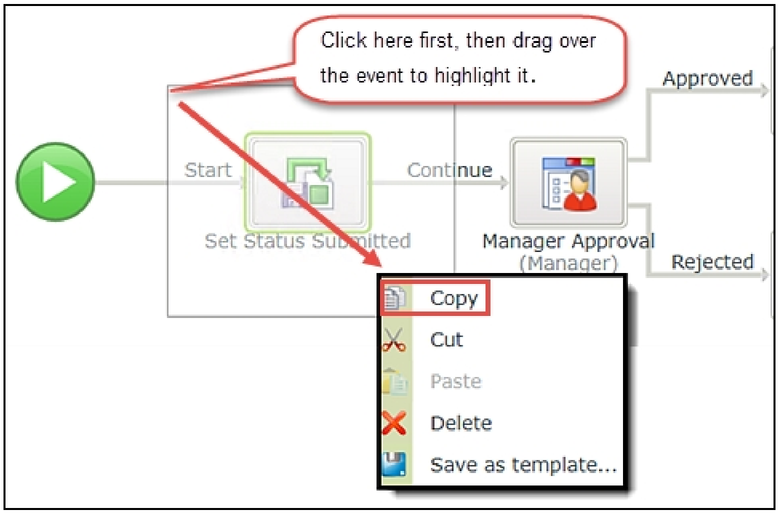
Step 8 continues with the addition of two System Tasks that will update the Request Status property in the Leave Request SmartObject depending on the decision made by the approving manager. K2 Designer makes reusing events very simple by allowing you to copy and paste existing events. With a few minor adjustments, your new events are ready to go!

Step 8 Tasks

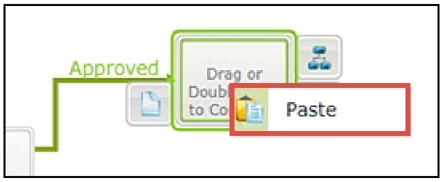
1. Copy the **Set Status Submitted** event and paste it into the event box for the Approved outcome line, then once again for the Rejected outcome line.
2. Edit both events and change the **Request Status** property to the appropriate outcome (*Approved* or *Rejected*). Change the event titles to match the outcome.
3. Add a **Placeholder** step to the new event boxes that are created.

Step 8 Walkthrough

a. Copy the **Set Status Submitted** event by click+dragging a box over it to highlight it. Select **Copy** when the option menu appears.

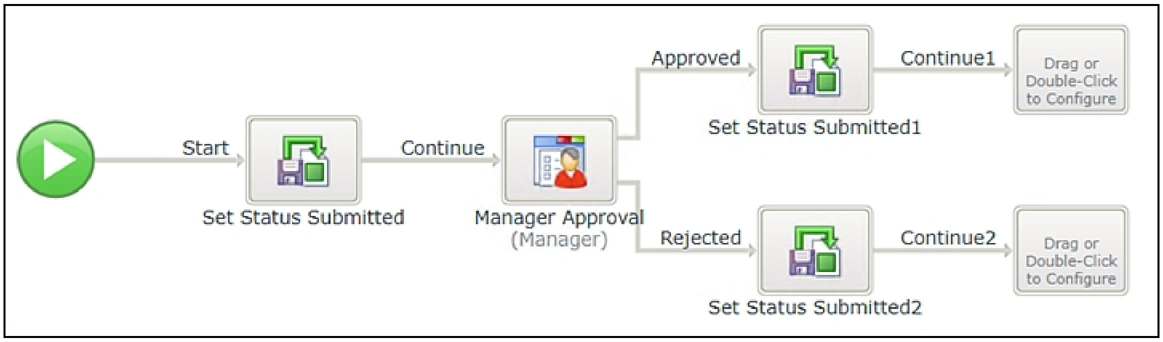


b. Hover over the Approved outcome event, right-click, then click **Paste**.



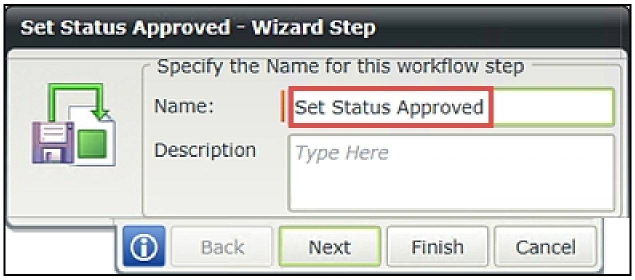
c. Hover over the Rejected outcome event, right-click, then click **Paste** once again.

Your screen should look like the image below. K2 added an additional step and event for each of the system tasks you just copied and pasted.



Notice that K2 copied the Save event as well as the Continue outcome line. In the next few steps, you will adjust each new event so that they are consistent with their outcome, then you'll add a Placeholder event to the last two empty event boxes. A Placeholder simply completes an outcome line without adding a formal event. This allows you to deploy and test your workflow without receiving errors of missing events.

d. Double-click inside of the **Set Status Submitted1** event (the Approval side) to open its wizard. Change the name to
Set Status Approved
then click **Next**.



Set Status Approved - Wizard Step

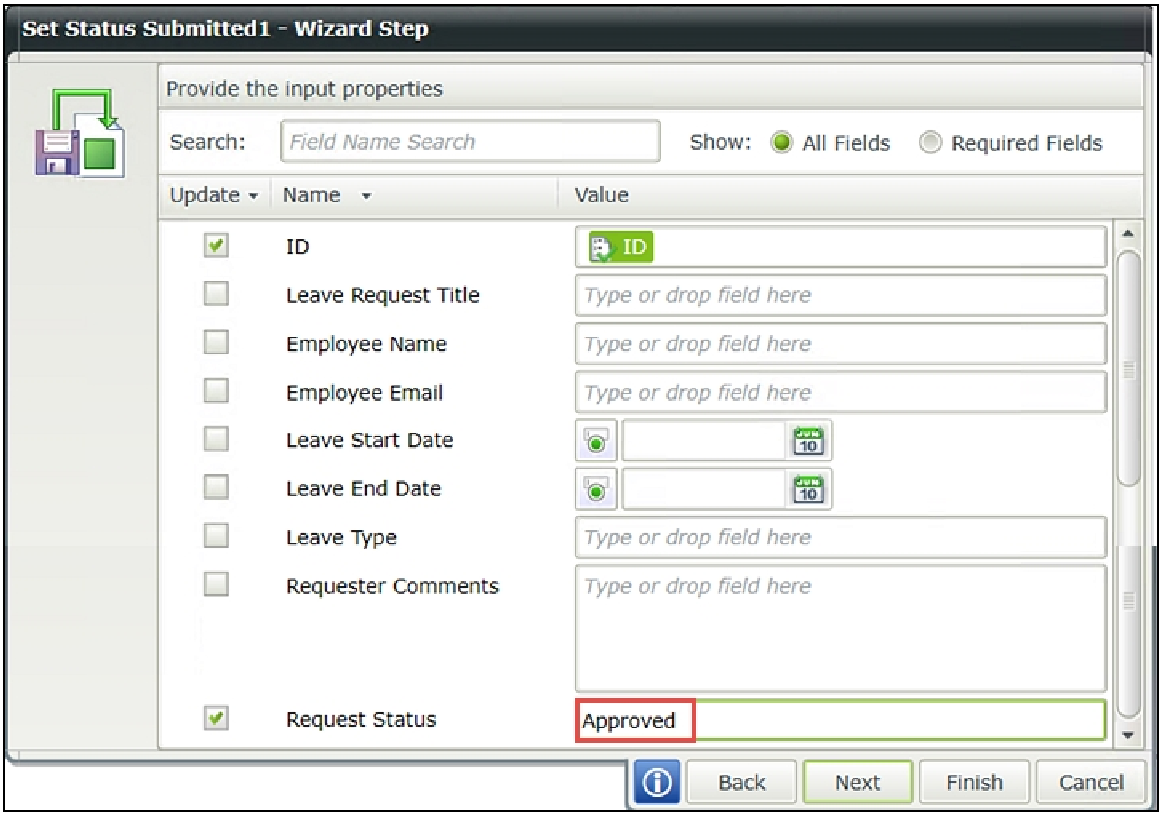
Specify the Name for this workflow step

Name: Set Status Approved

Description: *Type Here*

e. There are no necessary changes to the Workflow Step Outcomes screen, so click **Next**. (You can change Continue1 to just Continue if you like to keep your lines consistent.)

f. On the input properties screen, change the **Request Status** property from Submitted to
Approved
then click **Finish**. Be sure to keep the ID property as it is, since you have already configured it to reflect the current record. You just want to update the status property so that it reflects the outcome line for an Approved decision.



Set Status Submitted1 - Wizard Step

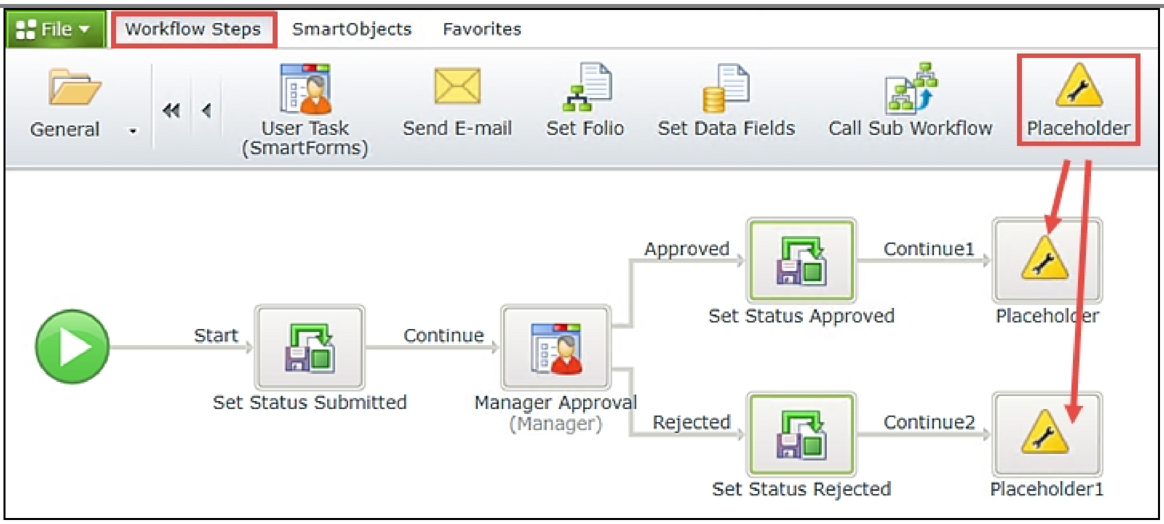
Provide the input properties

Search: *Field Name Search* Show: ● All Fields ○ Required Fields

Update	Name	Value
☑	ID	ID
☐	Leave Request Title	*Type or drop field here*
☐	Employee Name	*Type or drop field here*
☐	Employee Email	*Type or drop field here*
☐	Leave Start Date	
☐	Leave End Date	
☐	Leave Type	*Type or drop field here*
☐	Requester Comments	*Type or drop field here*
☑	Request Status	Approved

g. Repeat the steps above for the Rejected side. Change the **Request Status** to
Rejected
then click **Finish**.

h. Still on the Workflow Steps screen, drag a **Placeholder** into each of the empty event boxes (one for the Approved side and one for the Rejected side).



Save your work.

STEP 8 REVIEW

In this step, you reused the Save event for the Approved and Rejected outcomes. By copying and pasting the event, you could take advantage of configurations already in place (the ID property already mapped). Copying an event will also copy its associated outcome, so you added Placeholders into each of the new event boxes that were created. You cannot deploy or test your application with empty event boxes, so the Placeholders tell K2 that the event boxes are empty by design.

Step 9: Deploy the workflow

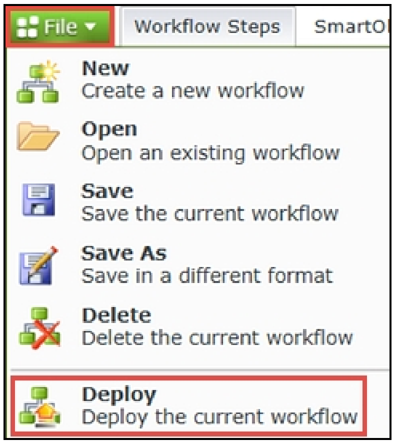
Next you will deploy your workflow. Deploying a workflow publishes the workflow to the K2 server which then allows it to be exposed to the forms configured for that workflow. If you make changes to your workflow, you must redeploy it before the changes are published to the K2 server.

Step 9 Tasks

1. Deploy the workflow.

Step 9 Walkthrough

a. Deploy the workflow by clicking (**File** > **Deploy**)


File
Workflow Steps
SmartO
New
Create a new workflow
Open
Open an existing workflow
Save
Save the current workflow
Save As
Save in a different format
Delete
Delete the current workflow
Deploy
Deploy the current workflow

b. You will see the Deploying Workflow dialog while the workflow is being compiled and deployed. When you see the **Deployment Successful** dialog, your workflow is deployed and ready for use! Click **Close & Exit**.

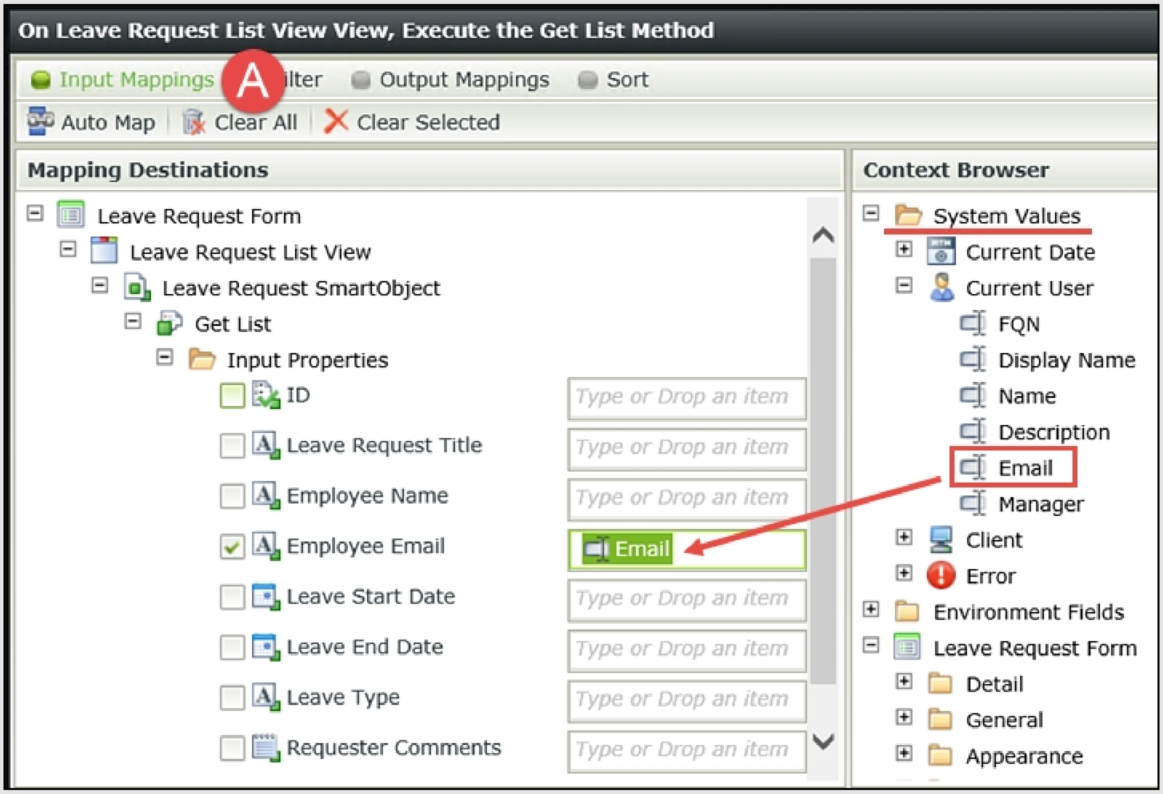
Step 10: Edit the Form Initializing rule to display the current user's employee details and previous leave requests

Now that the workflow has been deployed, you can go back into your Forms and tweak them a little based on where in the workflow the Form is being used (think of this as the "State" of the Form). For example, when the Form is used in the Manager Approval step, you want to load the Previous Leave Requests for the user who requested the leave. Or, when the Form is used to Start the workflow, you want to populate some of the controls on the Form with the current user's information.

In Step 10, you will edit the Form for the Start Workflow state. You will add an action that retrieves the current user's previous leave requests found in the Leave Request SmartObject. These records will populate your List View. The second step will be to configure K2 to automatically populate the Employee Name and Employee Email fields on the form with the current user's details.

Step 10 Tasks

1. Edit the Leave Request Form, **When the Form is Initializing** rule on the Leave Request Workflow (Default) state and add an execute a view method action to retrieve the previous leave requests for the current user. Use the system value **Email** as the input property, then map all of the return properties for the Leave Request List View.



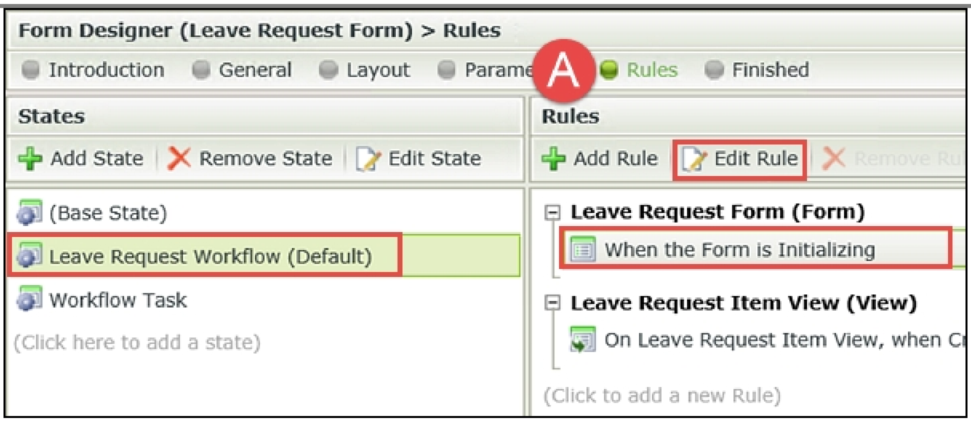
On Leave Request List View View, Execute the Get List Method

□ ID	Type or Drop an item
□ Leave Request Title	Type or Drop an item
□ Employee Name	Type or Drop an item
☑ Employee Email	Email
□ Leave Start Date	Type or Drop an item
□ Leave End Date	Type or Drop an item
□ Leave Type	Type or Drop an item
□ Requester Comments	Type or Drop an item

2. Add a **Transfer Data** action and configure it to use the system value **Display Name** for the **Employee Name Text Box** control and the **Email** for the **Employee Email Data Label** control.

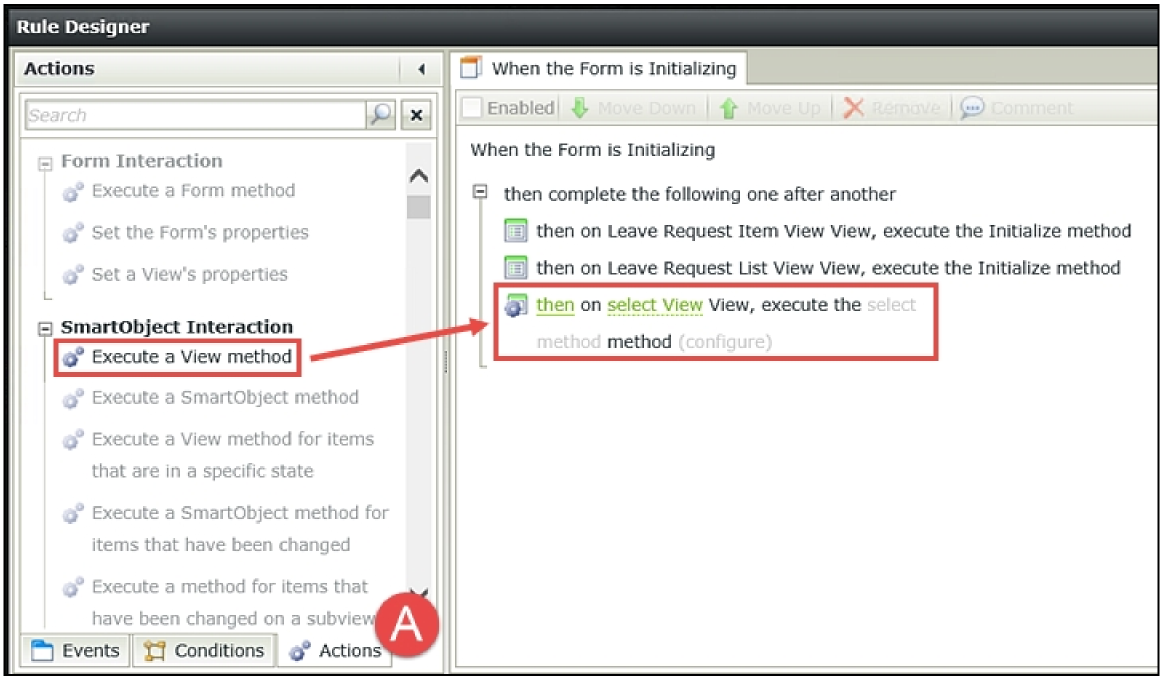
Step 10 Walkthrough

a. In K2 Designer, click on the **Leave Request Form** and select **Edit**. Click the **Rules** button in the Breadcrumb Bar (at the top of the screen). (**A** below) Highlight the **Leave Request Workflow (Default)** state, then the **When the Form is Initializing** rule. Click **Edit Rule**.


Form Designer (Leave Request Form) > Rules
Introduction
General
Layout
Parame
A
Rules
Finished
States
Add State
Remove State
Edit State
(Base State)
Leave Request Workflow (Default)
Workflow Task
(Click here to add a state)
Rules
Add Rule
Edit Rule
Remove Ru
Leave Request Form (Form)
When the Form is Initializing
Leave Request Item View (View)
On Leave Request Item View, when Cl
(Click to add a new Rule)

First, you want to populate the Leave Request List View. Recall that this is the view you created to output a list of all leave requests previously submitted by the form originator. To do this, you want to execute (call) a SmartObject method (Get List) for the list view. You want K2 to populate the previous leave requests when the form is loaded.

b. From the Rule Designer **Actions** pane (**A** below), click once on the action **Execute a View method** to add it to the Rule Definition pane.


Rule Designer
Actions
Search
Form Interaction
Execute a Form method
Set the Form's properties
Set a View's properties
SmartObject Interaction
Execute a View method
Execute a SmartObject method
Execute a View method for items that are in a specific state
Execute a SmartObject method for items that have been changed
Execute a method for items that have been changed on a subview
A
Events
Conditions
Actions
When the Form is Initializing
Enabled
Move Down
Move Up
Remove
Comment

When the Form is Initializing

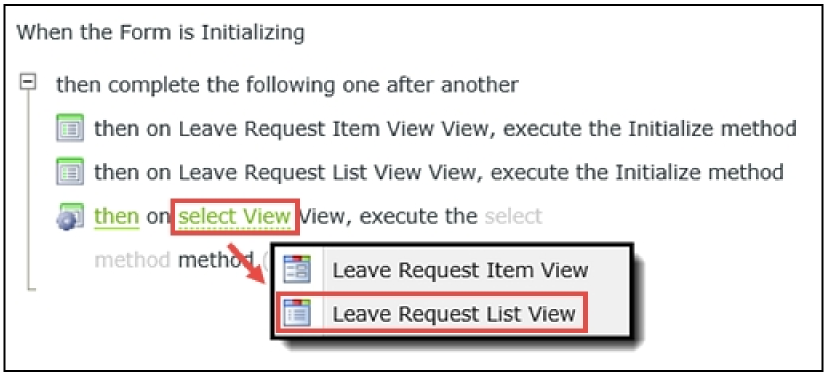
then complete the following one after another

then on Leave Request Item View View, execute the Initialize method

then on Leave Request List View View, execute the Initialize method

then on select View View, execute the select method method (configure)

c. Click the **select View** link so that you can tell K2 which view you want to apply this action. Select **Leave Request List View**.



When the Form is Initializing

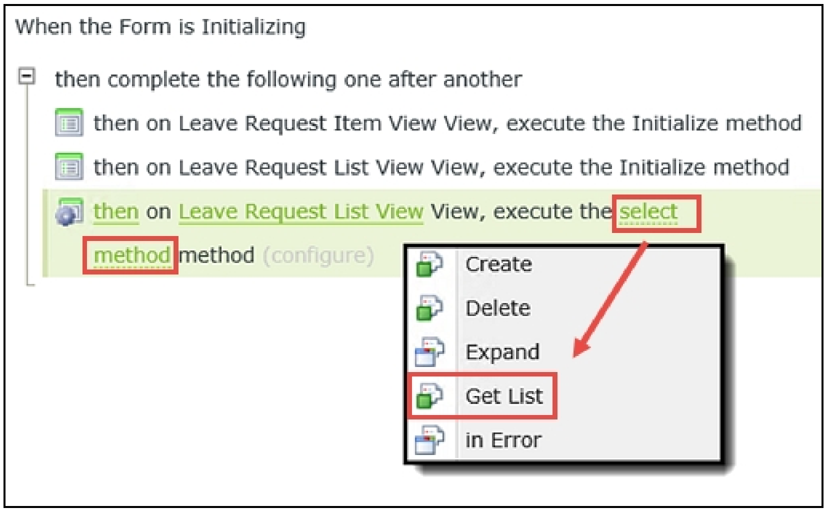
then complete the following one after another

then on Leave Request Item View View, execute the Initialize method

then on Leave Request List View View, execute the Initialize method

then on select View View, execute the select method method (

d. Click the **select method** link and choose **Get List**. The Get List method retrieves content from a data source.



When the Form is Initializing

then complete the following one after another

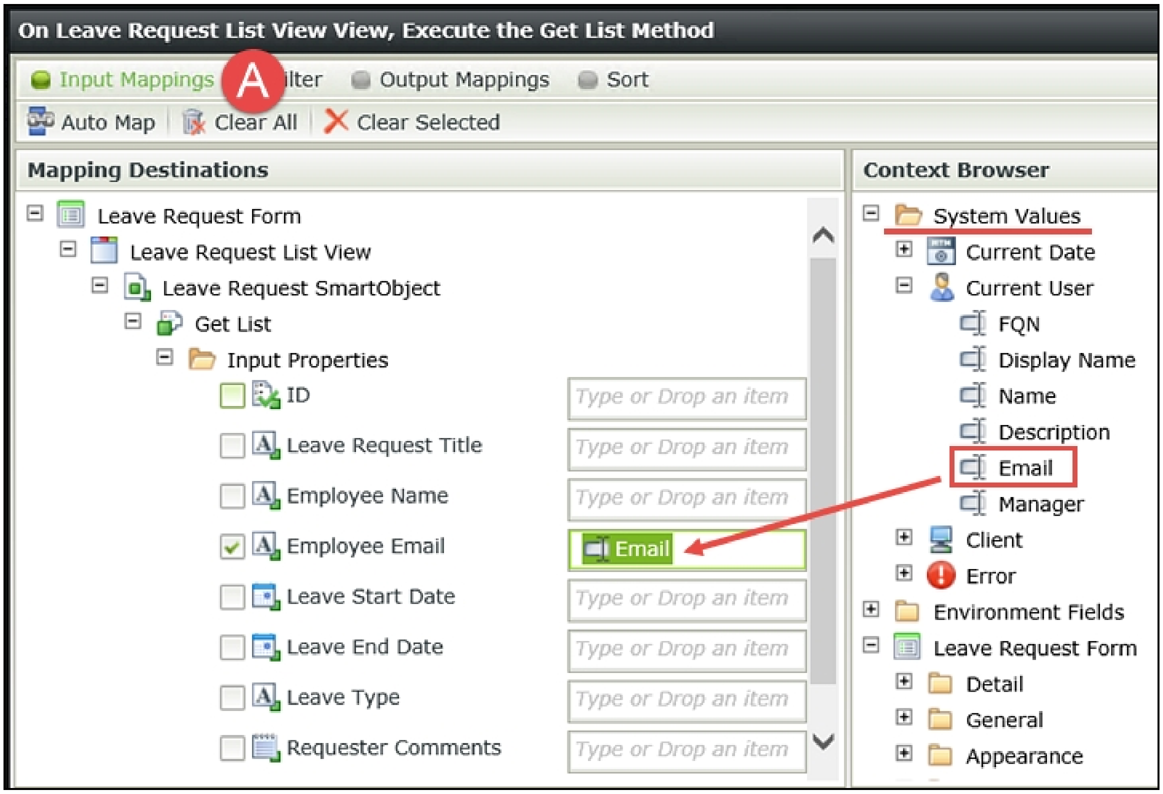
then on Leave Request Item View View, execute the Initialize method

then on Leave Request List View View, execute the Initialize method

then on Leave Request List View View, execute the select method method (configure)

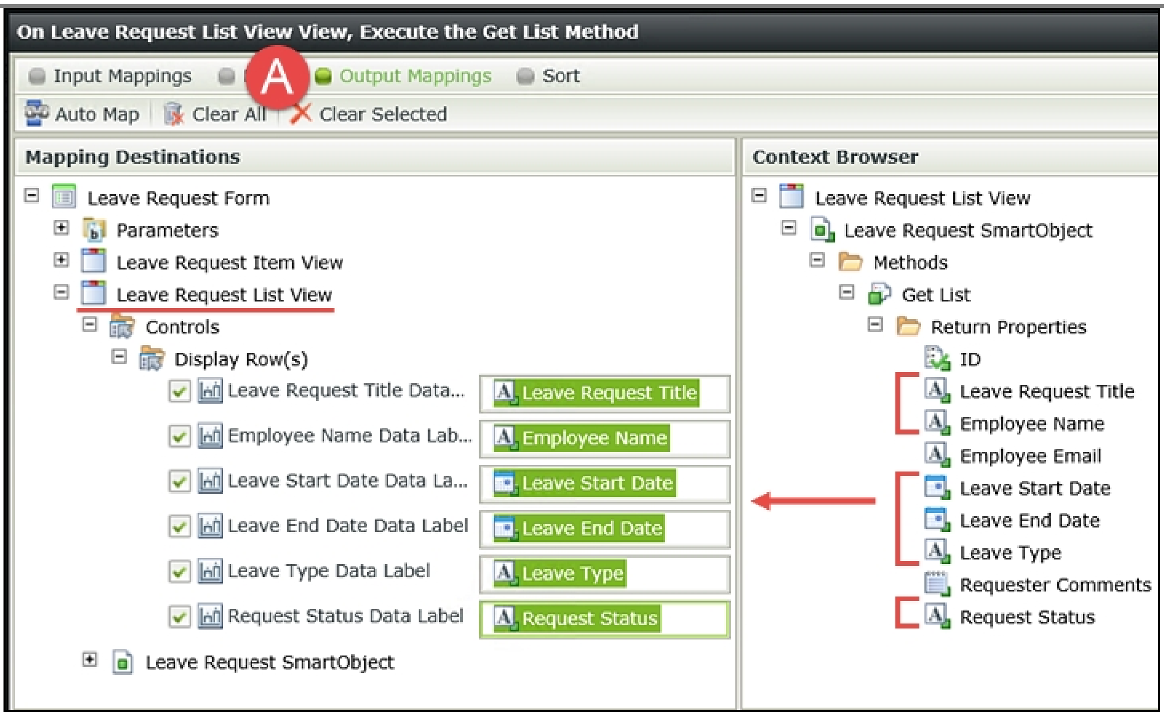
Now click the **(configure)** link. There are two primary sections to the editor that opens. First, you need to tell K2 *which* user records you want to retrieve (this is the input value) and second, you need to tell K2 which SmartObject properties you want to return (these are the output values).

e. On the **Input Mappings** screen (**A** below) expand the **System Values** in the **Context Browser** and drag the **Current User** > **Email** property into the Input Properties **Employee Email** field. Here, you are telling K2 to use the current user's email as the filter to search for previous leave requests. You want K2 to return all user records that have the same email as your current user.



On Leave Request List View View, Execute the Get List Method

f. Now click the **Output Mappings** button. (**A** below). On this screen, you want to tell K2 which properties you want returned to your form. Using the image below as a guide, drag the highlighted **Return Properties** into the Leave Request **List View** fields. (You may have to collapse the Item View in order to see the List View.) You are simply telling K2 which fields you want populated with returned records. In this case, you are returning all of the fields, but you don't have to.

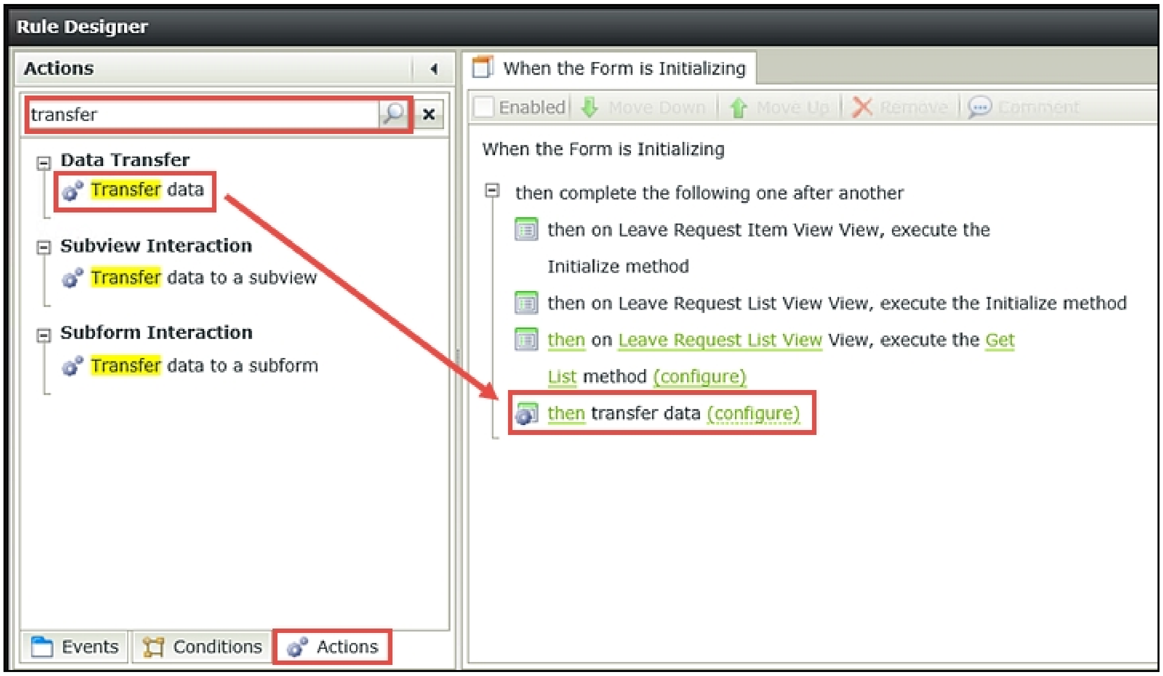


On Leave Request List View View, Execute the Get List Method

☑ Leave Request Title Data...	Leave Request Title
☑ Employee Name Data Lab...	Employee Name
☑ Leave Start Date Data La...	Leave Start Date
☑ Leave End Date Data Label	Leave End Date
☑ Leave Type Data Label	Leave Type
☑ Request Status Data Label	Request Status

Next, you want to display the current user's name and email on the Leave Request Item View. K2 will automatically populate these fields based on who is logged in. These values will then be saved to the Leave Request SmartObject when the form is submitted, along with the leave request details. This step is simply a matter of transferring system values to the Item View fields.

g. Still in the Rule Designer (Actions tab), search the keyword
transfer
then click on **Transfer data** to add it to the Rule Definition pane. (As you become more familiar with rules, conditions and actions, you will find searching for a keyword to be much more convenient than scrolling through all of the rule options.)



Rule Designer

When the Form is Initializing

then complete the following one after another

then on Leave Request Item View View, execute the Initialize method

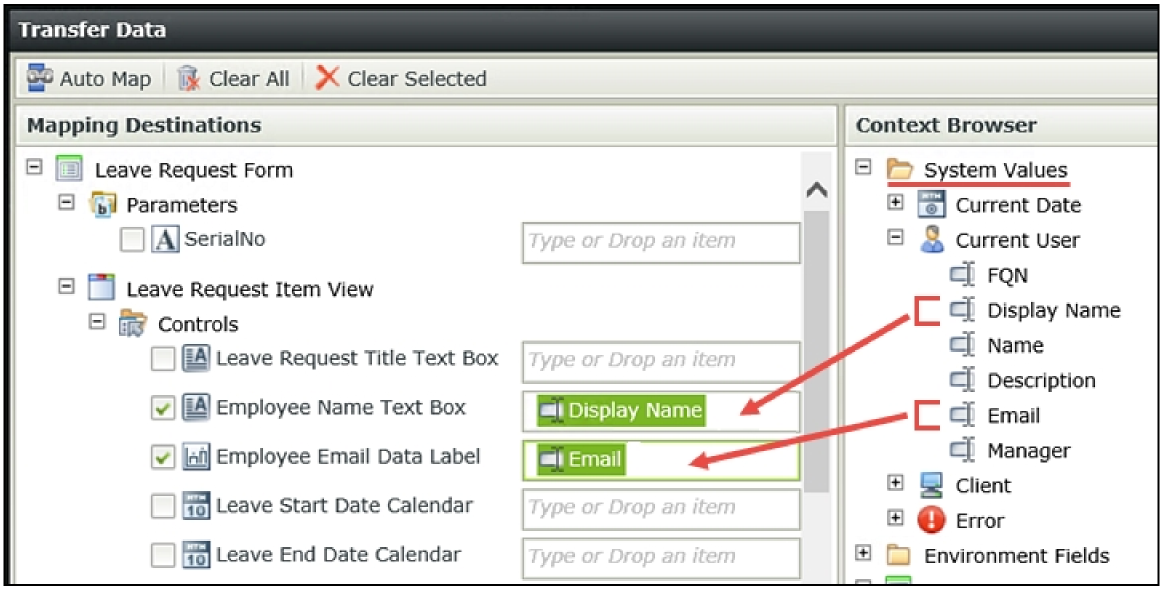
then on Leave Request List View View, execute the Initialize method

then on Leave Request List View View, execute the Get List method (configure)

then transfer data (configure)

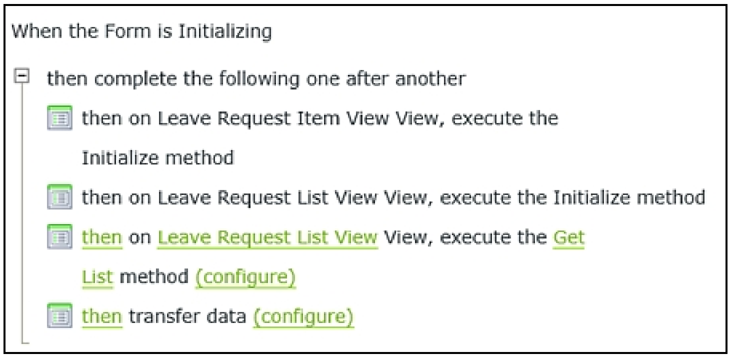
h. Click the **(configure)** link. In the Context Browser, expand the **System Values** node, then **Current User**. Drag the **Display Name** into the **Employee Name Text Box** field. Drag the **Email** into the **Employee Email Data Label** field. Click **OK**.

This step highlights the reason behind making the Email a data label and *not* a text box. Recall that a data label cannot be changed by the user. You need the email to be consistent throughout the entire workflow process, so that rules (like the one you are configuring now) will be stable.



SerialNo	Type or Drop an item
Leave Request Title Text Box	Type or Drop an item
☑ Employee Name Text Box	Display Name
☑ Employee Email Data Label	Email
Leave Start Date Calendar	Type or Drop an item
Leave End Date Calendar	Type or Drop an item

i. Your Rule Definition pane should now look like the image below. Click **OK** to close the Rule Designer.



When the Form is Initializing

then complete the following one after another

then on Leave Request Item View View, execute the Initialize method

then on Leave Request List View View, execute the Initialize method

then on Leave Request List View View, execute the Get List method (configure)

then transfer data (configure)

STEP 10 REVIEW

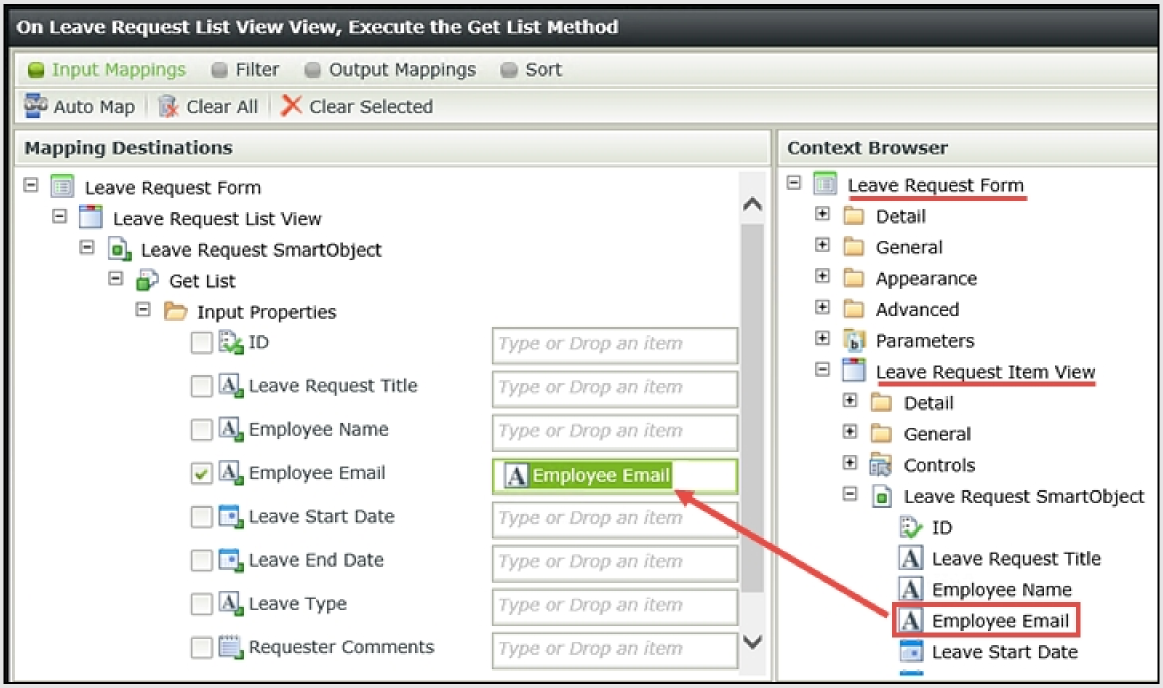
In this step, you edited the Form Initializing rule on the Leave Request Workflow (Default) state, so that the current user's previous leave requests will populate the Leave Request List View when the form is first loaded. The Default state is the form configuration that the form originator will see. You configured the current user's email as the value K2 will use to search for records in the Leave Request SmartObject. You also configured the system values for the Display Name and Email (again, for the current user) to populate their corresponding fields on the Item View. System Values are a convenient mechanism for using the current user's details as filters and values throughout the workflow.

Step 11: Add a rule to populate the leave request details on the Manager Approval state

In this step, you will edit a rule so that the approving manager sees the originator's previous leave requests. You will hide the Create button, since the approving manager doesn't need to create a record. K2 will automatically add a third view to the Leave Request form specifically for the approving manager, which will contain your actions (Approved/Rejected) and a submit button.

Step 11 Tasks

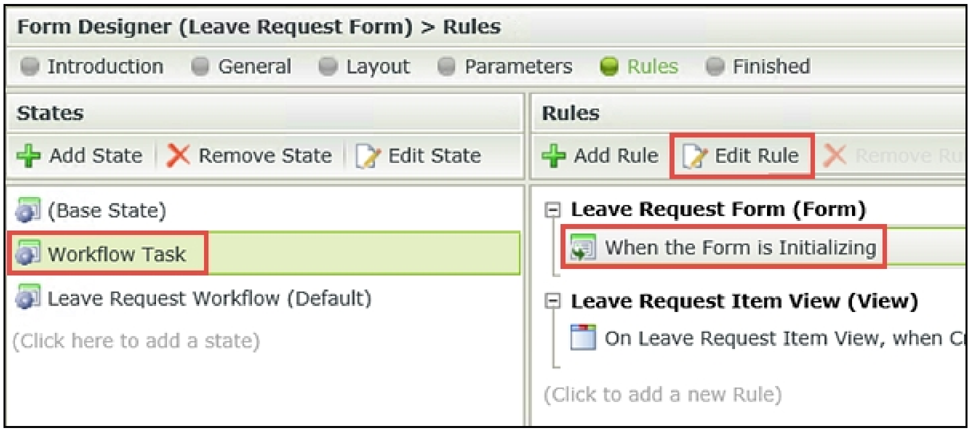
1. Edit the **Form Initializing** rule for the **Workflow Task** state. Execute the **Get List** method for the **Leave Request List View**.
2. For the **Input Property**, use the Leave Request SmartObject **Employee Email** property for the **Employee Email** input property.



3. Map all of the Return Properties for the Leave Request List View.
4. Still on the Workflow Task state, hide the **Create** button.

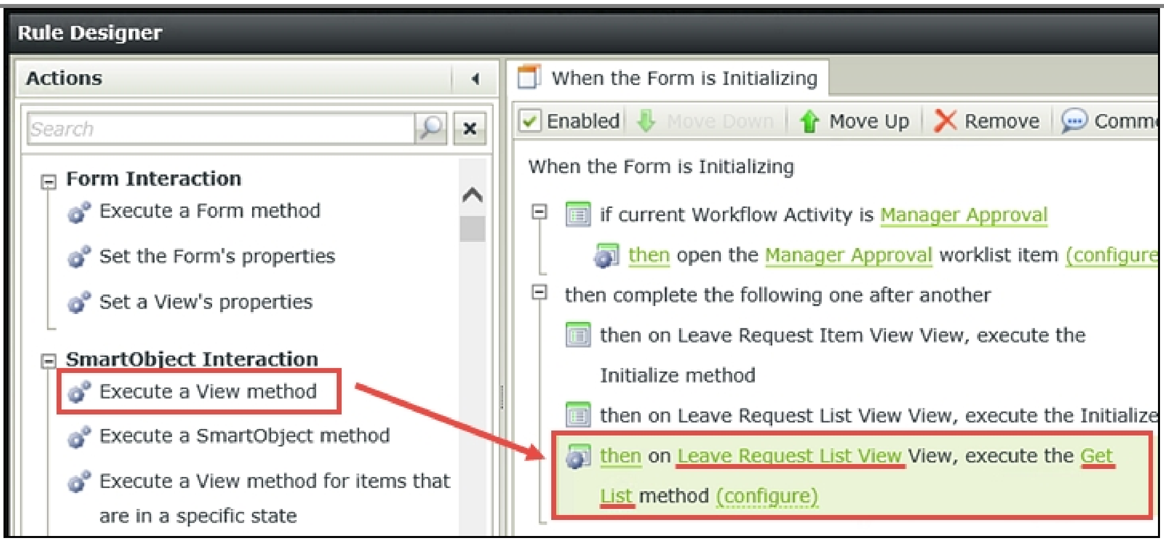
Step 11 Walkthrough

a. Continuing on with Rules, highlight the **Workflow Task** state and the **When the Form is Initializing** rule, then click **Edit Rule**. The Workflow Task state is the form and rule configuration that the manager will see when they approve or reject the request.


Form Designer (Leave Request Form) > Rules
Introduction
General
Layout
Parameters
Rules
Finished
States
Add State
Remove State
Edit State
(Base State)
Workflow Task
Leave Request Workflow (Default)
(Click here to add a state)
Rules
Add Rule
Edit Rule
Remove Ru
Leave Request Form (Form)
When the Form is Initializing
Leave Request Item View (View)
On Leave Request Item View, when C
(Click to add a new Rule)

First, you want to configure an action so that the originator's previous leave requests are loaded. Recall in previous steps, you told K2 to retrieve the previous requests of the current user. On the manager's form, you don't want the current user (that would be the manager), instead you need K2 to retrieve the *form originator's* records. You also want to hide the Create button for the manager, as it is not needed. K2 will add a third view to this form (the Workflow Task view) that contains the task actions (Approved, Rejected, etc.) and a submit button for the manager to use.

b. Click on **Execute a View method** to add it to the Rule Definition pane. Click the **select View** link and choose the **Leave Request List View**. Then click the **select method** link and choose **Get List**.



When the Form is Initializing

if current Workflow Activity is Manager Approval

then open the Manager Approval worklist item (configure

then complete the following one after another

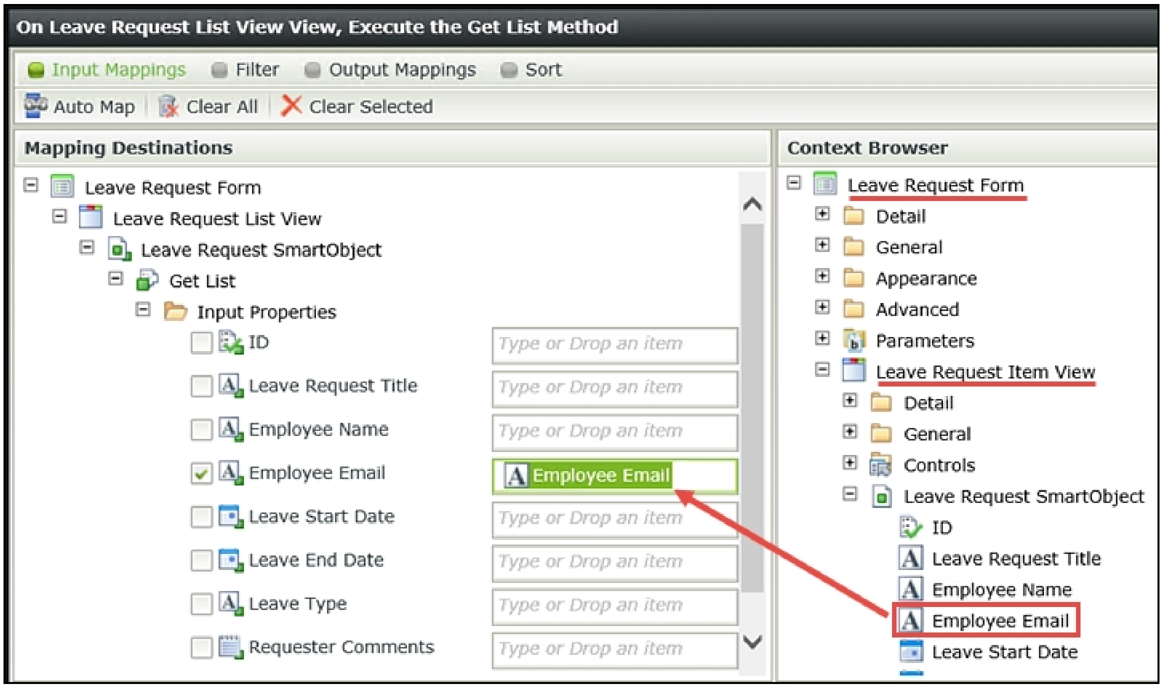
then on Leave Request Item View View, execute the Initialize method

then on Leave Request List View View, execute the Initialize

then on Leave Request List View View, execute the Get List method (configure)

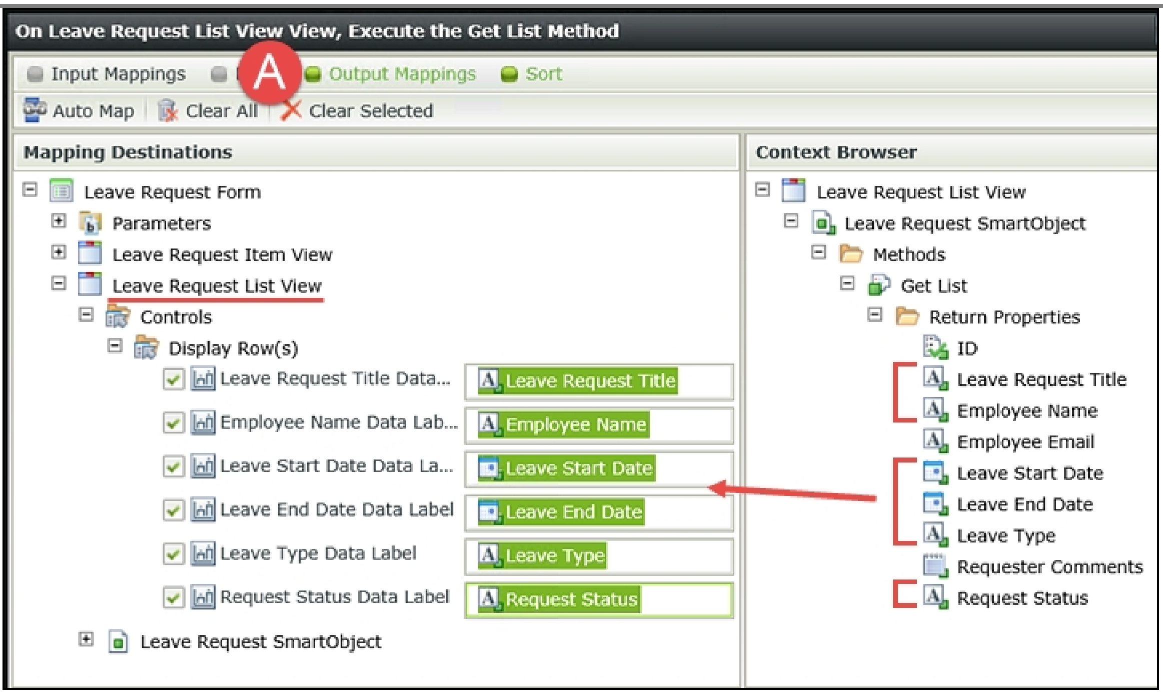
c. Click the **(configure)** link. Using steps very similar to those in Step 10, you will map the Input Properties and the Return Properties. But, instead of using the System Value for the email input, you will use the SmartObject value for the email input because you don't want the email of the current user, you want the email of the user who submitted the Leave Request.

In the Context Browser expand the **Leave Request Item View** node, then the **Leave Request** SmartObject. Drag the **Employee Email** into the Employee Email input property.

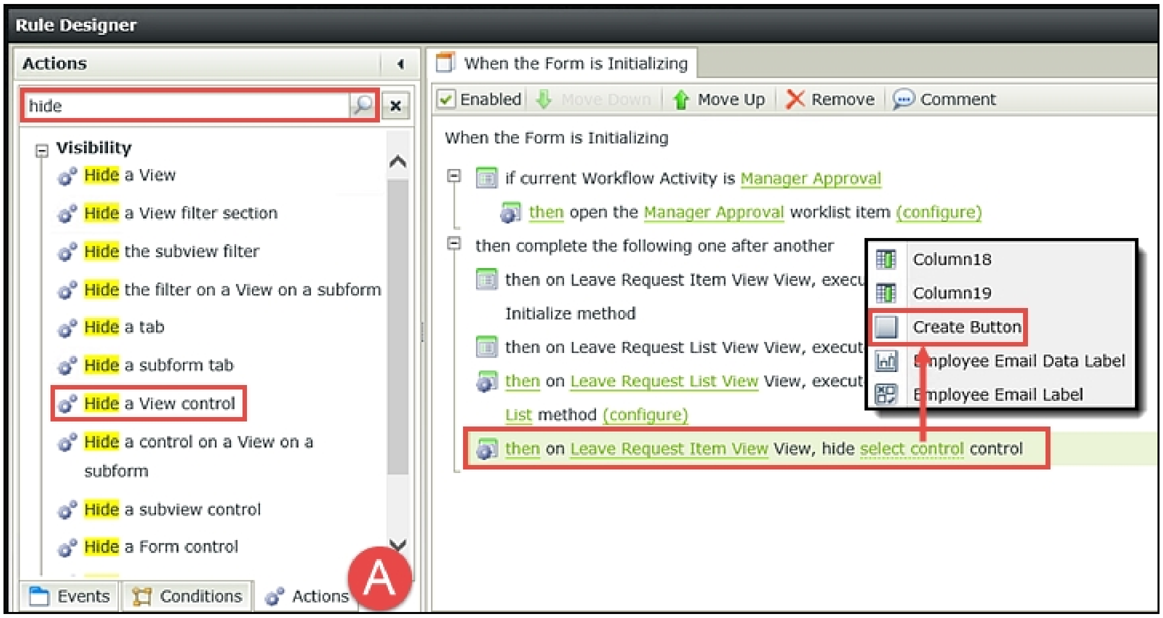


□ ID	Type or Drop an item
□ Leave Request Title	Type or Drop an item
□ Employee Name	Type or Drop an item
☑ Employee Email	Employee Email
□ Leave Start Date	Type or Drop an item
□ Leave End Date	Type or Drop an item
□ Leave Type	Type or Drop an item
□ Requester Comments	Type or Drop an item

d. Now switch to the **Output Mappings** screen (**A** below) and map the **Return Properties** for the Leave Request List View. Click **Finish** when ready to close the configuration screen, but do not exit the Rule Designer just yet.



e. The last configuration is to hide the Create button for the approving manager. With the Actions tab still highlighted (**A** below) search for the keyword
hide
then click once on **Hide a View control** to add it to the Rule Definition pane. Click the **select View** link and choose **Leave Request Item View**. Click the **select control** link and choose **Create Button**. Click **OK** to close the Rule Designer, then click **Finish** to complete the rule configurations.


Rule Designer
Actions
hide
Visibility
Hide a View
Hide a View filter section
Hide the subview filter
Hide the filter on a View on a subform
Hide a tab
Hide a subform tab
Hide a View control
Hide a control on a View on a subform
Hide a subview control
Hide a Form control
Events
Conditions
Actions
A
When the Form is Initializing
Enabled
Move Down
Move Up
Remove
Comment
When the Form is Initializing
if current Workflow Activity is Manager Approval
then open the Manager Approval worklist item (configure)
then complete the following one after another
then on Leave Request Item View View, execu
Initialize method
then on Leave Request List View View, execut
then on Leave Request List View View, execut
List method (configure)
then on Leave Request Item View View, hide select control control
Column18
Column19
Create Button
Employee Email Data Label
Employee Email Label

f. Right-click the Leave Request Form and select **Check In**. If you get a message about checking in associated views, click **OK** to continue.

STEP 11 REVIEW

In this final step to creating your basic Leave Request Approval application, you configured two actions for the Workflow Task state. This is the state that represents the form and rule configuration that the approving manager will interact with. First, you configured an action that will retrieve the originator's previous leave requests. Then, you hid the Create Button from the manager, as it is not needed.

Part 4: Test

Now it's time to test your basic Leave Request Approval application.

Step 12: Test the completed application

You will submit two Leave Request Forms. Using the link found in one of the task notification emails, you will open the worklist item and action it (mark it as "Approved" or "Rejected"). You will then open the Leave Request Form again to confirm that the previous two leave request records are appearing in the List View with the correct status value. (One record should say "Submitted" and one should say "Approved" or "Rejected".)

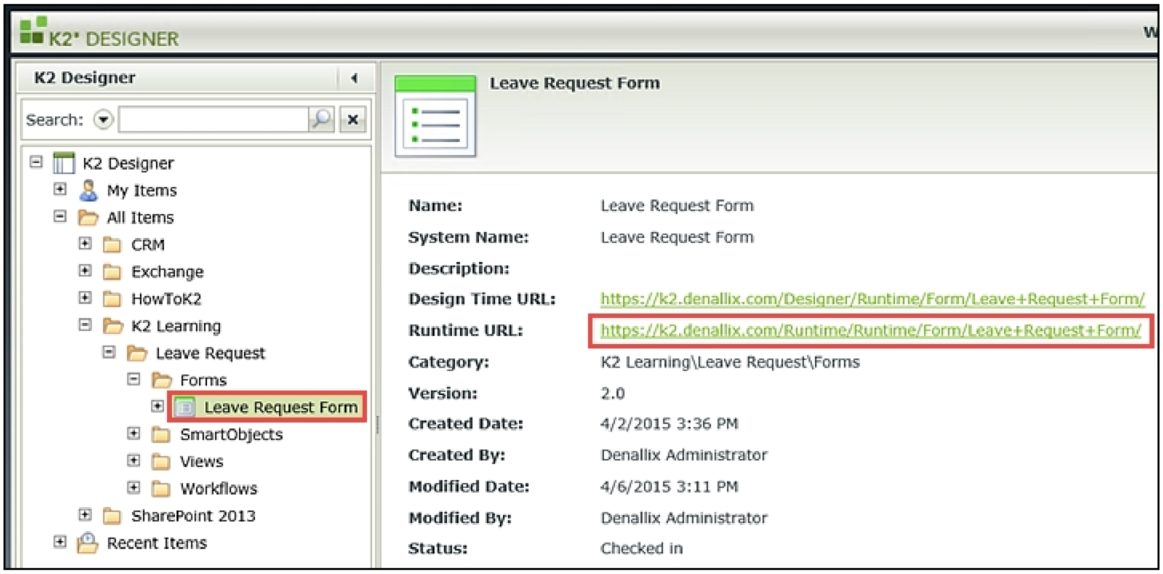
Step 12 Tasks

1. Use the **Runtime URL** to open and submit two Leave Request Forms.
2. Open the approving manager's Outlook, then confirm there are two task notification emails. **Action** one of the tasks.
3. Open the remaining worklist task and confirm the Request Status reflects the two results: Submitted and Approved/Rejected

Step 12 Walkthrough

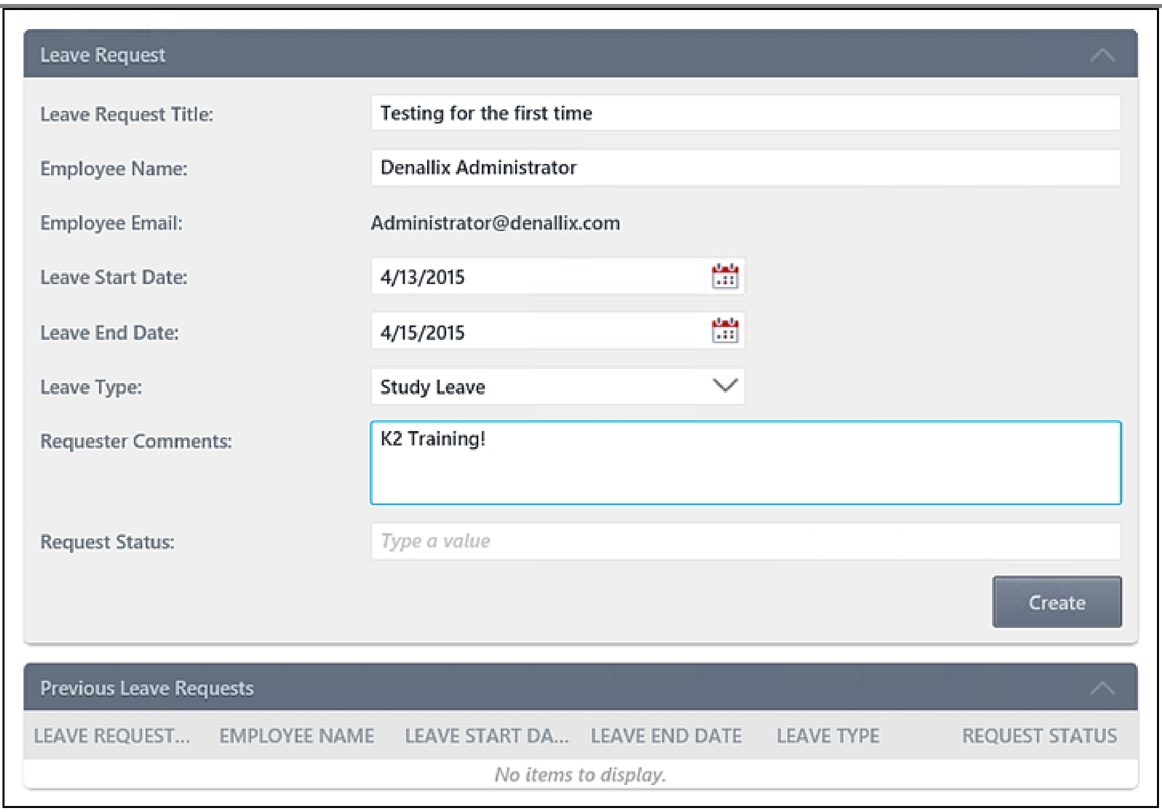
a. From K2 Designer, click to highlight the **Leave Request Form**. In the Properties pane, notice the **Runtime URL**. This is the link (or URL) to your form. You can provide this link to your end users so that they can submit their own forms. (You may need to click the little down-arrow icon in the top right of the panel to expand the properties for the form.)

b. Click the **Runtime URL** link. Depending on your environment, it may take a few seconds for the form to open.



Name: Leave Request Form
System Name: Leave Request Form
Description:
Design Time URL: https://k2.denallix.com/Designer/Runtime/Form/Leave+Request+Form/
Runtime URL: https://k2.denallix.com/Runtime/Runtime/Form/Leave+Request+Form/
Category: K2 Learning\Leave Request\Forms
Version: 2.0
Created Date: 4/2/2015 3:36 PM
Created By: Denallix Administrator
Modified Date: 4/6/2015 3:11 PM
Modified By: Denallix Administrator
Status: Checked in

c. Complete the form fields with the exception of the Request Status field, then click the **Create** button



Leave Request

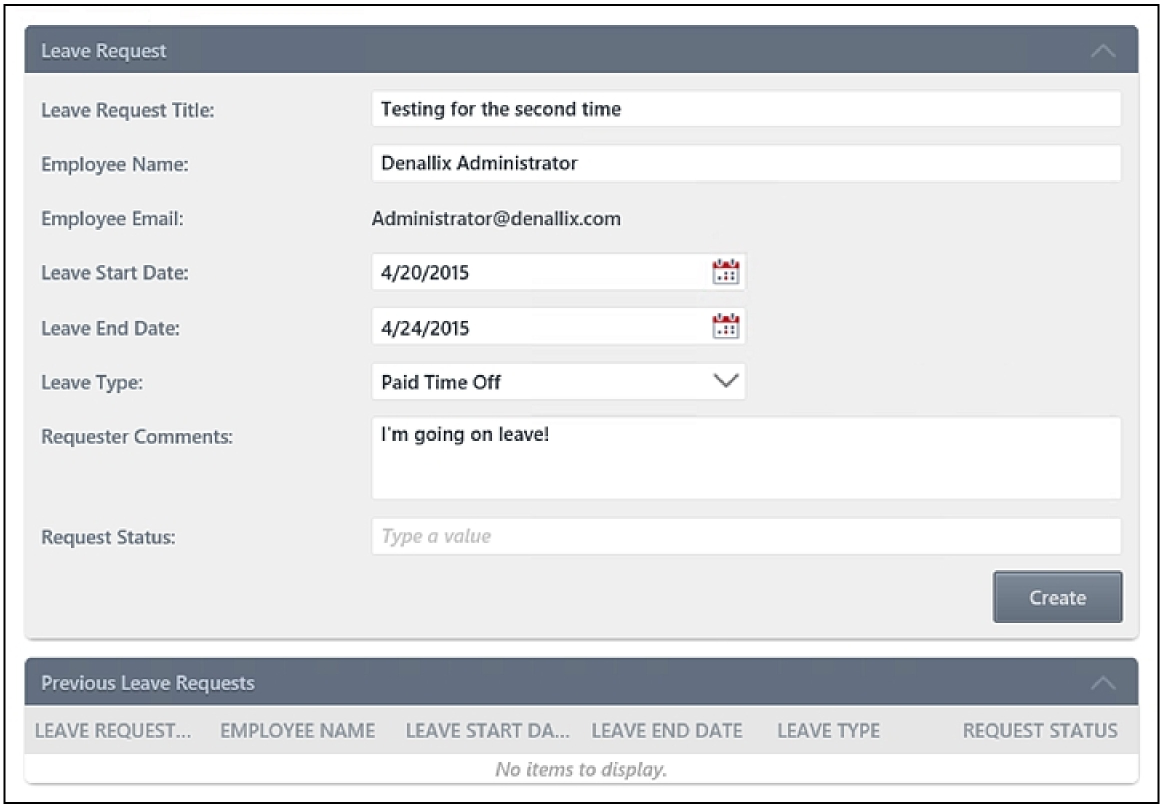
Leave Request Title:	Testing for the first time
Employee Name:	Denallix Administrator
Employee Email:	Administrator@denallix.com
Leave Start Date:	4/13/2015
Leave End Date:	4/15/2015
Leave Type:	Study Leave
Requester Comments:	K2 Training!
Request Status:	Type a value

Previous Leave Requests

LEAVE REQUEST...	EMPLOYEE NAME	LEAVE START DA...	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS

No items to display.

d. You should see an animated processing icon, then you will be returned to the form as you left it. Change the fields (title, dates, etc.) and click the **Create** button again. Once again, the form processes very quickly and you are returned to the form as you left it. If you did not receive any errors, close the browser.
(Usually, you would add some kind of confirmation message to indicate to the user that their request was submitted. You will add this functionality in the extended version of the Leave Request Approval application)



Leave Request

Leave Request Title:	Testing for the second time
Employee Name:	Denallix Administrator
Employee Email:	Administrator@denallix.com
Leave Start Date:	4/20/2015
Leave End Date:	4/24/2015
Leave Type:	Paid Time Off
Requester Comments:	I'm going on leave!
Request Status:	Type a value

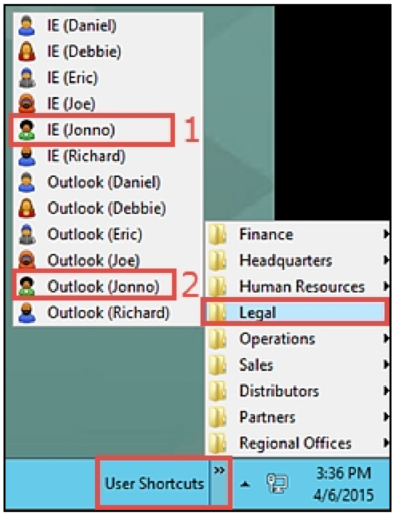
Previous Leave Requests

LEAVE REQUEST...	EMPLOYEE NAME	LEAVE START DA...	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS

No items to display.

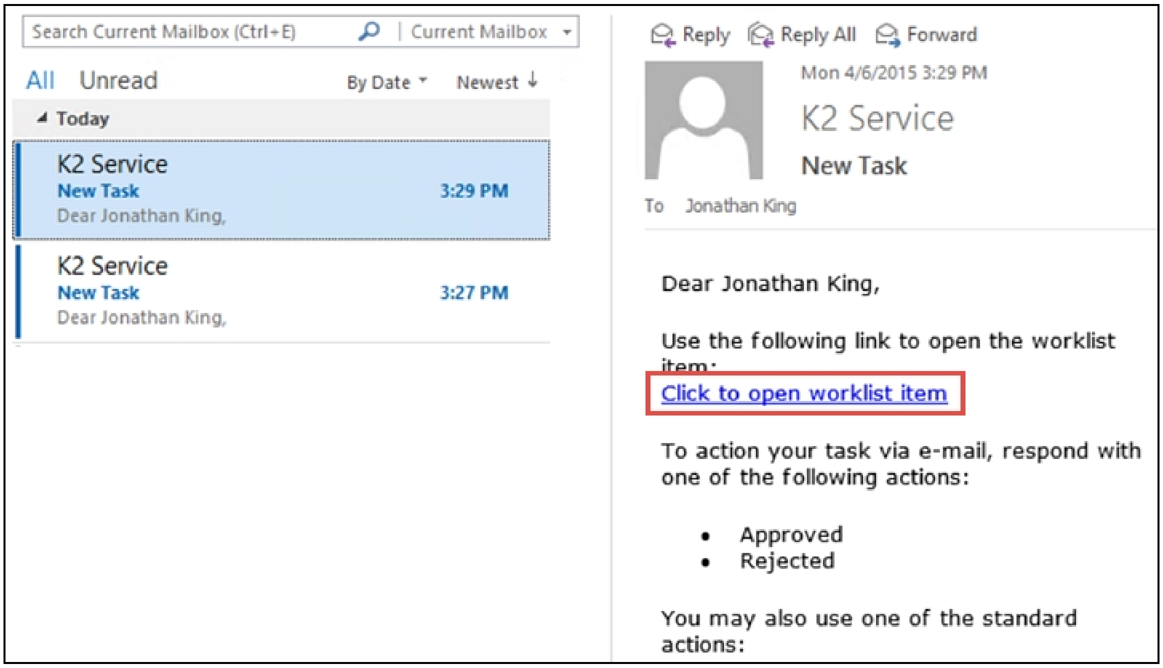
Now you need to open the approving manager's Outlook and confirm the task notification emails were received. Then you will action one of the tasks. If you are working on a K2-provided virtual environment (VM), you are likely logged in as Denallix Administrator. Administrator's manager is Jonno, so you'll open Jonno's Outlook. (If you are building this application in your own environment, you will need to access the appropriate Outlook account of the Manager for the user that submitted the leave request or whatever user you selected for the Manager Approval task when you designed the workflow.)

e. If using a K2-provided virtual machine, close all open Internet Explorer windows. Then, click the **User Shortcuts** arrows found in the lower-right corner of the screen. Expand the **Legal** folder and click **IE (Jonno)** as well as **Outlook (Jonno)** to open IE and Outlook as Administrator's manager, Jonno.



Jonno's Outlook will begin to open. This can take a minute or two when using a virtual machine environment. If you see a message about synchronizing CRM data, continue to wait. If you see a message about licensing, close to continue.

f. Confirm there are two task notification emails. In one of the emails, **Click to open the worklist item** link to open the form and action the request.



Mon 4/6/2015 3:29 PM

K2 Service

New Task

To Jonathan King

Dear Jonathan King,

Use the following link to open the worklist item:

Click to open worklist item

To action your task via e-mail, respond with one of the following actions:

- Approved
- Rejected

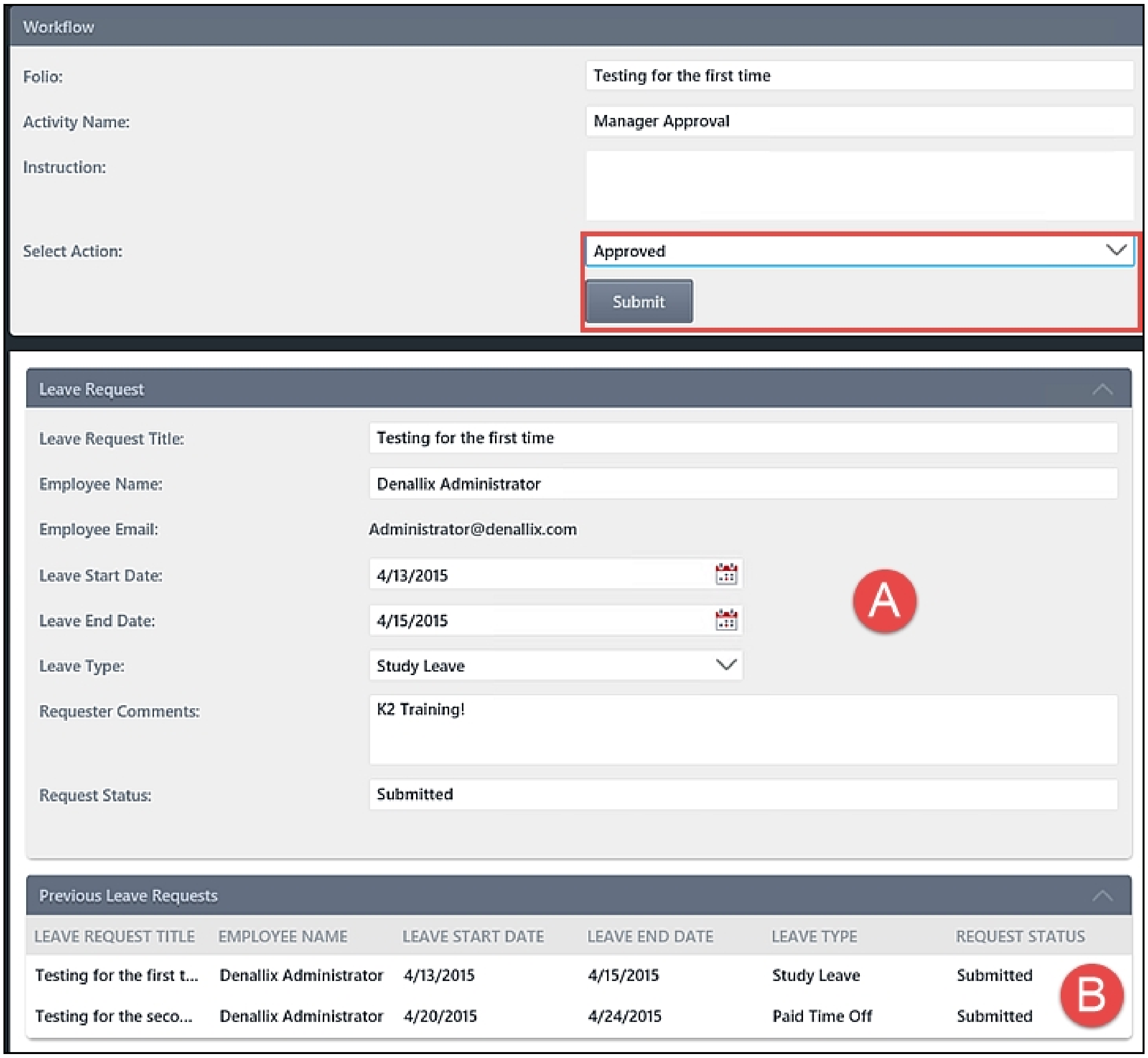
You may also use one of the standard actions:

Note
Notice the Workflow view that K2 has placed at the top of the form. This is the default panel that K2 added to

the form when you workflow-enabled the form by adding it to the Manager Approval task. The approver will select their decision from the drop-down list and then click the Submit button to complete the task. This workflow panel is not the only way to workflow-enable forms, you can also define your own controls and rules to workflow-enable a form. For the purposes of this exercise you are just using the simplest approach to workflow-enable forms.

Confirm that the Leave Request details are correct (**A** below) and then confirm that both form submissions show up in the **Previous Leave Requests** view with the Request Status as Submitted. (**B** below)

g. Action the request by selecting one of the Action items, then click **Submit**. This time, you will receive K2's generic confirmation message. Click **OK** to close the dialog and then exit the browser.



Workflow

Folio:	Testing for the first time
Activity Name:	Manager Approval
Instruction:	
Select Action:	Approved

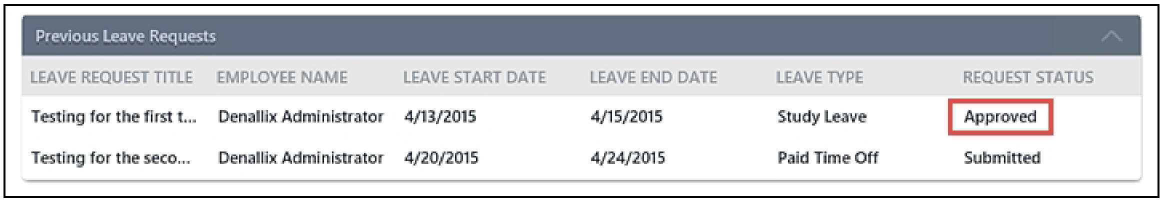
Leave Request

Leave Request Title:	Testing for the first time
Employee Name:	Denallix Administrator
Employee Email:	Administrator@denallix.com
Leave Start Date:	4/13/2015
Leave End Date:	4/15/2015
Leave Type:	Study Leave
Requester Comments:	K2 Training!
Request Status:	Submitted

Previous Leave Requests

LEAVE REQUEST TITLE	EMPLOYEE NAME	LEAVE START DATE	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
Testing for the first t...	Denallix Administrator	4/13/2015	4/15/2015	Study Leave	Submitted
Testing for the seco...	Denallix Administrator	4/20/2015	4/24/2015	Paid Time Off	Submitted

h. Open the second worklist item. In the Previous Leave Requests list, confirm that the Request Status value has now changed for the request that you approved in the previous step. If you wish, you can Approve or Reject this second leave request.



Previous Leave Requests

LEAVE REQUEST TITLE	EMPLOYEE NAME	LEAVE START DATE	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
Testing for the first t...	Denallix Administrator	4/13/2015	4/15/2015	Study Leave	Approved
Testing for the seco...	Denallix Administrator	4/20/2015	4/24/2015	Paid Time Off	Submitted

STEP 12 REVIEW

In this step, you tested your Leave Request Approval application by submitting two forms using the Runtime URL. The Runtime URL is the "user" link that allows your users to access and submit your forms. (Note: depending on your K2 environment setup, this link would probably only be available to internal users in your organization but there are ways of exposing SmartForms to the outside world.) You confirmed the task notification emails were sent and after actioning one of the worklist items, you confirmed the Request Status property was updated appropriately in the Leave Request SmartObject.

Summary

The exercises in the Leave Request Approval (Basic) Application are meant to provide an introduction to building K2 Applications and the K2 components of an Application: **Data**, **Forms** and **Workflows**. By completing the five parts, you should have a basic understanding of how data, forms and workflows integrate with each and how rules can be implemented to provide dynamic functionality to your user interface (for example, auto-populating form fields with the current user details). Key to these exercises are the following:

Data

- SmartObjects are "connectors" to data sources and are the bridge between providers of data (data sources) and consumers of data (forms, workflows, etc.).
- SmartBox is K2-provided storage for creating your own data source from "scratch".
- SmartObject methods can be called from views, forms and workflows (for example, Get List).

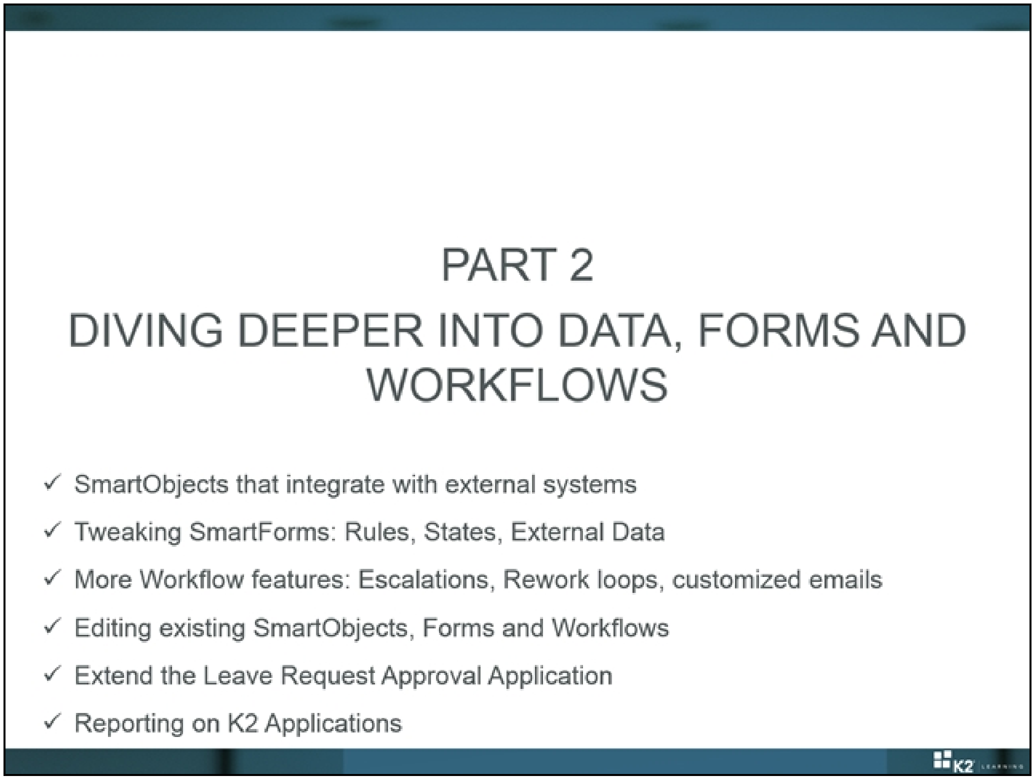
Forms

- Forms are containers for views and controls.
- Views are logical sections of form content.
- There are two types of views: Item Views contain the content from one record. List Views contain multiple records.
- Rules are comprised of Events, Conditions and Actions. Events are *when* something occurs, Conditions are *if a specific criteria has been met* and Actions *do something* if the Event and Condition has been met.
- Rules can be applied to different workflow states, essentially creating custom forms for workflow events.
- Forms and views must be checked in before they can be exposed to users.

Workflows

- Workflows created in K2 Designer are comprised of steps and events and are connected with outcome lines.
- There are two types of events: System Tasks are performed by the K2 server such as sending an email or updating a SmartObject. User, or Client Tasks are performed by a human, such as making a decision of some kind.
- Workflows must be deployed before they can be exposed to users.

Part 2: Diving Deeper into Data, Forms and Workflows



PART 2

DIVING DEEPER INTO DATA, FORMS AND WORKFLOWS

- ✓ SmartObjects that integrate with external systems
- ✓ Tweaking SmartForms: Rules, States, External Data
- ✓ More Workflow features: Escalations, Rework loops, customized emails
- ✓ Editing existing SmartObjects, Forms and Workflows
- ✓ Extend the Leave Request Approval Application
- ✓ Reporting on K2 Applications

Video

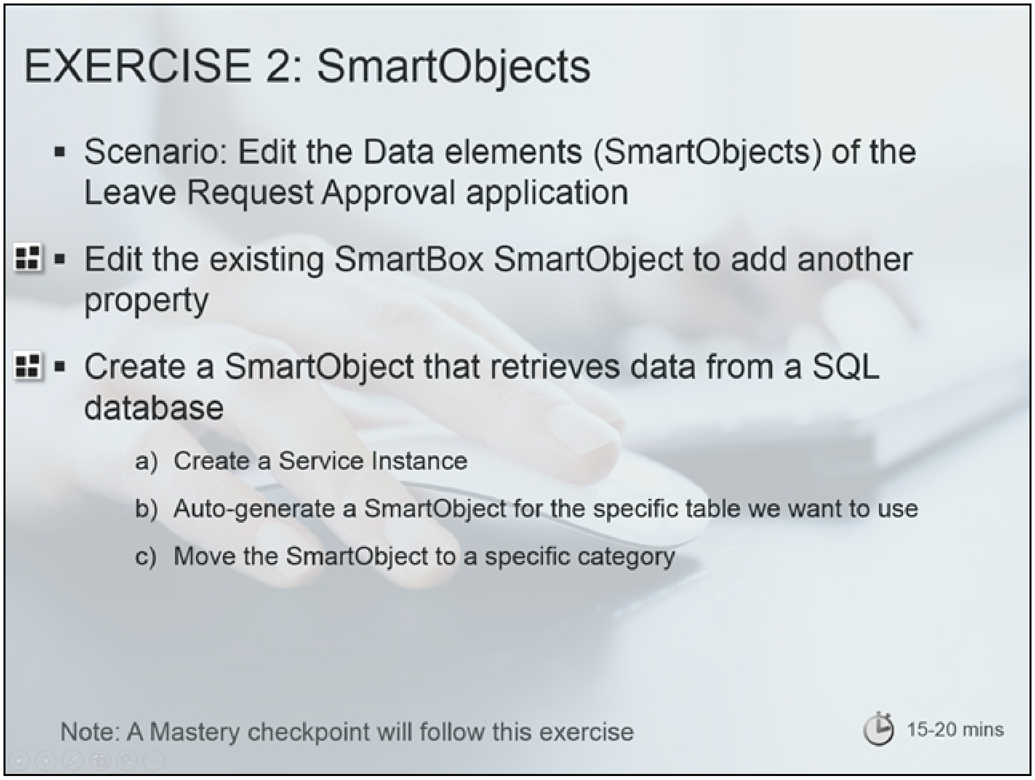
In Part 2 of this module we will dive deeper into the Data, Forms and Workflow components of K2 applications. We separate Part 2 into Data (with a exercise), then Forms (with an exercise) and finally Workflows (with an exercise). In the exercises for this module, you will expand on the basic version for the Leave Request Approval application that you created in Part 1, and add more functionality and features to the application.

You will learn:

- How to use SmartObjects to integrate with an external system
- A little more detail on working with SmartForms
- More detail on workflow components like:
  - Escalations
  - Rework loops
  - Customized emails and task notification messages
  - More advanced outcomes like multiple approvals
- How to report on K2 Applications

Note
Note: this module is not intended to cover K2 smartforms in-depth, so we will keep the SmartForms aspect light. If you will be using SmartForms, there is a separate training course that covers SmartForms in much more detail. We are using SmartForms since it is one of the easiest ways to build user interfaces without any third-party tools like SharePoint or Visual Studio.

EXERCISE 2: SmartObjects



EXERCISE 2: SmartObjects

- Scenario: Edit the Data elements (SmartObjects) of the Leave Request Approval application
- Edit the existing SmartBox SmartObject to add another property
- Create a SmartObject that retrieves data from a SQL database
  a) Create a Service Instance
  b) Auto-generate a SmartObject for the specific table we want to use
  c) Move the SmartObject to a specific category

Note: A Mastery checkpoint will follow this exercise

15-20 mins

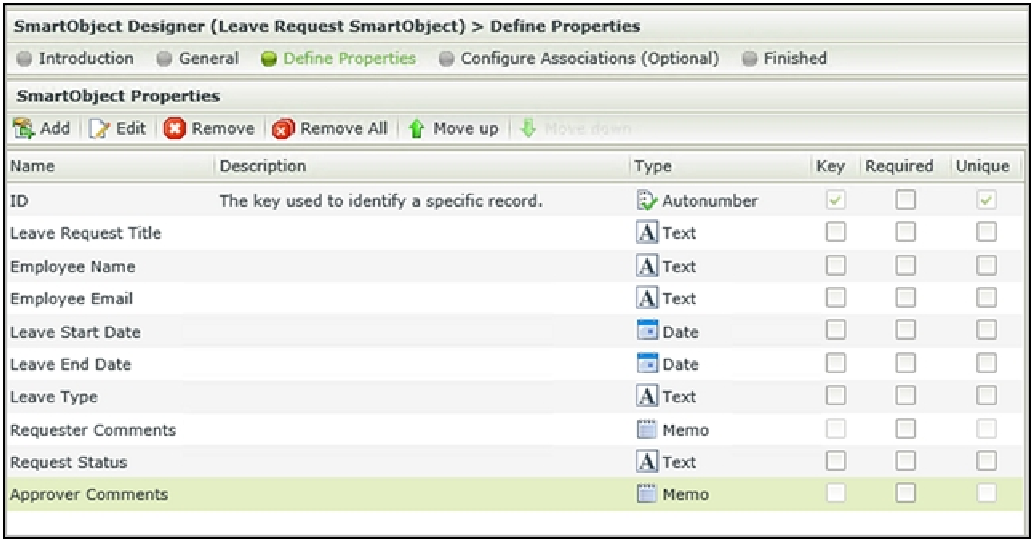
Video

This exercise goes into a little more depth on SmartObjects, the "Data" component of our Leave Request Approval application. We will edit the existing Leave Request SmartObject to add another field, and we will create a new SmartObject that integrates with a Microsoft Azure SQL database. (We will use this SmartObject in a later exercise to populate a drop-down list of leave types from the database).

You will also learn how to register a Service Instance, auto-generate SmartObjects and move SmartObjects in the Category System.

After this exercise we will do a quick mastery check on SmartObjects.

Adding an "Approver Comments" property to the existing SmartObject

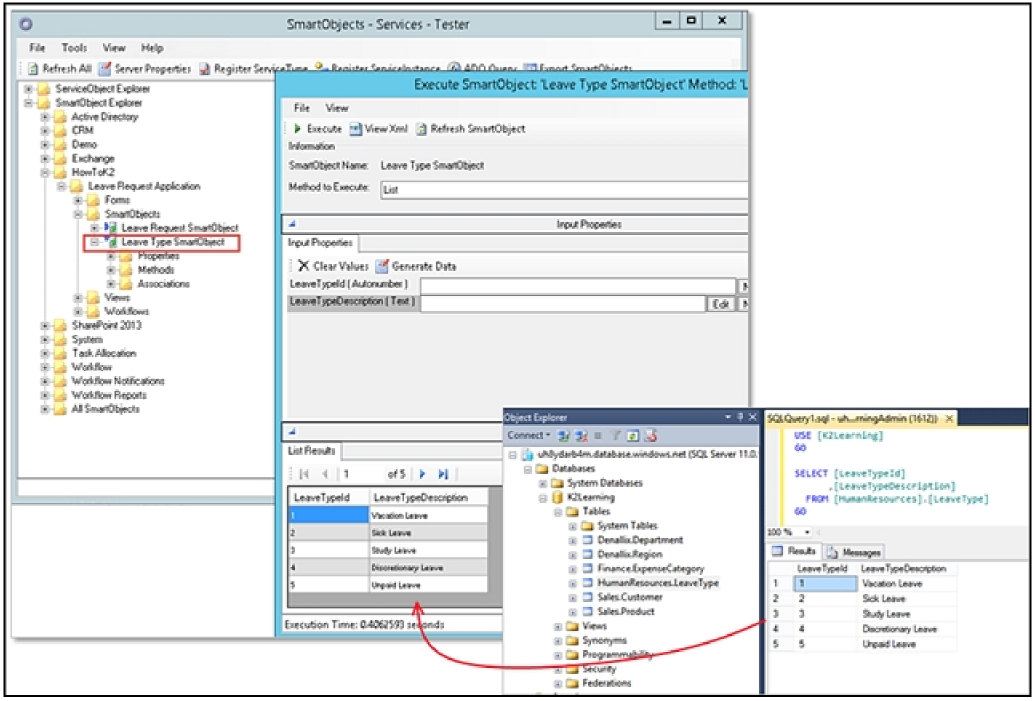


SmartObject Designer (Leave Request SmartObject) > Define Properties

SmartObject Properties

Name	Description	Type	Key	Required	Unique
ID	The key used to identify a specific record.	Autonumber	☑	□	☑
Leave Request Title		Text	□	□	□
Employee Name		Text	□	□	□
Employee Email		Text	□	□	□
Leave Start Date		Date	□	□	□
Leave End Date		Date	□	□	□
Leave Type		Text	□	□	□
Requester Comments		Memo	□	□	□
Request Status		Text	□	□	□
Approver Comments		Memo	□	□	□

The SmartObject we will use to retrieve leave types from the external SQL database



SmartObject Name: Leave Type SmartObject

Method to Execute: List

LeaveTypeId (Autonumber)

LeaveTypeDescription (Text)

LeaveTypeId	LeaveTypeDescription
1	Vacation Leave
2	Sick Leave
3	Study Leave
4	Discretionary Leave
5	Unpaid Leave

Execution Time: 0.4062593 seconds

```
USE [K2Learning]
GO

SELECT [LeaveTypeId]
      ,[LeaveTypeDescription]
  FROM [HumanResources].[LeaveType]
GO
```

	LeaveTypeId	LeaveTypeDescription
1	1	Vacation Leave
2	2	Sick Leave
3	3	Study Leave
4	4	Discretionary Leave
5	5	Unpaid Leave

When you are ready, continue on to " Leave Request Approval (V2 or "Extended" version)" below to review the design of the extended version of the Leave Request application, and then you can continue on to the "Part 1: Data" on page 61 exercise to build the enhanced SmartObjects for the application.

Leave Request Approval (V2 or "Extended" version)

This tutorial explains how to build a more advanced application by extending the basic version of the Leave Request application and provides more advanced learning in the following components of K2 applications: Data, Forms and Workflow. This tutorial is intended for users new to K2, or with little K2 experience.

Note
This tutorial assumes that you have already completed the Leave Request (V1 or "Basic" version). If you have not, please complete the basic version first, since this tutorial depends on and extends elements that were built in the basic version. If you are already familiar with the basic principals of K2 smartforms and workflow, you may download and deploy the basic solution in lieu of completing the step-by-step instructions. The solution for the basic version is provided with that tutorial.

This tutorial can be completed in any environment that has K2 version 4.7 or later installed. K2-delivered training events normally include access to a Virtual Server environment provided by K2 which you will use for the exercises. The screen shots and users in the tutorial reflect this K2-provided virtual environment. You can, however, complete this tutorial in any other environment as long as the necessary K2 components are installed and operational. You must also have the necessary rights and permissions to create the K2 artifacts in the target environment.

Some tutorials require that you have internet access to be able to connect to an external SQL database used for retrieving data. Those tutorials contain expanded information on internet connection requirements.

Tutorial Overview

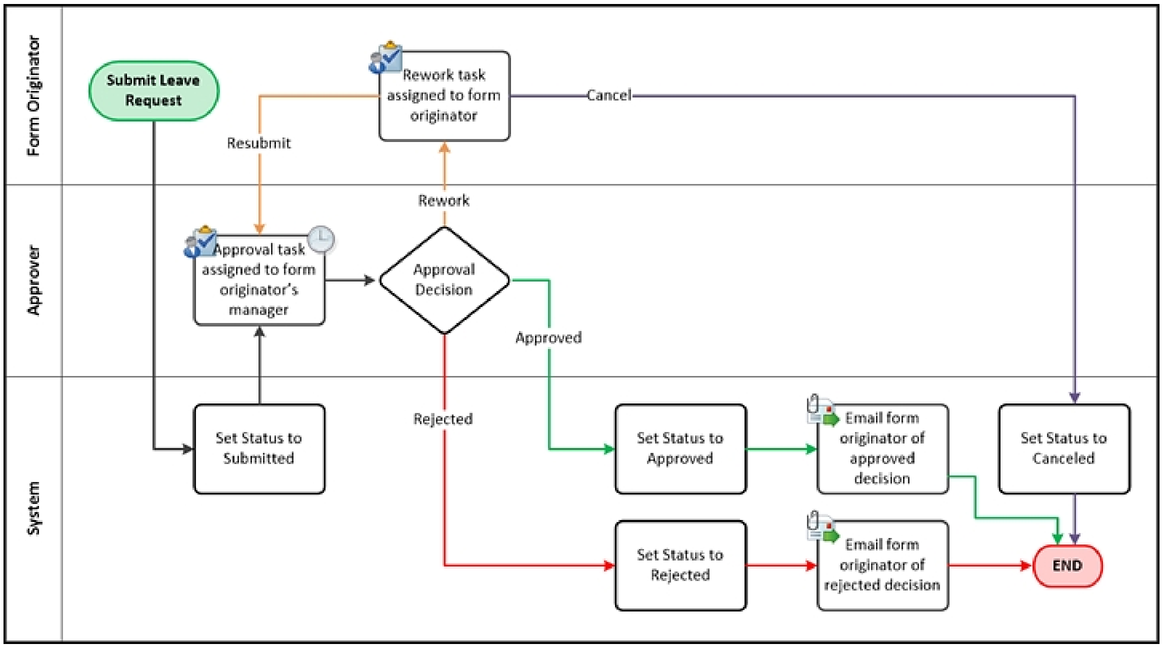
You may recall that K2 Applications consist of four main components: **Data**, **Forms**, **Workflow** and **Reports**.

In Part 1, you expand upon the **Data** component by editing the existing Leave Request SmartObject and adding an additional property. You'll open the K2 Management site and build a service instance based on the SQL Server Service Type. You will then auto-generate a new SmartObject called Leave Types, which will connect to an external SQL data source. In Part 2, you will expand the **Forms** component by adding the new Leave Types SmartObject as the data source for the Leave Type drop-down list. In Part 3, you will expand on the **Workflow** component by adding additional email events and a new outcome called Rework. You will also return to **Forms** where you will edit rules and add actions that will incorporate your data and workflow updates. In Part 4, you will test the new version of the Leave Request application.

Prior to jumping into building the application, it is important to understand what you want to achieve, by designing the necessary pieces of the application. Let's start with the workflow.

Leave Request Workflow

As before, you can use a logical flow diagram to represent the various steps and actors in the workflow. The diagram below shows the extended version of the workflow. Note that there is a rework loop so that the approver can send the request back for rework, and there are additional steps in the workflow to send e-mail notifications to the requester. You will also add an escalation to the manager approval step to remind the approver and originator if the request is not approved by two days before the start date of the leave request.

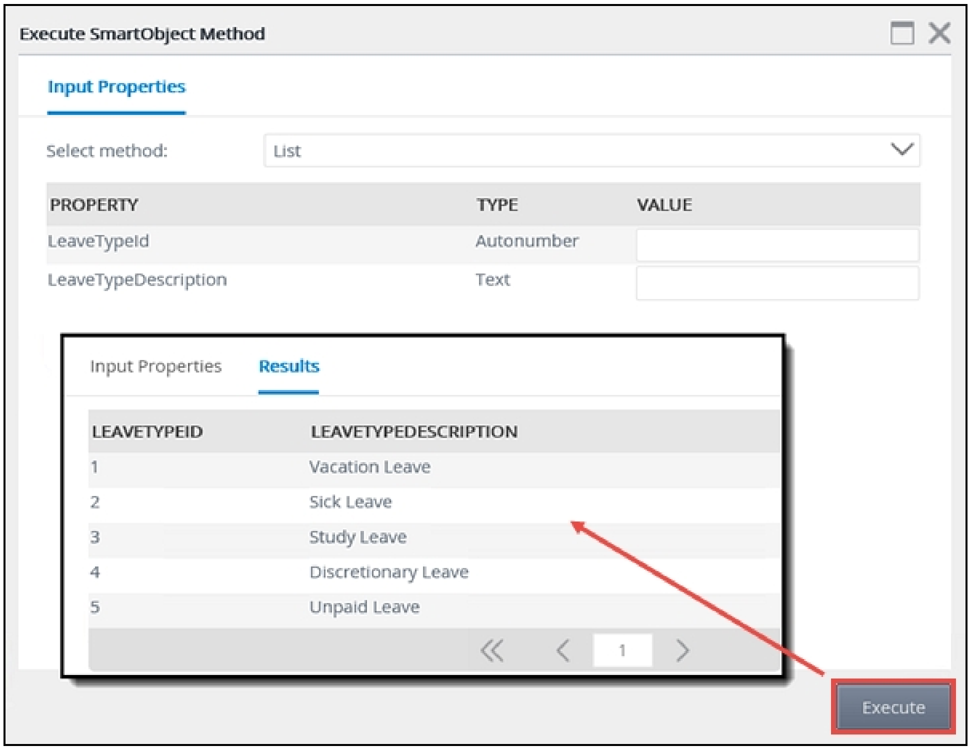


Leave Request Data

The first change to the data design is that you want to provide a field where the approver can capture comments so that the requester knows why their request was sent back or rejected. To do so, you will edit the existing Leave Request SmartObject and add an additional data field called Approver Comments.

You may recall that SmartObjects can also interact with other systems. SmartObjects are essentially a "middle layer or connector" that allows consumers of data (such as forms and workflows) to interact with providers of data (such as SQL databases, Active Directory or SharePoint, among others). In this version of the application, you will leverage this power by creating a SmartObject which retrieves data from a externally located SQL database.

The screen shot below shows the data that is returned after querying the external SQL database. You will replace the static Leave Type drop-down list values that you created in the basic version, with values from the new Leave Types SmartObject.



Execute SmartObject Method

Select method: List

PROPERTY	TYPE	VALUE
LeaveTypeId	Autonumber	
LeaveTypeDescription	Text	

LEAVETYPEID	LEAVETYPEDESCRIPTION
1	Vacation Leave
2	Sick Leave
3	Study Leave
4	Discretionary Leave
5	Unpaid Leave

Note

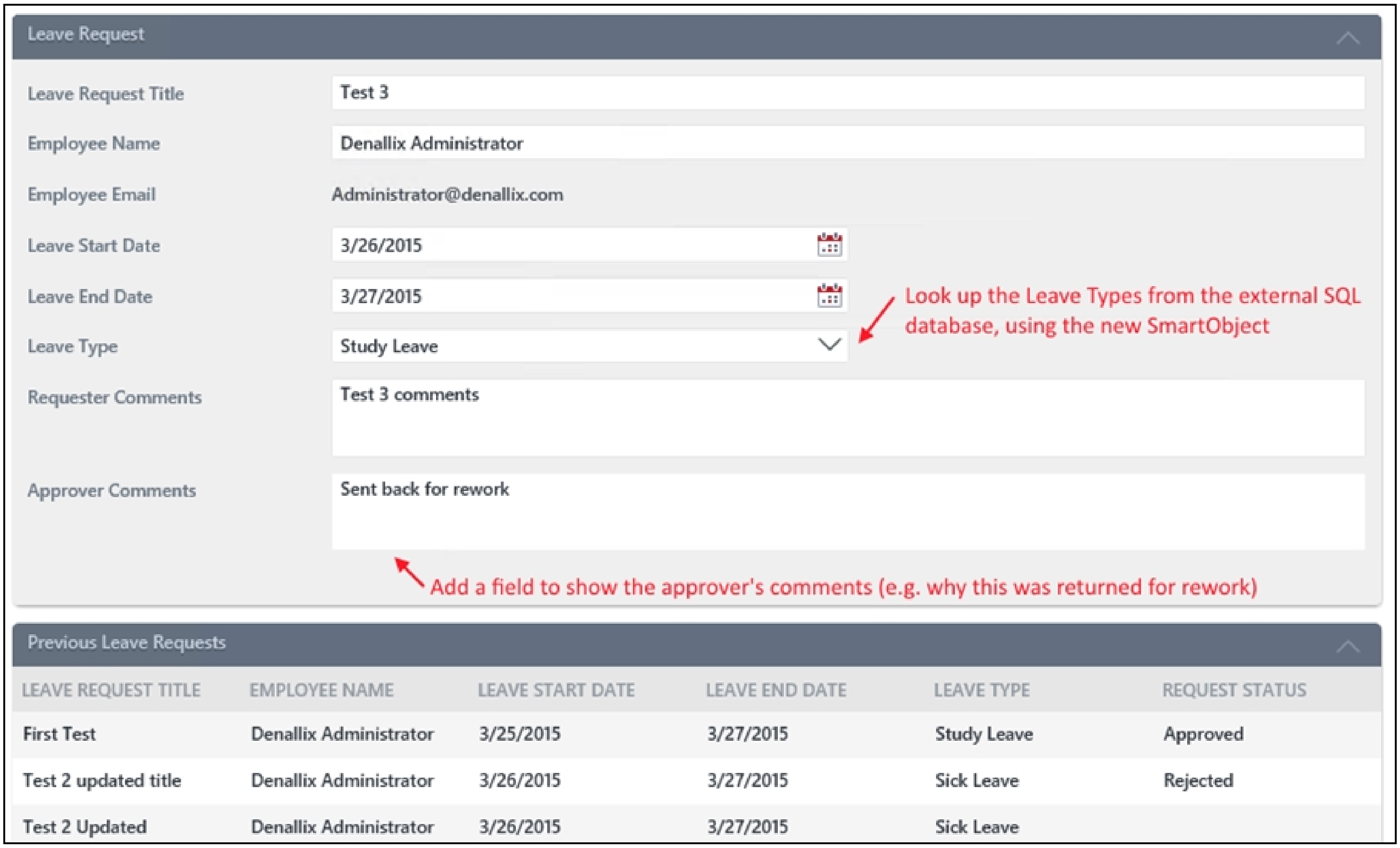
The integration data sources used in this application (the Azure SQL database that provides the Leave Type data) is intended only for tutorial and demonstration purposes. It is not intended for use in production applications and is not supported or guaranteed by K2. The SQL database used for this tutorial runs on SQL Azure. Connecting to the database will require an open port for port 1433 as described in this Microsoft KB article: http://support.Microsoft.com/KB/287932

If you are unable to connect to the provided SQL Azure database, you will need to open up the appropriate firewall rules or ports, or alternatively install the database on a local SQL environment using the provided SQL scripts located at http://help.k2.com/files/8553. Please contact your SQL database administrator for more information and help.

Leave Request Forms

The forms in the application will be adjusted to include the new Approver Comments field, and this field will be locked down so that it can only be changed when the approver is reviewing the request.

Additionally, you will update the Leave Type drop-down list to use the new SmartObject which retrieves the leave type values from the external SQL database. You will also make a few minor changes to the form's behavior so that it is a little more user-friendly, such as clearing out the data after a new request is submitted.



Leave Request

Leave Request Title	Test 3
Employee Name	Denallix Administrator
Employee Email	Administrator@denallix.com
Leave Start Date	3/26/2015
Leave End Date	3/27/2015
Leave Type	Study Leave
Requester Comments	Test 3 comments
Approver Comments	Sent back for rework

Look up the Leave Types from the external SQL database, using the new SmartObject

Add a field to show the approver's comments (e.g. why this was returned for rework)

Previous Leave Requests

LEAVE REQUEST TITLE	EMPLOYEE NAME	LEAVE START DATE	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
First Test	Denallix Administrator	3/25/2015	3/27/2015	Study Leave	Approved
Test 2 updated title	Denallix Administrator	3/26/2015	3/27/2015	Sick Leave	Rejected
Test 2 Updated	Denallix Administrator	3/26/2015	3/27/2015	Sick Leave	

Building the application

Now that you have designs for each of the elements of the extended version of the Leave Request application, you can start building them. As before, you will start with the data elements, then the form elements and finally the workflow enhancements.

When you are ready to start building the application, continue on to the Part 1: Data section to get started.

Part 1: Data

Part 1 of this tutorial will extend your knowledge of K2 SmartObjects and how you can define SmartObjects that interact with other systems. SmartObjects are essentially a "middle layer" or "connector" that allows consumers of data (such as forms and workflows) to interact with providers of data (such as SQL databases, Active Directory or SharePoint, among others).

In this part, you will extend the Leave Request SmartObject by adding a field to store approver comments. You will also create a new SmartObject (along with the underlying configuration) to connect to a SQL database. The results returned will be a list of leave types that will automatically populate the Leave Type drop-down list.

Step 1: Add the Approver Comments property to the Leave Request SmartObject

This step will demonstrate how you can add additional properties to existing SmartBox SmartObjects. You will add the Approver Comments property to the Leave Request SmartObject. (Later on, you will add this field to the Leave Request item view).

Step 1 Tasks

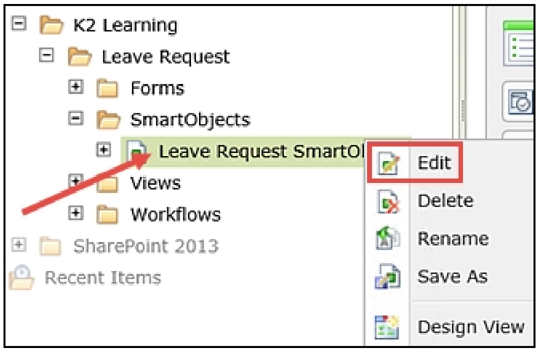
1. In K2 Designer, edit the **Leave Request SmartObject** and add a **Memo** type property called *Approver Comments*

Step 1 Walkthrough

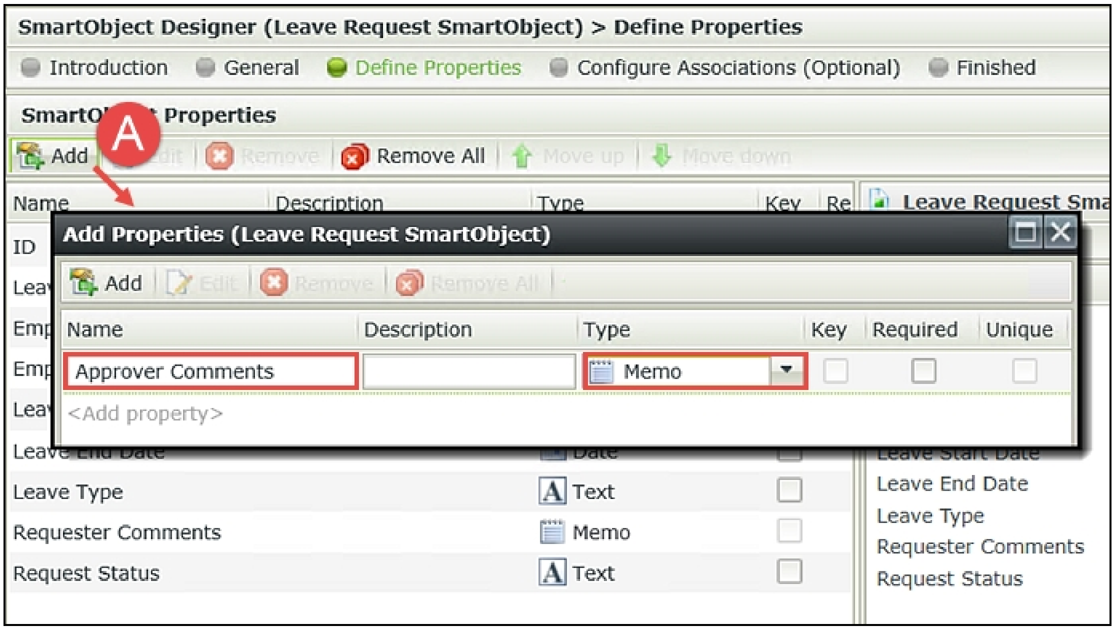
a. Launch K2 Designer (**Start Button** > **All Programs** > **K2 blackpearl** > **K2 smartforms** > **K2**

Designer).

b. Expand the **All Items** > **K2 Learning** > **Leave Request** > **SmartObjects** categories. Right-click the **Leave Request SmartObject** and select **Edit**.



c. On the SmartObject Properties screen, click **Add**. (**A** below) For the **Name**, enter *Approver Comments* then select the **Memo** option from the data **Type** drop-down. Click **OK**, then click **Finish** to save the SmartObject edit.



Note
Adding a new property to a SmartBox SmartObject in K2 Designer is a simple process and the change is published as soon as you **Finish** the SmartObject Designer. This means that the Approver Comments property you just added is ready for use in your Leave Request item view. Use caution when deleting properties: if you reference a deleted SmartObject property in a rule or workflow for example, that rule or workflow will likely throw an error because it is trying to read a property that does not exist anymore.

STEP 1 REVIEW

In this step, you added a new property to the Leave Request SmartBox SmartObject to hold the approver's comments. Adding a new property to a SmartBox is reflected as soon as you finish the designer, meaning the new property is ready for use in the Leave Request item view that is based off of the SmartBox SmartObject.

Step 2: Create a SQL Server service instance and auto-generate a SmartObject from the service instance

In Step 2, you will create a connection called a "service instance" to an external SQL database, then generate a SmartObject for a specific table in that database (leave types). The data from the leave types table will be used to automatically populate the Leave Type drop-down list in the Leave Request item view.

Note
New to K2 blackpearl 4.7 is the **K2 Management** site. If you are familiar with the K2 Workspace and the SmartObject Services Tester utility, you will find that many of the features and functionality in these tools are now located in the new Management site. In addition to managing processes and process instances, the K2 administrator can create service instances and SmartObjects directly from the Management site. You can also create SmartObjects using K2 Designer, K2 Studio and K2 for Visual Studio.

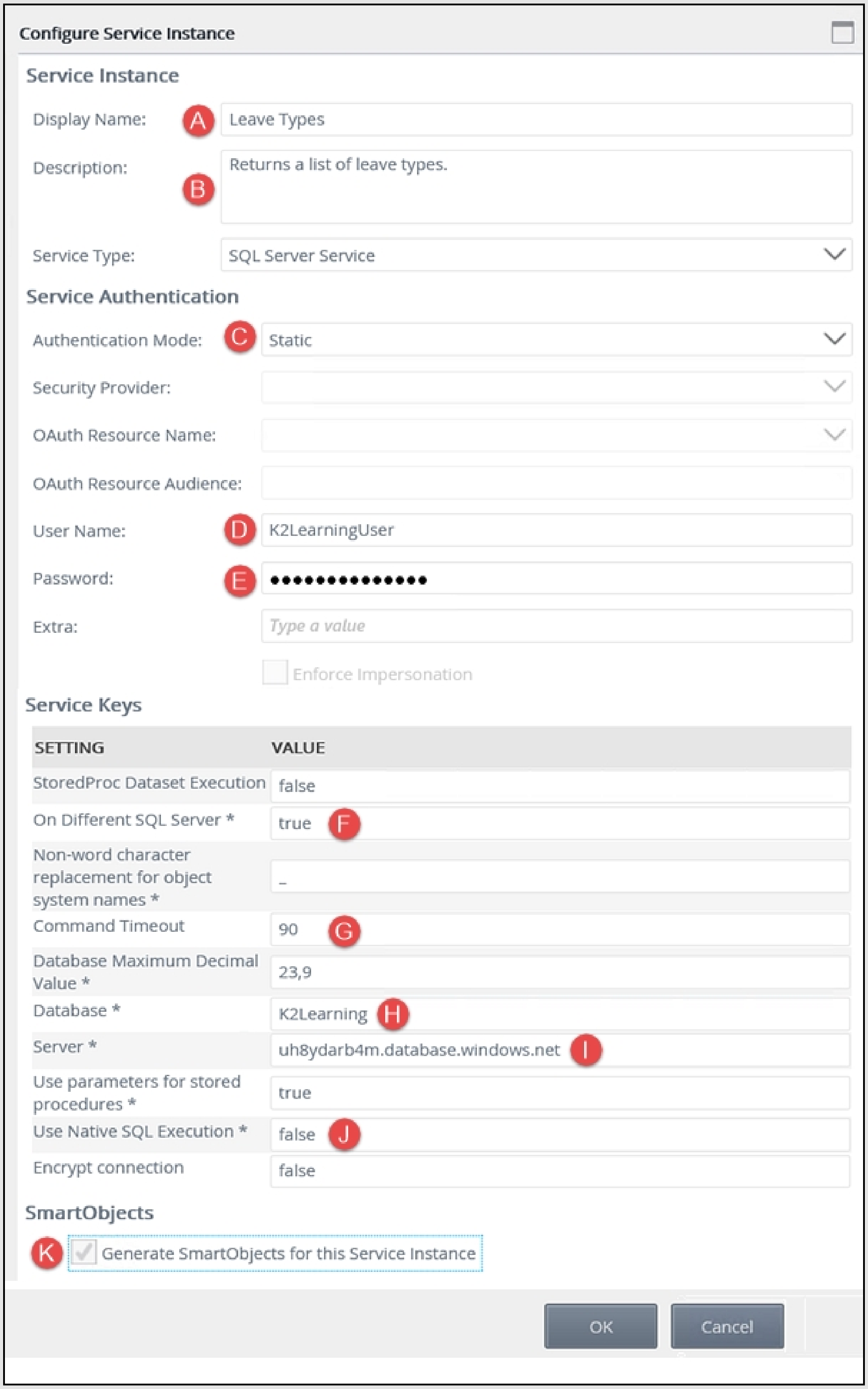
Note
In a K2 environment, registering a service instance for a particular system only needs to be performed once. If you are using a shared environment (for example, if you are completing this exercise in your own K2 environment), this step only needs to be performed once and may already have been completed by another user. If you prefer to create your own service instances, use a unique identifier for the service instance display names and system names. Otherwise, use the service instance that already exists.

Note
If you are attempting this tutorial from within a fire-walled environment, you may not be able to connect to the target SQL database. In this case, speak to your network administrator to determine if access can be opened to the target SQL database over port 1433 as described in this Microsoft KB article: http://support.microsoft.com/kb/287932, or alternatively you will need to create the database internally in a SQL server in your own environment. You can download a script to create the SQL database from the following location: http://help.k2.com/files/8553 . Please contact your SQL database administrator for more information and help.

Step 2 Tasks

1. Launch the **K2 Management** site. (**Start** > **All Programs** > **K2 blackpearl** > **K2 Management**)
2. Add a new service instance of the **SQL Server Service** Type using the table below as a guide for the necessary properties. If a property is not shown in the table below, then assume the default value.

	Field Name	Setting
A	Display Name	*Leave Types*
B	Description	*Returns a list of leave types.*
	Service Type	SQL Server Service (default)
C	Authentication Mode	Static
D	User Name	*K2LearningUser*
E	Password	*K2LearningPass*
F	On Different SQL Server	*true*
G	Command Timeout	*90*
H	Database	*K2Learning*
I	Server	*uh8ydarb4m.database.windows.net*
J	Use Native SQL Execution	*false*
K	Generate SmartObjects for this Service Instance	CHECKED



Service Instance

Display Name: Leave Types

Description: Returns a list of leave types.

Service Type: SQL Server Service

Service Authentication

Authentication Mode: Static

Security Provider:

OAuth Resource Name:

OAuth Resource Audience:

User Name: K2LearningUser

Password: ••••••••••••••

Extra: *Type a value*

□ Enforce Impersonation

Service Keys

SETTING	VALUE
StoredProc Dataset Execution	false
On Different SQL Server *	true
Non-word character replacement for object system names *	_
Command Timeout	90
Database Maximum Decimal Value *	23,9
Database *	K2Learning
Server *	uh8ydarb4m.database.windows.net
Use parameters for stored procedures *	true
Use Native SQL Execution *	false
Encrypt connection	false

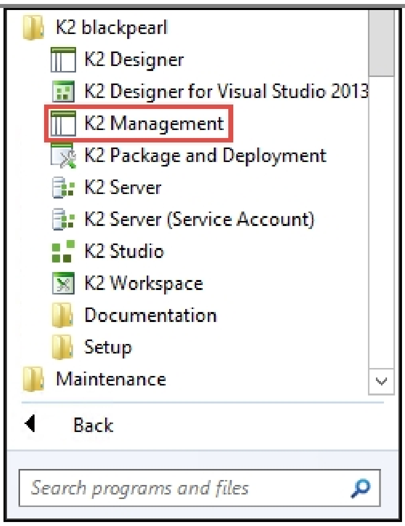
SmartObjects

☑ Generate SmartObjects for this Service Instance

3. Confirm the new Leave Types service instance is displayed in the Service Instances pane.

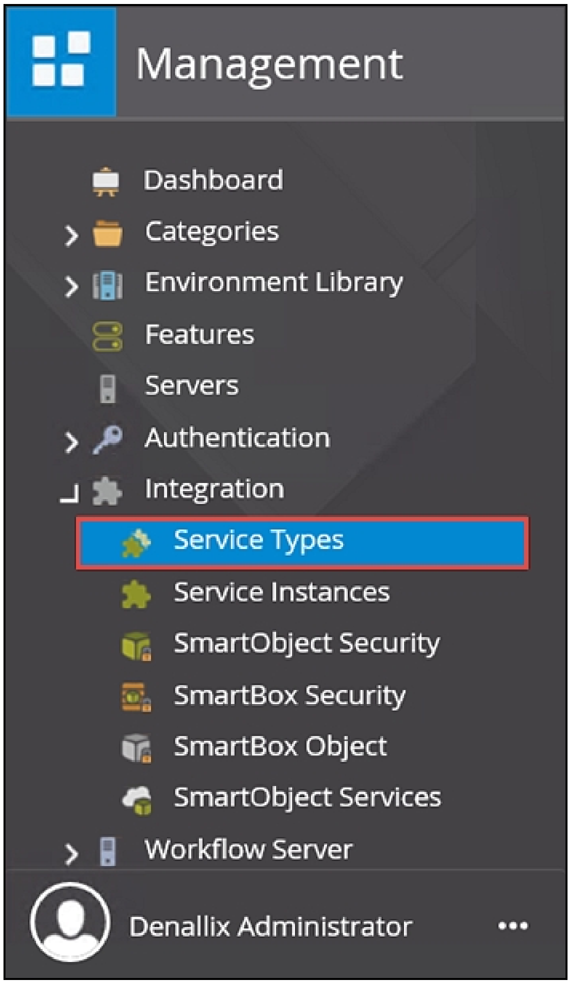
Step 2 Walkthrough

a. Launch the **K2 Management** site. (**Start** > **All Programs** > **K2 blackpearl** > **K2 Management**)


K2 blackpearl
K2 Designer
K2 Designer for Visual Studio 2013
K2 Management
K2 Package and Deployment
K2 Server
K2 Server (Service Account)
K2 Studio
K2 Workspace
Documentation
Setup
Maintenance
Back
Search programs and files

Note
This tutorial touches upon the **Integration** features found in the K2 Management site. For more detailed information on the K2 Management site, see the tutorial Administering K2, or the K2 blackpearl Product Documentation.

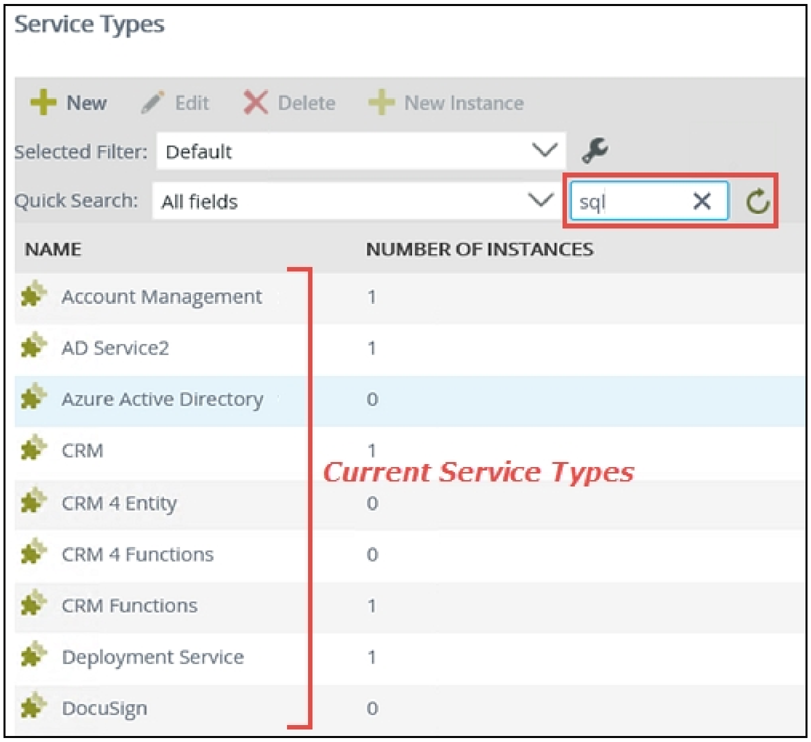
b. Expand the Integration category, then click **Service Types**.



c. The available service types are displayed in the central **Service Types** pane. You will be adding a service instance of the SQL Server Service Type for this step. To help you locate the correct service type, enter

sql
into the search text box, then click the green **refresh** icon.

Note
The **Service Types** node contains the available service types (or connectors). Instances of service types are individual connections containing specific configurations. For example, you will add an instance of the SQL Server Service Type and configure it to connect to an external SQL database. By creating the service instance, K2 will discover the properties and methods that define the SQL data source.

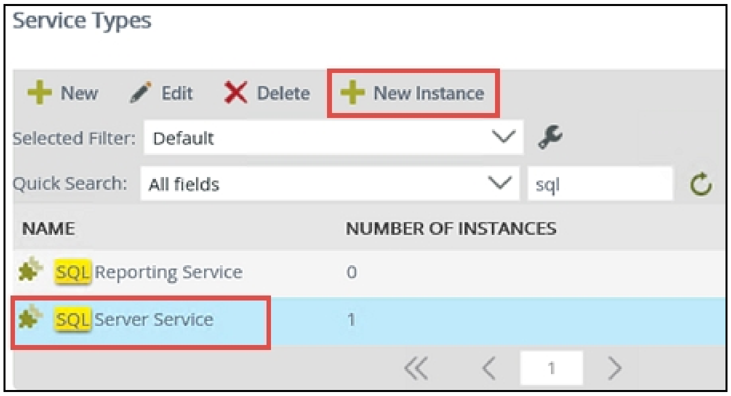


Selected Filter: Default

Quick Search: All fields | sql

NAME	NUMBER OF INSTANCES
Account Management	1
AD Service2	1
Azure Active Directory	0
CRM	1
CRM 4 Entity	0
CRM 4 Functions	0
CRM Functions	1
Deployment Service	1
DocuSign	0

d. At least two service types for SQL are returned: SQL Reporting Service and SQL Server Service. (There may be additional service types depending on your environment.) Click to highlight **SQL Server Service**, then click the **New Instance** button.



Selected Filter: Default

Quick Search: All fields | sql

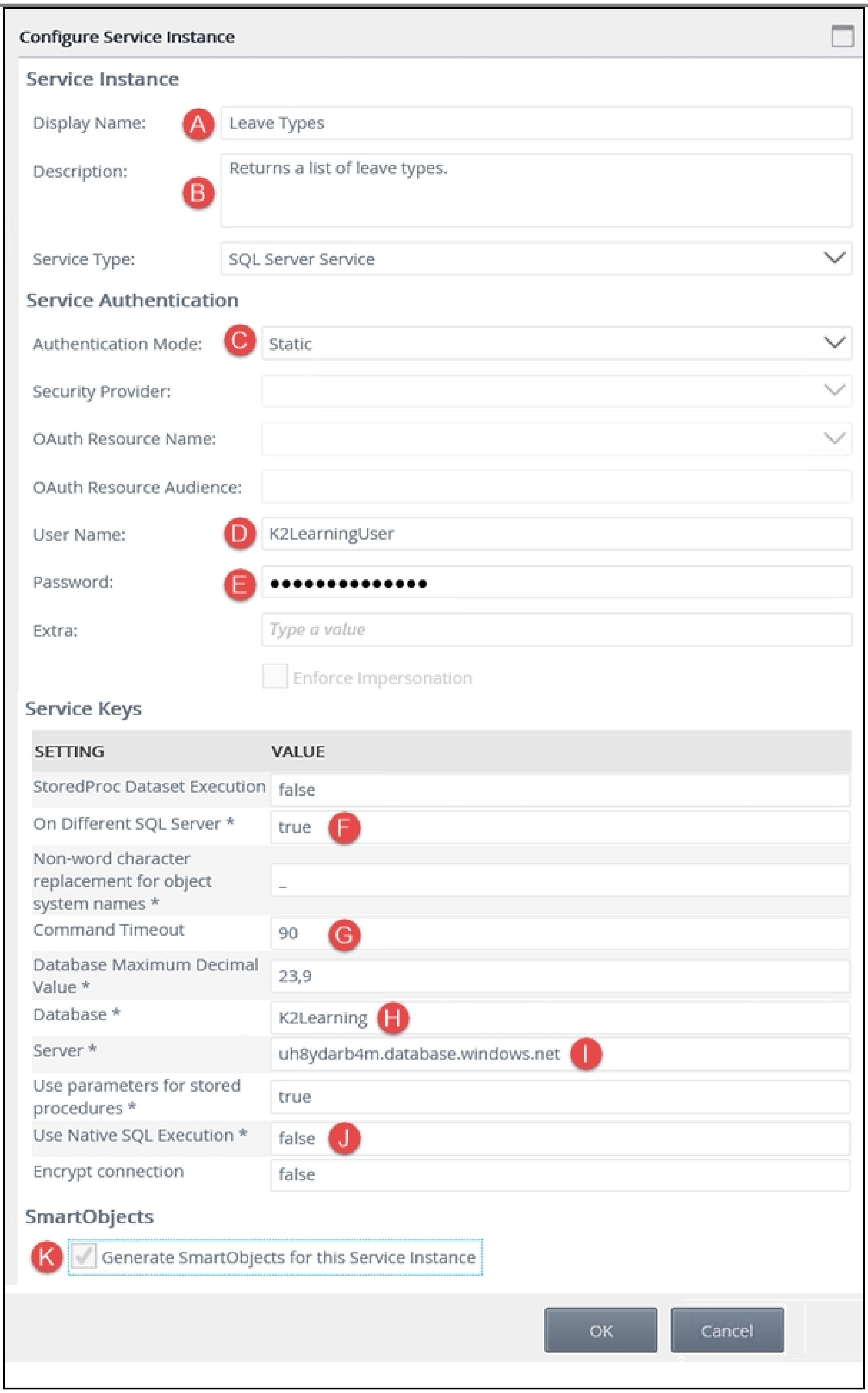
NAME	NUMBER OF INSTANCES
SQL Reporting Service	0
SQL Server Service	1

e. Use the table below as a guide for configuring the service instance. For additional reference, see the image below the table. Click **OK** after you have set the fields.

Note
In this step, notice that you will select the option to have K2 automatically generate a SmartObject from the SQL service instance (K). Because you are not customizing the new SmartObject, auto-generating is good choice and will return all of the properties and methods discovered. If you need to customize the SmartObject (for example, selecting which properties you want returned), use a SmartObject design tool such as K2

Studio, K2 Designer or K2 for Visual Studio.

	Field Name	Setting
A	Display Name	*Leave Types*
B	Description	*Returns a list of leave types*.
	Service Type	SQL Server Service (default)
C	Authentication Mode	Static
D	User Name	*K2LearningUser*
E	Password	*K2LearningPass*
F	On Different SQL Server	*true*
G	Command Timeout	*90*
H	Database	*K2Learning*
I	Server	*uh8ydarb4m.database.windows.net*
J	Use Native SQL Execution	*false*
K	Generate SmartObjects for this Service Instance	CHECKED



Configure Service Instance

Service Instance

Display Name: Leave Types

Description: Returns a list of leave types.

Service Type: SQL Server Service

Service Authentication

Authentication Mode: Static

Security Provider:

OAuth Resource Name:

OAuth Resource Audience:

User Name: K2LearningUser

Password: ●●●●●●●●●●●●●●●

Extra: *Type a value*

□ Enforce Impersonation

Service Keys

SETTING	VALUE
StoredProc Dataset Execution	false
On Different SQL Server *	true
Non-word character replacement for object system names *	_
Command Timeout	90
Database Maximum Decimal Value *	23,9
Database *	K2Learning
Server *	uh8ydarb4m.database.windows.net
Use parameters for stored procedures *	true
Use Native SQL Execution *	false
Encrypt connection	false

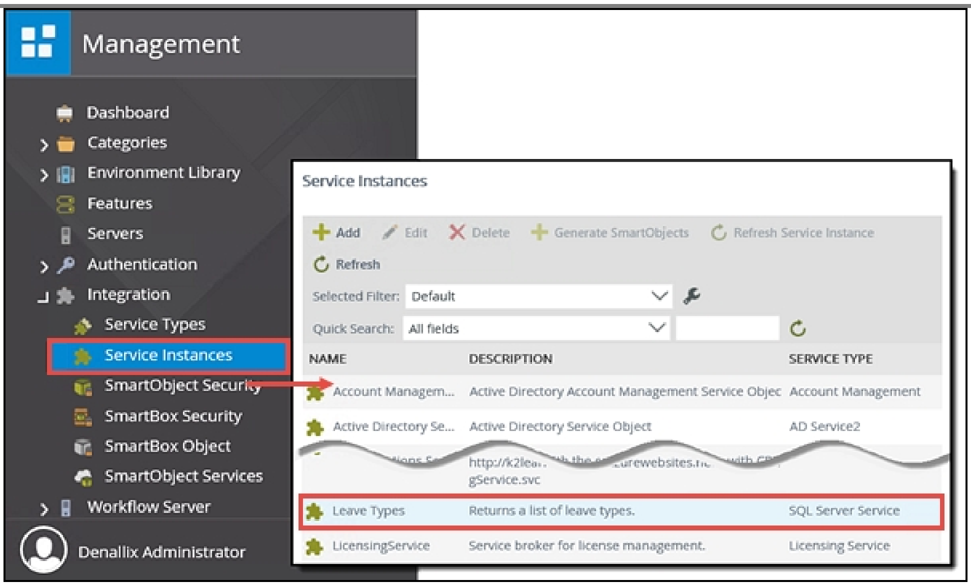
SmartObjects

☑ Generate SmartObjects for this Service Instance

f. You should see a confirmation dialog box. Click **OK** to close the dialog box.

Take a moment to locate the new service instance you just created using the following steps.

g. Still in the Integration node, click **Service Instances**. Scroll down and locate the **Leave Types** service instance. Notice the name, description and service type correspond to your configuration settings.



Selected Filter: Default

Quick Search: All fields

NAME	DESCRIPTION	SERVICE TYPE
Account Managem...	Active Directory Account Management Service Objec	Account Management
Active Directory Se...	Active Directory Service Object	AD Service2
...tions Se...	http://k2lea... ...urewebsites.n... with CR... gService.svc	
Leave Types	Returns a list of leave types.	SQL Server Service
LicensingService	Service broker for license management.	Licensing Service

STEP 2 REVIEW

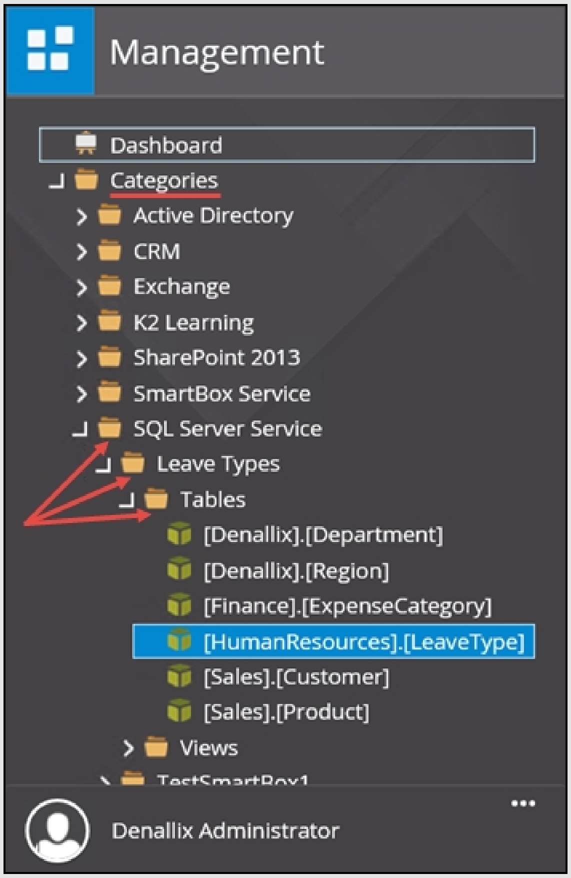
In Step 2 you created a new service instance based on the SQL Server Service Type. The new service instance contains connection settings that are specific to an external SQL database. As part of the configuration, you selected the option for K2 to automatically generate SmartObjects for the data objects that were discovered. In the next step, you will test the SmartObject, then rename it and move it to the appropriate Leave Request category.

Step 3: Test the SmartObject, then rename and move the SmartObject to the Leave Request category

Before you use the new SmartObject in your item view, you will test it to confirm the connection configuration. Then, just to keep your environment organized, you will rename the SmartObject and move it to the Leave Request > SmartObjects category that was created in the basic tutorial.

Step 3 Tasks

1. Expand the Categories node, then the SQL Server Service node. Locate the **[HumanResources].[LeaveType]** table under the Leave Types Service Instance, then execute the **List** method. Confirm results are returned with two properties shown: **LeaveTypeid** and **LeaveTypeDescription**.

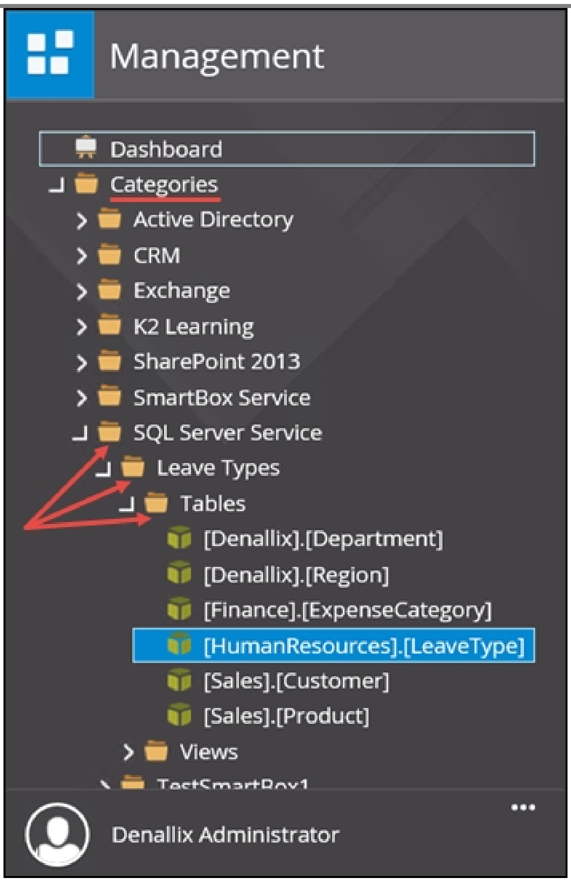


2. Click the Design link to edit the SmartObject. Rename the SmartObject to
   Leave Types
   then move the **Leave Types** SmartObject to the **K2 Learning** > **Leave Request** > **SmartObjects** category.

Step 3 Walkthrough

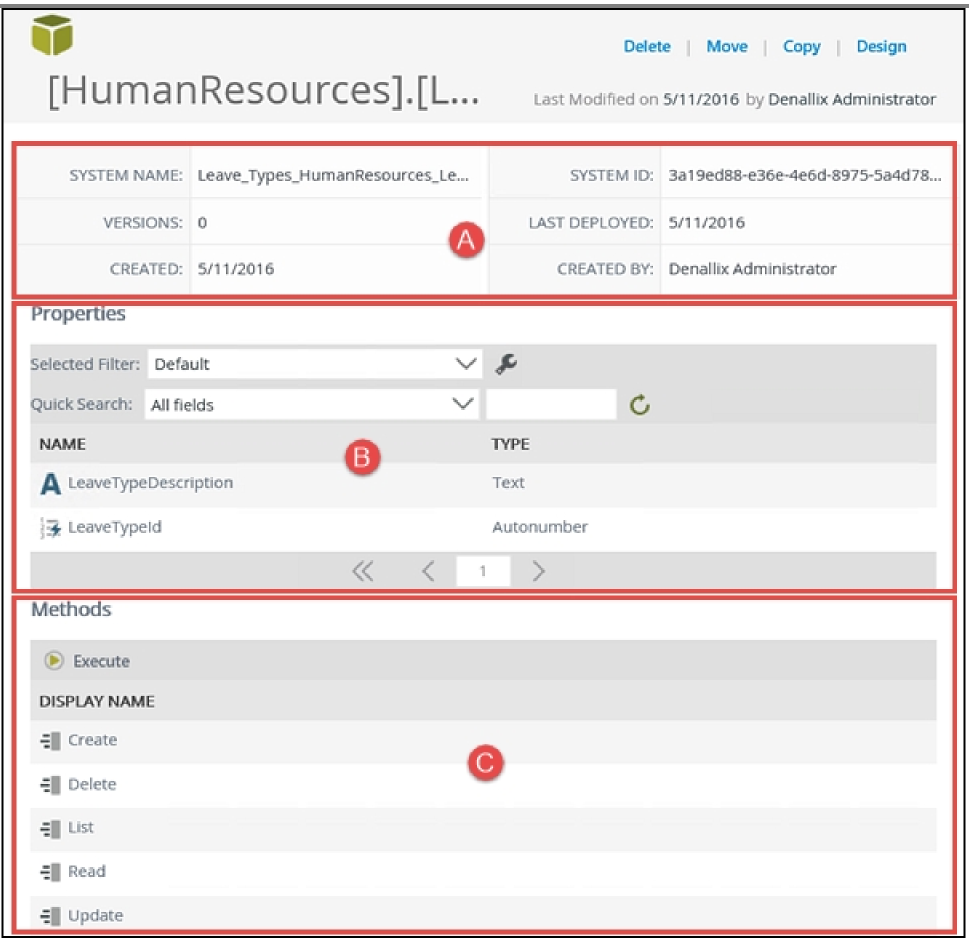
a. Expand the **Categories** node, then expand the **SQL Server Service** node. Notice the Leave Types SmartObject. Expand **Leave Types**, then **Tables**. There are six tables discovered with this SmartObject.

> **Note**
> When the service instance was created, K2 discovered the tables within the SQL database. Each table in return, has properties. Properties can be compared in theory to the columns in a spreadsheet. For example, Name, Department, Email and so forth. Each table will have its own set of properties and those properties may or may not contain actual values. The SmartObject becomes the communication mechanism between your forms and workflows and the values found in the SQL tables. For example, you will use the values found in the [HumanResources].[LeaveType] table as the values for your Leave Type drop-down list on your item view.



b. Double-dlick the **[HumanResources].[LeaveType]** table to expose the properties and methods in the central pane. Take a minute to review the following sections:

- A: Displays the system details such as the system name, date created and created by values.
- B: Displays the properties: LeaveTypeDescription and LeaveTypeid (think spreadsheet column titles).
- C: Displays the methods for this table. You will use the List method to return a list of leave types for the Leave Type drop-down list on the item view.



[HumanResources].[L...

Last Modified on 5/11/2016 by Denallix Administrator

SYSTEM NAME:	Leave_Types_HumanResources_Le...	SYSTEM ID:	3a19ed88-e36e-4e6d-8975-5a4d78...
VERSIONS:	0	LAST DEPLOYED:	5/11/2016
CREATED:	5/11/2016	CREATED BY:	Denallix Administrator

Properties

Selected Filter: Default

Quick Search: All fields

NAME	TYPE
LeaveTypeDescription	Text
LeaveTypeId	Autonumber

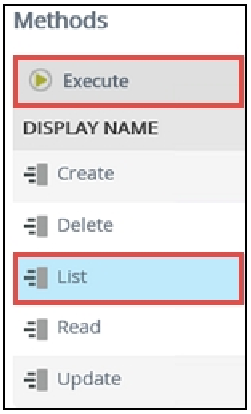
Methods

Execute

DISPLAY NAME
Create
Delete
List
Read
Update

Next, you will test the SmartObject connection by executing the List method.

c. In the Methods pane, click to highlight the **List** method, then click **Execute**.

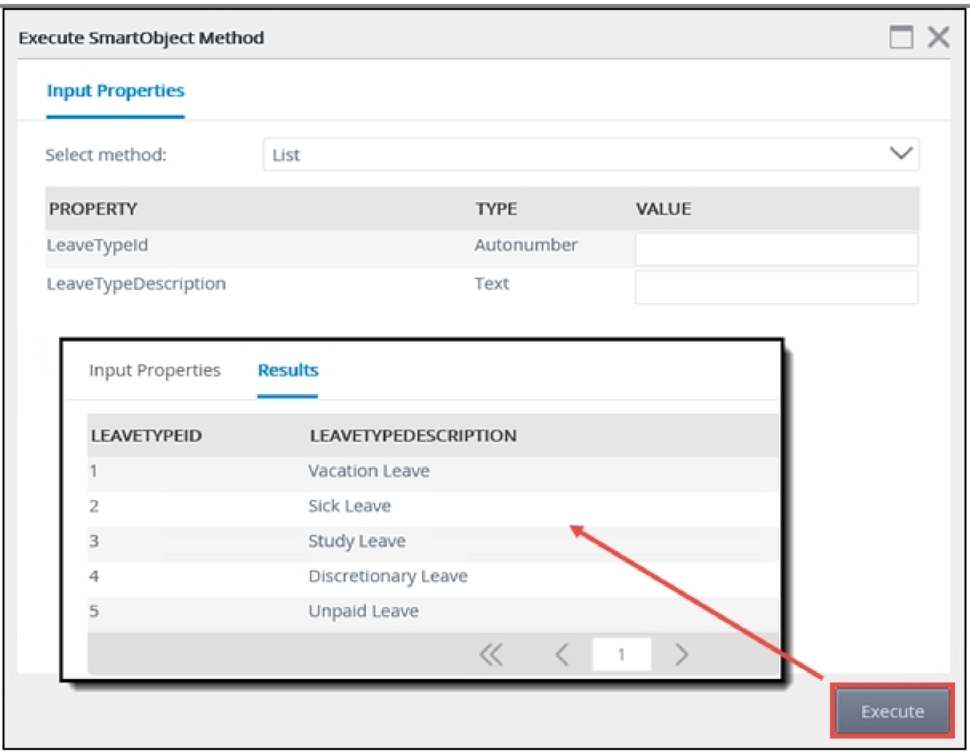


Methods

Execute

DISPLAY NAME
Create
Delete
List
Read
Update

d. There are no input properties for this test, so click **Execute**. The Results pane will appear with the values from the [HumanResources].[LeaveType] table. This confirms the external SQL database connection is correct. Exit the **Execute SmartObject Method** screen.



Execute SmartObject Method

Input Properties

Select method: List

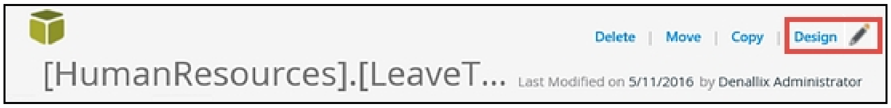
PROPERTY	TYPE	VALUE
LeaveTypeId	Autonumber	
LeaveTypeDescription	Text	

Input Properties | Results

LEAVETYPEID	LEAVETYPEDESCRIPTION
1	Vacation Leave
2	Sick Leave
3	Study Leave
4	Discretionary Leave
5	Unpaid Leave

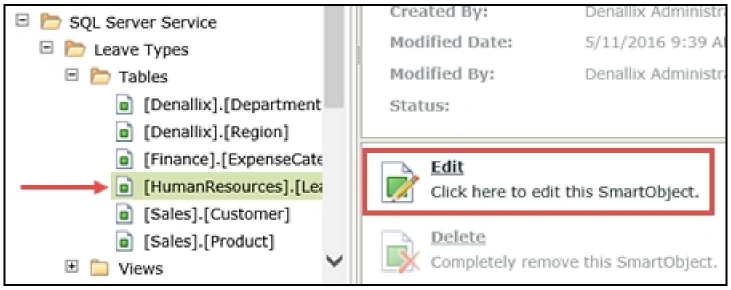
The last step in this series is to rename, then move the new SmartObject to the Leave Request category that you created using K2 Designer in the basic tutorial. This will keep all of your K2 artifacts organized and easy to locate as you move through the next steps.

e. Click to highlight the **[HumanResources].[LeaveType]** table to expose its properties (if they are not already). Click the **Design** link in the upper right corner of the screen.



[HumanResources].[LeaveT... Last Modified on 5/11/2016 by Denallix Administrator

f. You will be redirected to the K2 Designer. (The K2 Designer will open in a new tab on your browser.) Confirm the **[HumanResources].[LeaveType]** table is highlighted, then click **Edit**.



Created By: Denallix Administr

Modified Date: 5/11/2016 9:39 A

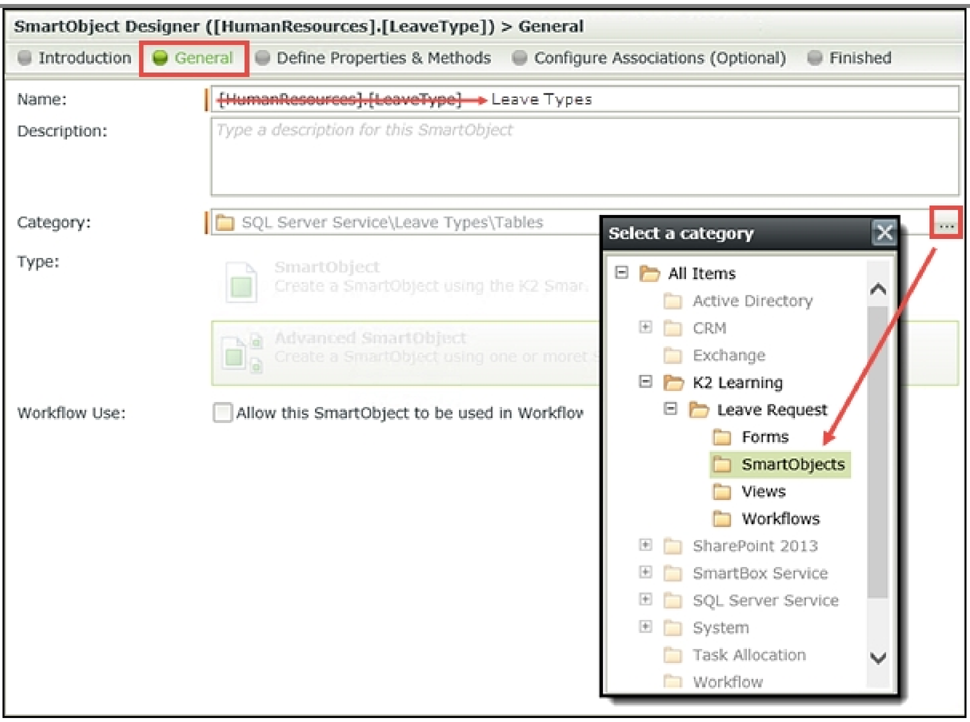
Modified By: Denallix Administr

Status:

Edit
Click here to edit this SmartObject.

Delete
Completely remove this SmartObject.

g. Switch to the **General** settings screen. Change the **Name** of the SmartObject to
Leave Types
then change the **Category** to the **K2 Learning/Leave Request/SmartObjects** category. (Your category names may be different if you chose different names in the basic tutorial.) There are no other edits to this screen, so click **Finish** to republish the SmartObject.


SmartObject Designer ([HumanResources].[LeaveType]) > General

Introduction
General
Define Properties & Methods
Configure Associations (Optional)
Finished

Name: ~~[HumanResources].[LeaveType]~~ Leave Types

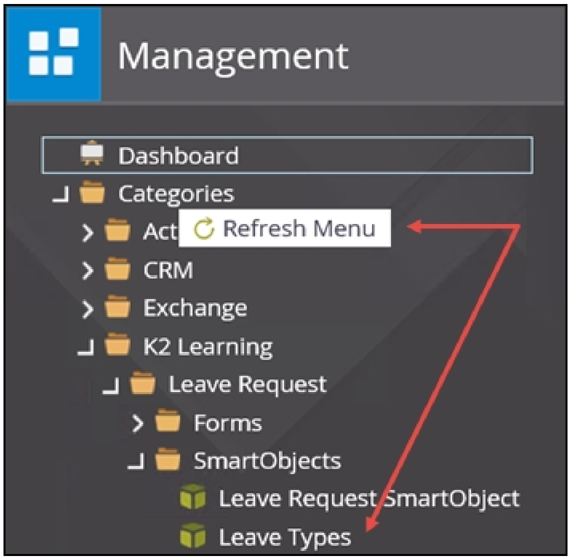
Description: Type a description for this SmartObject

Category: SQL Server Service\Leave Types\Tables

Type: SmartObject Create a SmartObject using the K2 Smar... Advanced SmartObject Create a SmartObject using one or more...

Workflow Use: □ Allow this SmartObject to be used in Workflow

h. Switch back to the **Management** site tab in your browser. Right-click the **Categories** name and select **Refresh Menu**. Expand the **K2 Learning** > **Leave Request** > **SmartObjects** categories and confirm the **Leave Types** SmartObject is shown. Notice too, that the Leave Request SmartObject also appears in Management site menu. This is the SmartBox that you created in the basic tutorial.

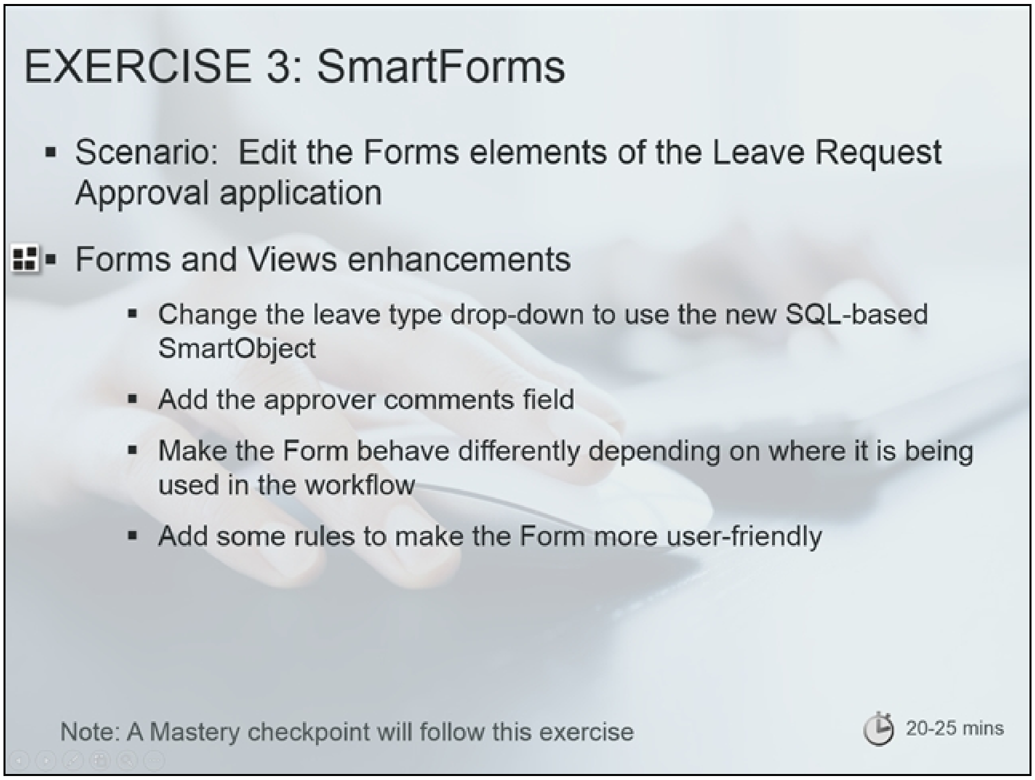


STEP 3 REVIEW

In this step you confirmed the service instance configuration is working correctly by executing the SmartObject list method. The results returned are the values that are found in the external database (leave type) table. The leave type values will be used to automate the Leave Type drop-down list on the item view. You concluded this step by renaming the SmartObject to a more friendly name, then moving the SmartObject to the appropriate category.

This concludes the data changes you will make for this new version of the Leave Request application. In Part 2: Forms, you will edit the item view and add the two new data sources just created: Approver Comments and Leave Types.

EXERCISE 3: SmartForms



EXERCISE 3: SmartForms

- Scenario: Edit the Forms elements of the Leave Request Approval application
- Forms and Views enhancements
  - Change the leave type drop-down to use the new SQL-based SmartObject
  - Add the approver comments field
  - Make the Form behave differently depending on where it is being used in the workflow
  - Add some rules to make the Form more user-friendly

Note: A Mastery checkpoint will follow this exercise

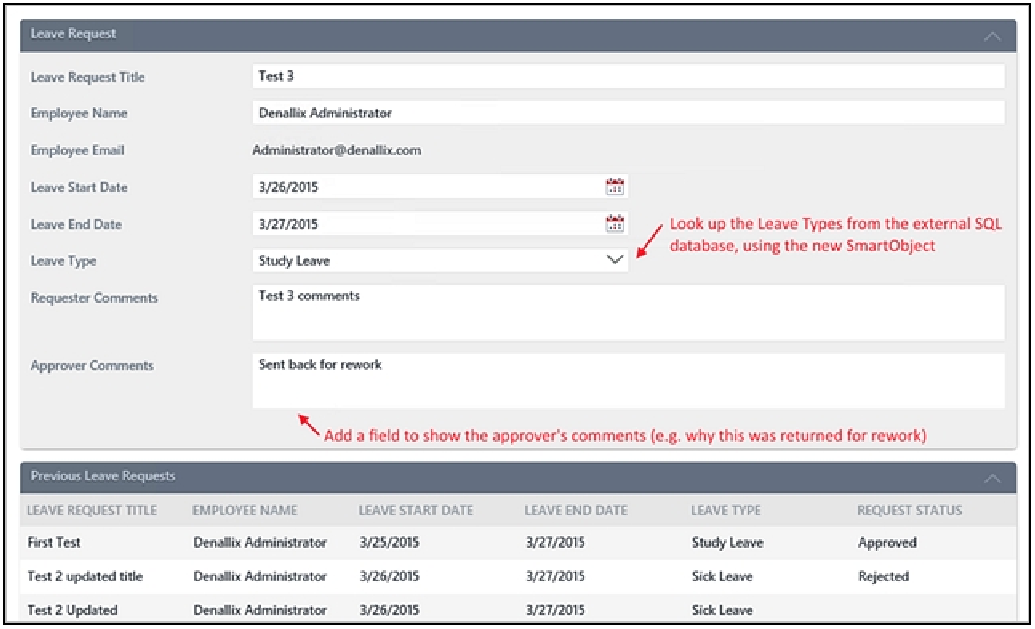
20-25 mins

Video

This exercise goes into a little more depth on SmartForms. We are going to edit the “Forms” components of our Leave Request Approval application in this exercise to add some more functionality to our application, specifically to add the Approver Comments field (but not show the field when the Form is being used to start the workflow), and to change the Leave Type drop-down list to show the values retrieved from the external database, instead of a static list of values.

You will also learn how to edit existing Views and Forms, and how to use concepts like Rules to make Forms more user-friendly.

The enhancements we will make to the Leave Request Form



Leave Request

Leave Request Title	Test 3
Employee Name	Denallix Administrator
Employee Email	Administrator@denallix.com
Leave Start Date	3/26/2015
Leave End Date	3/27/2015
Leave Type	Study Leave
Requester Comments	Test 3 comments
Approver Comments	Sent back for rework

Look up the Leave Types from the external SQL database, using the new SmartObject

Add a field to show the approver's comments (e.g. why this was returned for rework)

Previous Leave Requests

LEAVE REQUEST TITLE	EMPLOYEE NAME	LEAVE START DATE	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
First Test	Denallix Administrator	3/25/2015	3/27/2015	Study Leave	Approved
Test 2 updated title	Denallix Administrator	3/26/2015	3/27/2015	Sick Leave	Rejected
Test 2 Updated	Denallix Administrator	3/26/2015	3/27/2015	Sick Leave	

After this exercise we will do a quick mastery check on SmartForms.

When you are ready, continue on to Part 2: Forms to edit the Forms for our Leave Request application.

Part 2: Forms

Part 2 of this tutorial expands on the forms component created in the basic Leave Request application.

In this part, you will learn:

- How to add a SmartObject as a control's data source (in this sample, using a SmartObject to populate a drop-down list control)
- How to add a new control to a view and make the control read-only

If you have not completed Part 1 of this tutorial yet, please do so now since you will be using the updated SmartObjects from Part 1.

Step 4: Add the Approver Comments to the Leave Request item view and configure the Leave Types SmartObject as the data source for the Leave Type drop-down list

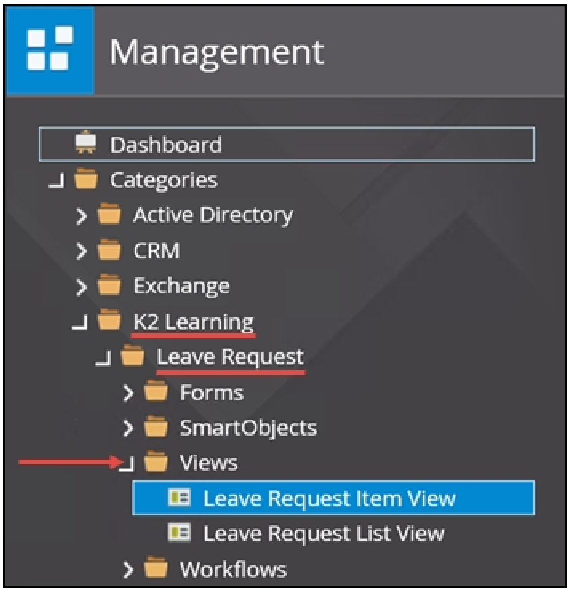
Step 4 Tasks

1. Edit the Leave Request item view and change the data source for the **Leave Type** drop-down list from static to the Leave Types SmartObject. Make the **LeaveTypeDescription** property the control's **Display** and **Value**.
2. Add a new row to the view layout just above the **Create** button row. Add the **Approver Comments** field to the new row, moving the label to the first cell and the field to the second cell. Make the field read-only by default.

Step 4 Walkthrough

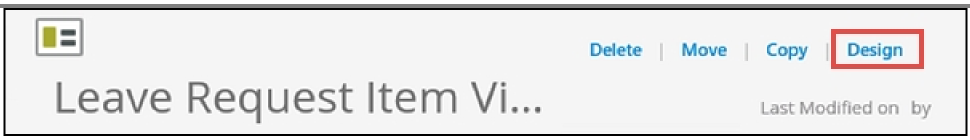
a. Right-click and check out the **Leave Request Item View**, then select **Edit** from the Properties pane. (You may have to select the title again to see the **Edit** link.) The View should open in Edit mode and jump to the **Layout** screen.

b. In the K2 Management site, expand the **Categories** > **K2 Learning** > **Leave Request** > **Views** nodes.

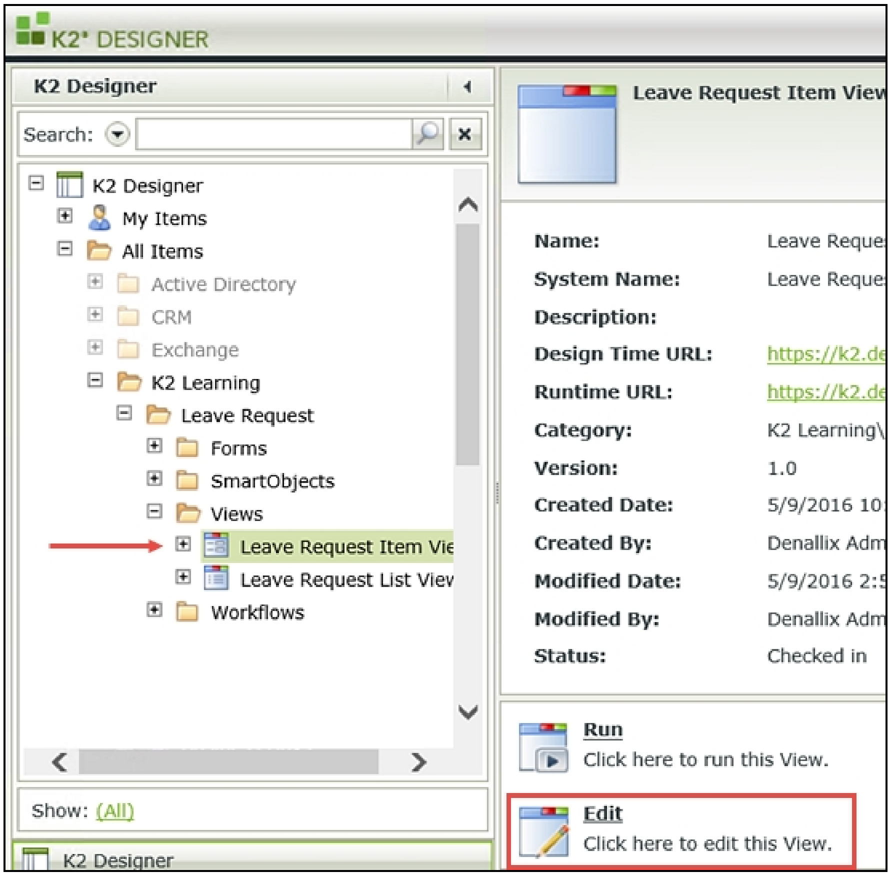


In the next few steps, you are going to replace the static values for the Leave Type drop-down list to the Leave Types SmartObject data source values.

c. Click to highlight the **Leave Request Item View** in the Management site menu. The view properties are displayed in the central pane. Click the **Design** link in the upper right corner of your screen.



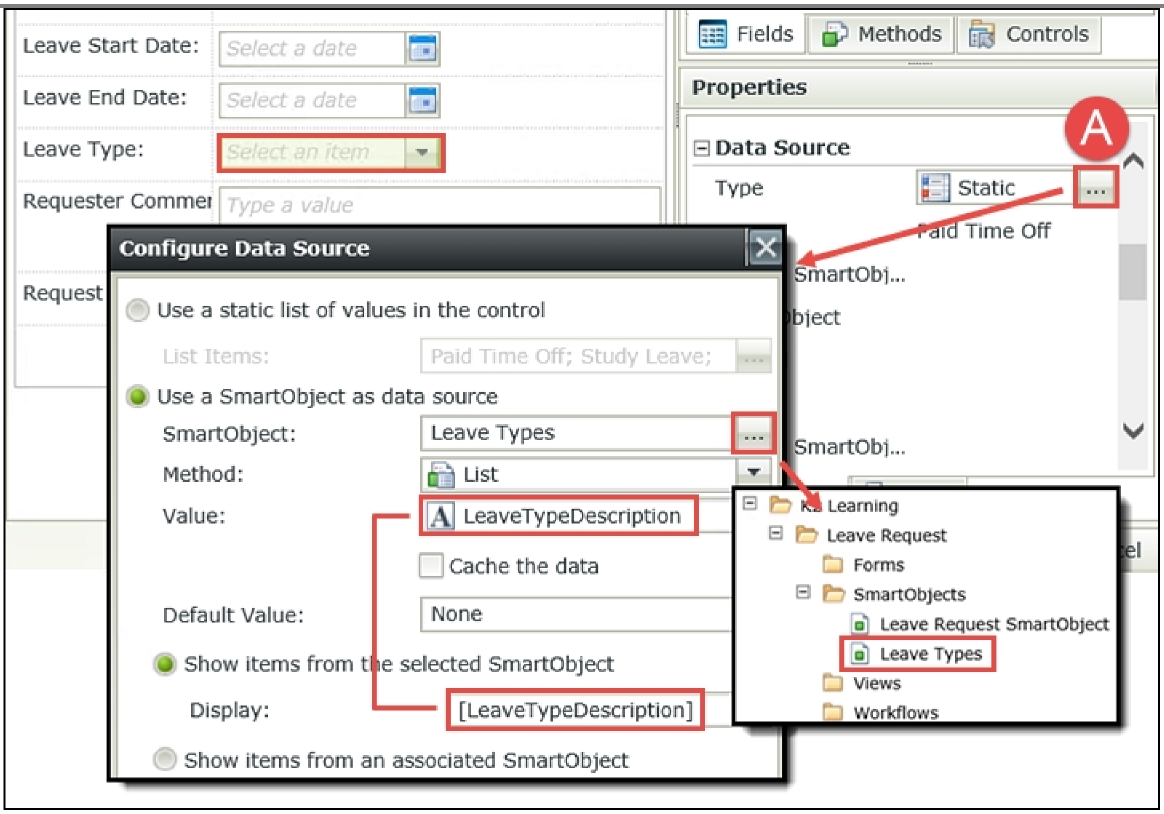
d. The K2 Designer will open in a new tab on your browser. With the item view highlighted, click **Edit**.



Name: Leave Reque
System Name: Leave Reque
Description:
Design Time URL: https://k2.de
Runtime URL: https://k2.de
Category: K2 Learning\
Version: 1.0
Created Date: 5/9/2016 10:
Created By: Denallix Adm
Modified Date: 5/9/2016 2:5
Modified By: Denallix Adm
Status: Checked in

e. Click the **Leave Type** drop-down list to highlight it. (If you see a dialog indicating the view is not checked out, click **Yes** to check out the view and continue.) In the **Properties** pane, open the **Data Source** editor. (**A** below)

- Select the **Use a SmartObject as data source** option, then click the SmartObject browse button
- Navigate to, then select the **Leave Types** SmartObject that you created in Part 1. Click **OK**.
- Change the **Value** to the **LeaveTypeDescription** property so that it matches the **Display**
- Click **OK** to return to the View Designer canvas



○ Use a static list of values in the control

List Items: Paid Time Off; Study Leave;

● Use a SmartObject as data source

SmartObject: Leave Types

Method: List

Value: LeaveTypeDescription

□ Cache the data

Default Value: None

● Show items from the selected SmartObject

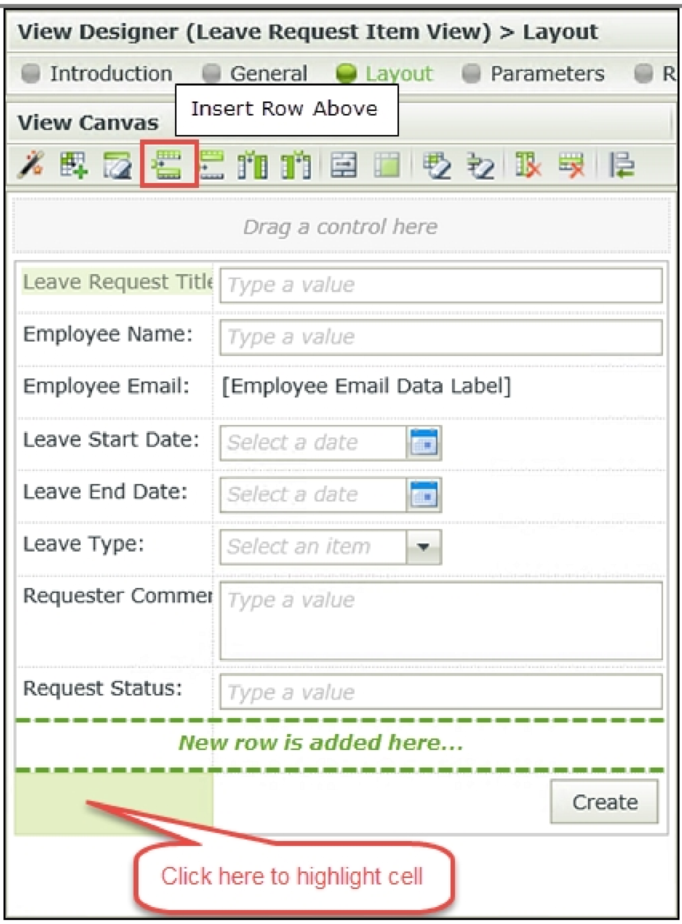
Display: [LeaveTypeDescription]

○ Show items from an associated SmartObject

Tip
One advantage to using a SmartObject as a data source for the values in a drop-down list is that you simply update the SmartObject itself if you need to change the list values. For example, if you have several forms that all contain a Leave Type field, you would want to keep the values consistent across all forms. By using a single SmartObject as the data source, you can easily manage and administer the properties and their values. You also gain access to K2's tools for using the data source properties and methods in rules, if required. An example of this would be if you allowed the user to add their own value to the drop-down list, you could then use the save method in an action to update the SmartObject with the new value. (This is just an example, not actually part of this tutorial.)

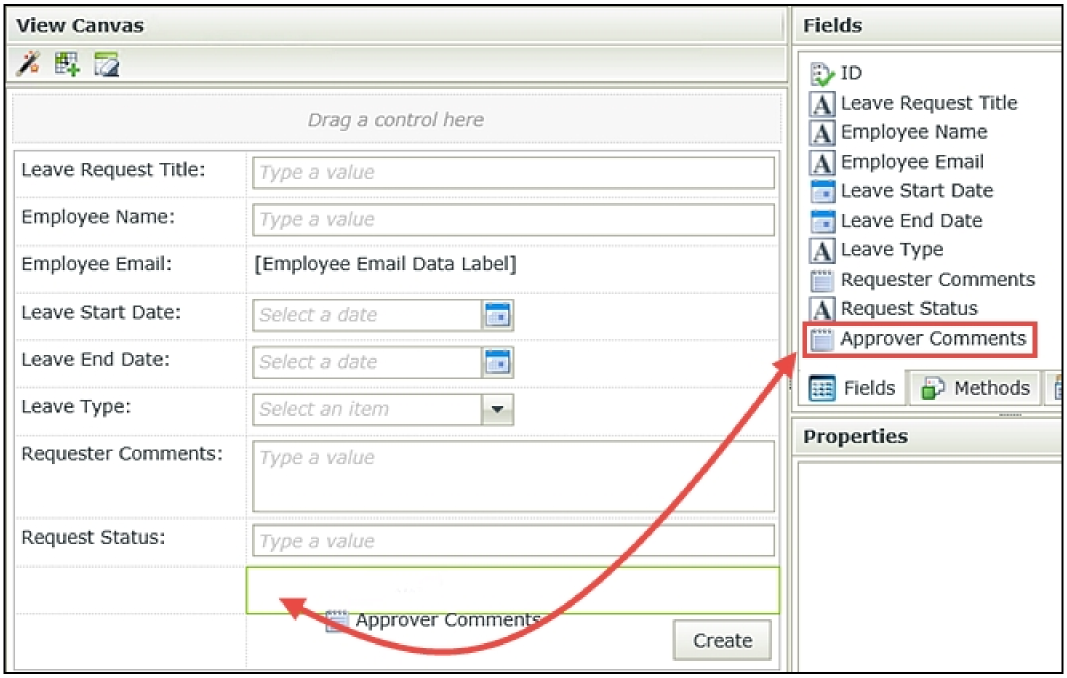
In the next step, you will edit the view layout and add the new Approver Comments field.

f. First, you will add a new row to your view layout table so that you have a location for the new Approver Comments field and label. Click once in the bottom row, first cell, to highlight the cell. Click the **Insert Row Above** icon found in the View Menu to add a new row.

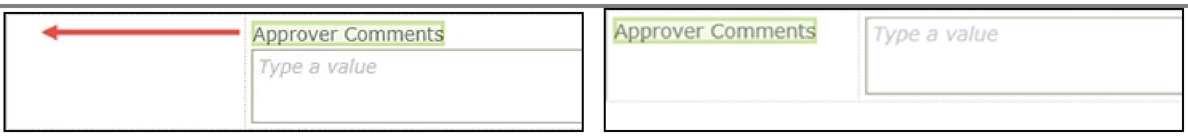


Click here to highlight cell

g. In the **Fields** pane, scroll through the fields until you find the new **Approver Comments** field. Drag the field into the second cell of the new row you just created in the layout table.

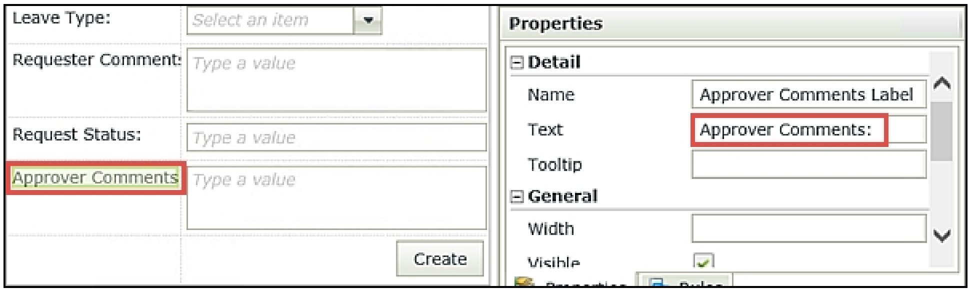


h. Notice that K2 has placed the Approver Comments label to just above the field. Click and drag the **label** into the empty cell to the left.



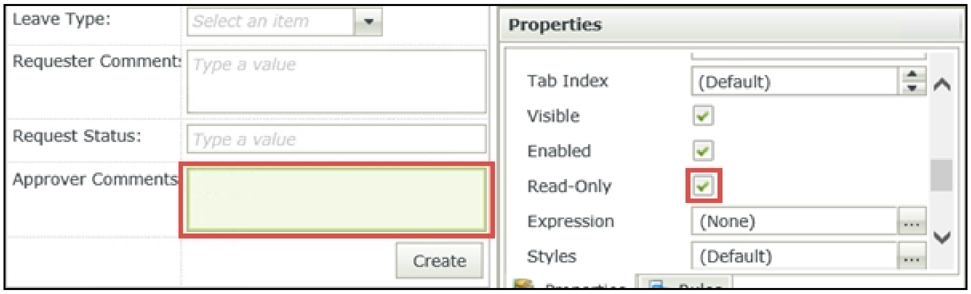
i. You have two minor adjustments to make to the new label and field. First, highlight the label, then locate the **Text** value in the Properties pane. Add a colon (:) to the end of the text value so that it matches the other labels in the view.

> Note
> The **Name** value in the Properties pane is the control's identity that K2 uses to reference a specific control. In this case, the name value is Approver Comments Label. The text value is what is shown to the user who is accessing the form.



Name: Approver Comments Label
Text: Approver Comments:
Tooltip:

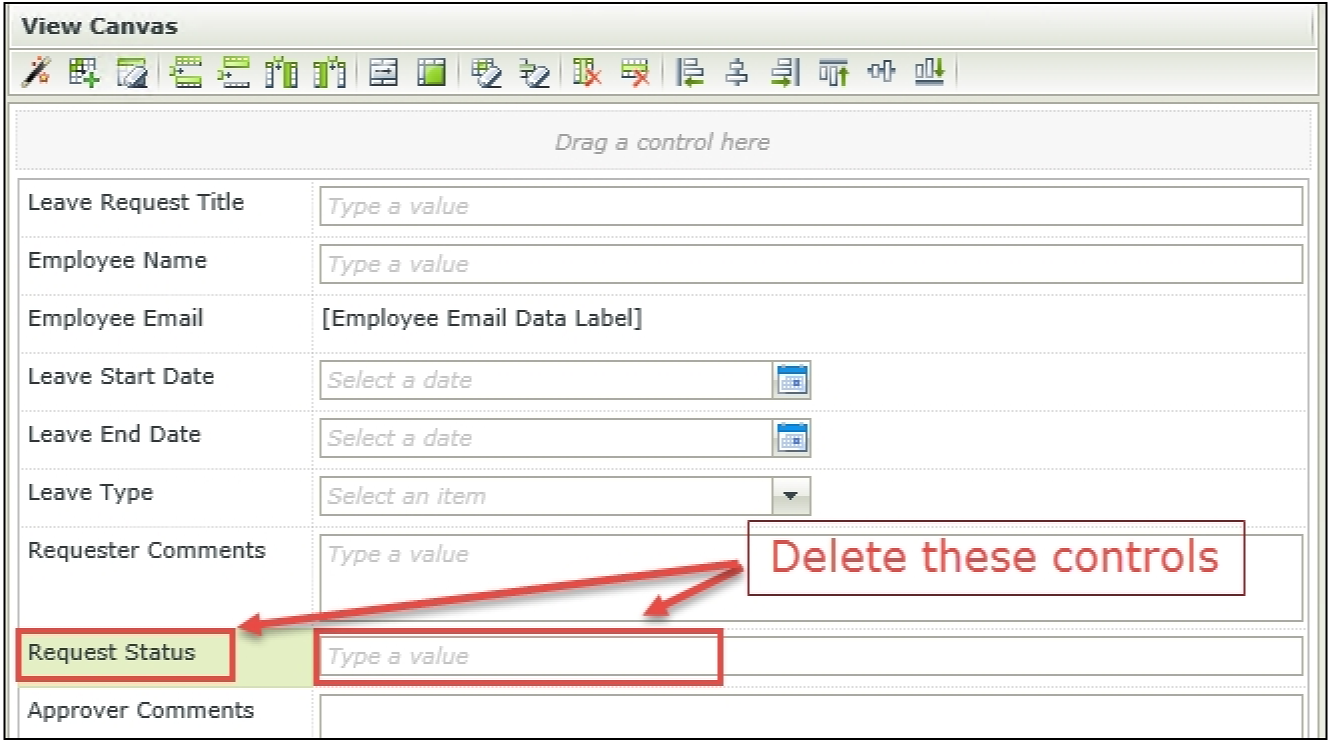
j. Now highlight the field itself and locate and CHECK the **Read-Only** option (again, in the Properties pane) so that the field cannot be edited by the form originator. (Later on, you will use a rule to enable the field for the manager so they can add comments.)



Tab Index: (Default)
Visible: ☑
Enabled: ☑
Read-Only: ☑
Expression: (None)
Styles: (Default)

k. In this step, you will delete the Request Status label and field. This field value is updated by the workflow and is displayed in the Previous Leave Requests list view. It could be confusing to the user if it is displayed on the Leave Request form, so you will remove it. Select the **Request Status Text Box** to highlight it, then click the delete button (on your keyboard). Repeat this process for the **Request Status Label**. (You can delete the table row if you wish using the Delete Row icon from the View

Menu.)



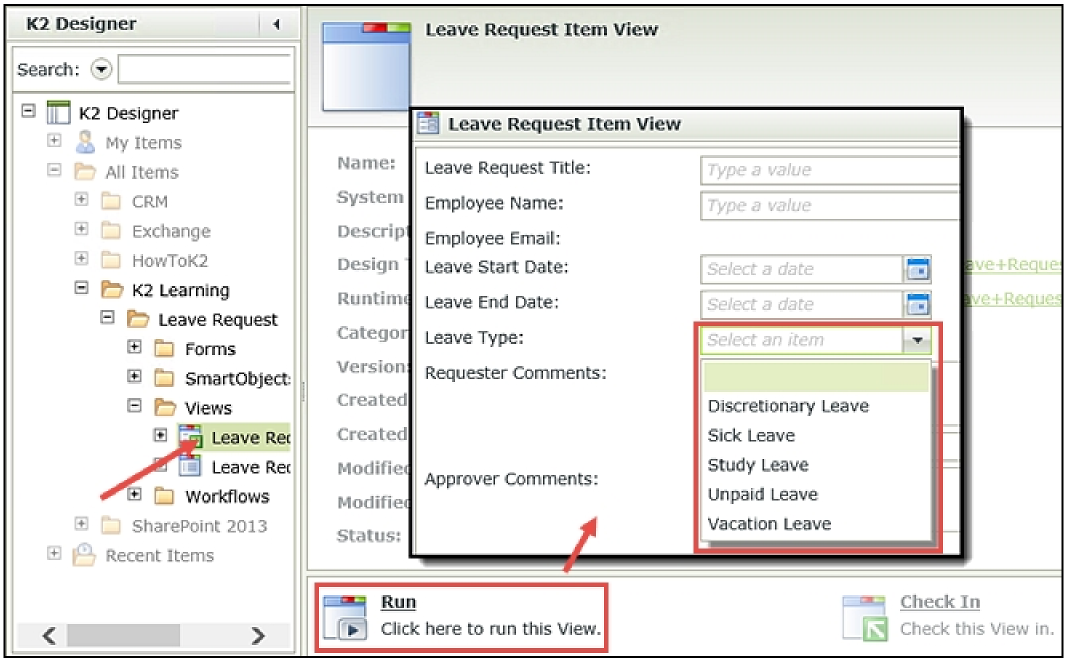
View Canvas

Drag a control here

Leave Request Title	Type a value
Employee Name	Type a value
Employee Email	[Employee Email Data Label]
Leave Start Date	Select a date
Leave End Date	Select a date
Leave Type	Select an item
Requester Comments	Type a value
Request Status	Type a value
Approver Comments	

Delete these controls

l. Click **Finish** to save and exit the view. Highlight the view title in the Category Browser (if it is not already) then click the **Run** link found in the Properties pane. Confirm the Leave Type drop-down list now contains the values from the Leave Types SmartObject, as shown below.



Leave Request Item View

Leave Request Title:	Type a value
Employee Name:	Type a value
Employee Email:	
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Select an item
Requester Comments:	
Approver Comments:	

Discretionary Leave
Sick Leave
Study Leave
Unpaid Leave
Vacation Leave

Run
Click here to run this View.

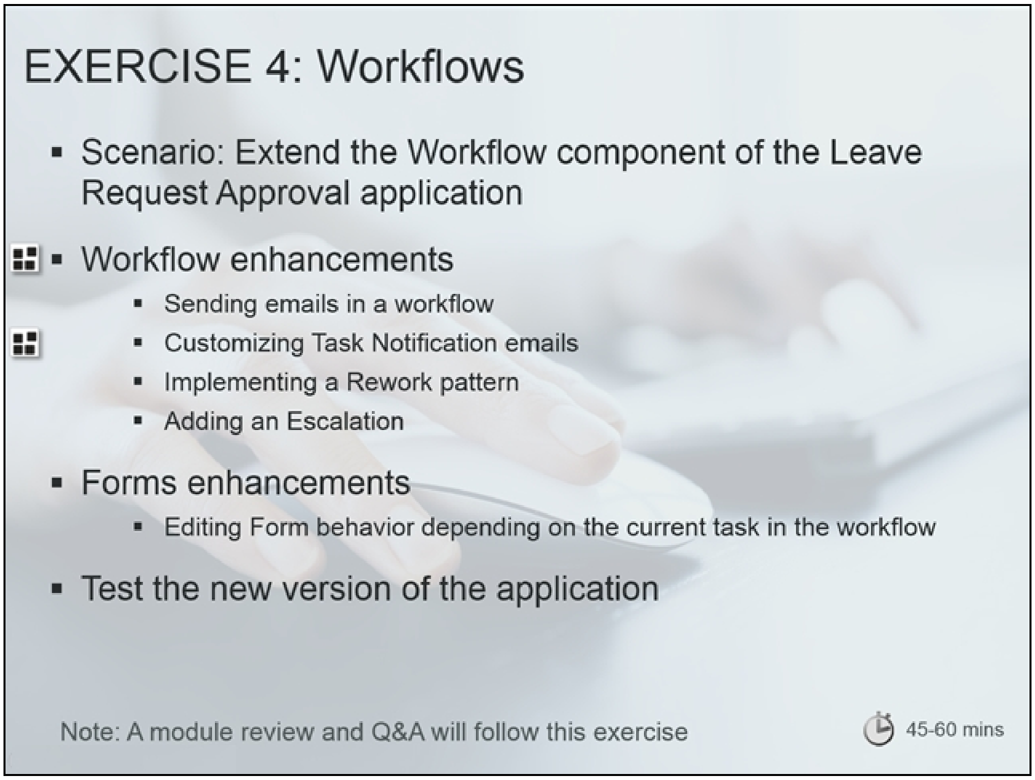
m. Click once again on the **Leave Request Item View** title to close the Run screen.

STEP 4 REVIEW

In this exercise, you edited the Leave Request item view by changing the data source for the Leave Type drop-down list from a static value to a dynamic value that is retrieved from an external SQL database. Then you added the Approver Comments field to the view and made it read-only by default. You will enable the control in a later step so that the manager can add comments. Finally, you deleted the Request Status field to eliminate any confusion to the user submitting the form. This field is updated by the workflow and not the user.

This concludes the initial view changes you will make for this tutorial. In Part 3: Workflow, you will edit the workflow to add more steps, and then come back and edit your forms for the additional workflow tweaks.

EXERCISE 4: Workflows



EXERCISE 4: Workflows

- Scenario: Extend the Workflow component of the Leave Request Approval application
- Workflow enhancements
  - Sending emails in a workflow
  - Customizing Task Notification emails
  - Implementing a Rework pattern
  - Adding an Escalation
- Forms enhancements
  - Editing Form behavior depending on the current task in the workflow
- Test the new version of the application

Note: A module review and Q&A will follow this exercise

45-60 mins

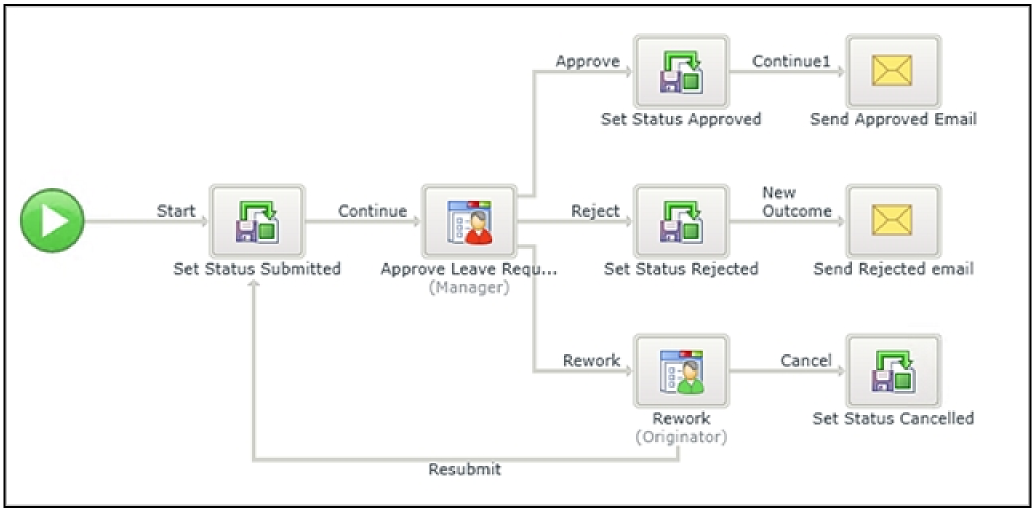
Video

This exercise goes into a little more depth on Workflows and we will edit and enhance the workflow component of the Leave Request Approval application. As part of this exercise, we will also test our workflow. This exercise is a little more involved since we are making a number of changes to the workflow and some tweaks to the Forms and Views used in the application.

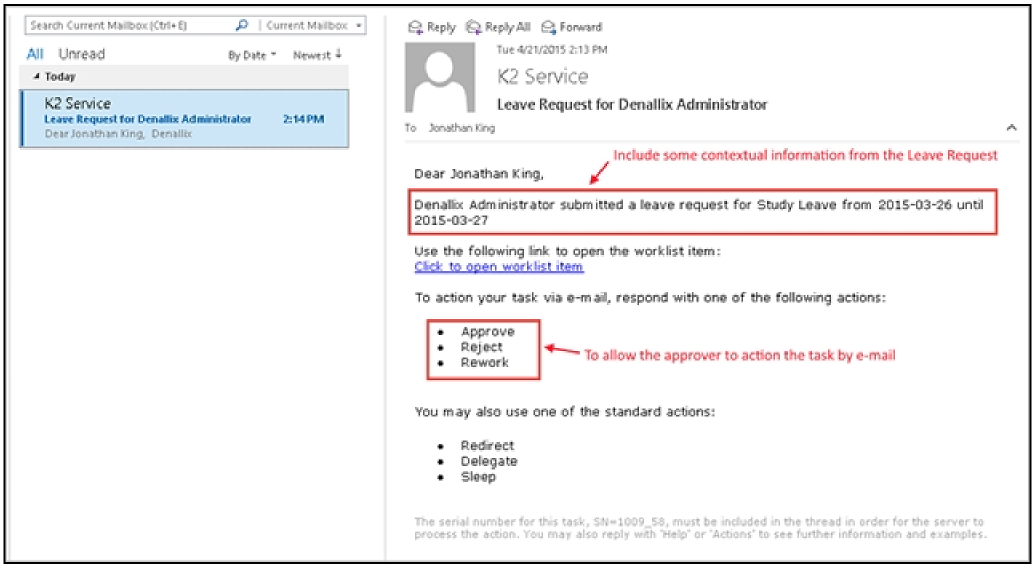
In this exercise you will learn about the following workflow functionality:

- Sending Emails in a workflow
- Customizing Task Notification emails so that users can use SmartActions to complete their tasks
- Implementing a Rework pattern
- Adding an Escalation
- We will also make some edits to the Forms in the application due to the new workflow steps that are added, to make the form behave differently if it is being used for the Approval task or the Rework task.

The workflow diagram below illustrates the updated version of the Leave Request Approval workflow:



This screenshot shows the SmartAction-enabled e-mail that will allow the approver to approve the leave request by replying to the task notification e-mail.


Search Current Mailbox (Ctrl+E)
Current Mailbox
All
Unread
By Date
Newest
Today
K2 Service
Leave Request for Denallix Administrator
2:14 PM
Dear Jonathan King, Denallix
Reply
Reply All
Forward

Tue 4/21/2015 2:13 PM

K2 Service

Leave Request for Denallix Administrator

To Jonathan King

Include some contextual information from the Leave Request

Dear Jonathan King,

Denallix Administrator submitted a leave request for Study Leave from 2015-03-26 until 2015-03-27

Use the following link to open the worklist item:
Click to open worklist item

To action your task via e-mail, respond with one of the following actions:

- Approve
- Reject
- Rework

To allow the approver to action the task by e-mail

You may also use one of the standard actions:

- Redirect
- Delegate
- Sleep

The serial number for this task, SN=1009_58, must be included in the thread in order for the server to process the action. You may also reply with "Help" or "Actions" to see further information and examples.

When you are ready, continue on to the Part 3: Workflow exercise to extend the workflow, and then you can continue with the Part 4: Testing the Application exercise to test the new version of your application.

Part 3: Workflow

In Part 3 of this tutorial, you will edit your Leave Request Workflow by expanding the system tasks and by adding an additional manager approval outcome to rework the request. You will learn about escalations and how they can be used to remind a user of a task waiting to be actioned. You will also be tweaking the forms in the application to behave differently depending on where in the workflow the form is being used.

In this part, you will learn:

- How to add additional system tasks for sending email notifications
- How to customize an email task notification message body
- How to add an additional outcome and configure a new user task for responding to that outcome
- How to implement a rework loop in a workflow
- How to add an escalation to a user task
- Tweaking form rules for workflow integration

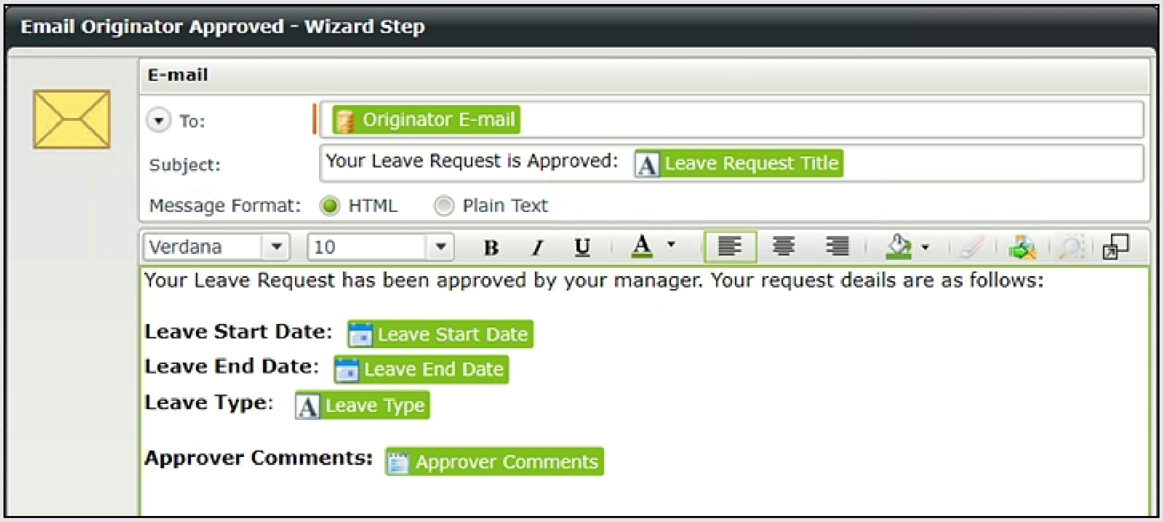
(If you have not completed Part 2: Forms of this tutorial yet, please do so before continuing with this part.)

Step 5: Add and configure email events for the manager outcomes

In Step 5, you will replace the two Placeholders with email events that notify the originator with the manager's decision.

Step 5 Tasks

1. Add a **Send E-mail** event to the approved outcome event, replacing the existing Placeholder. Route the email to the **Originator Email**.
2. Customize the subject line and message body using Item References properties. You may structure the content in any manner you like; the image below is an example to use as a guide.



Email Originator Approved - Wizard Step

E-mail

To: Originator E-mail

Subject: Your Leave Request is Approved: Leave Request Title

Message Format: ● HTML ○ Plain Text

Your Leave Request has been approved by your manager. Your request deails are as follows:

Leave Start Date: Leave Start Date

Leave End Date: Leave End Date

Leave Type: Leave Type

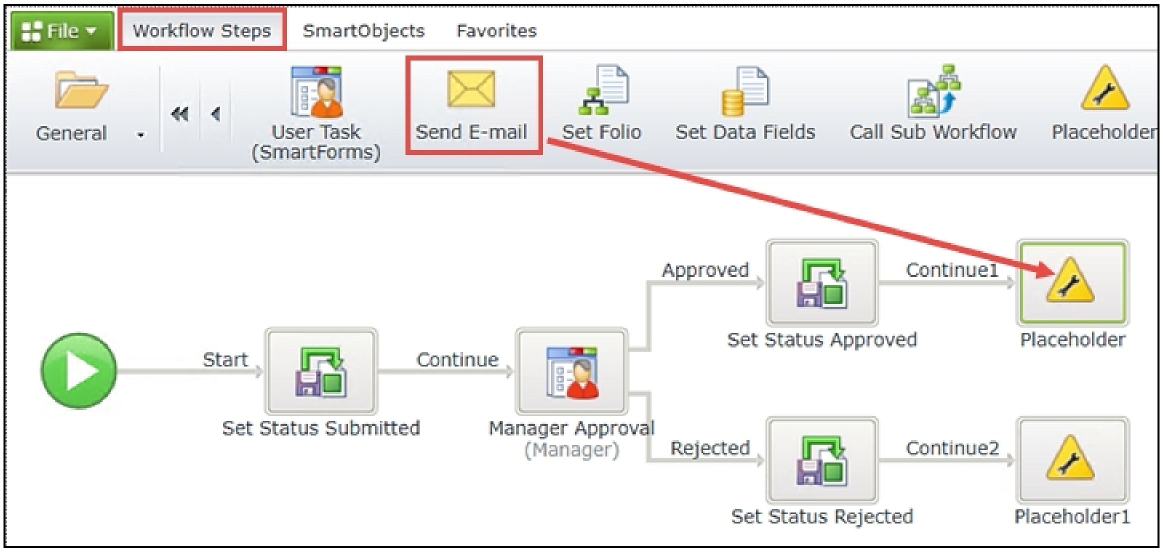
Approver Comments: Approver Comments

3. Delete the **Placeholder** event for the rejected outcome. **Add** a new continue outcome, then copy and paste the approved email event into the rejected event box. Edit the rejected email event to reflect the rejected decision.

Step 5 Walkthrough

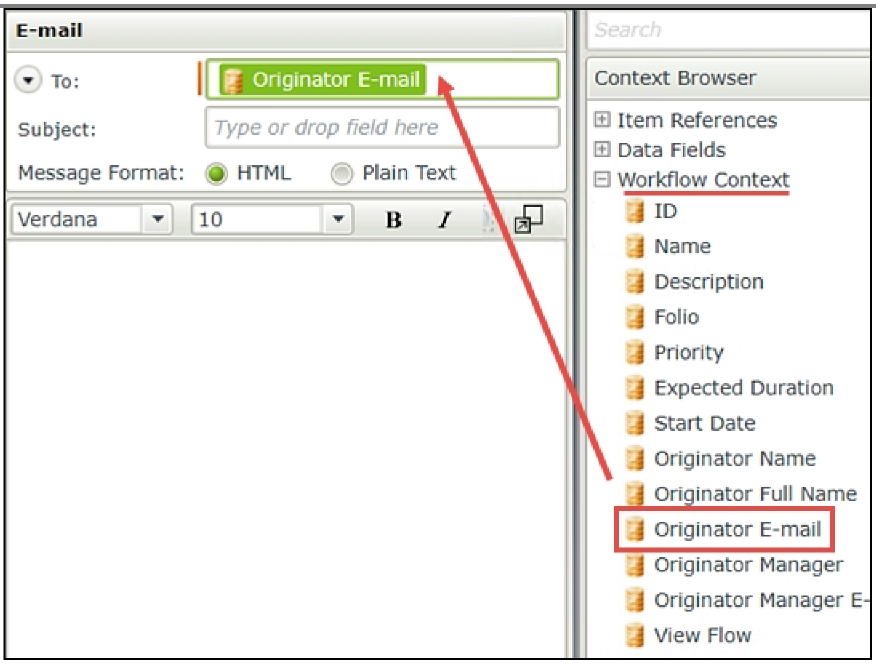
a. In K2 Designer, edit the **Leave Request Workflow**. (Right-click the workflow name in the category tree, then select **Edit**.)

b. Click the **Workflow Steps** tab to expose the wizards if it is not already open. Drag a **Send E-mail** event into the approved outcome Placeholder event. Click **Yes**, for the confirmation dialog that asks if you want to change this step.



c. The e-mail event wizard opens. Expand the **Context Browser**, then **Workflow Context**. Drag the **Originator Email** into the **To** field.

> **Note**
> Recall that the context browser provides dynamic values. Context browser properties are displayed as green blocks of text, so that you can distinguish them from other properties. At runtime, the context browser properties are replaced by "real" values. For example, in the image below, the Originator E-mail will be replaced by the actual email address of the form originator. Within the Workflow Context node, you will find a variety of properties that all relate to the current workflow in progress. Simply drop and drag any of these properties to use throughout your workflow.



E-mail

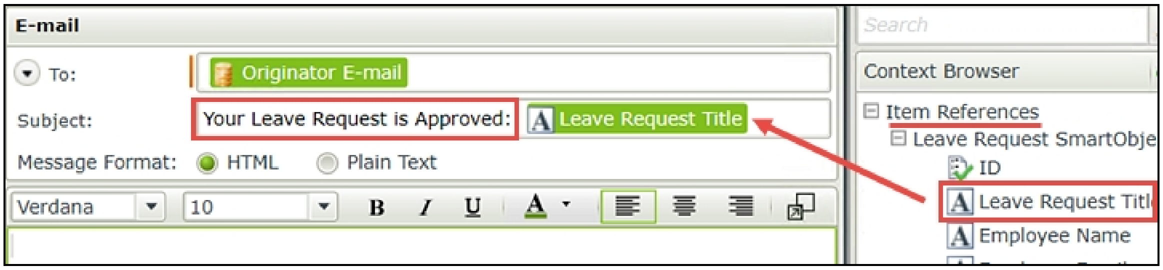
To: Originator E-mail

Subject: Type or drop field here

Message Format: ● HTML ○ Plain Text

d. For the **Subject** line, enter the following text, replacing the bracketed text [Title] with the **Item Reference** > **Leave Request SmartObject** > **Leave Request Title** found in the **Context Browser**.
   Your Leave Request is Approved: [Title]

> Note
> Item References are "pointers" to values found in the SmartObject associated with the workflow. Where context browser properties are generally system-related values, item references are specific to the associated SmartObject. Once again, at runtime, K2 will replace the item reference with the actual value of the SmartObject record for the specific workflow instance.



E-mail

To: Originator E-mail

Subject: Your Leave Request is Approved: Leave Request Title

Message Format: ● HTML ○ Plain Text

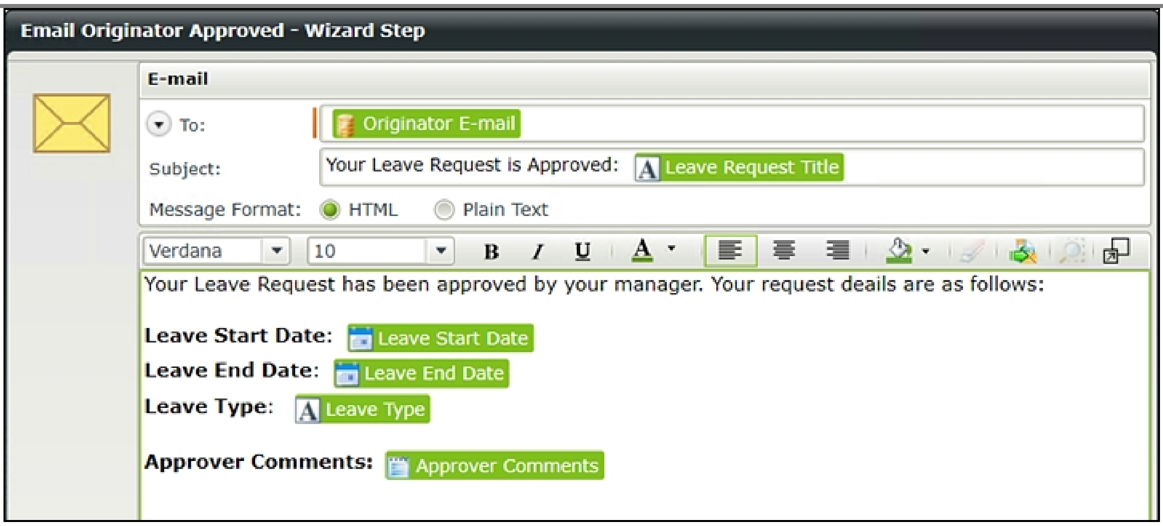
e. Using the same approach as above, enter the following content into the message body, again replacing the bracketed text with the appropriate **Item References** properties from the context browser.

   Your Leave Request has been approved by your manager. Your request details are as follows:

   Leave Start Date: [Leave Start Date]
   Leave End Date: [Leave End Date]
   Leave Type: [Leave Type]

   Approver Comments: [Approver Comments]

Your email should look like the image below. Click **OK** when ready.



Email Originator Approved - Wizard Step

E-mail

To: Originator E-mail

Subject: Your Leave Request is Approved: Leave Request Title

Message Format: ● HTML ○ Plain Text

Your Leave Request has been approved by your manager. Your request deails are as follows:

Leave Start Date: Leave Start Date

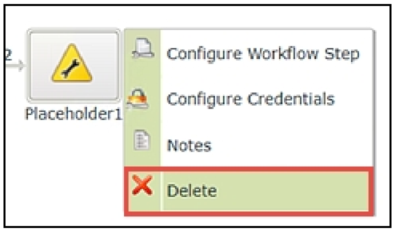
Leave End Date: Leave End Date

Leave Type: Leave Type

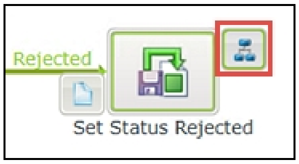
Approver Comments: Approver Comments

f. Double-click the Placeholder title of the step and change it to
Email Originator Approved

g. In the basic Leave Request tutorial, you copied and pasted events to speed up the build process. You will do the same here. Begin by deleting the rejected side Placeholder so that you can paste the e-mail event you just created in its place. Right-click the **Placeholder1** event and select **Delete**. Click **Yes** for the confirmation dialog that appears.

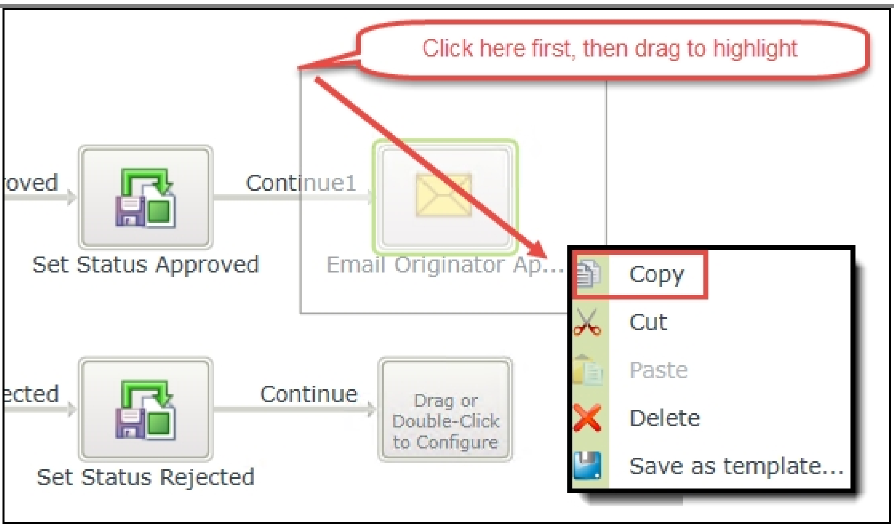


h. Next, you need to add back in the Continue outcome line, which will add a corresponding event holder. Hover over the Set Status Rejected event until you see the **Workflow Step Outcomes** icon (it looks like a blue mini-flowchart). (If you don't see the icon when hovering, click anywhere on the design canvas to refocus your mouse to the canvas, then try hovering again.) Click the icon when you can see it.



i. On the Workflow Step Outcomes configuration screen, click **Add**. Replace **New Outcome** with
Continue
then click **OK**. You should now see the continue line with an empty event box.

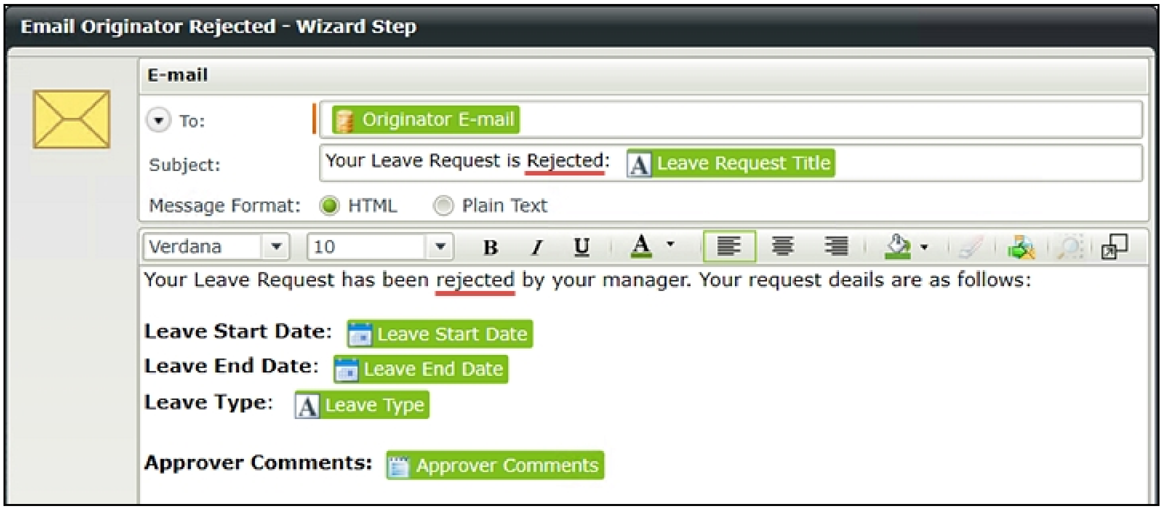
j. Copy the **E-mail event** you just configured. To copy an event in K2 Designer, click outside of the event box, then drag over the event to highlight it. Click **Copy** when the option menu opens. (If you get a dialog about allowing access to the clipboard, click **Yes** to continue.)



k. Right-click inside the empty event box and click **Paste**. You should now see the Email Originator event has been replicated. You need to make a few adjustments to reflect the rejected outcome.

l. Double-click the event to open its wizard. Change the **Name** to
Email Originator Rejected
then click **Next**.

m. There are no outcomes to configure, so click **Next**. Change the **Subject** line and the message body to reflect the manager's rejected decision. Use the image below as a guide, the changed values are underlined in red. Click **Finish** when ready.


Email Originator Rejected - Wizard Step

E-mail

To: Originator E-mail

Subject: Your Leave Request is Rejected: Leave Request Title

Message Format: ● HTML ○ Plain Text

Verdana 10

Your Leave Request has been rejected by your manager. Your request deails are as follows:

Leave Start Date: Leave Start Date
Leave End Date: Leave End Date
Leave Type: Leave Type

Approver Comments: Approver Comments

STEP 5 REVIEW

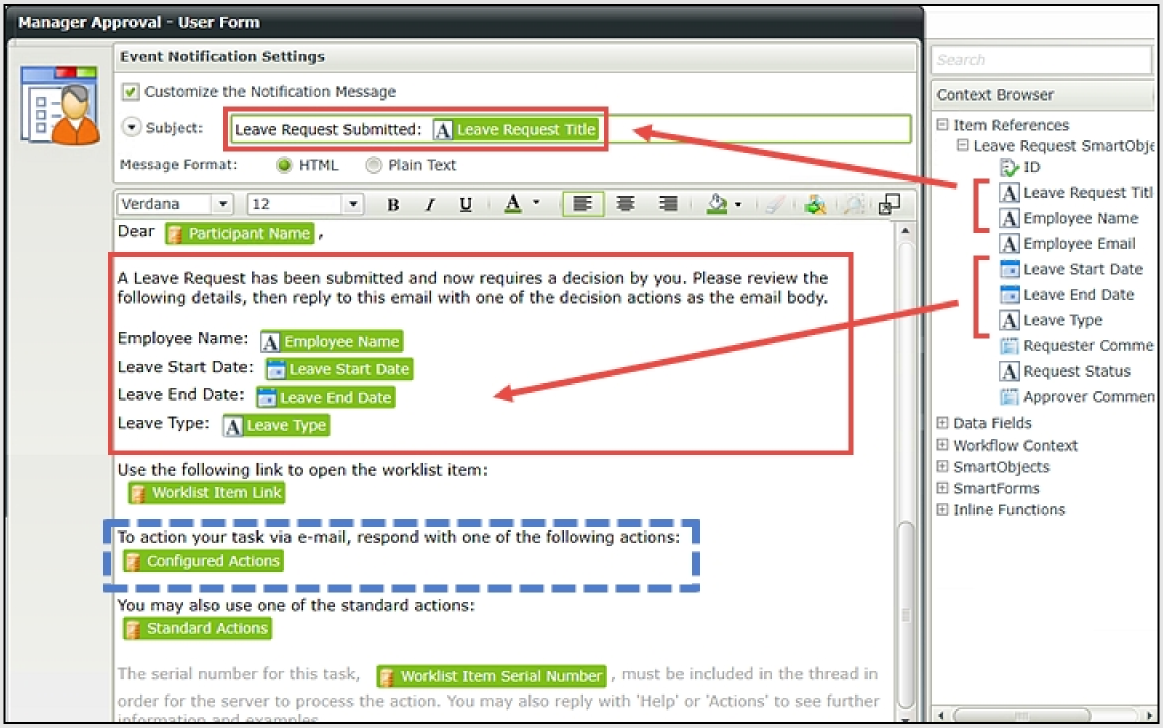
In this step, you added an email event and customized the email using properties found in the context browser. The placeholder values from the context browser (indicated by green blocks in the wizard screens) will be replaced at runtime by K2 with the actual values for that instance of the workflow. The workflow context offers properties that contain details about the workflow itself, such as the workflow originator, originator's email, manager, folio and so on. Item references contain SmartObject property references. For example, the [Leave Request Title] item reference is replaced at runtime with whatever value was entered into the form's Leave Request Title field. This allows you to customize and personalize your workflow. You also discovered the ease of reusing content by copying and pasting events.

Step 6: Add a third manager outcome to rework the request and configure a new user rework task

In this step, you are going to add an additional outcome for the manager approval task. This third outcome, called Rework, will send the request back to the originator where it can either be resubmitted or canceled. You will add a new user task for the form originator to action the rework task, then add events to complete the rework decision outcomes.

Step 6 Tasks

1. Add a third outcome to the manager approval task and name it
   Rework
   and allow the user to action the task without opening the form.
2. Customize the manager's task notification email in any manner you like. You may use the image below as a guide.



Manager Approval - User Form

Event Notification Settings

☑ Customize the Notification Message

Subject: Leave Request Submitted: Leave Request Title

Message Format: ● HTML ○ Plain Text

Dear Participant Name ,

A Leave Request has been submitted and now requires a decision by you. Please review the following details, then reply to this email with one of the decision actions as the email body.

Employee Name: Employee Name
Leave Start Date: Leave Start Date
Leave End Date: Leave End Date
Leave Type: Leave Type

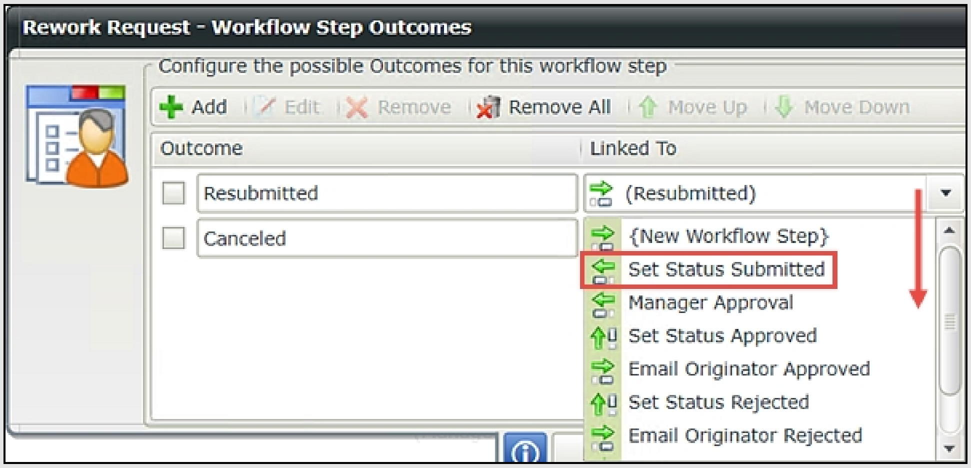
Use the following link to open the worklist item:
Worklist Item Link

To action your task via e-mail, respond with one of the following actions:
Configured Actions

You may also use one of the standard actions:
Standard Actions

The serial number for this task, Worklist Item Serial Number , must be included in the thread in order for the server to process the action. You may also reply with 'Help' or 'Actions' to see further information and examples

3. Add a **User Task (SmartForms)** to the rework event box. Name the new task
   Rework Request
   then add the following actions:
   Resubmitted
   Canceled
4. Use the Leave Request form for the rework form and route the task to the **Originator**.
5. Change the **Resubmitted** outcome so that it routes to the Set Status Submitted event.



Rework Request - Workflow Step Outcomes

Configure the possible Outcomes for this workflow step

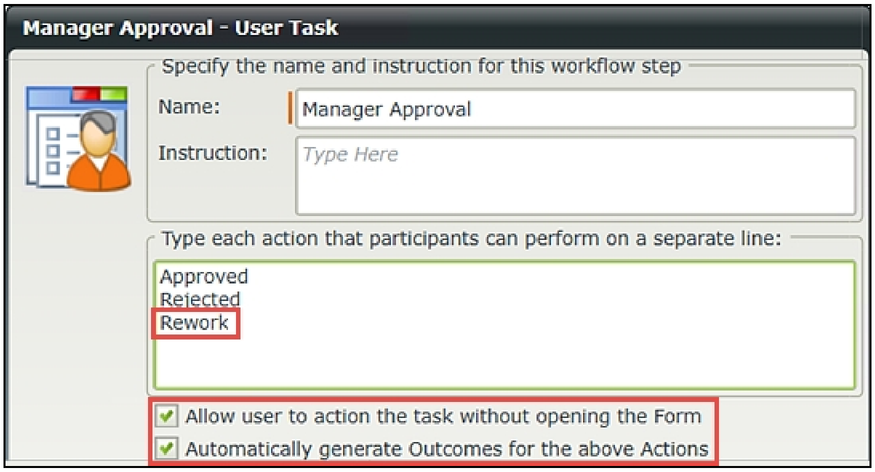
Outcome	Linked To
□ Resubmitted	(Resubmitted)
□ Canceled	

6. Copy and paste the **Set Status Submitted** event into the empty event for the **Canceled** outcome. **Remove** the continue outcome for the cancel event as you want the workflow to end at this point. Change the **Request Status** value to
Canceled
leaving the **ID** value set as it is.

Step 6 Walkthrough

a. Double-click the **Manager Approval** user task to open its wizard. Enter a third action and name it
Rework
then confirm the option to **Allow user to action the task without opening the Form** is CHECKED. Confirm the option to **Automatically generate Outcomes for the above Actions** is CHECKED. Click **Next** to continue.

Tip
By allowing the user to action the task without opening the form, you are enabling **SmartActions**. SmartActions are a K2 feature that allows the user to simply reply to the email with the action entered in the message body. This eliminates the need for the approver to open the form, select an action, then submit the form. You will edit the task notification email message to let the user know they can action the task using SmartActions. Selecting the option to automatically generate outcomes will tell K2 to add another outcome line for the rework action.



Manager Approval - User Task

Specify the name and instruction for this workflow step

Name: Manager Approval

Instruction: Type Here

Type each action that participants can perform on a separate line:

Approved
Rejected
Rework

☑ Allow user to action the task without opening the Form

☑ Automatically generate Outcomes for the above Actions

b. There are no changes to the **Linked To** values, so click **Next**. The Leave Request Form is correct, so click **Next**. There are no changes to the work item rules, so click **Next**. There are no changes to the Participants, so click **Next**.

c. On the **Event Notification Settings** screen (email message/body content) CHECK the box to **Customize the Notification Message**. Notice that after you check the box, K2 adds the default task notification content in the message body. You are going to add additional text to the email so that the manager can approve, reject or rework the request using SmartActions.

Note
Notice the paragraph in the email that contains the Work Item Context called *Configured Actions*. (Refer to the image below, blue dashed line.) At runtime, configured actions will be replaced with the actual actions you have configured (Approved, Rejected, Rework). Using SmartActions, the user can simply reply to the email with one of the actions as the message body.

For this step, you will add text to the notification email and customize the subject line. For the subject line, enter
Leave Request Submitted: [Leave Request Title]
then drag the **Item References** > **Leave Request SmartObject** > **Leave Request Title** to the end of the subject line, replacing the bracketed text. (See the image below as a reference if necessary.)

In the message body, just below the Dear Participant line, enter the following, dragging values from the context browser as needed for the fields contained in brackets.

A Leave Request has been submitted and now requires a decision by you. Please review the following details, then reply to this email with one of the decision actions as the email message body.

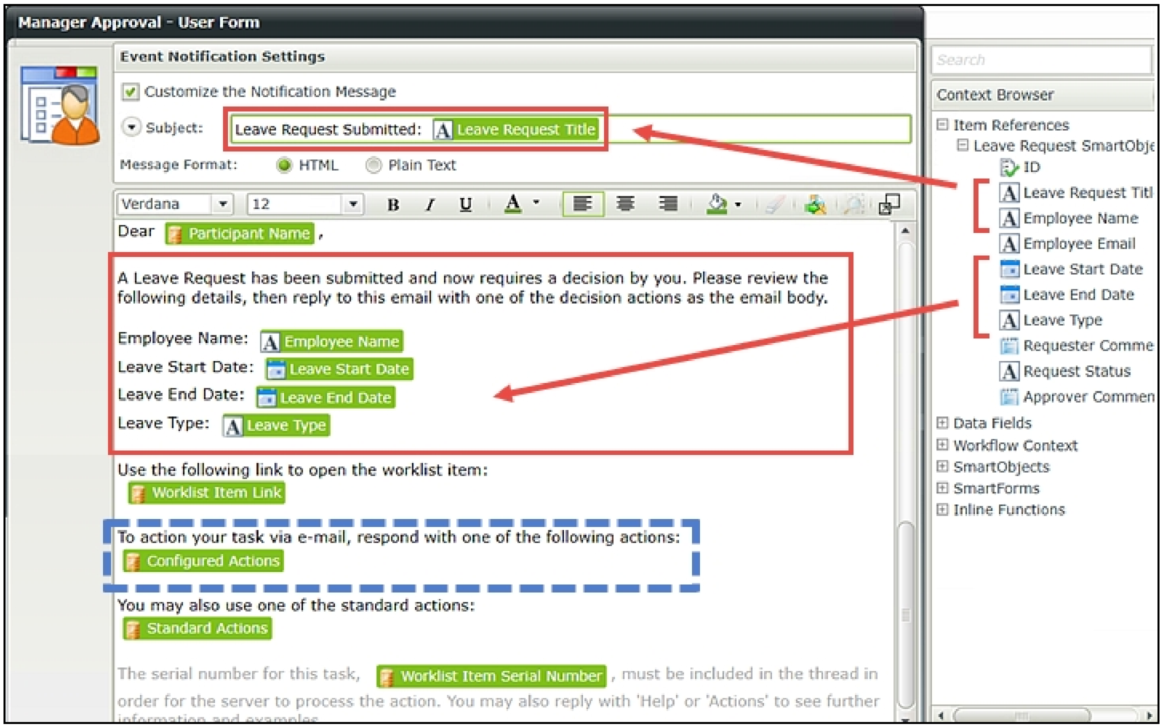
Employee Name: [Employee Name]

Leave Start Date: [Leave Start Date]

Leave End Date: [Leave End Date]

Leave Type: [Leave Type]

(Click **Finish** when ready.)



Manager Approval - User Form

Event Notification Settings

☑ Customize the Notification Message

Subject: Leave Request Submitted: Leave Request Title

Message Format: ● HTML ○ Plain Text

Dear Participant Name ,

A Leave Request has been submitted and now requires a decision by you. Please review the following details, then reply to this email with one of the decision actions as the email body.

Employee Name: Employee Name

Leave Start Date: Leave Start Date

Leave End Date: Leave End Date

Leave Type: Leave Type

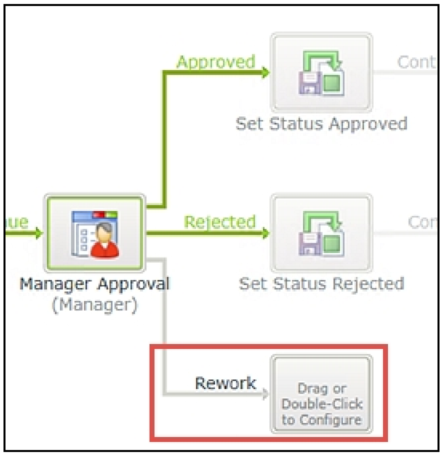
Use the following link to open the worklist item:
Worklist Item Link

To action your task via e-mail, respond with one of the following actions:
Configured Actions

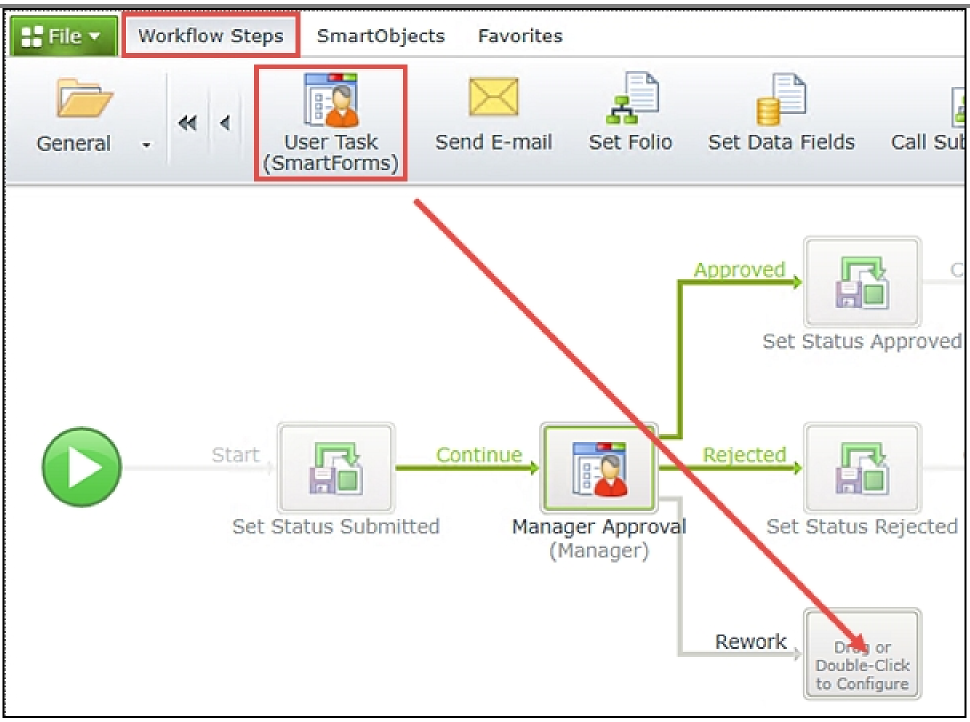
You may also use one of the standard actions:
Standard Actions

The serial number for this task, Worklist Item Serial Number , must be included in the thread in order for the server to process the action. You may also reply with 'Help' or 'Actions' to see further information and examples

Notice that K2 has added a third action outcome with an event box. You will now add a user task for the form originator to either resubmit or cancel the request.



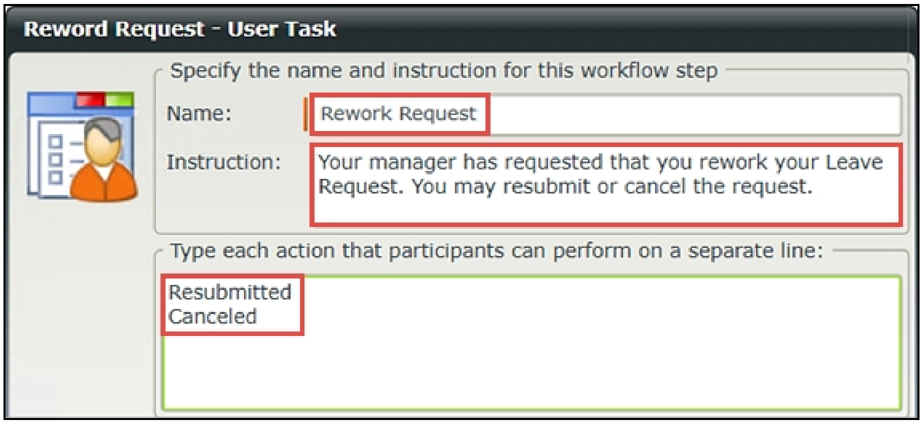
d. Drag a **User Task (SmartForms)** event into the new rework event box.



e. The User Task wizard opens. Name the new event
Rework Request
and enter the following for the Instruction:

Your manager has requested that you rework your Leave Request. You may resubmit or cancel the request.

For the actions, enter
Resubmitted
Canceled
then click **Next**.


Reword Request - User Task

Specify the name and instruction for this workflow step

Name: Rework Request

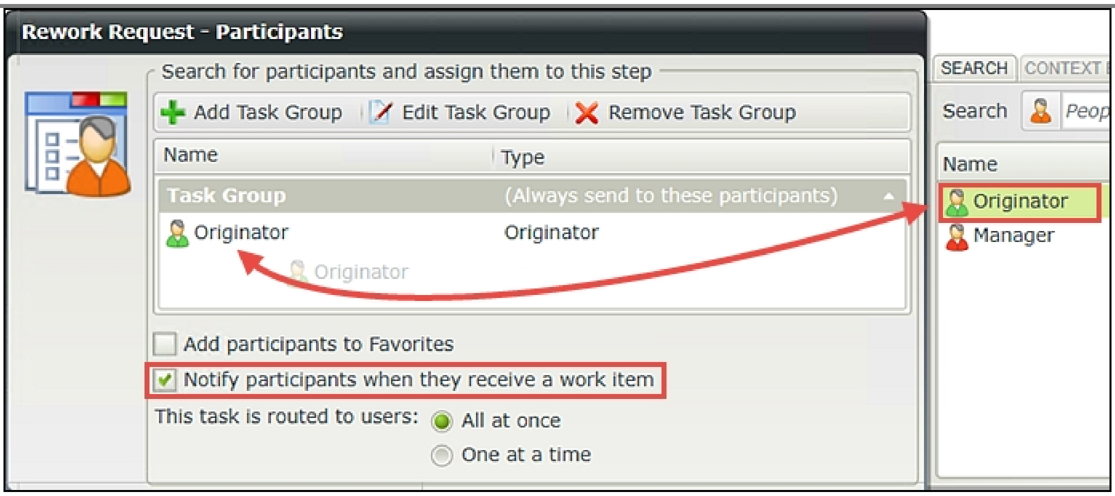
Instruction: Your manager has requested that you rework your Leave Request. You may resubmit or cancel the request.

Type each action that participants can perform on a separate line:

Resubmitted
Canceled

f. The Leave Request Form is correct, so click **Next**. There are no changes to the work item rules, so click **Next**.

g. On the Participants screen, drag the **Originator** into the Task Group box. CHECK the box to **Notify participants when they receive a work item**. To save time you will not customize the email content, so click **Finish**.



Rework Request - Participants

Search for participants and assign them to this step

Add Task Group | Edit Task Group | Remove Task Group

Name	Type
Task Group	(Always send to these participants)
Originator	Originator

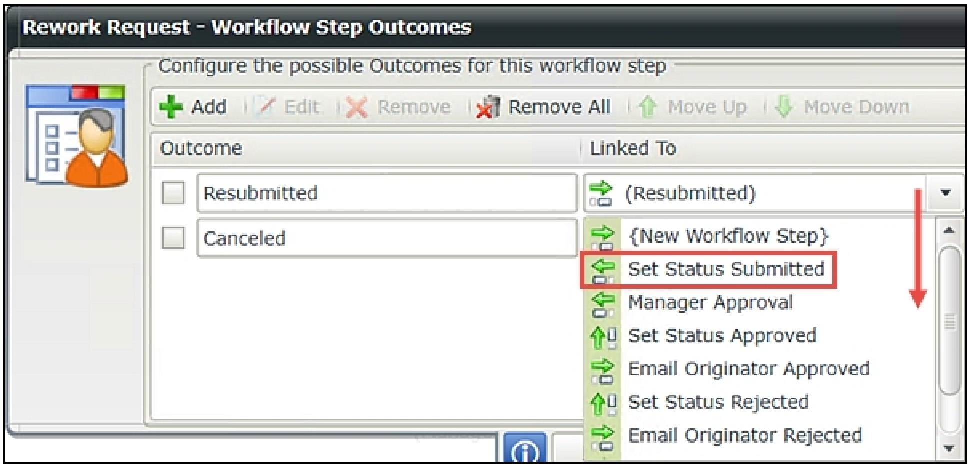
☐ Add participants to Favorites

☑ Notify participants when they receive a work item

This task is routed to users: ● All at once ○ One at a time

Now you want to configure the events for the two rework outcomes (**Resubmitted** and **Canceled**). First, you will route the resubmitted outcome back to the **Set Status Submitted** event so that the workflow returns to a previous step in the workflow. For the canceled outcome, you will copy and paste a **Set Status** event, then update the **Request Status** value so that it indicates the request was canceled.

h. Double-click the **Rework Request** user task to open its wizard. Click **Next** from the home screen. Change the **Linked To** value for the **Resubmitted** outcome to **Set Status Submitted**. Click **Next** until you can click **Finish** to complete this step (three times).



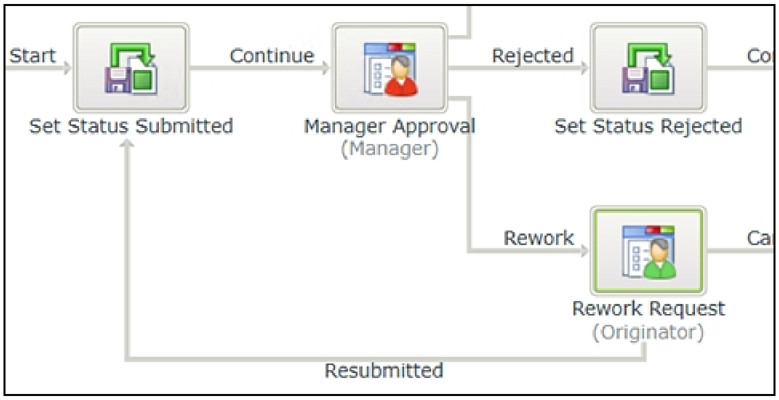
Rework Request - Workflow Step Outcomes

Configure the possible Outcomes for this workflow step

Add | Edit | Remove | Remove All | Move Up | Move Down

Outcome	Linked To
☐ Resubmitted	(Resubmitted)
☐ Canceled	

Notice now that K2 routes the **Resubmitted** outcome back to the **Set Status Submitted** event. This will create a loop effect in the workflow between the rework and resubmitted outcomes until a different action is selected.

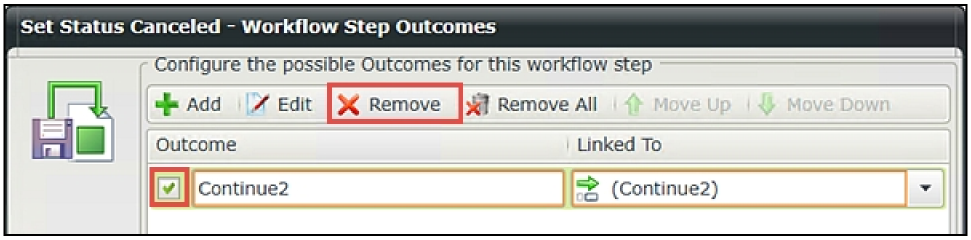


i. Copy and paste the **Set Status Submitted** event into the Canceled event. (See Step 5 (j) if you need a reminder on how to copy and paste events in K2 Designer.)

j. Double-click the new **Set Status Submitted1** event to open its wizard. Change the name to
Set Status Canceled
then click **Next**.

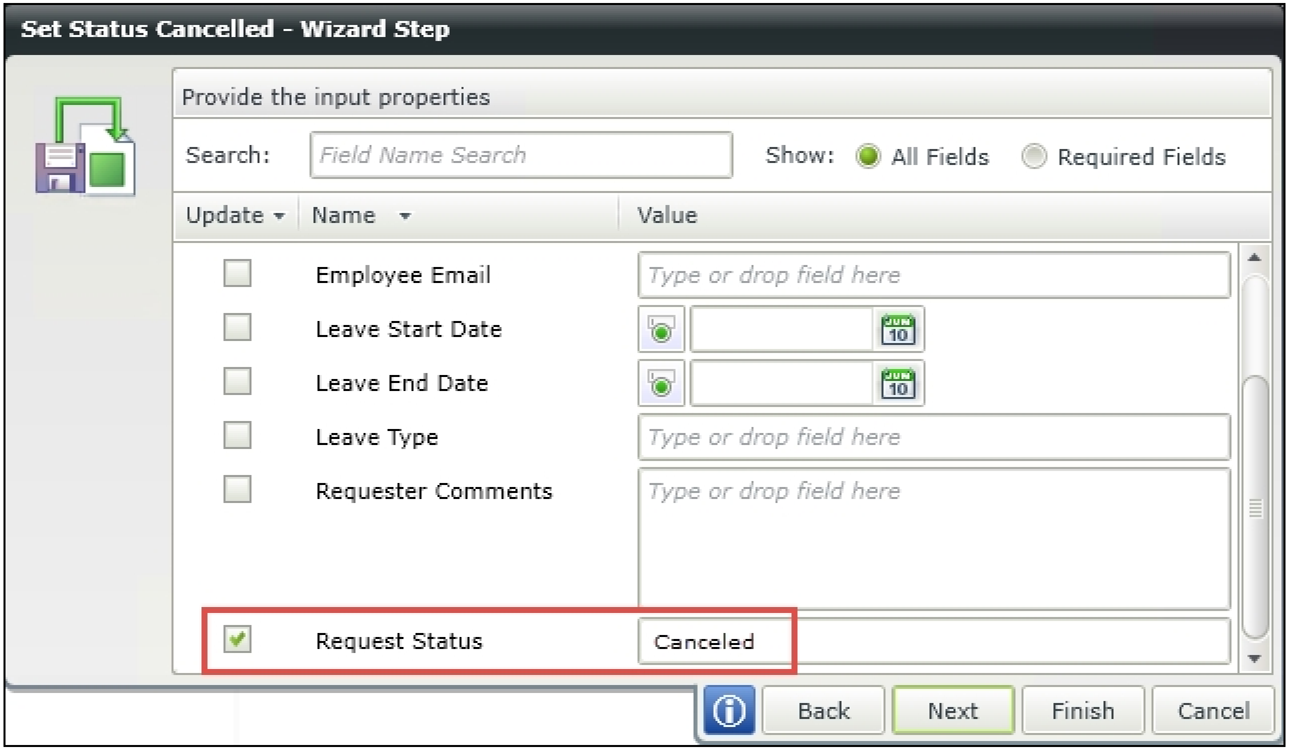
k. On the Workflow Step Outcomes screen, CHECK the box next to the **Continue2** outcome and click **Remove**. (Depending on how you've worked through the steps, your outcome may have a slightly different name from Continue2.) Click **Next**.

You are removing this outcome because you don't need to continue the workflow after this step. If the decision is to cancel the request, then after the request status has been updated, the workflow will be complete.


Set Status Canceled - Workflow Step Outcomes
Configure the possible Outcomes for this workflow step
Add
Edit
Remove
Remove All
Move Up
Move Down
Outcome
Linked To

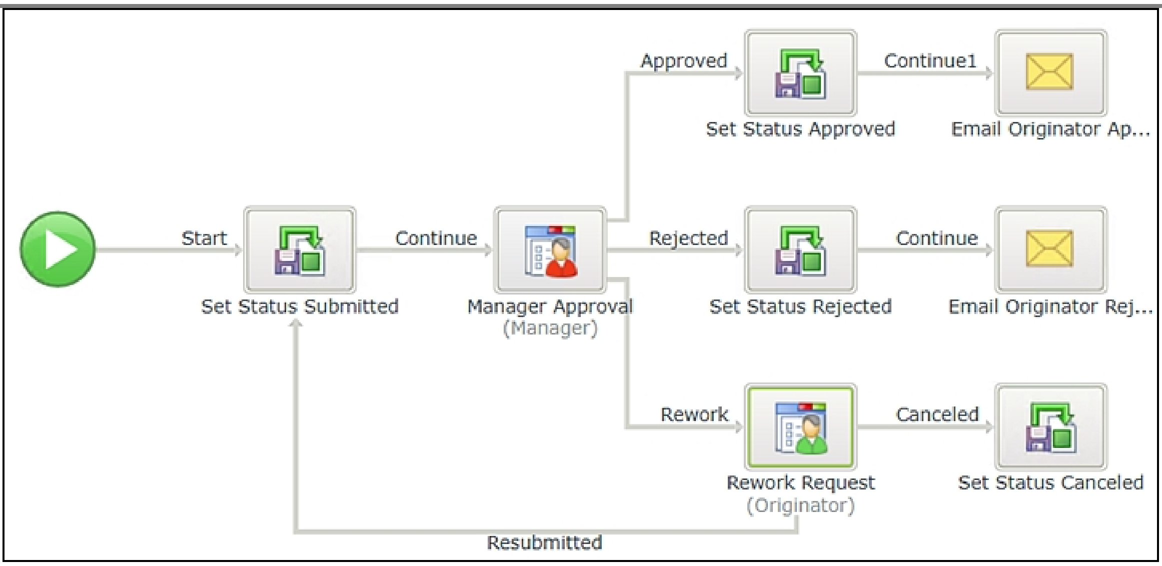
Outcome	Linked To
☑ Continue2	(Continue2)

l. Click **Next** to reach the input properties screen, then scroll down and change the **Request Status** value to
Canceled
then click **Next**. You do not need to change the **ID** value as it is telling K2 *which* record you want to update (the current record). Click **Finish** when ready.



Update	Name	Value
□	Employee Email	Type or drop field here
□	Leave Start Date	
□	Leave End Date	
□	Leave Type	Type or drop field here
□	Requester Comments	Type or drop field here
☑	Request Status	Canceled

m. Your completed workflow should look like the image below.



n. **Save** your work. (**File** > **Save**)

STEP 6 REVIEW

In this step you added a third action for the manager approval task. You customized the task notification email so that the manager can use SmartActions to action the task directly from the email. You added a user task for the form originator in the event the manager decides the request needs to be reworked. You routed the resubmitted outcome so that the workflow essentially starts over again if the originator resubmits the request. Finally, you added a system event to update the request status if the originator decides to cancel the request.

Step 7: Add an escalation for the manager approval task

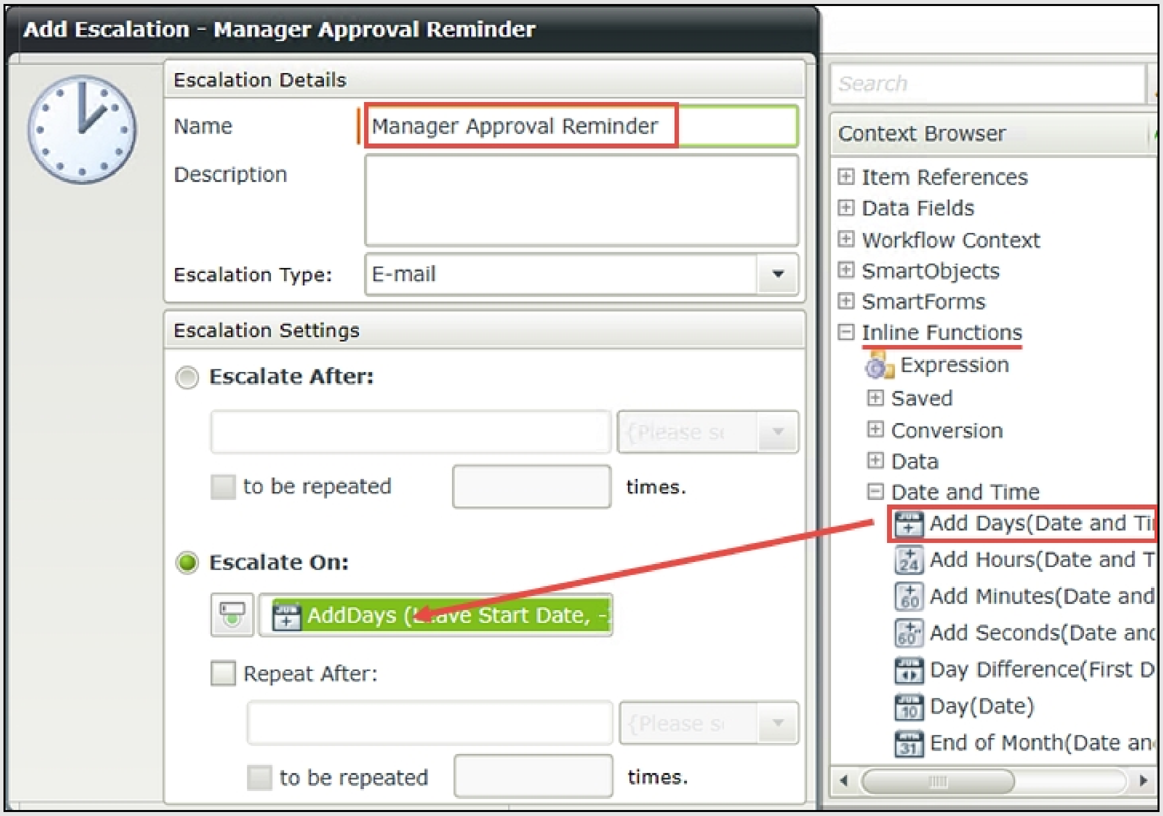
In this step, you will explore escalations and how you can use them to keep your workflows running smoothly and in a timely manner. An escalation kicks in when a user has not responded to a task assigned to them within a specified time period. An escalation can be as simple as an email reminder or more complex, such as automatically redirecting the workflow to another user. Escalations can be set up to repeat themselves (for example, an email reminder every two days) and they can have repeat limits applied (such as send two email reminders, then redirect to another user).

Caution
One word of caution: you don't want to crowd users' in-boxes with reminder notices, so take care in which events you apply escalations to and give some thought as to who you are reminding. For example, it might be a better choice to send the escalation notice to the originator so that they can follow up with their manager, or limit the number of escalations to only what is really necessary.

In this step, you will apply an escalation to the manager approval user task that will send a reminder email to both the manager and the form originator if the manager has not responded within two days of the Leave Start Date.

Step 7 Tasks

1. **Add** a new escalation to the manager approval event and name it
   Manager Approval Reminder
   then keep the default **E-mail** escalation type.
2. Select the **Escalate On** option then using the Inline Functions to subtract two days from the Leave Start Date value.



Add Escalation - Manager Approval Reminder

Escalation Details

Name: Manager Approval Reminder

Description:

Escalation Type: E-mail

Escalation Settings

○ Escalate After:

□ to be repeated times.

● Escalate On:

AddDays (Leave Start Date, -

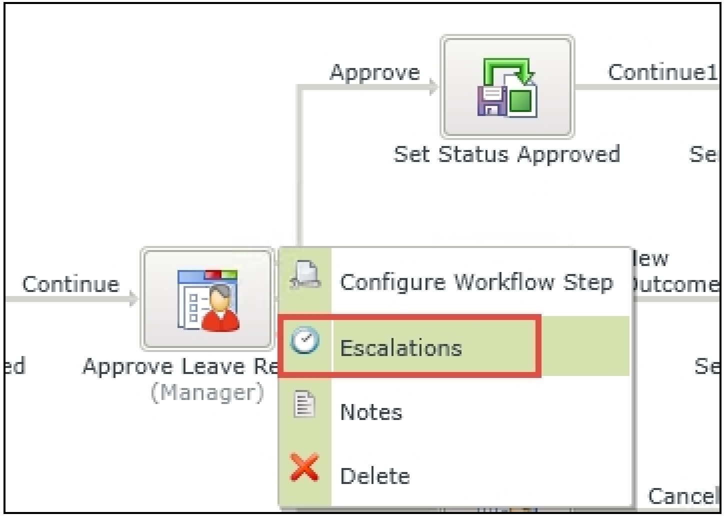
□ Repeat After:

□ to be repeated times.

3. Configure the email notification to be sent to both the manager and the form originator. Use properties from the context browser to customize the message.

Step 7 Walkthrough

a. Click the **Manager Approval** event to highlight it. Right-click and select **Escalations**.

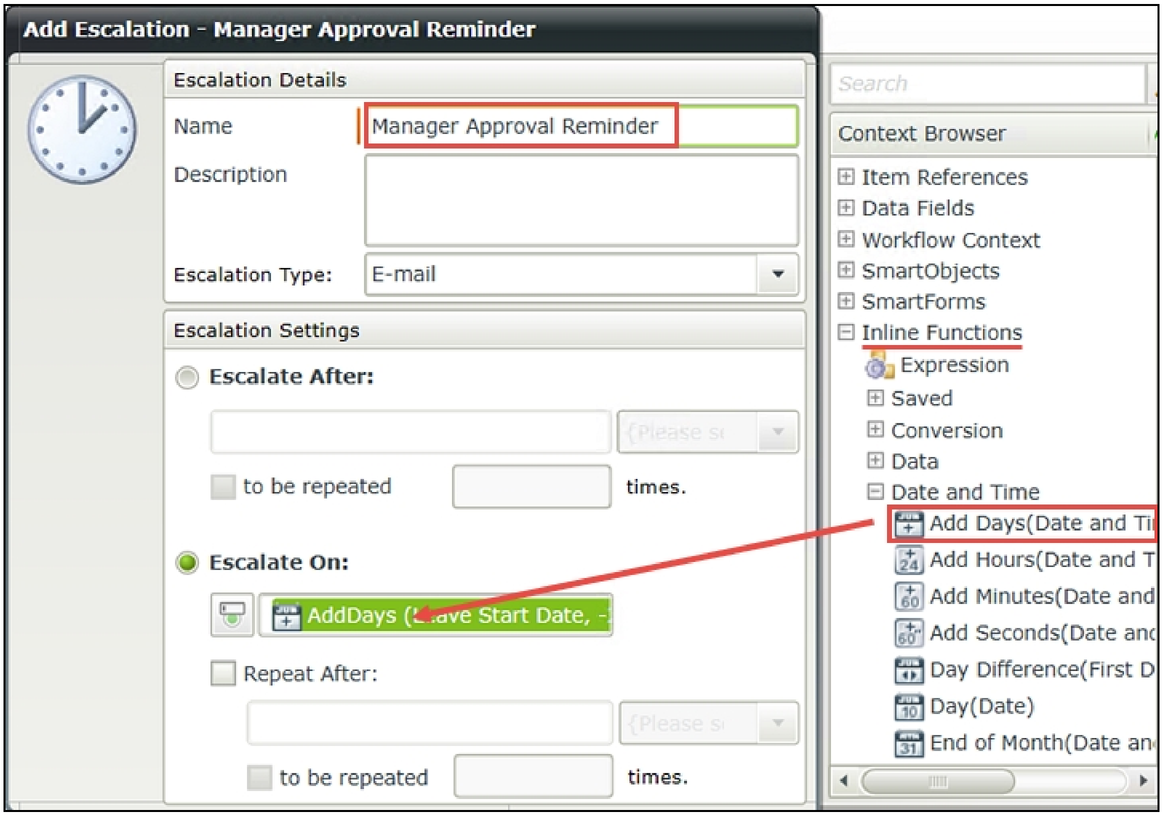


b. When the Add Escalation screen opens, click **Add**.

c. On the Escalation Settings screen, name the escalation
Manager Approval Reminder
then select the option **Escalate On**.

d. You want the escalation to fire 2 days before the leave start date. You will use an inline function to achieve this. In the context browser, expand the **Inline Functions** > **Date and Time** node and drag the **AddDays** function into the date field as shown below.

Note
There are three options for escalations. The most common perhaps, is **Email**, where an email reminder is sent to the destination user (and optionally, the form originator) letting them know they have an outstanding task. You can choose to **Redirect** the task to another user or **Expire** the task, which essentially marks the task complete and moves the workflow on to the next step.



Add Escalation - Manager Approval Reminder

Escalation Details

Name	Manager Approval Reminder
Description	
Escalation Type:	E-mail

Escalation Settings

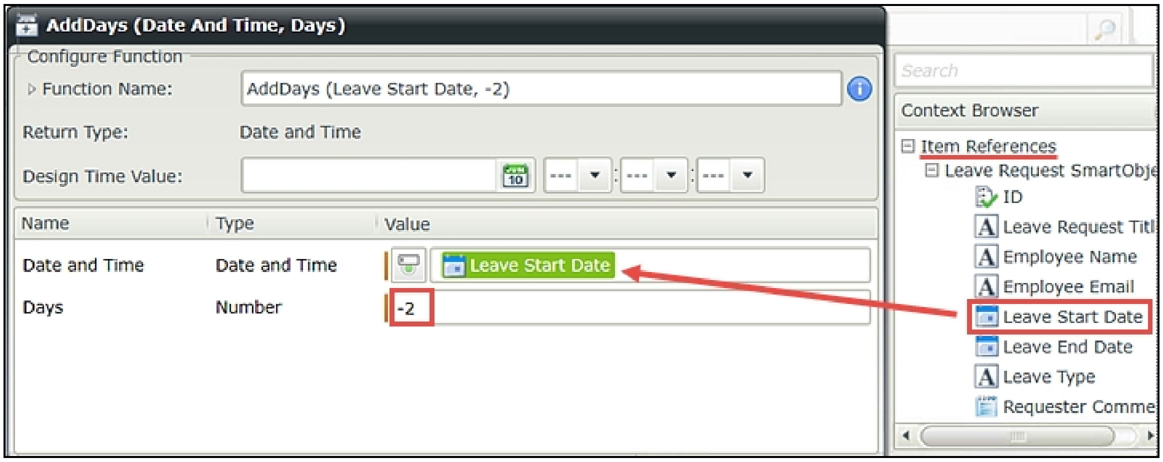
○ Escalate After: {Please s
□ to be repeated times.

● Escalate On: AddDays (Leave Start Date, -

□ Repeat After: {Please s
□ to be repeated times.

e. On the **AddDays** screen, expand **Item References** in the context browser and drag the **Leave Start Date** property into the **Date and Time** field. For the **Days** value, enter (minus 2)
-2
then click **OK**, then click **Next**.

Here, you are simply telling K2 that you want to kick off this escalation two days prior to the Leave Start Date.



AddDays (Date And Time, Days)

Configure Function

Function Name:	AddDays (Leave Start Date, -2)
Return Type:	Date and Time
Design Time Value:	--- : --- : ---

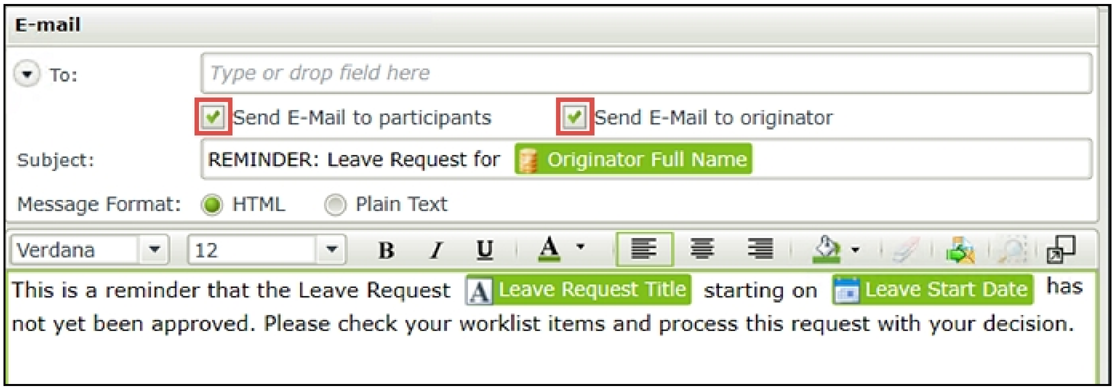
Name	Type	Value
Date and Time	Date and Time	Leave Start Date
Days	Number	-2

f. On the Escalation E-mail screen, CHECK the boxes to send the notification to the **participant** and the **originator**. This will send the email to the manager and the form originator. See the screen shot below for guidance, if needed.
For the subject line, enter
REMINDER: Leave Request for [Originator Name]

then expand the workflow context node in the context browser and drag the **Originator Full Name** into the subject line (replacing the bracketed text above).
For the message body, enter the following (again replacing the bracketed text with item references).

This is a reminder that the Leave Request [Leave Request Title] *starting on* [Leave Start Date] *has not yet been approved. Please check your worklist items and process this request with your decision.*

Click **Finish** when you are done to complete the Escalation wizard screen.



E-mail

To: Type or drop field here

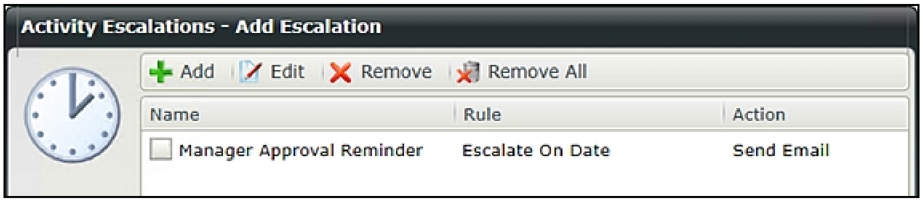
☑ Send E-Mail to participants ☑ Send E-Mail to originator

Subject: REMINDER: Leave Request for Originator Full Name

Message Format: ● HTML ○ Plain Text

This is a reminder that the Leave Request Leave Request Title starting on Leave Start Date has not yet been approved. Please check your worklist items and process this request with your decision.

g. Your screen should look like the image below. Click **OK** to close the Activity Escalations screen.



Activity Escalations - Add Escalation

Name	Rule	Action
□ Manager Approval Reminder	Escalate On Date	Send Email

h. **Save** your work.

STEP 7 REVIEW

In this step, you added an escalation to send an email reminder two days prior to the leave start date in the event the manager has not actioned the request yet. You customized the email to include details about the specific request and configured the email to route to the manager and the form originator.

Step 8: Deploy the workflow

Before the workflow changes will be applied, you must deploy the workflow to the K2 server. Deploying a workflow publishes it and makes the current version available to your users. You must redeploy your workflow each time you makes changes.

Caution
Be aware that any workflow instances that are currently active will continue on with the workflow version from which they were started. Redeploying a workflow does not 'update' any active workflow instances. If you are making significant changes, consider waiting until all previous workflow instances are complete before deploying a new version.

Step 8 Tasks

1. Deploy the workflow.

Step 8 Walkthrough

a. Deploy the workflow (**File** > **Deploy**)

Step 9: Edit a rule to enable the Approver Comments field for the manager

Now that the workflow changes have been published, you need to make a couple of tweaks to the forms used by the workflow, because you want the forms to behave differently depending on the current task. Recall in a previous step you added the Approver Comments field to the Leave Request item view and made it read-only by default. In this step, you will edit a rule to enable the comments field for the manager approval task. To do this, you will edit a rule on the Workflow Task *state*.

> Note
> A *state* represents the configuration and behavior of a form at a given point in the workflow. States are most commonly configured at the user task event level. For example, you might want the form configuration to contain certain fields for the manager approval task, then contain different fields for the originator rework task. K2 can use the same form and views with adjusted configurations 'assigned' to a specific state. Breaking this down a little further, each user task can be assigned a state of its own. The form designer can then manipulate the form's styling and rules specific to that single state.

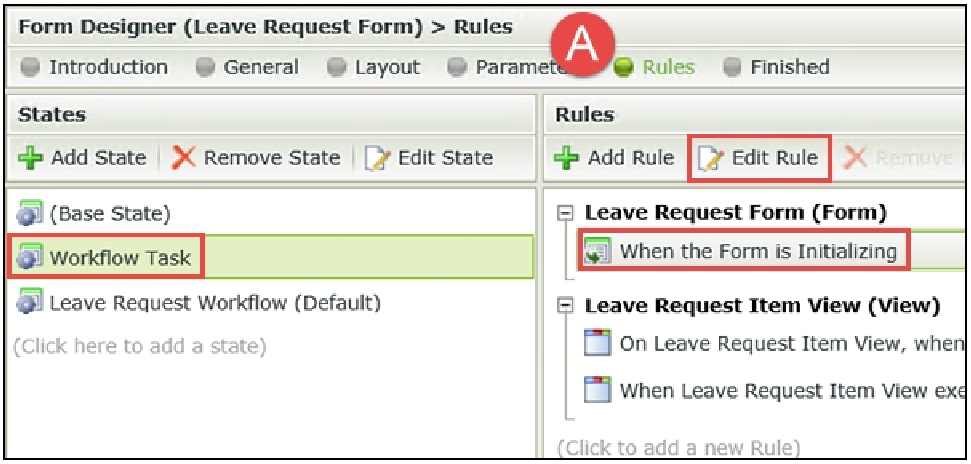
Step 9 Tasks

1. Check out and then edit the **Leave Request Form**
2. Edit the **Form Initializing** rule for the **Workflow Task** state.
3. In the **if current Workflow Activity is Manager Approval**condition, add an action to **set a View control's properties** on the Leave Request item view that enables the Approver Comments Text Area.

Step 9 Walkthrough

a. In K2 Designer, right-click the **Leave Request Form** and check it out (if it is not already), then **Edit** it.

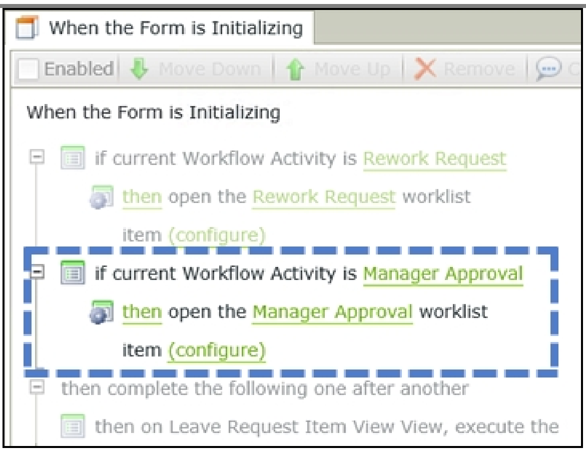
b. Click the **Rules** tab in the Breadcrumb Bar. (**A** below) Highlight the **Workflow Task** state, then highlight the **When the Form is Initializing** rule. Click **Edit Rule**. (Once again, the Workflow Task state represents the form and view configuration that the approving manager will see.)


Form Designer (Leave Request Form) > Rules
A
Introduction
General
Layout
Paramete
Rules
Finished
States
Add State
Remove State
Edit State
(Base State)
Workflow Task
Leave Request Workflow (Default)
(Click here to add a state)
Rules
Add Rule
Edit Rule
Remove
Leave Request Form (Form)
When the Form is Initializing
Leave Request Item View (View)
On Leave Request Item View, when
When Leave Request Item View exe
(Click to add a new Rule)

c.

> Note
> **Rules: Events, Conditions and Actions**
> As a reminder, rules consist of events, conditions and actions.
> - **Events** are *when something occurs*, such as clicking a button or when the form or view loads.
> - **Conditions** evaluate *if a criteria has been met*: if the condition is true, continue. For example, evaluating a required field to determine if the field contains a value.
> - **Actions** *do something*. If the event and condition passes, then perform the action, such as starting a workflow, or calling another rule, or enabling a form field.

In the Rule Definition pane, locate the condition that says **if current Workflow Activity is Manager Approval**. Notice that this condition already has an action to open the Manager Approval worklist item. K2 added this condition and action when you configured the Manager Approval user task. You want to add an action at this point to enable the Approver Comments field.



When the Form is Initializing

if current Workflow Activity is Rework Request

then open the Rework Request worklist item (configure)

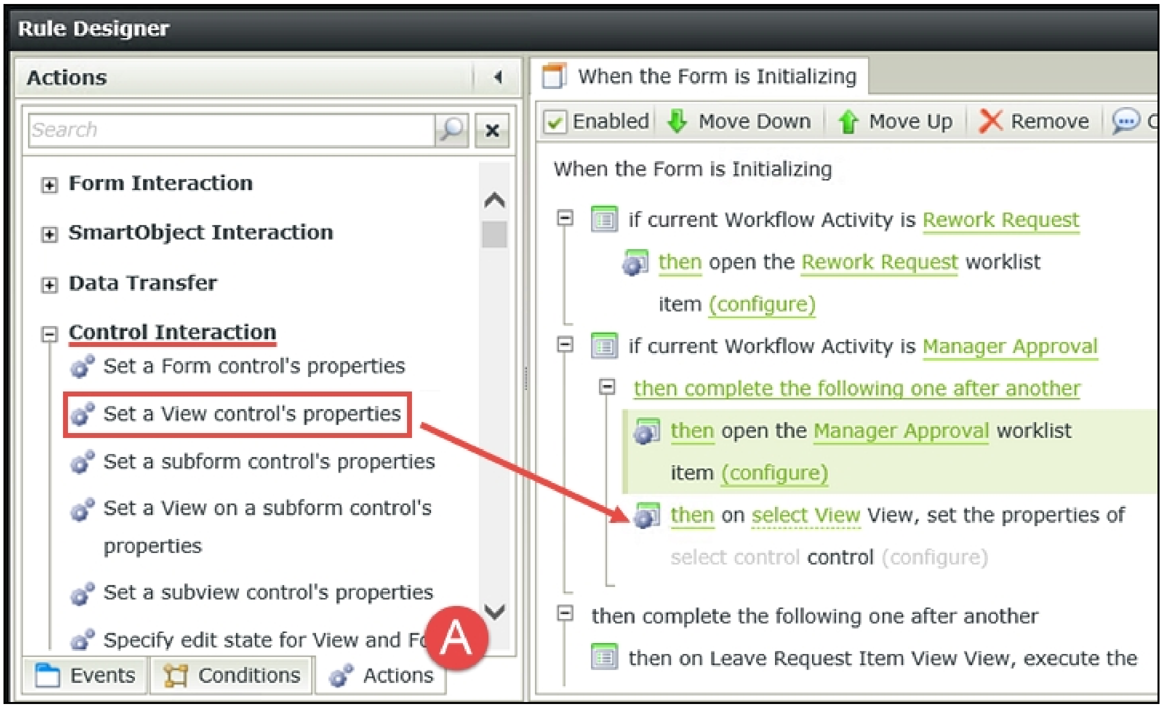
if current Workflow Activity is Manager Approval

then open the Manager Approval worklist item (configure)

then complete the following one after another

then on Leave Request Item View View, execute the

d. Confirm the **Actions** tab is highlighted. (**A** below) Click on the current action (**then open the Manager Approval worklist item**) to highlight it. In the **Actions** pane, scroll down until you see the **Control Interaction** heading. Click on the **Set a View control's properties** action to add it to the Workflow Activity condition, just below the open worklist item action as shown below.


Rule Designer
Actions
Search
Form Interaction
SmartObject Interaction
Data Transfer
Control Interaction
Set a Form control's properties
Set a View control's properties
Set a subform control's properties
Set a View on a subform control's properties
Set a subview control's properties
Specify edit state for View and Fo
A
Events
Conditions
Actions
When the Form is Initializing
Enabled
Move Down
Move Up
Remove

When the Form is Initializing

if current Workflow Activity is Rework Request

then open the Rework Request worklist item (configure)

if current Workflow Activity is Manager Approval

then complete the following one after another

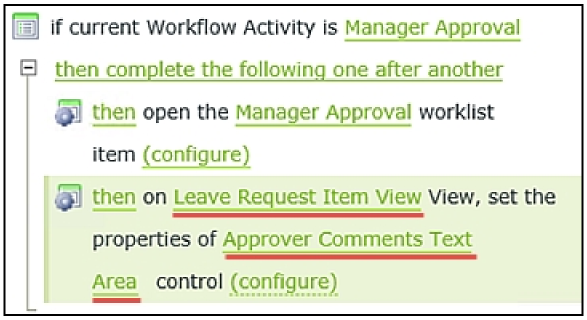
then open the Manager Approval worklist item (configure)

then on select View View, set the properties of select control control (configure)

then complete the following one after another

then on Leave Request Item View View, execute the

e. Click the **select View** link and select **Leave Request Item View**. Click the **select control** link and select **Approver Comments Text Area**.



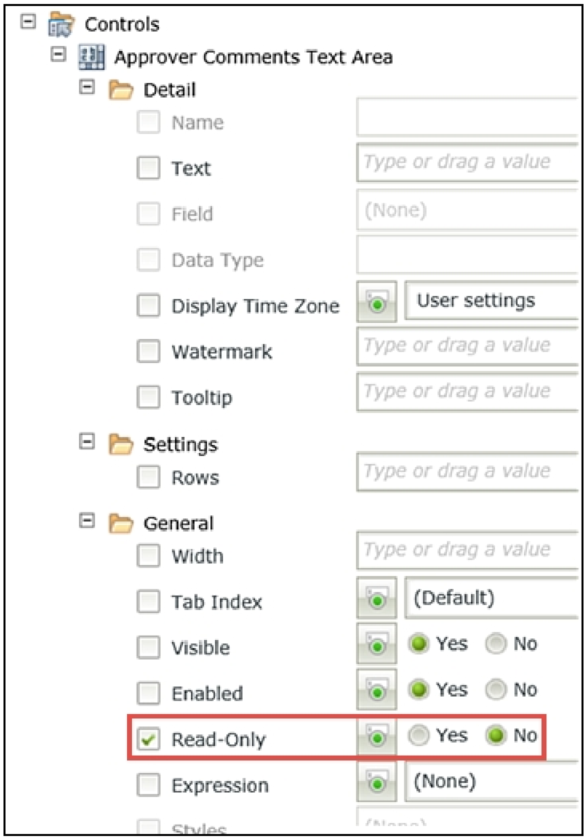
if current Workflow Activity is Manager Approval

then complete the following one after another

then open the Manager Approval worklist item (configure)

then on Leave Request Item View View, set the properties of Approver Comments Text Area control (configure)

f. Click the **(configure)** link. On the **Mapping Destinations** screen, scroll down until you can see the **General** heading. Change the **Read-Only** setting from Yes to **No**. By turning off the read-only setting, the manager will be able to add comments when they open the form. Click **OK**. Click **OK** to return to the Rules home screen.



- Controls
  - Approver Comments Text Area
    - Detail
      - □ Name
      - □ Text — Type or drag a value
      - □ Field — (None)
      - □ Data Type
      - □ Display Time Zone — User settings
      - □ Watermark — Type or drag a value
      - □ Tooltip — Type or drag a value
    - Settings
      - □ Rows — Type or drag a value
    - General
      - □ Width — Type or drag a value
      - □ Tab Index — (Default)
      - □ Visible — ● Yes ○ No
      - □ Enabled — ● Yes ○ No
      - ☑ Read-Only — ○ Yes ● No
      - □ Expression — (None)
      - □ Styles — (None)

STEP 9 REVIEW

In Step 9, you edited the Workflow Task state, Form Initializing rule and enabled the Approver Comments field for the Manager Approval task. This will allow the approving manager to add comments to the request if they choose. This step demonstrated the power of states and how you can edit individual states to customize the form and view configurations for a specific task or event.

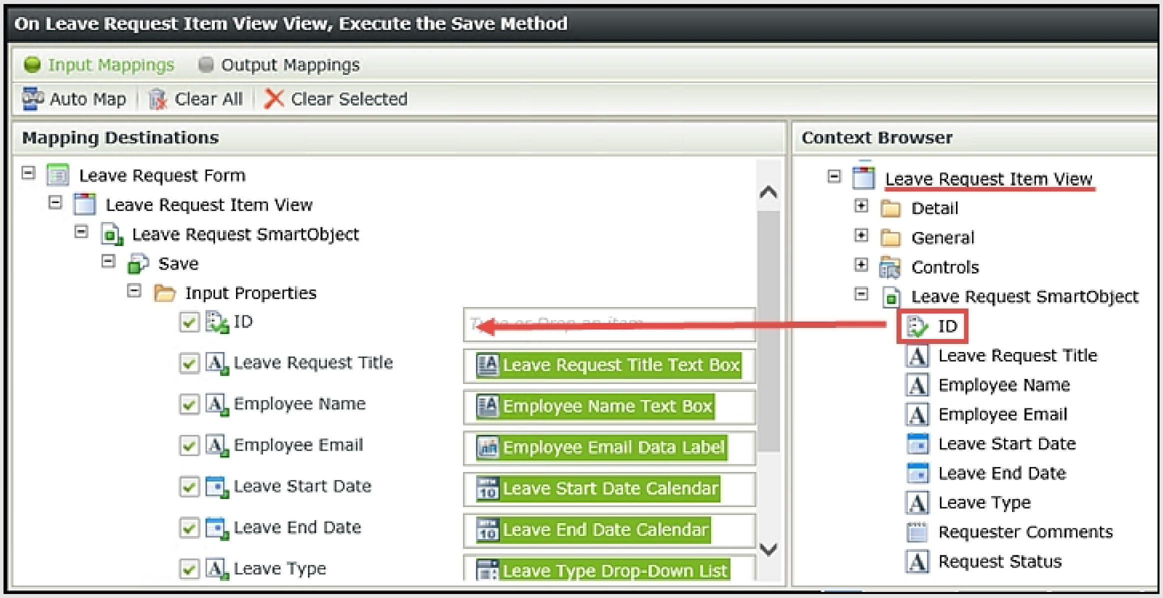
Step 10: Add a save method to update the SmartObject with changes made by the manager

Now you will add a new rule to this state. This rule will allow you to save any changes the manager might have made to the form, such as adding approver comments. (*Save* is the equivalent of update. When adding a new record, the Create method is used.)

Step 10 Tasks

1. Add a new rule to the Workflow Task state. Add the event **After the Workflow action was submitted**.
2. Add a **Execute a View method** action to call the **Save** method of the Leave Request item view.
3. You may **Auto-Map** the Input Properties, but you MUST delete the auto-mapped ID value and replace it with the **Leave Request Item View SmartObject ID** property so that K2 knows you are

referencing the current record.

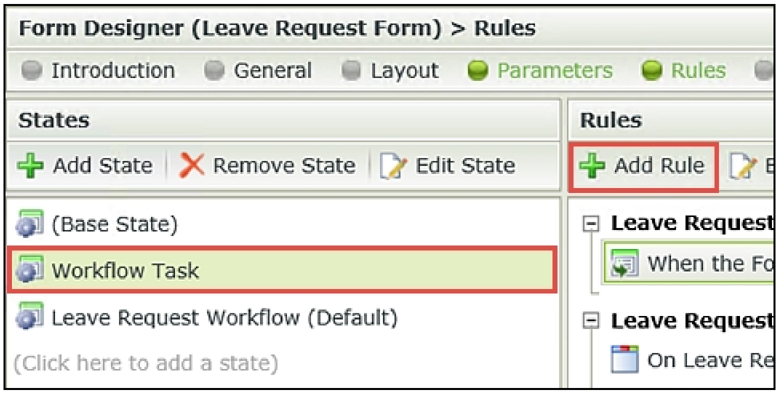


Leave Request Form
Leave Request Item View
Leave Request SmartObject
Save
Input Properties

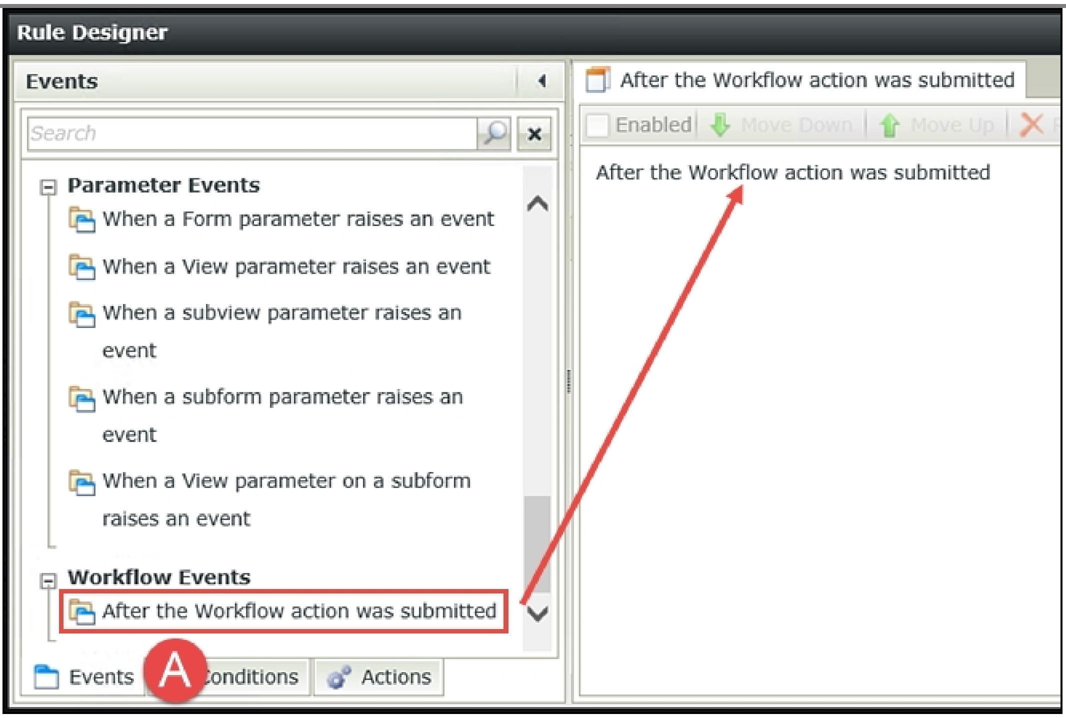
☑ ID	Type or Drop an item
☑ Leave Request Title	Leave Request Title Text Box
☑ Employee Name	Employee Name Text Box
☑ Employee Email	Employee Email Data Label
☑ Leave Start Date	Leave Start Date Calendar
☑ Leave End Date	Leave End Date Calendar
☑ Leave Type	Leave Type Drop-Down List

Step 10 Walkthrough

a. On the Rules home screen, confirm the **Workflow Task** state is highlighted, then click **Add Rule**.



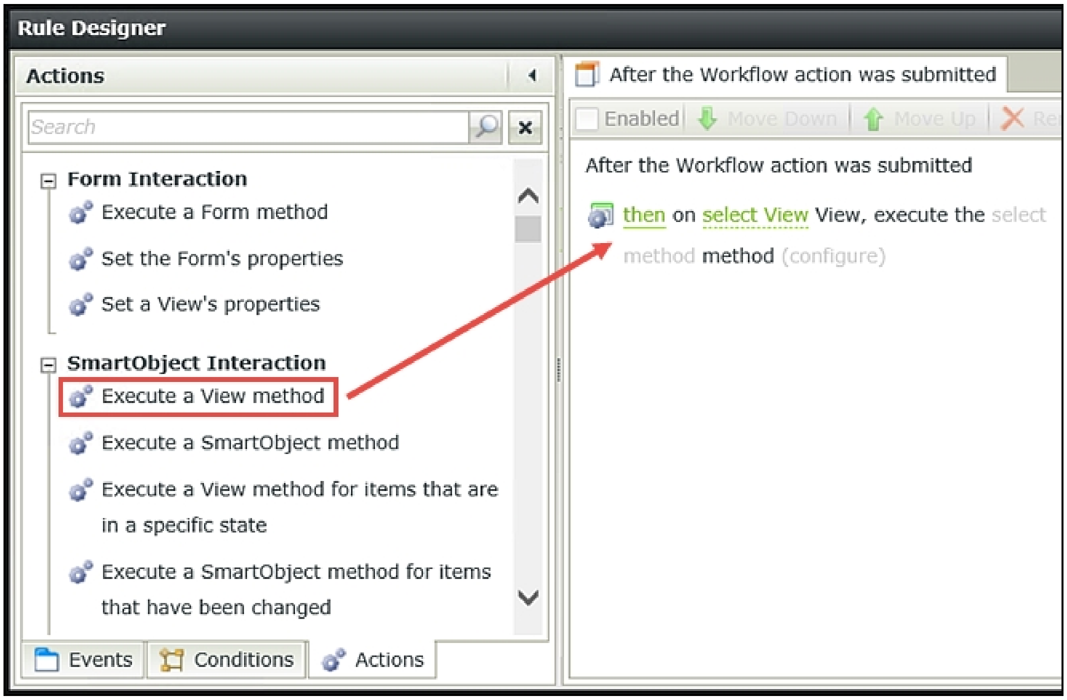
b. Click the **Events** tab. (**A** below) Scroll down until you see the Workflow Events heading. Locate and click the **After the Workflow action was submitted** event to add it to the Rule Definition pane.


Rule Designer
Events
Search
Parameter Events
When a Form parameter raises an event
When a View parameter raises an event
When a subview parameter raises an event
When a subform parameter raises an event
When a View parameter on a subform raises an event
Workflow Events
After the Workflow action was submitted
Events
A
Conditions
Actions
After the Workflow action was submitted
Enabled
Move Down
Move Up

After the Workflow action was submitted

c. Now click the **Actions** tab and click to add the **Execute a View method** action to the Rule Definition pane. Click the **select View** link and select **Leave Request Item View**. Click the **select method** link and select **Save**.

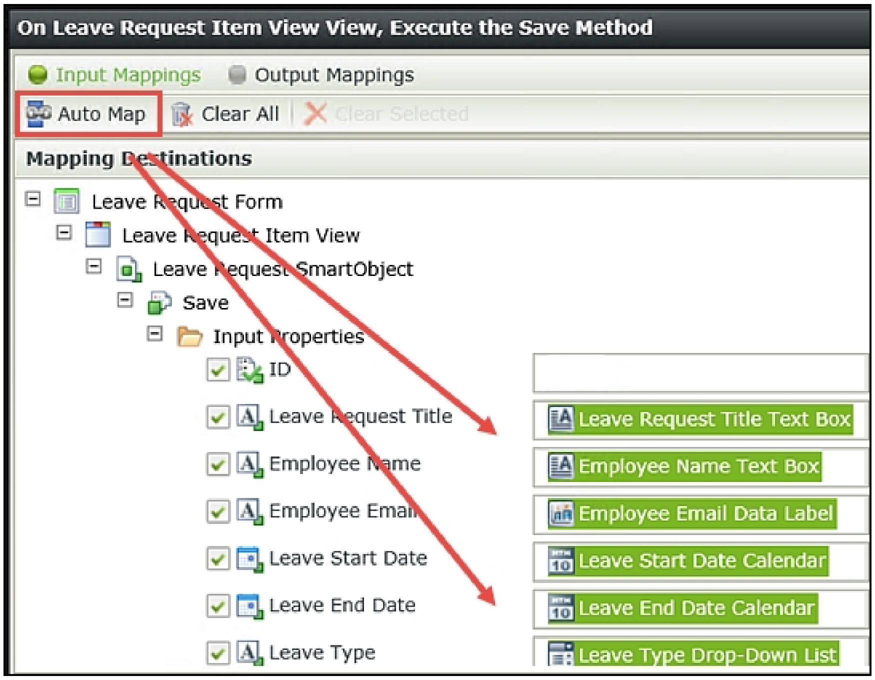
Note
The Save action saves the form field entries (Employee Name, Leave Start Date, etc.) to the Leave Request SmartObject. You will first specify the item view as the 'container' of the field entries you want to save. Then you will auto-map the view fields to the SmartObject. Auto-map is simply a time saver, as K2 will map all of the view fields to their corresponding SmartObject properties. K2 knows which fields correspond to which properties because the view was generated from the SmartObject in the first place.


Rule Designer
Actions
Search
Form Interaction
Execute a Form method
Set the Form's properties
Set a View's properties
SmartObject Interaction
Execute a View method
Execute a SmartObject method
Execute a View method for items that are in a specific state
Execute a SmartObject method for items that have been changed
Events
Conditions
Actions
After the Workflow action was submitted
Enabled
Move Down
Move Up
Re

After the Workflow action was submitted

then on select View View, execute the select method method (configure)

d. Click the **(configure)** link. Click **Auto-Map**. (In some versions of K2 blackpearl, the ID property may be auto-mapped as well. Continue to the next step on editing the ID property.)



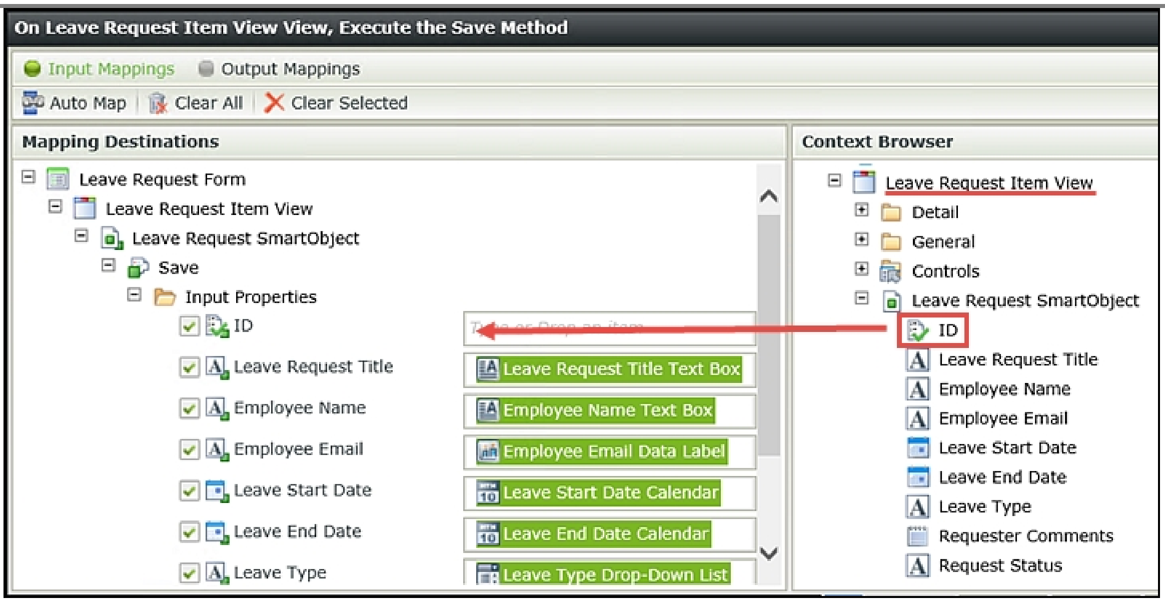
If the ID property has a value, you need to replace it with the correct SmartObject ID. If the ID property was left empty, you need to map the SmartObject ID to it.

> Note
> When you auto-map the input properties, K2 creates a new record in the SmartObject and saves the form field entries to their corresponding SmartObject properties. While we want to save the form field entries, we don't necessarily want to create a new SmartObject record. Why is this? When you started the workflow, K2 created a placeholder of sorts in the SmartObject. (This is where the item references come from as well.) We want to save our record to the SmartObject that was started with the workflow, not create a new one. That is why we want to reference the SmartObject ID that already exists.

e. If there is an auto-mapped ID, click to the right of the **ID** value and backspace until you have removed it. With the field empty, expand the **Leave Request Item View** in the **Context Browser**, then expand the **Leave Request SmartObject.** Drag the SmartObject **ID** into the **Input Properties** > **ID** field.

> Caution
> **IMPORTANT**: Be sure to complete this step. You want to use the SmartObject ID so that K2 knows you are updating the current record. If you use the auto-mapped ID, K2 will not recognize the value as the current record.

Click **Finish**. Click **OK** to close the Rule Designer.



On Leave Request Item View View, Execute the Save Method

☑ ID: Type or Drop an item
☑ Leave Request Title: Leave Request Title Text Box
☑ Employee Name: Employee Name Text Box
☑ Employee Email: Employee Email Data Label
☑ Leave Start Date: Leave Start Date Calendar
☑ Leave End Date: Leave End Date Calendar
☑ Leave Type: Leave Type Drop-Down List

STEP 10 REVIEW

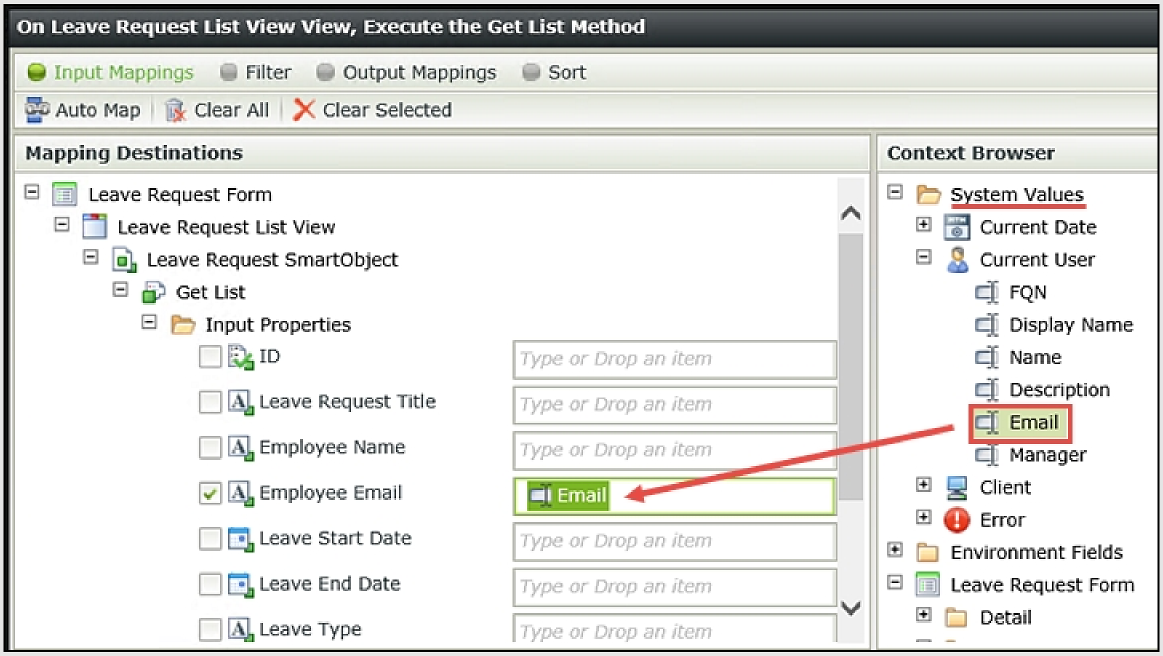
In this step, you added a rule that will fire off the save method (which is the equivalent of update) for the Leave Request SmartObject with any changes the approving manager might have made.

Step 11: Add a clear method to clear the item view and refresh the list view upon submit

The last few edits you are going to make to the rules will be to clear the form entries after the form originator has clicked the Create button, then add the new entry to the list view. (The list view contains all of previous leave requests.) Recall in the basic tutorial, when the form originator submitted the form there wasn't any visual indication of the form being submitted. By clearing the form entries and updating the list view, the user will have visual confirmation that their request was submitted.

Step 11 Tasks

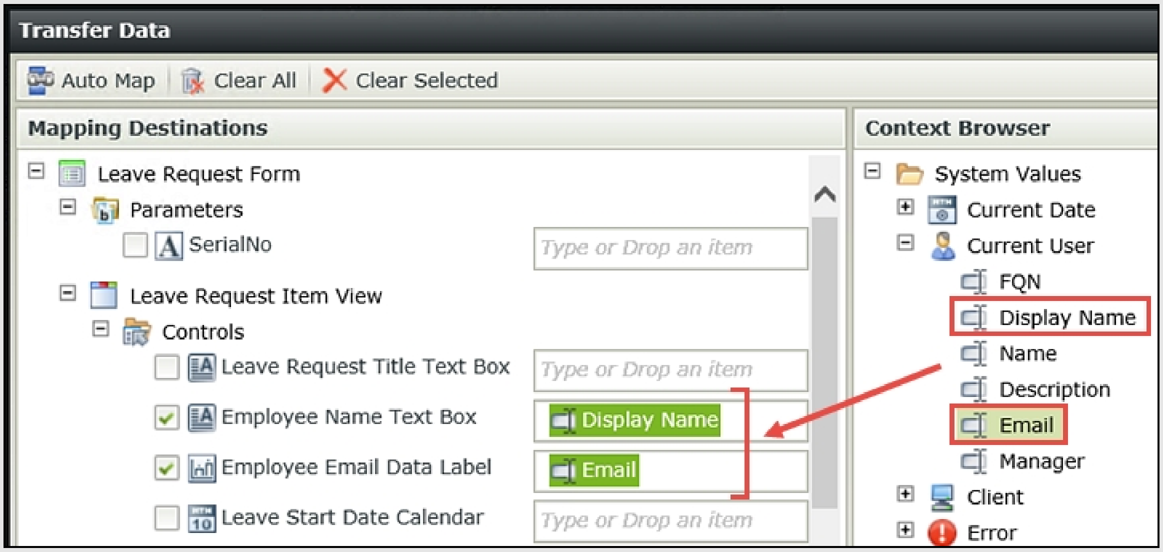
1. Edit the rule for the **Leave Request Workflow (Default)** state, when the **Create Button is clicked**. Add a View method action to **Clear** the Leave Request item view.
2. Add another View method action to populate (**Get List**) the Leave Request list view, using the System Value Email as the Input Property.



On Leave Request List View View, Execute the Get List Method

☐ ID: Type or Drop an item
☐ Leave Request Title: Type or Drop an item
☐ Employee Name: Type or Drop an item
☑ Employee Email: Email
☐ Leave Start Date: Type or Drop an item
☐ Leave End Date: Type or Drop an item
☐ Leave Type: Type or Drop an item

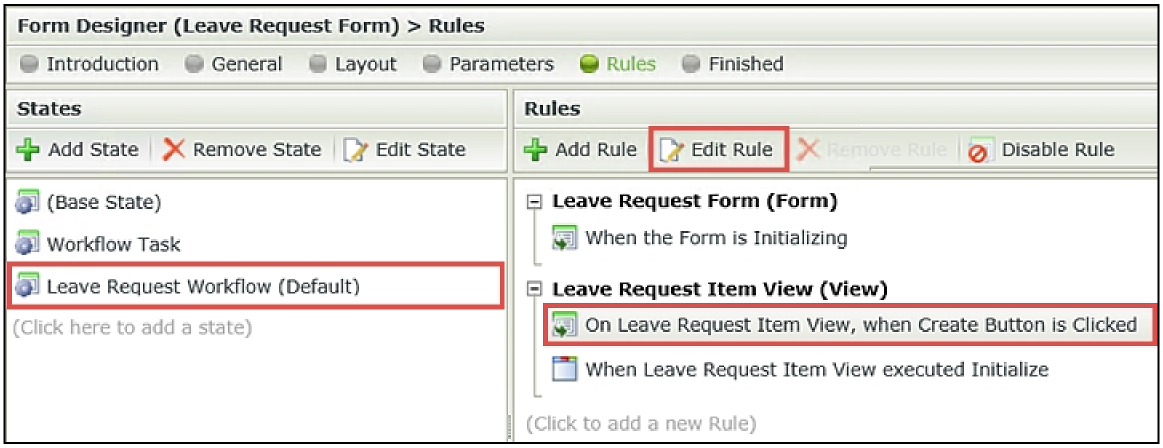
3. Add a **Transfer data** action and map the Current User Display Name and Email to their respective form controls.



4. Check In the **Leave Request Form**.

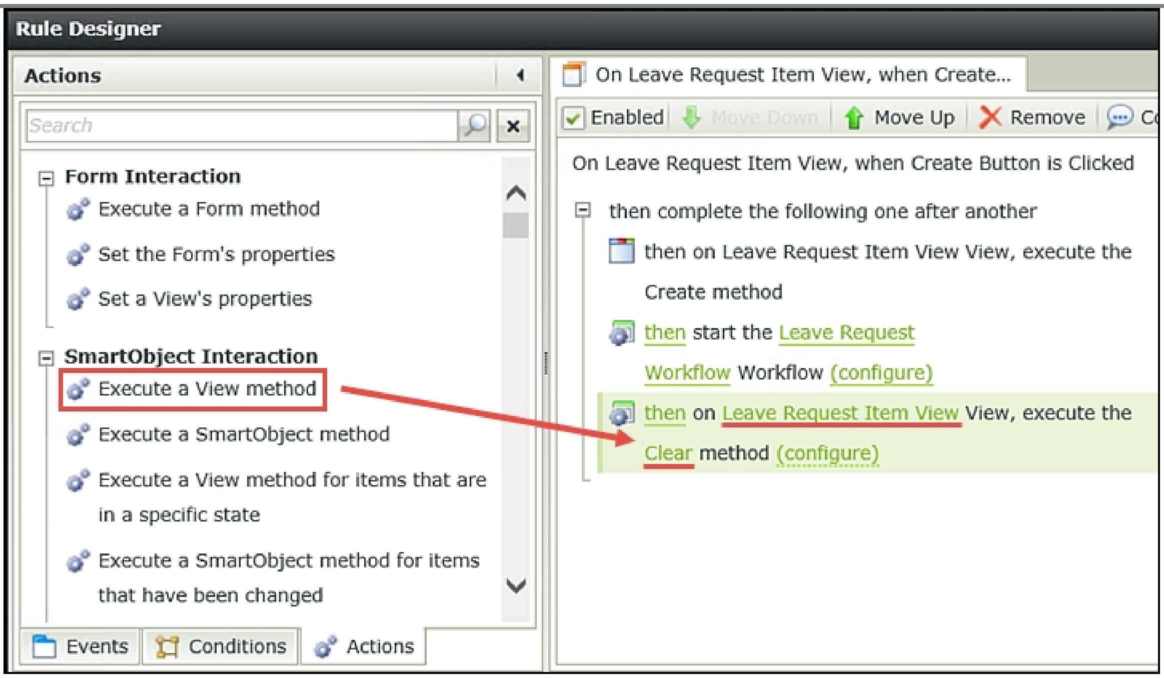
Step 11 Walkthrough

a. First you will configure an action to clear the form entries upon submit. On the **Rules** home screen, highlight the **Leave Request Workflow (Default)** state, then the **Create Button is Clicked** rule. (The Leave Request Workflow (Default) state is the configuration for the form originator.) Click **Edit Rule**.


Form Designer (Leave Request Form) > Rules
Introduction
General
Layout
Parameters
Rules
Finished
States
Add State
Remove State
Edit State
(Base State)
Workflow Task
Leave Request Workflow (Default)
(Click here to add a state)
Rules
Add Rule
Edit Rule
Remove Rule
Disable Rule
Leave Request Form (Form)
When the Form is Initializing
Leave Request Item View (View)
On Leave Request Item View, when Create Button is Clicked
When Leave Request Item View executed Initialize
(Click to add a new Rule)

b. Confirm the **Actions** tab is highlighted, then click the **Execute a View method** action to add it to the Rule Definition pane. Click the **select View** link and select **Leave Request Item View**. Click the **select method** link and select **Clear**.

Sorting through events and actions: first you see an event for when the Create Button is Clicked. Then the Create method for the SmartObject fires and creates the record in the Leave Request SmartBox SmartObject. Then the workflow starts. Finally the Leave Request Item View is cleared.



Rule Designer
Actions
Search
Form Interaction
Execute a Form method
Set the Form's properties
Set a View's properties
SmartObject Interaction
Execute a View method
Execute a SmartObject method
Execute a View method for items that are in a specific state
Execute a SmartObject method for items that have been changed
Events
Conditions
Actions
On Leave Request Item View, when Create...
Enabled
Move Down
Move Up
Remove
Co

On Leave Request Item View, when Create Button is Clicked

then complete the following one after another

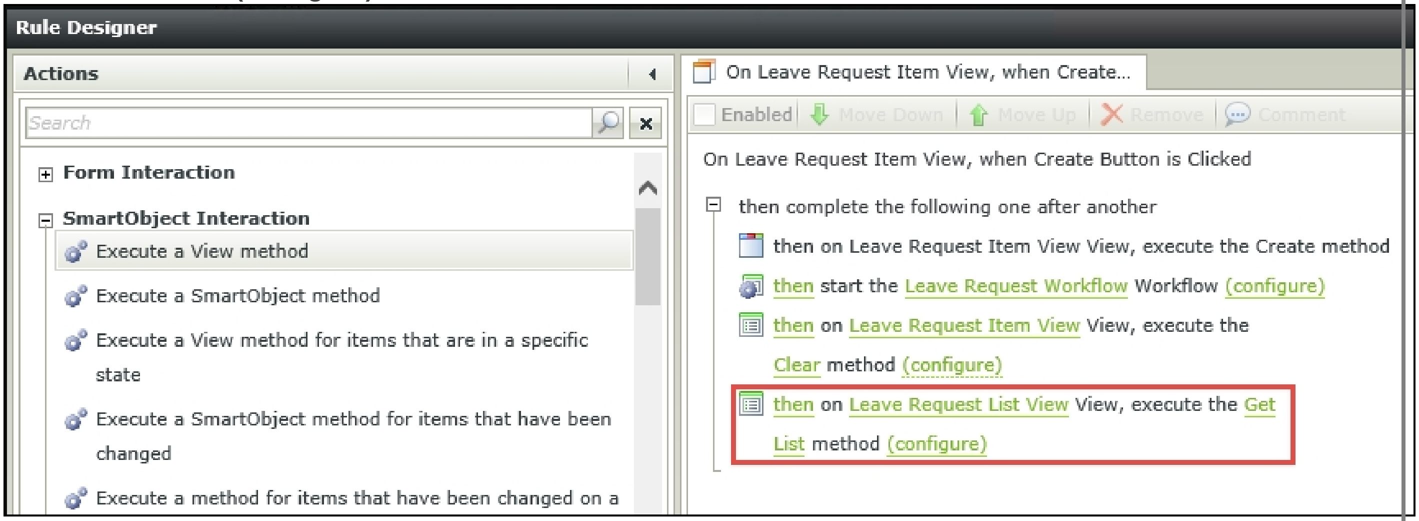
then on Leave Request Item View View, execute the Create method

then start the Leave Request Workflow Workflow (configure)

then on Leave Request Item View View, execute the Clear method (configure)

Now you will update the Leave Request list view so that the new entry is displayed.

c. Click **Execute a View method** once again to add another instance to the Rule Definition pane. Click the **select View** link and select **Leave Request List View**. Click the **select method** link and select **Get List**. Click the **(configure)** link.



Rule Designer
Actions
Search
Form Interaction
SmartObject Interaction
Execute a View method
Execute a SmartObject method
Execute a View method for items that are in a specific state
Execute a SmartObject method for items that have been changed
Execute a method for items that have been changed on a
On Leave Request Item View, when Create...
Enabled
Move Down
Move Up
Remove
Comment

On Leave Request Item View, when Create Button is Clicked

then complete the following one after another

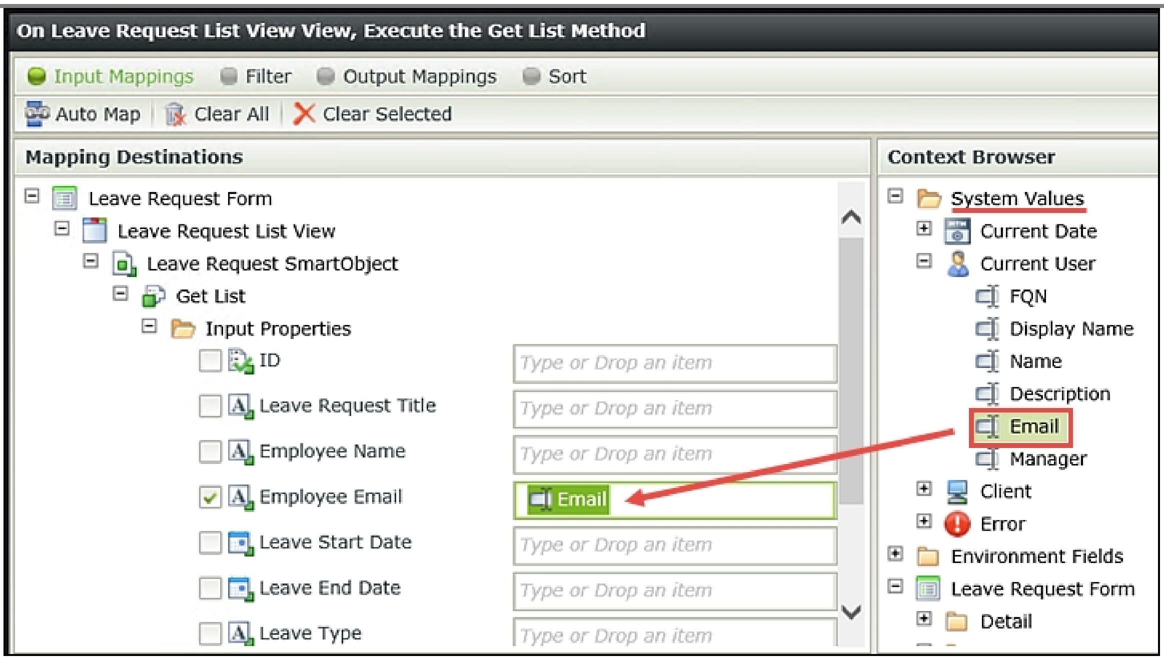
then on Leave Request Item View View, execute the Create method

then start the Leave Request Workflow Workflow (configure)

then on Leave Request Item View View, execute the Clear method (configure)

then on Leave Request List View View, execute the Get List method (configure)

d. In this step, you will map the current user's email as the input property so that K2 knows to search the SmartBox records for those that have the same email value as the current user. In the context browser, expand **System Values**, then **Current User** and drag the **Email** property into the **Employee Email** Input Property. Click **Finish**.



On Leave Request List View View, Execute the Get List Method

Mapping Destinations:

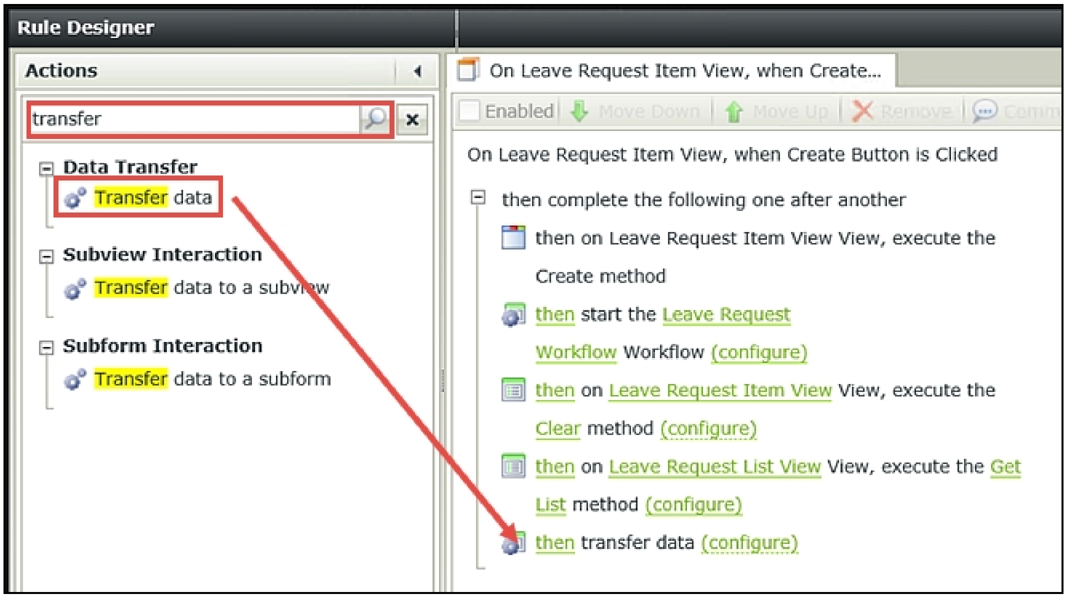
- Leave Request Form
  - Leave Request List View
    - Leave Request SmartObject
      - Get List
        - Input Properties
          - □ ID: Type or Drop an item
          - □ Leave Request Title: Type or Drop an item
          - □ Employee Name: Type or Drop an item
          - ☑ Employee Email: Email
          - □ Leave Start Date: Type or Drop an item
          - □ Leave End Date: Type or Drop an item
          - □ Leave Type: Type or Drop an item

Context Browser:

- System Values
  - Current Date
  - Current User
    - FQN
    - Display Name
    - Name
    - Description
    - Email
    - Manager
  - Client
  - Error
- Environment Fields
- Leave Request Form
  - Detail

You have one additional step you need before you close out of your form. In the previous step, you cleared the form entries when the user clicked the Create button. Now you need to add back in the Employee Name and Employee Email values so the refreshed form appears exactly the same as a brand new form. Recall that you configured the Employee Name and Employee Email values to be the current user values, which you again, obtain from the context browser, system values.

e. Still on the Actions tab, search for
transfer
then click **Transfer data** to add it to the Rule Definition pane. Click the **(configure)** link.



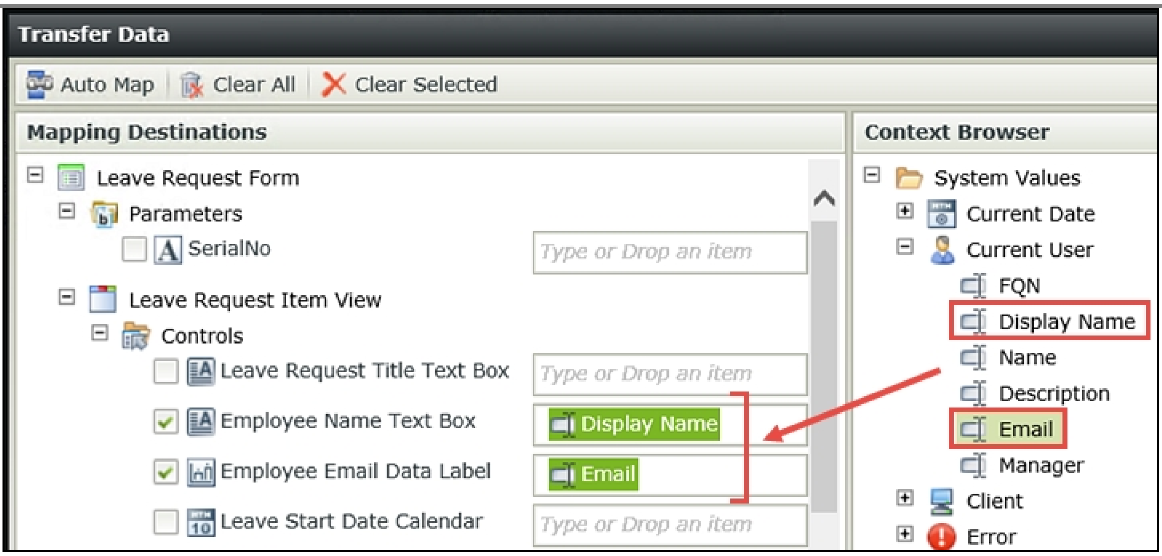
Search: transfer

- Data Transfer
  - Transfer data
- Subview Interaction
  - Transfer data to a subview
- Subform Interaction
  - Transfer data to a subform

On Leave Request Item View, when Create Button is Clicked

- then complete the following one after another
  - then on Leave Request Item View View, execute the Create method
  - then start the Leave Request Workflow Workflow (configure)
  - then on Leave Request Item View View, execute the Clear method (configure)
  - then on Leave Request List View View, execute the Get List method (configure)
  - then transfer data (configure)

f. In the context browser, expand the **System Values** node, then **Current User**. Drag the **Display Name** into the **Employee Name Text Box** field and the **Email** into the **Employee Email Text Box** field. Click **OK** twice, then click **Finish** to save and exit the form.



g. Right-click the **Leave Request Form** and select **Check In**. If you get a message about associated Views, click **OK** to continue.

STEP 11 REVIEW

In this final step of configuring actions, you added a clear method so that the user will have a visual confirmation that their form was submitted. There are a number of mechanisms for letting your users know they have successfully submitted a form, including adding a confirmation pop-up message or redirecting them to a web page, for example. As you build more applications within your own environment, you will gain knowledge of what works best for your target users.

Now that your application changes are done, you will move on to Part 4: Test, to try out your application!

Part 4: Testing the Application

In Part 4, you will test the updated Leave Request application.

Note
You must complete Parts 1, 2 and 3 to continue with Part 4.

Step 12: Testing the application

For this test, you will submit two Leave Request forms. You will access the approving manager's Outlook and action one of the requests using SmartActions to send the request back for rework. As the form originator, you will resubmit the request, then reject it as the approving manager. The second form that you submit, you will not action.

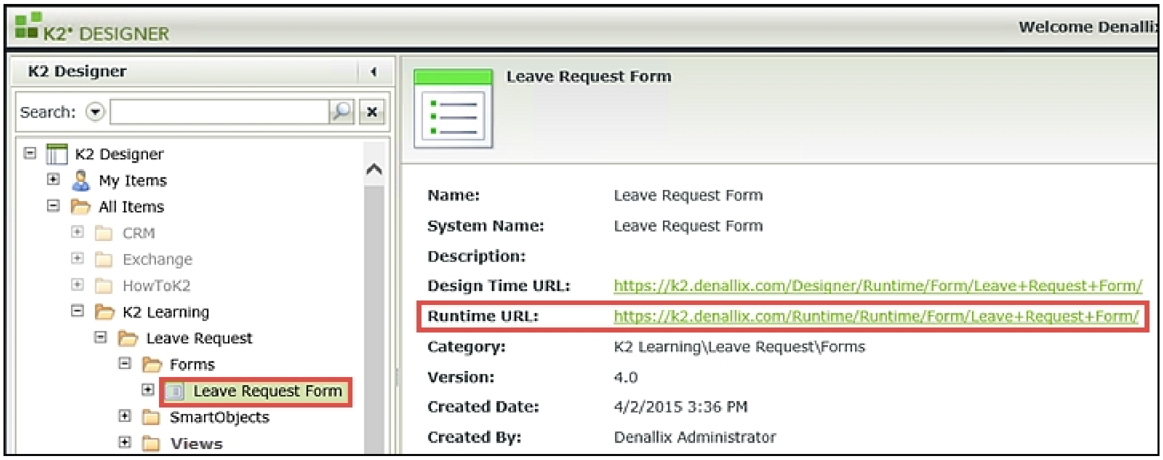
Step 12 Tasks

1. Using the Runtime URL, submit two Leave Request forms. Make the **Leave Start Date** for one of the requests today's date. This will fire off the escalation since you have it configured to start within two days of the Leave Start Date.
2. Access the approving manager's Outlook and using **SmartActions**, reply to the request with **Rework**.
3. Access the form originator's email and open the form. Edit the form entry in any manner you like, then resubmit it.
4. As the approving manager, add some comments, then reject the resubmitted request. Confirm the email notification that is sent back to the form originator contains the manager's comments within the rejected email.

Step 12 Walkthrough

a. Using the **Runtime URL**, open and submit *two* Leave Request forms. Make the **Leave Start Date** for one of the forms today's date. This will fire off the escalation that you configured (remember it executes if within two days of the start date). Confirm you cannot edit the Approver Comments field (it should be read-only for the form originator). Enter any leave type or comments you like for each form.

If you have time and would like to submit more leave requests so that you have additional content when you work through the Reports tutorial, feel free to do so! (The Reports tutorial uses the workflow process instances from the Leave Request tutorials for content.)

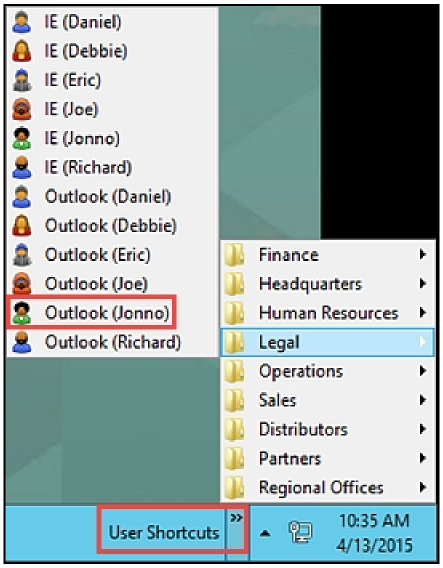


Name: Leave Request Form
System Name: Leave Request Form
Description:
Design Time URL: https://k2.denallix.com/Designer/Runtime/Form/Leave+Request+Form/
Runtime URL: https://k2.denallix.com/Runtime/Runtime/Form/Leave+Request+Form/
Category: K2 Learning\Leave Request\Forms
Version: 4.0
Created Date: 4/2/2015 3:36 PM
Created By: Denallix Administrator

After you have submitted a form, confirm that the field entries are cleared (except the Employe Name and Email) and your new entry now shows up in the Previous Leave Requests List View. After you have submitted at least two forms, close your browser.

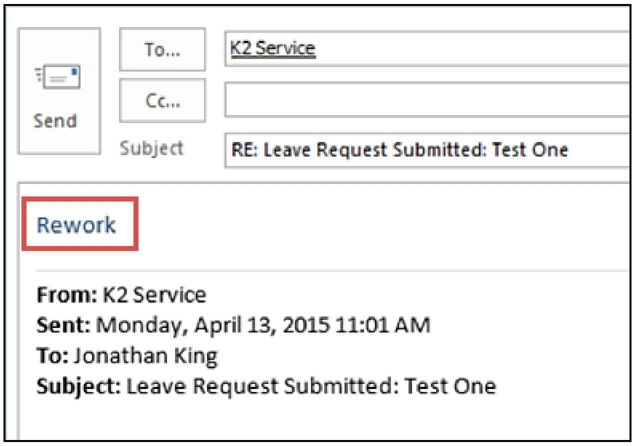
At this point, you need to open the approving manager's Outlook. If you are building this application on a K2-provided VM, you are most likely logged in as Denallix Administrator. Administrator's manager is Jonno, so you'll open Jonno's Outlook. If you are building this application within your own environment, open the Outlook account for the user assigned the manager approval task. The screenshots and usernames used in this tutorial assume you are using a K2-provided virtual machine, so your screens and users may be different if working through the steps in another environment.

b. Begin by closing any open browsers and any open instances of Outlook. Click on the **User Shortcuts** link found in the lower right corner of the screen. Expand the Legal folder, then click on **Jonno's Outlook**. Allow a minute or two for Outlook to open. If you see a message about syncing CRM, just let it complete. If you see a message about licensing, click **Close** to continue.



c. Confirm there are two task notification emails for Jonno. Open one of the emails and reply with *Rework* as the email message body. Send the email.

As a reminder, you are going to action this request using **SmartActions**. SmartActions allow you to simply reply to a task notification email with one of the actions as the message body. K2 will read the action and move the workflow along accordingly.



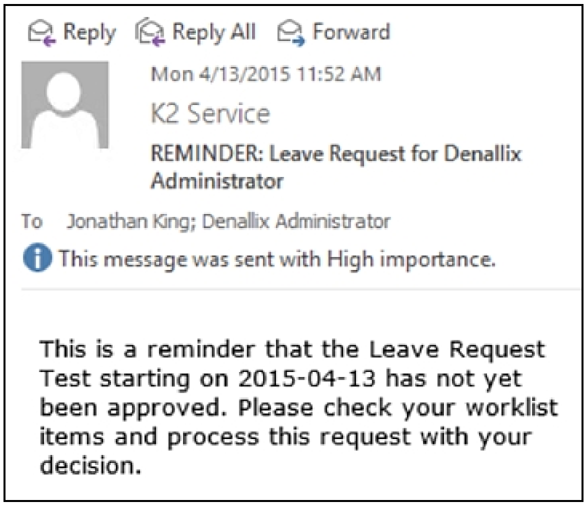
To... K2 Service

Subject RE: Leave Request Submitted: Test One

Rework

From: K2 Service
Sent: Monday, April 13, 2015 11:01 AM
To: Jonathan King
Subject: Leave Request Submitted: Test One

d. Confirm there is an escalation email for the task with today's date as the Leave Start Date.



Mon 4/13/2015 11:52 AM
K2 Service
REMINDER: Leave Request for Denallix Administrator
To Jonathan King; Denallix Administrator
This message was sent with High importance.

This is a reminder that the Leave Request Test starting on 2015-04-13 has not yet been approved. Please check your worklist items and process this request with your decision.

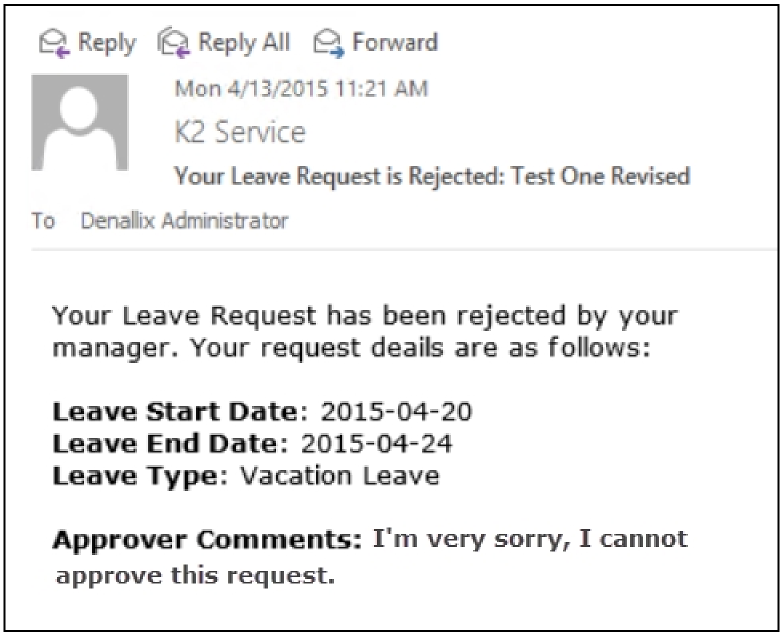
e. Now you will switch back and view the email that is sent back to the user for reworking their request. Close Jonno's Outlook and open **Administrator's Outlook**. (Start, then search for Outlook). Allow a minute or so for Outlook to open. Confirm there is a task notification email for Administrator. This time, click on the link **Click to open worklist item** so that you can open and edit your original request. (Remember if you are working in your own environment, you most likely will need to open your own Outlook.)

f. Make a change of some kind to your form entries. Select **Resubmitted** from the Action options and click **Submit**. You should see a confirmation dialog, click **OK**. Close out of the browser.

g. **Close Administrator's Outlook** and open Jonno's Outlook. Once again, you should see a new task notification email. This time, click the link **Click to open worklist item** so that you can access the form.

h. Confirm that you can now enter content into the **Approver Comments** field. Recall that you edited the Workflow Task state and turned off the read-only option for the comments. Enter a comment. Select **Rejected** from the action options and submit the form. Click **OK** for the confirmation dialog, then close out of the browser.

i. Close Jonno's Outlook and open **Administrator's Outlook**. Confirm the rejected email arrived and there is content for the Approver Comments.



Mon 4/13/2015 11:21 AM

K2 Service

Your Leave Request is Rejected: Test One Revised

To Denallix Administrator

Your Leave Request has been rejected by your manager. Your request deails are as follows:

Leave Start Date: 2015-04-20
Leave End Date: 2015-04-24
Leave Type: Vacation Leave

Approver Comments: I'm very sorry, I cannot approve this request.

STEP 12 REVIEW

If your testing was successful, then congratulations! You have extended the Leave Request application and have seen the Data, Forms and Workflow co-operate in a complete cycle from start to finish. You have completed the extended version of the Leave Request application.

Summary

The exercises in this tutorial expanded on the functionality and interaction of K2's components: **Data**, **Forms**, **Workflows**. Key to these exercises are the following:

Data

- SmartObjects can leverage connectivity with internal and external data sources.
- SmartObjects can be associated with form controls, creating an association that can automatically populate control values.
- SmartObject methods can be called from views, forms and workflows and are commonly used to create, save (update) or retrieve (list) SmartObject properties.

Forms

- Forms are containers for views and controls
- Forms can contain multiple views
- Views are logical sections of form content
- There are two types of views: Item Views contain the content from one record. List Views contain multiple records.
- Rules are comprised of Events, Conditions and Actions. Events are *when* something occurs, Conditions are *if a specific criteria has been met* and Actions *do something* if the Event and Condition has been met.
- Rules can be applied to different workflow states, essentially creating custom forms for workflow tasks
- Forms and views must be checked in before they can be exposed to users

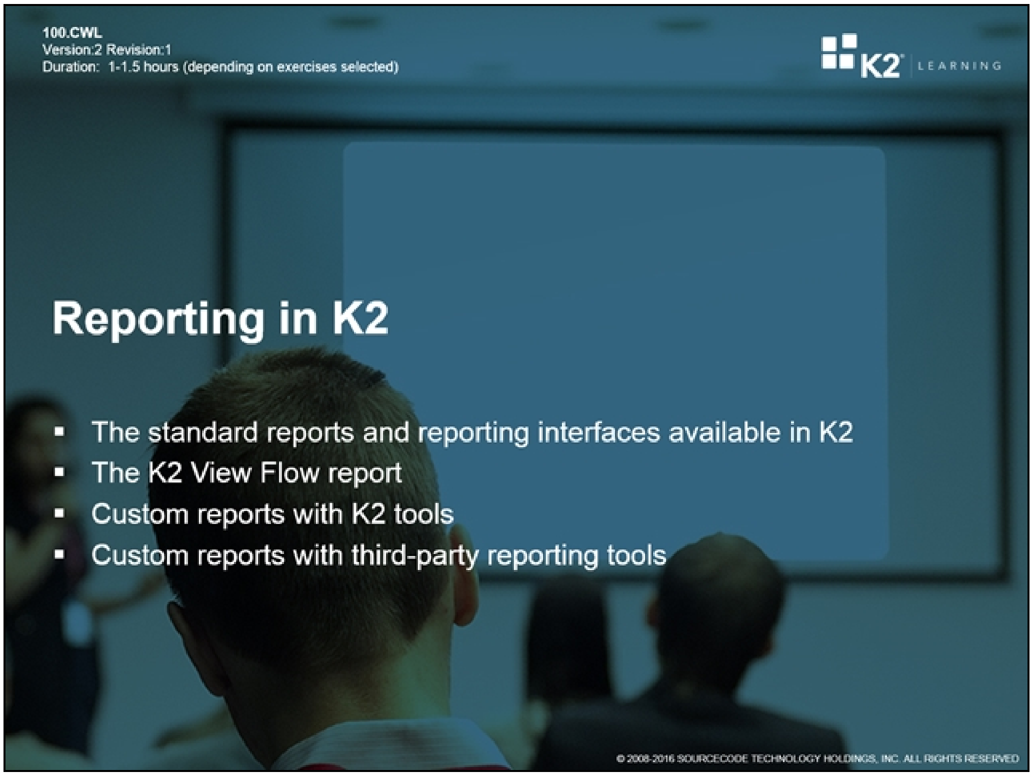
Workflows

- Workflows are comprised of steps which in turn, have associated events. The main container for a workflow is called a process.
- There are two types of events: System tasks are performed by the K2 server such as sending an email. User tasks are performed by a human, such as making a decision of some kind.
- Escalations keep your workflows flowing by sending email reminder notices, redirecting the unactioned task to

another user or expiring the task altogether.

- Workflows must be deployed before they can be exposed to users

100.CWL: Reporting in K2



100.CWL
Version:2 Revision:1
Duration: 1-1.5 hours (depending on exercises selected)

K2 LEARNING

Reporting in K2

- The standard reports and reporting interfaces available in K2
- The K2 View Flow report
- Custom reports with K2 tools
- Custom reports with third-party reporting tools

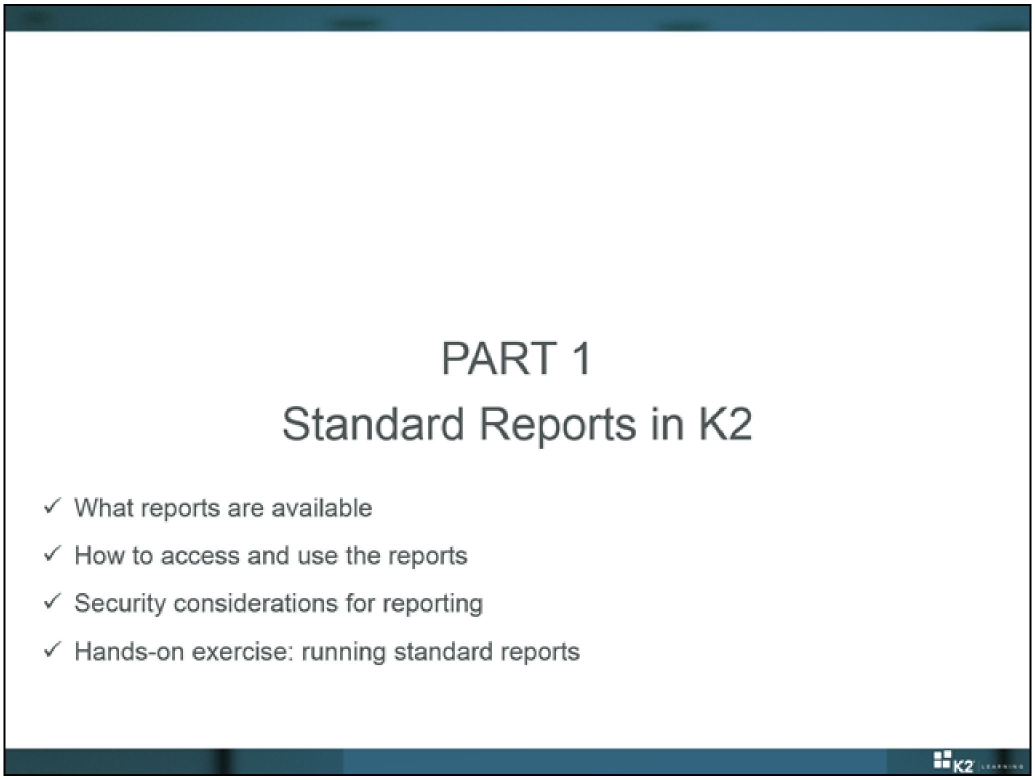
Video

The *100.CWL: Reporting in K2* training module explains how to use the available standard and custom reporting in K2 to report on workflows. The module should take around 1 to 1.5 hours to complete, depending on how many of the hands-on exercises you choose to do.

This module covers the following concepts:

- The standard reports that are available in K2 and where you find and run those reports
- Using the K2 View Flow report
- Creating custom reports in K2 Workspace and K2 smartforms
- Creating custom reports with other third-party reporting tools

Part 1: Standard Reports in K2



PART 1

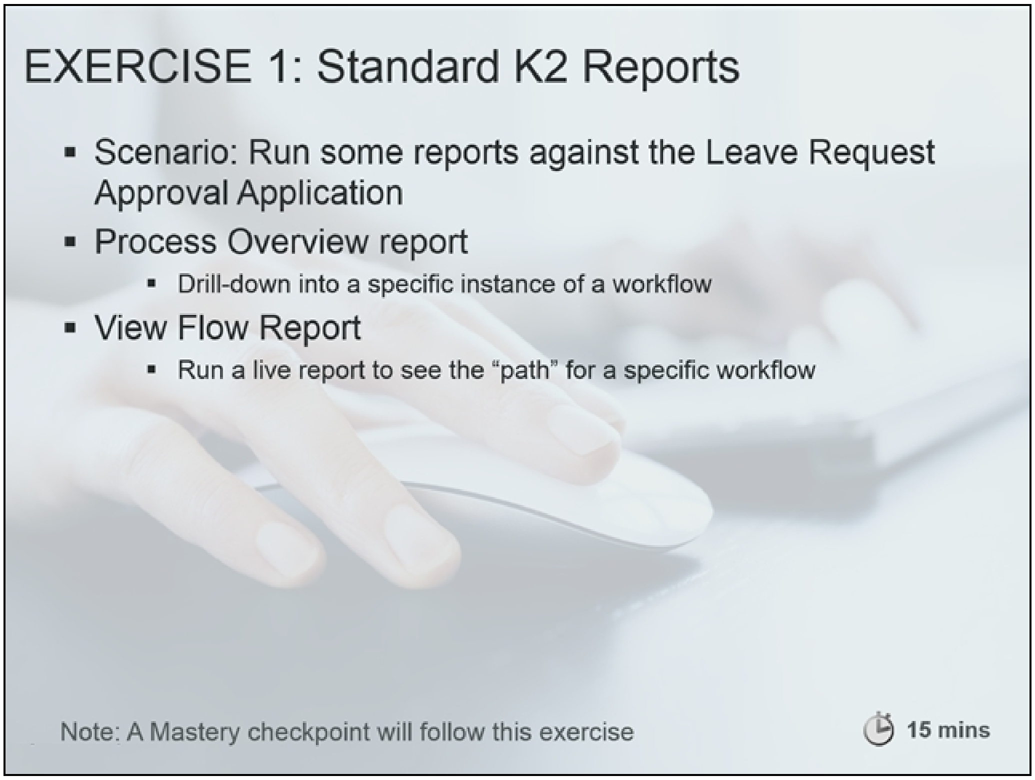
Standard Reports in K2

- ✓ What reports are available
- ✓ How to access and use the reports
- ✓ Security considerations for reporting
- ✓ Hands-on exercise: running standard reports

Video

In Part 1 we will look at the standard reports that are available in K2, how to access and use these reports and some security considerations that affect how reporting data is shown to users. At the end of Part 1 there will be a hands-on exercise followed by a mastery check.

EXERCISE 1: Standard K2 Reports



EXERCISE 1: Standard K2 Reports

- Scenario: Run some reports against the Leave Request Approval Application
- Process Overview report
    - Drill-down into a specific instance of a workflow
- View Flow Report
    - Run a live report to see the “path” for a specific workflow

Note: A Mastery checkpoint will follow this exercise

15 mins

Video

In this exercise you will run some of the standard reports against an existing solution in your K2 environment. You will learn how to use the Process Overview report to "drill-down” into a specific workflow instance and how to use the View Flow report to see the “path” of a workflow in near real-time.

> **Note**
> This exercise assumes that you have completed the Leave Request Approval application from the *100BHX: Introduction to K2 Applications with K2 Designer* learning module. If you have not, you can follow the same steps to run reports against another workflow in your environment, but the report data and screenshots will look different.

When you are ready, continue on to Reporting in K2 Workspace and then The View Flow Report to complete the two parts for this exercise.

Reporting in K2 Workspace

K2 Workspace is the primary Administration and Reporting tool for K2 blackpearl installations. From the Management Console, you can determine the current version of a deployed workflow, set workflow permissions (Process Rights), stop and start Process Instances and redirect them to another user if necessary. All workflows, whether built in K2 Designer, K2 Studio or K2 Visual Studio can be managed from K2 Workspace.

> **Note**
> The steps and screenshots in this tutorial are based on the Leave Request (Extended) application. You can, however, use these same steps and instructions to report on other workflows in your K2 environment If you are working through the tutorials in your own environment, simply use a Process or Processes that already exist. The steps will be the same in either environment.

> **Note**
> Required Permissions for these tutorials:
> You must have the necessary process permissions to complete these tutorials. If you are logged into a K2-provided VM as Denallix Administrator, you will have the necessary rights. If you are working from within your own environment, take note of the permissions described below and if necessary, request additional permissions from your K2 Administrator.

- **View**: Users with View rights on a Process can run Reports against all instances of a workflow. This is the recommended permission if you want to run statistical reports against a Process.
- **View Participate**: Users with View Participate rights can only report on those workflow instances where they are the originator (started the workflow), or actioned a user task. With this permission, you are likely to only see a subset of all the statistical reporting information for a specific process, because only those instances that meet these criteria will be included in the report. data
- **Admin**: Required to access the *Management Console* in K2 Workspace in order to assign workflow permissions. If you are working in your own environment and you are not a K2 Administrator,either request this permission from your K2 administrator or just read through the exercises and do not perform the exercise tasks. You will be able to complete the majority of the exercises in this tutorial without needing the Admin permission.

Step 1: Overview and basic navigation of K2 Workspace

In this step, we will take a brief tour of the Management Console in K2 Workspace. We will look at the Process Rights for the Leave Request Workflow.

Step 1 Tasks

1. In this step, we are taking a brief tour of the Management Console, specifically looking at the **Process Rights** for the **Leave Request Workflow** Process. If you are familiar with the Management Console and Process Rights, feel free to move on to the next step.

Step 1 Walkthrough

a. Launch K2 Workspace. (**Start** > **All Programs** > **K2 blackpearl** > **K2 blackpearl Workspace**). If you do not see this link in your start menu, contact your K2 Administrator to obtain the URL of the K2 Workspace in your environment.

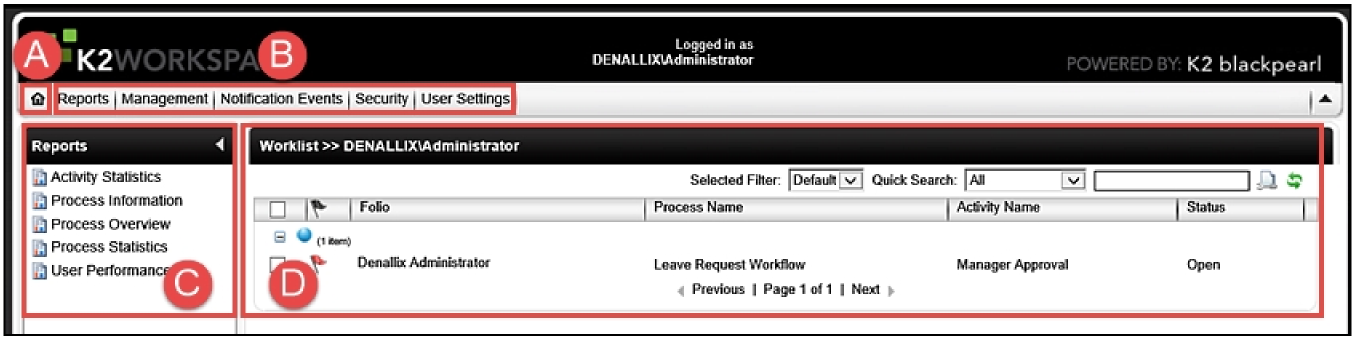
b. Notice the following sections of the landing page:

  A. Home Page Button

  B. Menu Bar

  C. Reports Console Menu

  D. Worklist for current user. (If there are any Process Instances waiting to be actioned by the current user, they are listed here. There may not be any. In the image below, DENALLIX\Administrator has one Process Instance waiting to be actioned.)

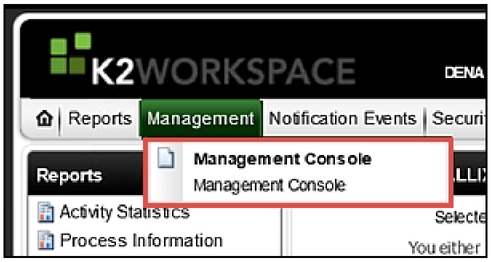


Worklist >> DENALLIX\Administrator

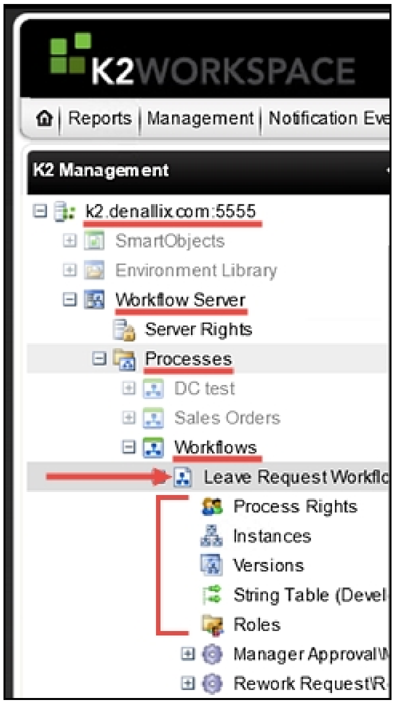
Selected Filter: Default Quick Search: All

Folio	Process Name	Activity Name	Status
(1 item)			
Denallix Administrator	Leave Request Workflow	Manager Approval	Open

c. Click on the **Management Menu** button in the Menu Bar, then click **Management Console**.



d. Expand the K2 Server tree (your server name may be different than the image below), then expand **Workflow Server** > **Processes** > **Workflows** > **Leave Request Workflow**. This is the Leave Request Workflow that was deployed as part of the Leave Request (Extended) tutorial.

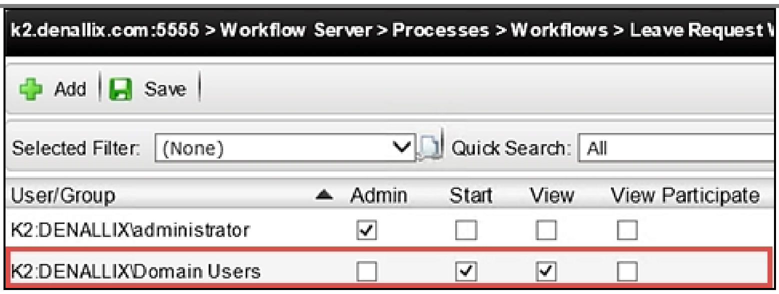


Notice the Workflow Server options for the Leave Request Workflow Process.

- **Process Rights**: Assign workflow permissions to a Process.
  - Admin: Full-control over the Process. This should not be confused with Admin Server Rights, where the user has full control in K2 Workspace.
  - Start: Users can start a workflow. Without Start (or greater) permissions, users are likely to get an error when they submit a form.
  - View: Users can view, run and create reports for the Process.
  - View Participate: Users can view standard reports for workflows they started or where they completed a task.
- **Instances**: View running instances of the current Process. This option gives you an overview of how many Process Instances are currently running, as well as those that are in an error state.
- **Versions**: By default, K2 makes the most recently deployed workflow the default. You can change the default to another version if desired. When running reports, you have the option of changing the version you want to report on.
- **String Table**: Shows the K2 blackpearl connection strings.
- **Roles**: Roles are initially created globally and then can be added to a specific workflow if desired. Roles are predefined groups of users. For example, you might create a role called 'Finance Approvers', then assign a user task to the Finance Approvers role, instead of one or more individual users.

In this exercise we will be working with the Process Rights option.

e. Click on **Process Rights**. With Start and View rights, all Domain Users can start this Process (the Leave Request Workflow) as well as access and run Standard and Custom Reports. The **View Participate** option would be in lieu of the View option and would allow them view Standard Reports for workflows they started or where they completed a task. (You would select one or the other, but in most cases, the View option is selected.) If you don't want users to have access to reports, simply remove the View option by deselecting it and clicking Save.

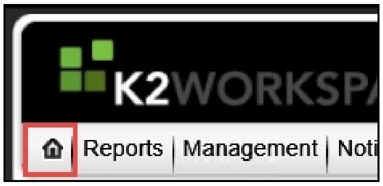


k2.denallix.com:5555 > Workflow Server > Processes > Workflows > Leave Request W

Selected Filter: (None) Quick Search: All

User/Group	Admin	Start	View	View Participate
K2:DENALLIX\administrator	☑	□	□	□
K2:DENALLIX\Domain Users	□	☑	☑	□

f. Click on the **Home** icon to return to the K2 Workspace landing page.



STEP 1 REVIEW

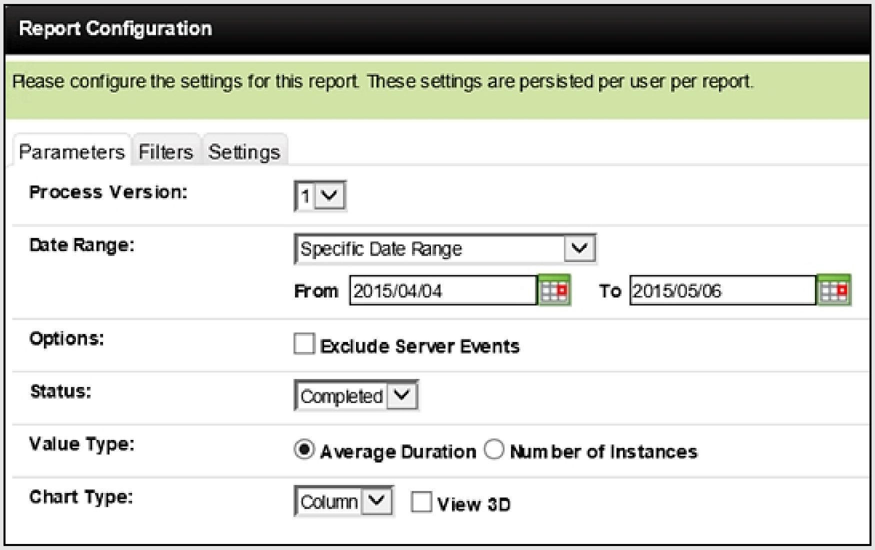
In this step, we took a brief look at the Management Console, specifically the Process Rights for the Leave Request Workflow Process. This is important to be familiar with, as you will use Process Rights to grant users the ability to view, run and create Standard and Custom Reports. If you do not want your users accessing reports, confirm their only Process Right is the Start option. This will allow them to submit a form (and start a workflow), but not access reports.

Step 2: The Activity Statistics Report

In the next several steps, we will learn more about three Standard Reports and how you might use them for analyzing and monitoring your Processes. The first report we are going to work with is the Activity Statistics Report. When you drill down into it, the Activity Statistics Report displays in a graphical format.

Step 2 Tasks

1. Launch the **Activity Statistics** Report. Using the **Leave Request Workflow**, run a report that displays the **Average Duration** for **Completed** tasks. Be sure to include Server Events.



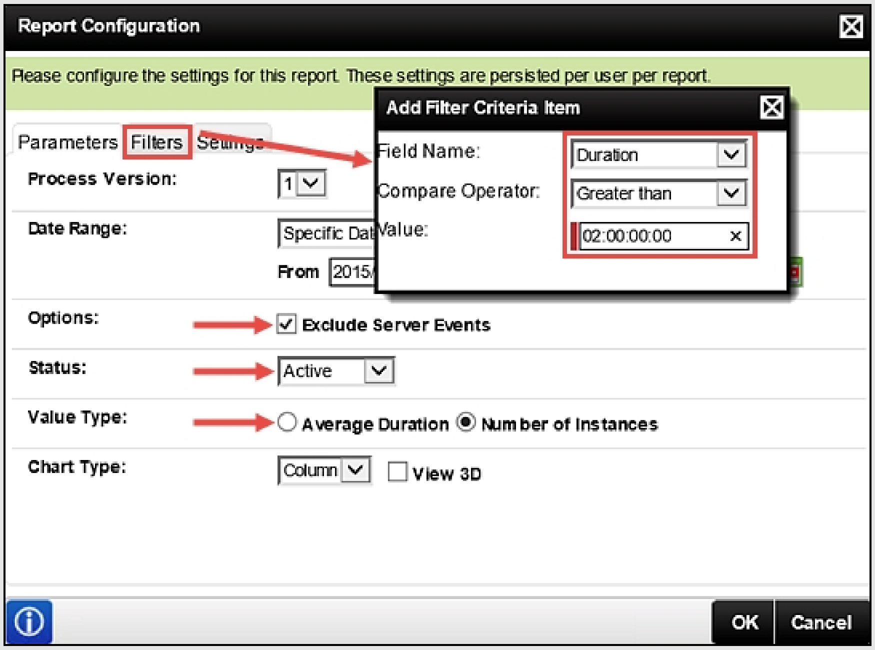
Report Configuration

Please configure the settings for this report. These settings are persisted per user per report.

Process Version:	1
Date Range:	Specific Date Range From 2015/04/04 To 2015/05/06
Options:	□ Exclude Server Events
Status:	Completed
Value Type:	● Average Duration ○ Number of Instances
Chart Type:	Column □ View 3D

2. Run a second Activity Statistics Report to determine if there are any tasks still running after two

days. This time, exclude the server events.



Please configure the settings for this report. These settings are persisted per user per report.

Field Name: Duration

Compare Operator: Greater than

Value: 02:00:00:00

Process Version: 1

Date Range: Specific Dat

From 2015/

Options: ☑ Exclude Server Events

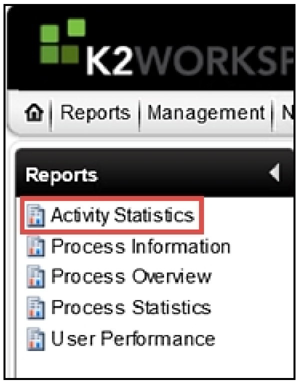
Status: Active

Value Type: ○ Average Duration ● Number of Instances

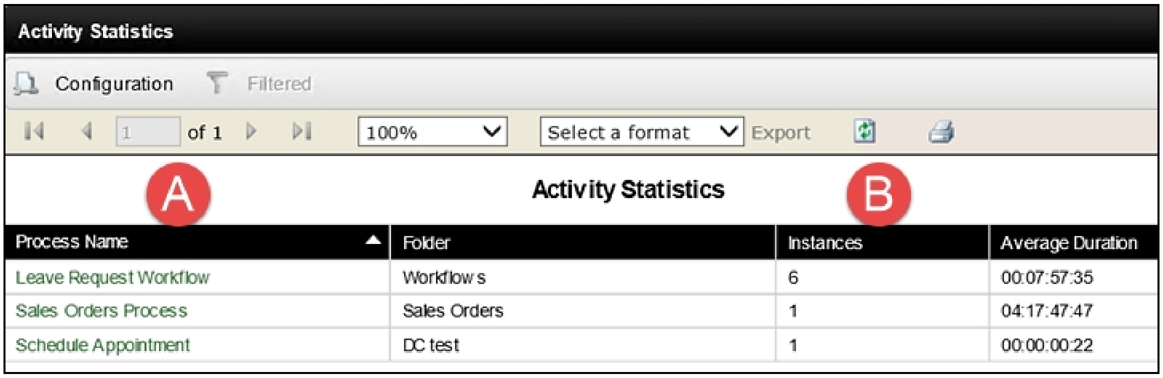
Chart Type: Column ☐ View 3D

Step 2 Walkthrough

a. From the K2 Workspace landing page, click the **Activity Statistics** Report link in the Reports pane. (It may take a minute or so for the report to open.)



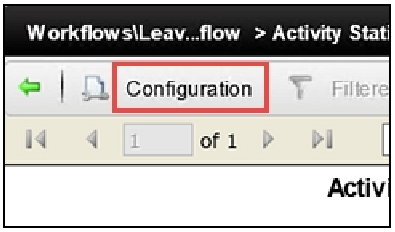
The Activity Statistics landing page will open. Here you will find a list of all **Processes** that have been deployed. (**A** below) This high-level overview also indicates the number of **Instances** that have been started for each Process. (**B** below) The number of instances includes Processes that are active and non-active.



Activity Statistics

Process Name	Folder	Instances	Average Duration
Leave Request Workflow	Workflows	6	00:07:57:35
Sales Orders Process	Sales Orders	1	04:17:47:47
Schedule Appointment	DC test	1	00:00:00:22

b. Click on the **Leave Request Workflow** Process Name to open the report. The report opens using the system default filters, or the last configuration of parameters and filters if they were set. Click the **Configuration** link.



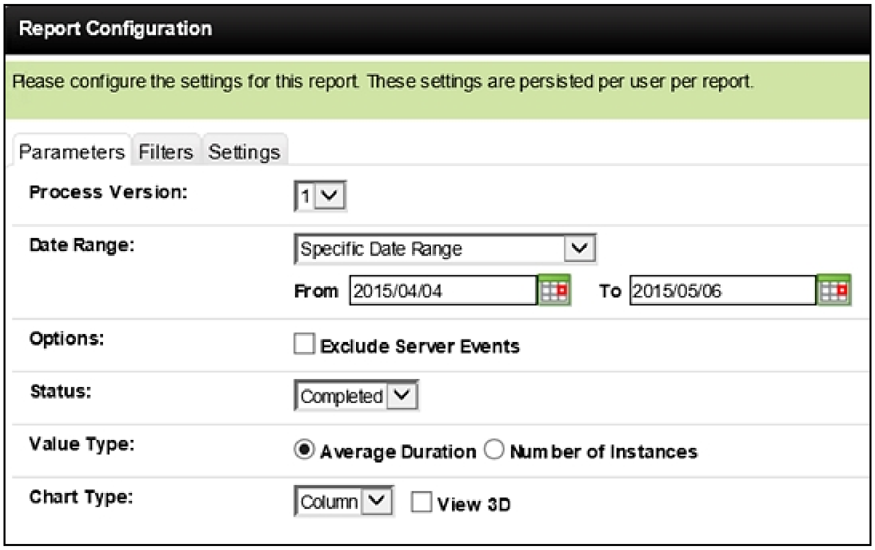
For our first report, we want to see which tasks take the longest to complete compared to the other tasks in the Process.

The Report Configuration screen opens on the Parameters page. This is where you can narrow down the results returned by specifying Parameters, Filters and Deleted Process Settings for the Process Instance you are reporting on. The following is a guide to the options available:

Parameter	Notes
Process Version	Allows you to select the Process Version you want to report on. By default, K2 will make the last workflow deployed the default. When a user submits a form, the default workflow will be the Process started. Select the version number you want to report on.
Date Range	Use the date range setting to narrow down the reporting time frame. For example, you may want to run the report weekly, then save a copy for later comparison. This will allow you analyze a Process over time to help determine the time frames the Process is used the most, or least. HINT: Use the Date Range drop-down for predefined time frames. For example, This Week, Last Week, etc.
Options	This option allows you to include or exclude Server Events. Server, or System Events are those tasks that K2 runs behind the scenes. For example, sending an email or updating a list. System tasks are generally performed quite fast. If you have multiple user tasks, you may not want to take up chart space with the system tasks.
Status	Select the Process Instance status you want to report on. Options include Active, Completed, Expired and Waiting. Generally speaking, you will report on Completed Process Instances, as they can give a more accurate view of the time a task has taken. This, when combined with multiple Process Instances, can indicate if any tasks are taking a longer time than you had anticipated.
Value Type	You have two types of values you can report on. The Average Duration will calculate the average time a task has taken, or is taking. The Number of Instances displays the total number Process Instances for the selected Status.
Chart Type	Allows you to choose the chart type you would like to output for your report.
Filters	**Notes**
Add, Edit, Remove	The Filter option allows you to further define your report results by applying filters to either the Activity Name, Duration, Full Name or Instance. For example, you might run a report based pulling Active Process Instances that have been running for longer than 1 day. In this case, you would apply a Filter on the Duration option setting to be greater than 1 day.
Settings	**Notes**
Deleted Process Instances	Select whether you want to include deleted (or removed) Process Instances in your report. By default, this is set to True. You may want to set this to False so that you are reporting on only the workflows that are viable (Active or Completed).

c. On the **Report Configuration Parameters** screen, set the options as follows: (Use the image below as a guide if necessary.)

- **Process Version**: Select the Process Version you want to report on. For the Leave Request (Extended) Process, you might only have one choice.
- **Date Range**: Make sure the **From** value is at least two-to-three weeks back. This will be dependent on how long ago you built the Leave Request (Extended) Application and/or submitted test Processes. Adjust the dates so that they encompass the time frame around building and testing the application.
- **Option**: UNCHECK **Exclude Server Events**
- **Status**: Select **Completed**
- **Value Type**: SELECT **Average Duration**
- **Chart Type**: Select **Column**
- We will not specify any Filters or Deleted Process Settings

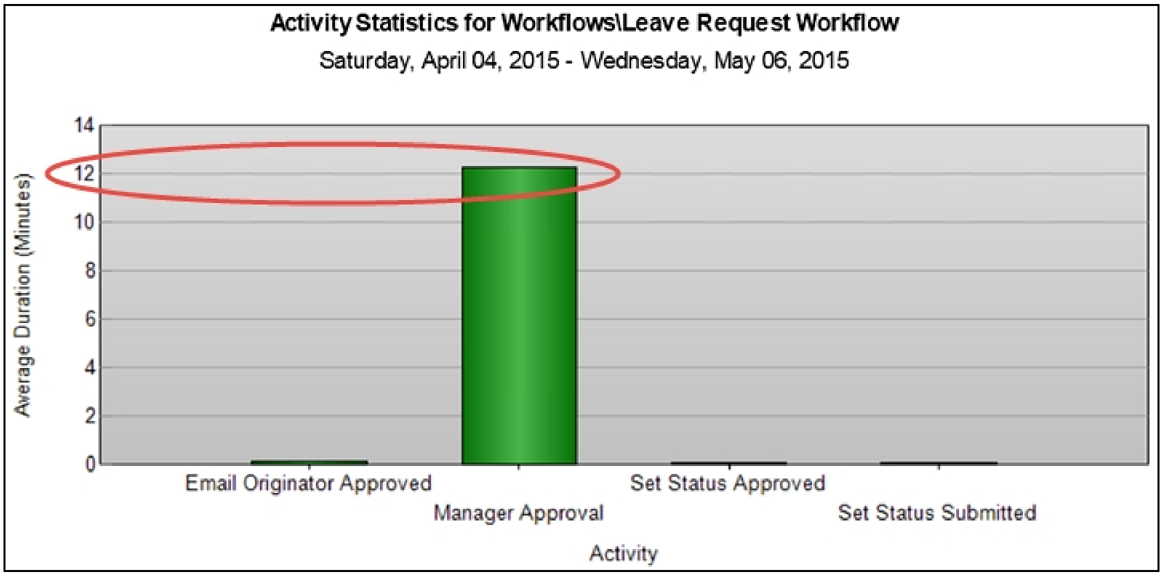


Report Configuration

Please configure the settings for this report. These settings are persisted per user per report.

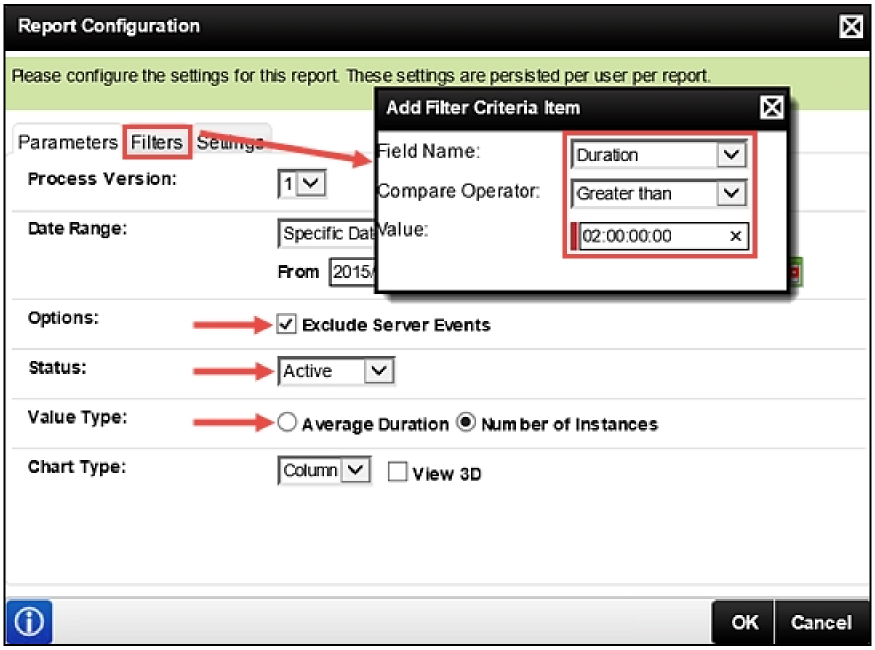
Process Version:	1
Date Range:	Specific Date Range **From** 2015/04/04 **To** 2015/05/06
Options:	□ **Exclude Server Events**
Status:	Completed
Value Type:	● **Average Duration** ○ **Number of Instances**
Chart Type:	Column □ **View 3D**

While your chart will not look exactly like the image below, we should see a representation of one user task (Manager Approval) and three other system tasks. The system tasks are processed very quickly. Notice the average time it takes for the Manager Approval task. Since we are using test applications, the 12 minutes it's taking is very reasonable. If, in the real world, if this average indicated 3-4 days, then you might look at the complexity of the task and see if the communication around the task could be made clearer. Another option to consider is to add an Escalation to the task. Escalations can be as simple as sending an email reminder of the unfinished task, or more complex, where the task is automatically redirected to another user.



d. Now we want to see if there are any Process Instances still running after two days. Open the **Report Configuration** and make the following changes to the Parameter and Filter options:

- **Options**: CHECK the box to Exclude Server Events
- **Status**: Active
- **Value Type**: Number of Instances
- **Filter** (use the Filter tab): Duration is Greater Than 2 days (02:00:00:00)



Report Configuration

Please configure the settings for this report. These settings are persisted per user per report.

Parameters | Filters | Settings

Process Version: 1

Date Range: Specific Dat

From 2015/

Options: ☑ Exclude Server Events

Status: Active

Value Type: ○ Average Duration ● Number of Instances

Chart Type: Column □ View 3D

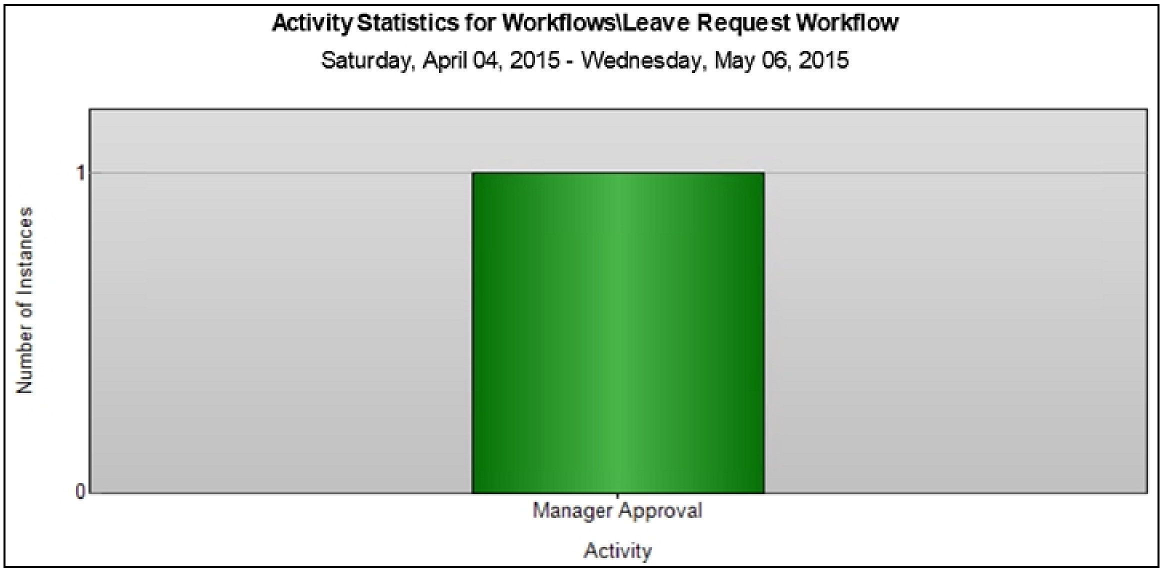
Add Filter Criteria Item

Field Name: Duration

Compare Operator: Greater than

Value: 02:00:00:00

In the chart returned, we see there is one Process Instance running that is older than two days. (Your results may not be the same.) Notice that in this report, we do not see any system tasks. This is because we selected the option to Exclude Server Events.



STEP 2 REVIEW

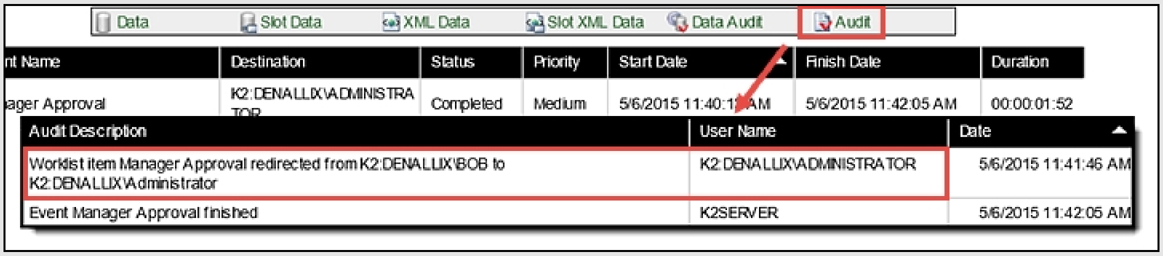
The Activity Statistics report is very useful for displaying duration-related statistics on individual Activities (events or steps). Use this report to show bottlenecks (which tasks are taking longer than others) or are running longer than desired. You can also use this report to show the number of instances that have been processed in a given time period (for example, how many Leave Request tasks were completed this week).

Step 3: The Process Overview Report

One of the most-widely used reports, the Process Overview Report allows you to view the details of events (or steps) in a Process Instance. The Process Overview Report initially displays a high-level view of a each Process Instance (Folio, Originator, Status, Start and Finish Dates and Duration) of the selected Process. Within the Process Overview report, you can drill down into each Activity Instance to determine the exact path each step took, and even the audit trail for each. You can also choose to view the report content via the View Flow option, which displays the information in a flow-chart format and is almost real-time. (We will cover the View Flow report in the next Part.) This report displays in a spreadsheet or table format.

Step 3 Tasks

1. Open the **Process Overview** Report, then the **Leave Request Workflow** Process.
2. Click on any **Process Instance** that shows a **Completed** Status. Was this Leave Request approved or not approved? How can you tell this?
3. Access the **Audit** trail for the Manager Approval step. Check to see if there were any other Process details for this Activity. (In the example below, the Manager Approval task was redirected from Bob to Denallix Administrator.



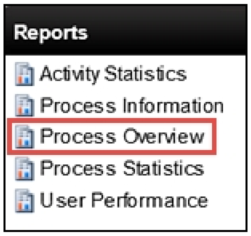
nt Name	Destination	Status	Priority	Start Date	Finish Date	Duration
ager Approval	K2:DENALLIX\ADMINISTRA TOR	Completed	Medium	5/6/2015 11:40:1[illegible] AM	5/6/2015 11:42:05 AM	00:00:01:52

Audit Description	User Name	Date
Worklist item Manager Approval redirected from K2:DENALLIX\BOB to K2:DENALLIX\Administrator	K2:DENALLIX\ADMINISTRATOR	5/6/2015 11:41:46 AM
Event Manager Approval finished	K2SERVER	5/6/2015 11:42:05 AM

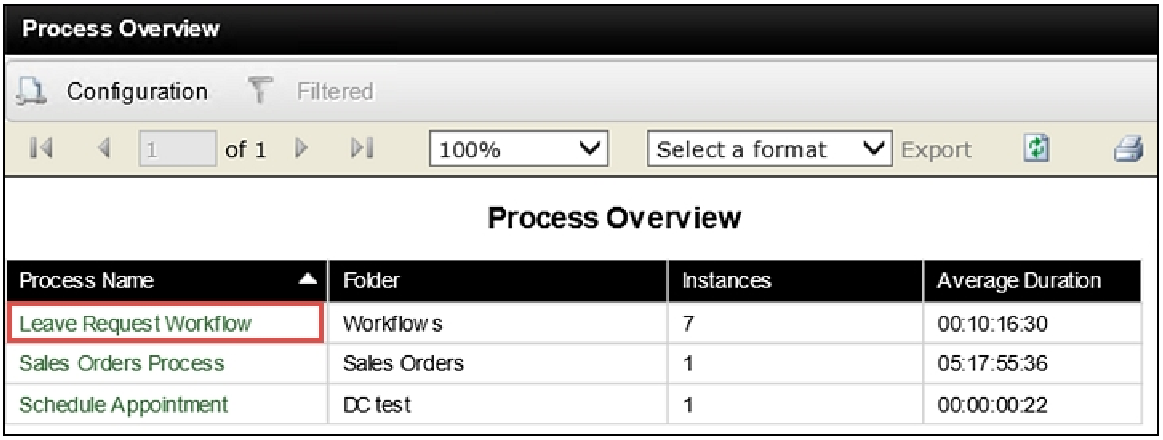
4. Returning to the **Process Instances** screen for the Leave Request Workflow, **Export** the data to **Excel**.
5. Review the details for a Process Instance that is still **Active**. What Activity is the workflow currently on; how long has it been at that Activity?

Step 3 Walkthrough

a. In the Reports pane, click on the **Process Overview** link to open the report.



On the Report's landing page, you will see a list of all the Processes that have been deployed. This list indicates the number of Instances for each Process, including Active and Completed Instances. You can also see an Average Duration for the Process.

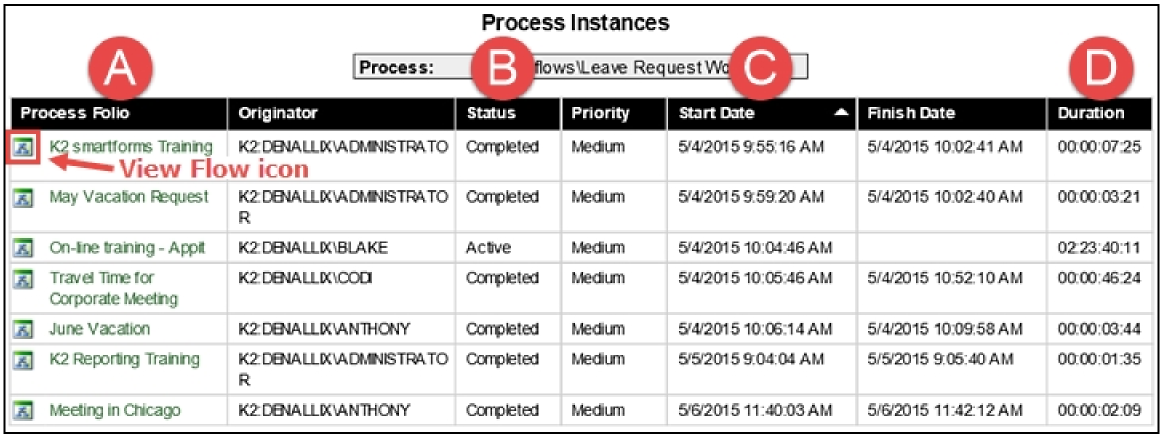


Process Overview

Process Name	Folder	Instances	Average Duration
Leave Request Workflow	Workflows	7	00:10:16:30
Sales Orders Process	Sales Orders	1	05:17:55:36
Schedule Appointment	DC test	1	00:00:00:22

b. Click on the **Leave Request Workflow** Process Name.

The Process Overview > **Process Instances** screen opens. This is a list of all Leave Request Workflows that have been submitted. In the image below, notice the **Process Folio** column. (**A** below) Recall that we assigned the Leave Request Title as the Folio that helps provide a unique identifier (along with the Originator name) for each Process Instance.



Process Instances

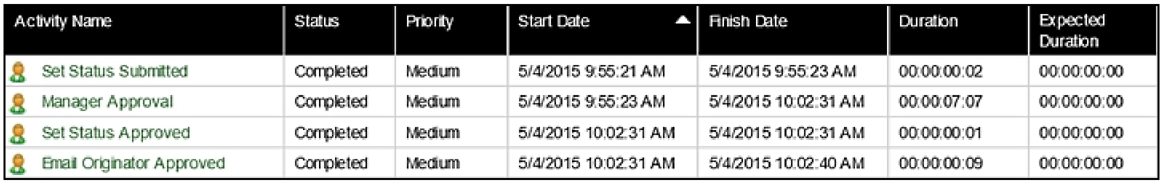
Process: flows\Leave Request Wo

Process Folio	Originator	Status	Priority	Start Date	Finish Date	Duration
K2 smartforms Training	K2:DENALLIX\ADMINISTRATO	Completed	Medium	5/4/2015 9:55:16 AM	5/4/2015 10:02:41 AM	00:00:07:25
May Vacation Request	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/4/2015 9:59:20 AM	5/4/2015 10:02:40 AM	00:00:03:21
On-line training - Appt	K2:DENALLIX\BLAKE	Active	Medium	5/4/2015 10:04:46 AM		02:23:40:11
Travel Time for Corporate Meeting	K2:DENALLIX\CODI	Completed	Medium	5/4/2015 10:05:46 AM	5/4/2015 10:52:10 AM	00:00:46:24
June Vacation	K2:DENALLIX\ANTHONY	Completed	Medium	5/4/2015 10:06:14 AM	5/4/2015 10:09:58 AM	00:00:03:44
K2 Reporting Training	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/5/2015 9:04:04 AM	5/5/2015 9:05:40 AM	00:00:01:35
Meeting in Chicago	K2:DENALLIX\ANTHONY	Completed	Medium	5/6/2015 11:40:03 AM	5/6/2015 11:42:12 AM	00:00:02:09

c. The Status (**B** above) column tells us if the Process is still active or has been completed. The Start Date (**C** above) and Duration (**D** above) give us an idea of how long this Process has been running, or how long it ran. Notice too, that each Process Instance has a View Flow icon as well.

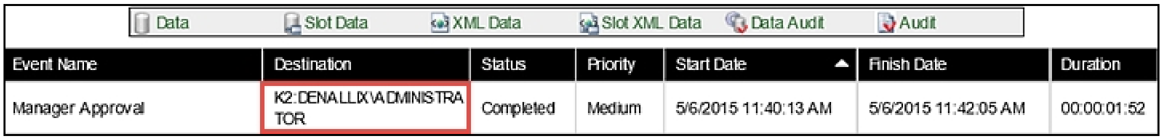
d. Click on a **Process Folio** name for a Process Instance that has a **Completed** Status. The Process Overview > **Activity Instances** screen opens. (Keep a mental note of the Process Instance you are opening. We will want to open the same Process in Part 2: View Flow, so that we can observe the same Process Instance details in a flowchart format.)

Notice the Activity Name column. Recall building the workflow. These are the events put in place for completing the workflow if the manager *Approved* the request. We know this request was approved because of the Set Status Approved activity shown. If the request had been rejected, the activity would be the Set Status Rejected. You can view when each step began and ended and the time it took to complete that step.



Activity Name	Status	Priority	Start Date	Finish Date	Duration	Expected Duration
Set Status Submitted	Completed	Medium	5/4/2015 9:55:21 AM	5/4/2015 9:55:23 AM	00:00:00:02	00:00:00:00
Manager Approval	Completed	Medium	5/4/2015 9:55:23 AM	5/4/2015 10:02:31 AM	00:00:07:07	00:00:00:00
Set Status Approved	Completed	Medium	5/4/2015 10:02:31 AM	5/4/2015 10:02:31 AM	00:00:00:01	00:00:00:00
Email Originator Approved	Completed	Medium	5/4/2015 10:02:31 AM	5/4/2015 10:02:40 AM	00:00:00:09	00:00:00:00

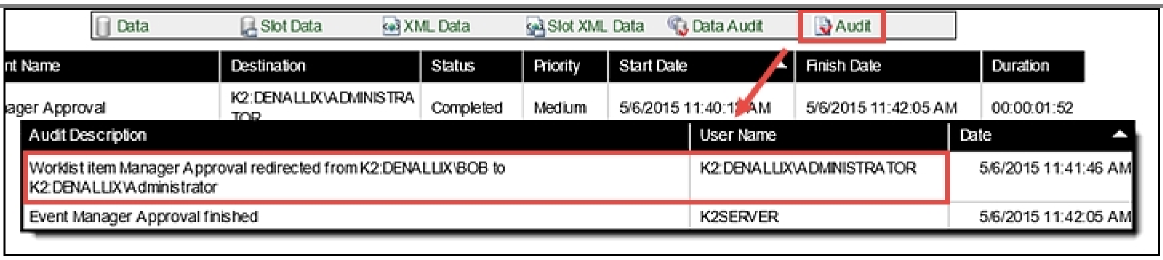
e. Click on the **Manager Approval Activity Name**. These are the final details for the Manager Approval step and it shows that Denallix Administrator approved the request.



Event Name	Destination	Status	Priority	Start Date	Finish Date	Duration
Manager Approval	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/6/2015 11:40:13 AM	5/6/2015 11:42:05 AM	00:00:01:52

We know that Anthony was the form originator (for this example). But we also know that Denallix Administrator is *not* Anthony's manager. We want to find know why Administrator approved the request and not Anthony's manager. One way to do this is to review the Audit trail for this step.

f. Click on the **Audit** link. Now we are looking at the fine details for the Manager Approval step. If there were any deviations from the normal path, they would show up here. And indeed, we can see that the Manager Approval task was redirected from Bob (Anthony's manager) to Denallix Administrator. The User Name column shows us the user that did the task redirection.

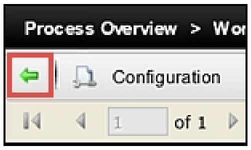


nt Name	Destination	Status	Priority	Start Date	Finish Date	Duration
ager Approval	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/6/2015 11:40:1 AM	5/6/2015 11:42:05 AM	00:00:01:52

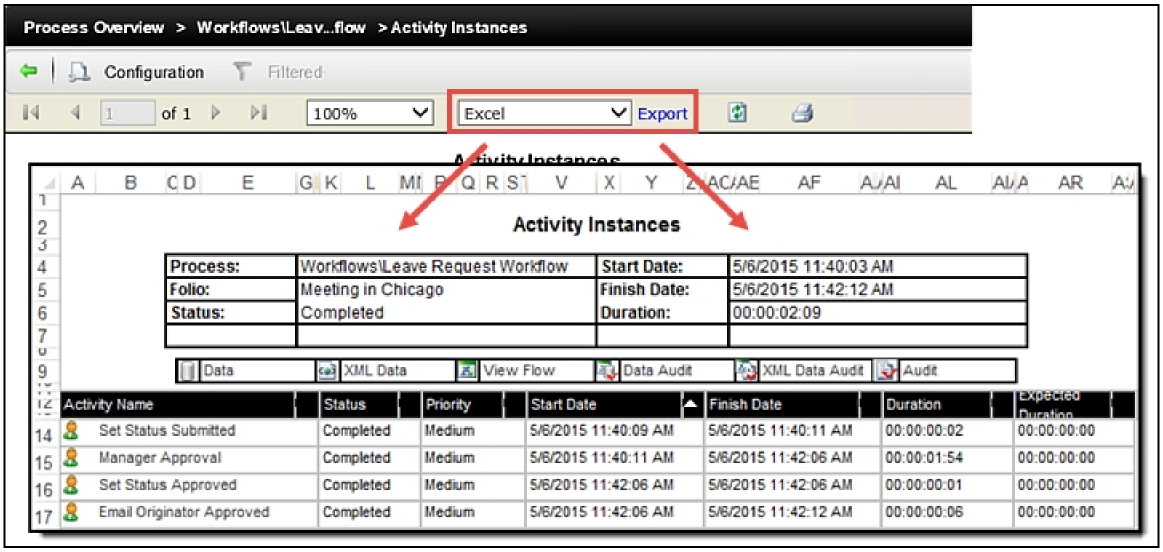
Audit Description	User Name	Date
Worklist item Manager Approval redirected from K2:DENALLIX\BOB to K2:DENALLIX\Administrator	K2:DENALLIX\ADMINISTRATOR	5/6/2015 11:41:46 AM
Event Manager Approval finished	K2SERVER	5/6/2015 11:42:05 AM

One last feature to mention that is very useful is the ability to Export reports into Excel or as a PDF file. If using Excel, this would allow you to compile reports from multiple Processes into one worksheet. It also creates a format that you can easily share with others. Most Standard Reports can be exported to Excel and PDF file formats.

g. Use the **Previous** button to return to the Activity Instances screen.



h. Select **Excel** from the Export drop-down options in the tools bar. Click **Export**. Your report has now been exported to Excel, where you can work with the data as you need to.


Process Overview > Workflows\Leav...flow > Activity Instances
Configuration
Filtered
1
of 1
100%
Excel
Export
Activity Instances
A B C D E G K L M P Q R S V X Y Z AC AE AF AJ AL AR
Data
XML Data
View Flow
Data Audit
XML Data Audit
Audit

Activity Instances

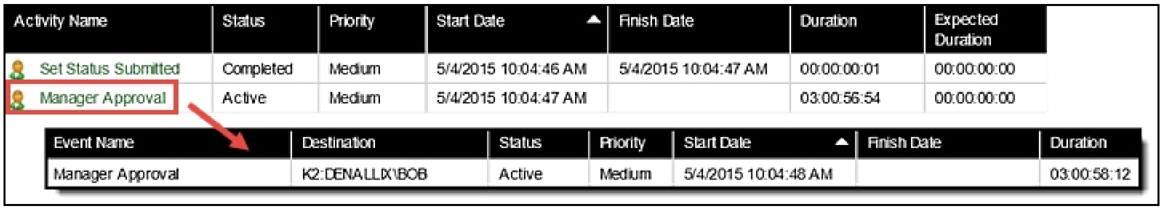
Process:	Workflows\Leave Request Workflow	Start Date:	5/6/2015 11:40:03 AM
Folio:	Meeting in Chicago	Finish Date:	5/6/2015 11:42:12 AM
Status:	Completed	Duration:	00:00:02:09

Activity Name	Status	Priority	Start Date	Finish Date	Duration	Expected Duration
Set Status Submitted	Completed	Medium	5/6/2015 11:40:09 AM	5/6/2015 11:40:11 AM	00:00:00:02	00:00:00:00
Manager Approval	Completed	Medium	5/6/2015 11:40:11 AM	5/6/2015 11:42:06 AM	00:00:01:54	00:00:00:00
Set Status Approved	Completed	Medium	5/6/2015 11:42:06 AM	5/6/2015 11:42:06 AM	00:00:00:01	00:00:00:00
Email Originator Approved	Completed	Medium	5/6/2015 11:42:06 AM	5/6/2015 11:42:12 AM	00:00:00:06	00:00:00:00

We have just reviewed the details for a Process Instance that has been completed. In the next few steps, we will look at a Process Instance that is still Active. This will tell us what event or step the workflow is currently on, and how long it's been there.

i. Using the green **Previous** arrow, make your way back to the **Overview Instances** landing page. (This is the page that lists all of the Process Instances for the Leave Request Workflow Process.) Click on any **Process Instance** that shows an **Active** Status.

In the Process Instance we selected, we can see that the first system task (Set Status Submitted) has completed. The next task, Manager Approval, is still waiting to be actioned. If we click on the Manager Approval title to view the details, we can see that Bob is the Destination user and the Duration is in excess of 3 days.



Activity Name	Status	Priority	Start Date	Finish Date	Duration	Expected Duration
Set Status Submitted	Completed	Medium	5/4/2015 10:04:46 AM	5/4/2015 10:04:47 AM	00:00:00:01	00:00:00:00
Manager Approval	Active	Medium	5/4/2015 10:04:47 AM		03:00:56:54	00:00:00:00

Event Name	Destination	Status	Priority	Start Date	Finish Date	Duration
Manager Approval	K2:DENALLIX\BOB	Active	Medium	5/4/2015 10:04:48 AM		03:00:58:12

STEP 3 REVIEW

In this step, we explored the Process Overview Report and learned how we can drill down into Process Instances and Activity Instances to get a very complete picture of the workflow and its status. The Process Overview landing page gives a high-level view of each Process Instance, displaying the current Status (Active, Completed in our example), the Starting date and Finish date (if Completed) and the Duration. With this report, we are able to drill down into the Activity Instances to view the status of each event, or step, within the workflow Process. Using the Audit feature, we can determine if any additional steps occurred, such as a task being redirected.

Viewing an Active Process Instance shows us the current event or step for the Process.

We can also export report content to Excel or as a PDF file, for distribution or for further compiling report data.

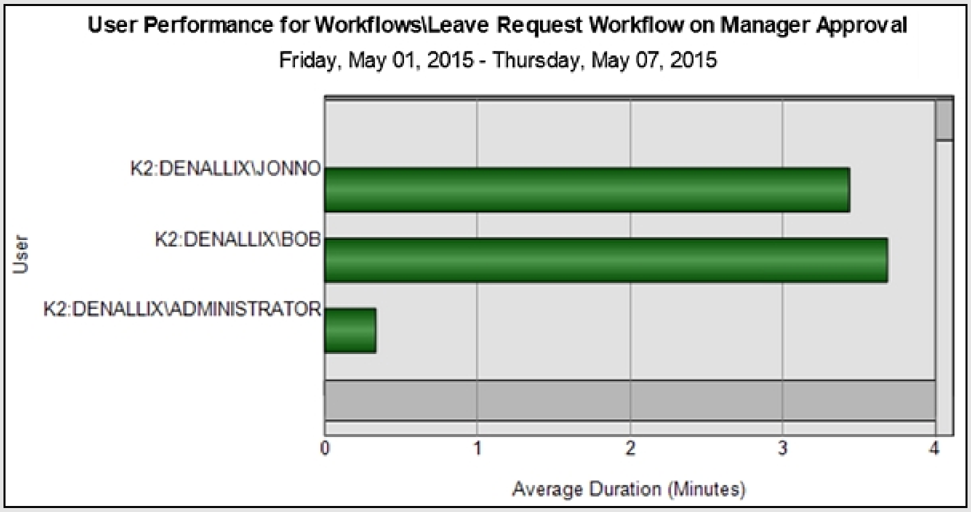
Step 4: The User Performance Report

The User Performance Report shows either the number of instances or the duration for a single Activity Instance of a Process, per user. This report can be useful for determining work load or training needs, if a user consistently takes longer to complete the same task compared to other users. An example might be a Help Desk scenario, where requests are submitted by users, and assigned to a group of Help Desk staff. This report would tell you who is responding to the most requests, or who is taking a longer than expected time to resolve requests. This report displays in a graphical format.

Step 4 Tasks

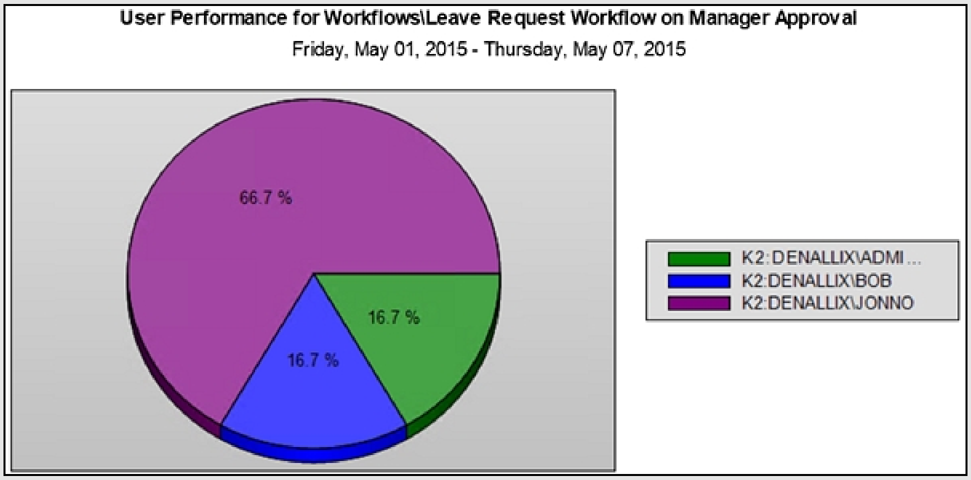
In this step, we want to generate two versions of the **User Performance** Report.

1. The first report will display how long (on average) it takes users to complete the Manager Approval task.



2. The second report should display the number of Manager Approval tasks each user has com-

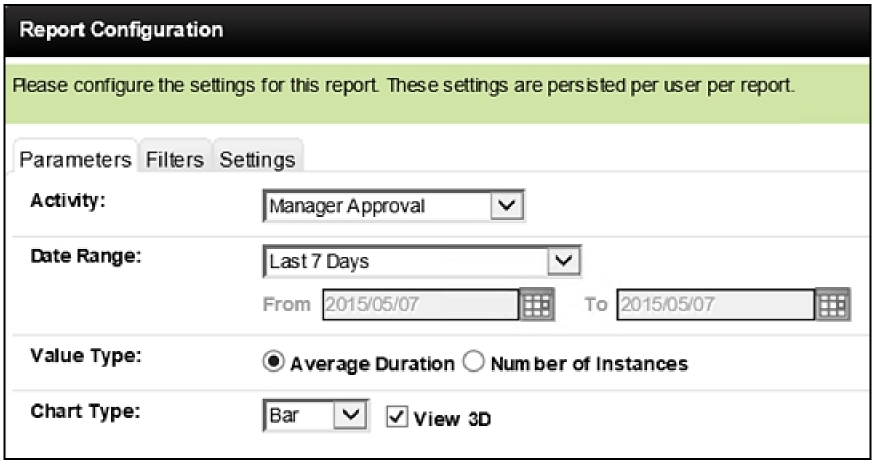
pleted within a week's time frame.



Step 4 Walkthrough

For our first report, we want to see how long (on average) it takes our users to complete the Manager Approval task.

a. Click on the **User Performance** Report. On the landing page, click the **Leave Request Workflow** Process. The default chart opens. Open the **Report Configuration** editor and make the following settings:
   - **Activity**: Manager Approval
   - **Date Range**: Last 7 Days (or any range that would contain completed Processes)
   - **Value Type**: Average Duration
   - **Chart Type**: Bar (or any chart type you prefer)



Report Configuration

Please configure the settings for this report. These settings are persisted per user per report.

Activity: Manager Approval

Date Range: Last 7 Days

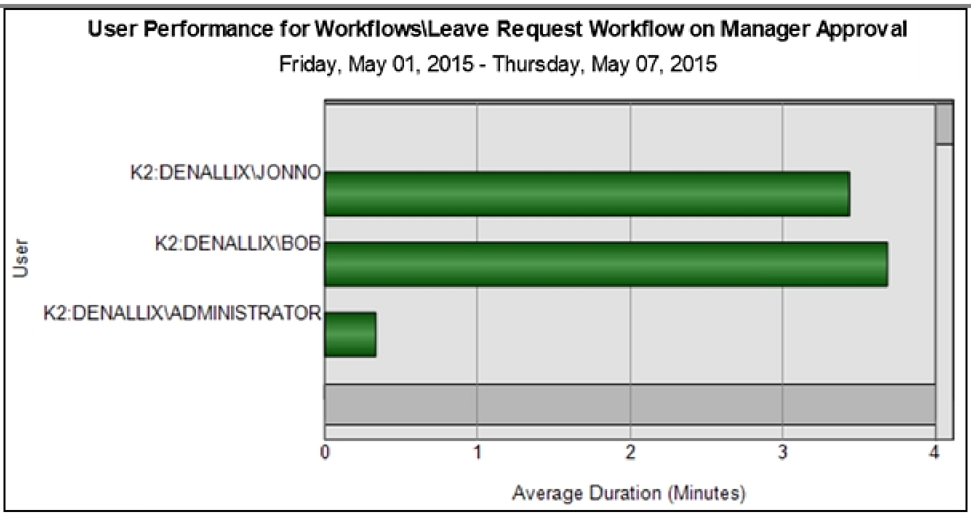
From 2015/05/07 To 2015/05/07

Value Type: ● Average Duration ○ Number of Instances

Chart Type: Bar ☑ View 3D

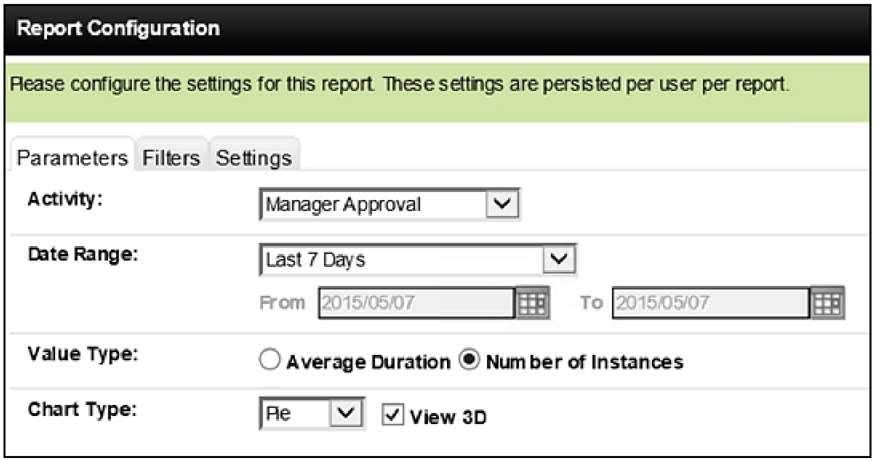
After the report is generated, we can see a comparison of the Manager Approval task average times for three users. Points we might consider are:

- Workload consistency between users
- Acceptable completion times versus unacceptable completion times (is there a standard?)
- User training for completing the task
- Escalations (reminders, redirection, etc.) for the task



For the next report, we want to see how the workload is spread out among the three users. This report will tell us the number of Manager Approval tasks each user has completed within a given time frame.

b. Once again, open the **Report Configuration** editor. Change the **Value Type** to **Number of Instances**. For this chart, we are going to use a **Pie** Chart Type, in 3D format.



Report Configuration

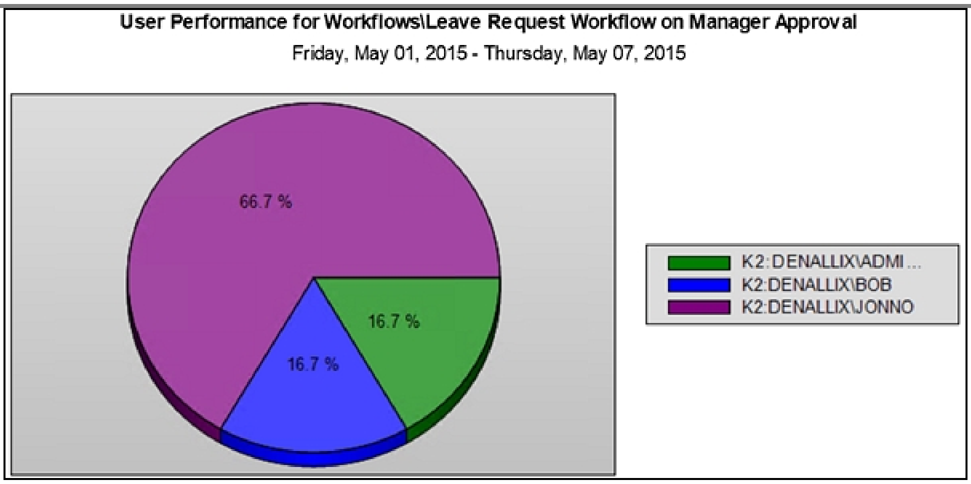
Please configure the settings for this report. These settings are persisted per user per report.

Parameters | Filters | Settings

Activity:	Manager Approval
Date Range:	Last 7 Days From 2015/05/07 To 2015/05/07
Value Type:	○ Average Duration ● Number of Instances
Chart Type:	Pie ☑ View 3D

The report is generated and we can view the distribution of Manager Approval tasks for our three users. Let's consider chart below using the Help Desk scenario where requests are assigned to a group of Help Desk staff, who in turn take ownership of tasks and complete them. Now we might want to consider:

- Why is one user completing more tasks than the others? Is this a performance issue or a training issue? Do the other users have workloads that are preventing them from taking ownership of tasks?



STEP 4 REVIEW

The User Performance report gives us an overview of task assignments and user volume performance. In the real world, this report would be a starting point for analyzing a number of workplace factors including workload, training (knowledge), performance, time management, etc.

This concludes Part 1: K2 Workspace. When you are ready, continue on to Part 2: View Flow, working with the View Flow Report.

The View Flow Report

In the Reporting in K2 Workspace exercise, we worked with the Process Overview report and observed how you can drill down into the Activity Instance details in a linear fashion. In this tutorial, we will see how this same information can be displayed in a flowchart format by using the View Flow Report. First, we'll take a tour of the View Flow Report, then we will action the Active Leave Request Workflow Process and observe the View Flow chart as it updates in almost real-time.

Note
This tutorial picks up from the Reporting in K2 Workspace tutorial. Be sure to complete that tutorial before continuing.

Step 5: Comparing View Flow results with the Process Overview Report

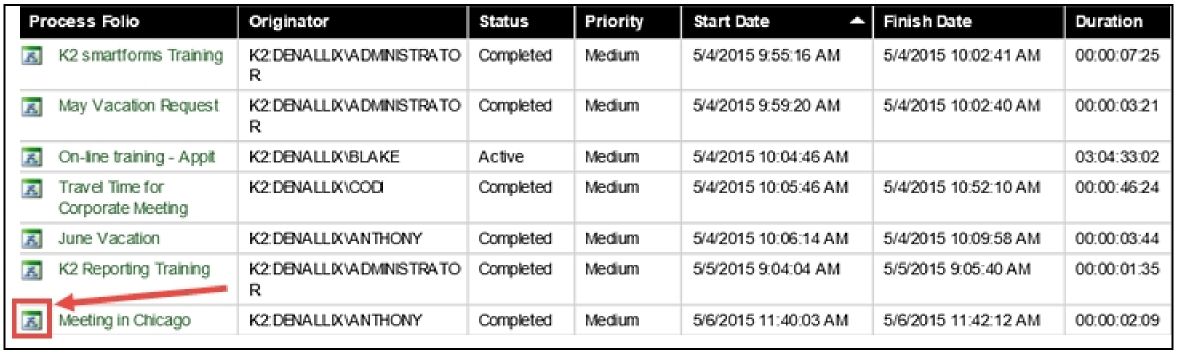
In this step, we will compare the View Flow results with the Completed Leave Request Workflow Process we observed in Step 3 of Part 1.

Step 5 Tasks

1. Open the **View Flow** Report for the same *completed* Leave Request Workflow Process Instance that you used for the Process Overview Report steps in Part 1.
2. Observe how the workflow is visually presented, and that you can see the paths/decisions that the workflow followed.
3. Open the **Process Summary** details (double-click the Start button) and observe how the same information found in the Process Overview Report is presented in the View Flow Process Summary.
4. Open the Manager Approval **Activity Summary** details and again, observe how the Process Overview Report details are presented in the View Flow format.

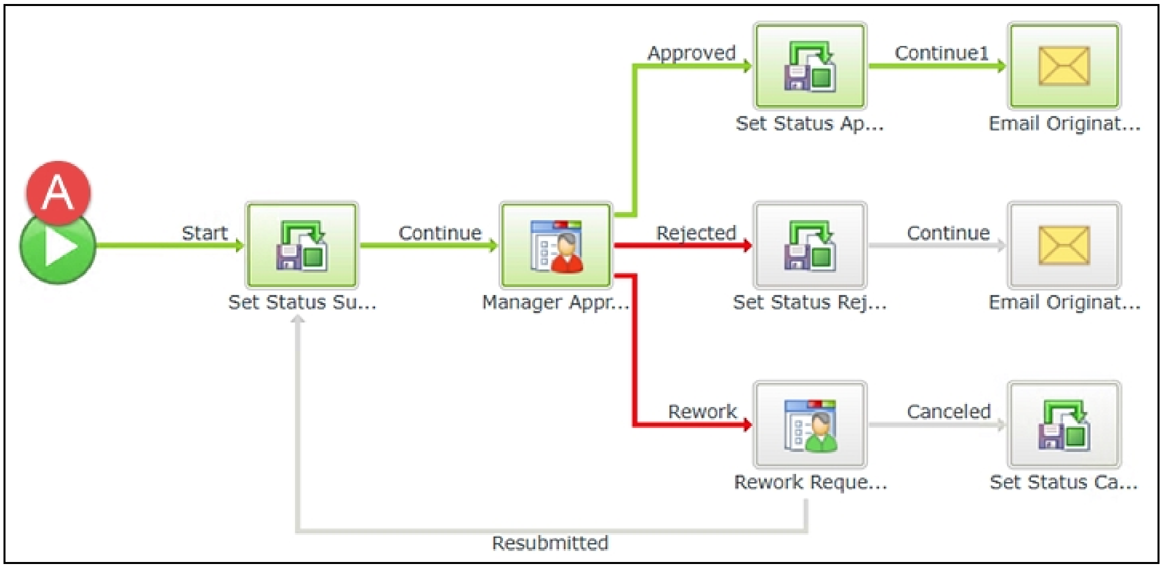
Step 5 Walkthrough

a. Click on the **Process Overview** Report to open it. Click on the **Leave Request Workflow** Process to expose the Process Instances. Each Process Instance has a View Flow icon to the left of the Process Folio title. Each Process Instance can be viewed in View Flow format. Click the **View Flow** icon that is associated with the **Completed Process Instance** that you worked with in Part 1.

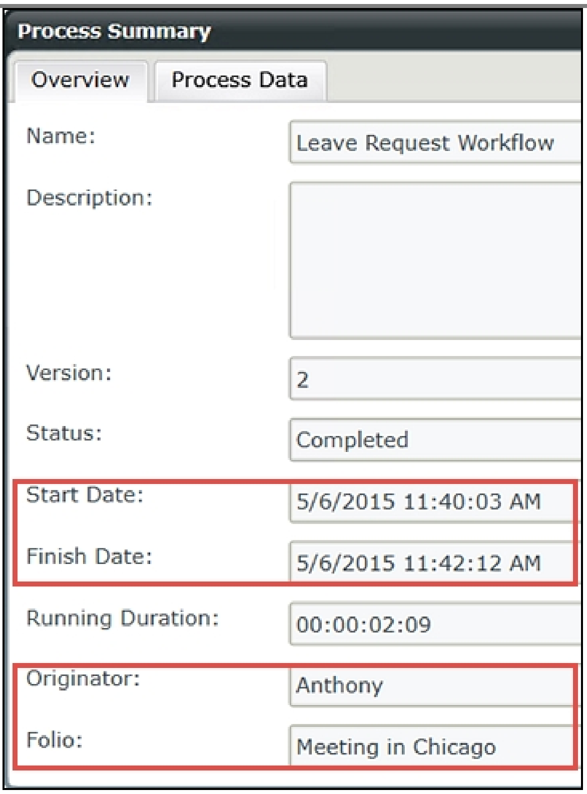


Process Folio	Originator	Status	Priority	Start Date	Finish Date	Duration
K2 smartforms Training	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/4/2015 9:55:16 AM	5/4/2015 10:02:41 AM	00:00:07:25
May Vacation Request	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/4/2015 9:59:20 AM	5/4/2015 10:02:40 AM	00:00:03:21
On-line training - Appt	K2:DENALLIX\BLAKE	Active	Medium	5/4/2015 10:04:46 AM		03:04:33:02
Travel Time for Corporate Meeting	K2:DENALLIX\CODI	Completed	Medium	5/4/2015 10:05:46 AM	5/4/2015 10:52:10 AM	00:00:46:24
June Vacation	K2:DENALLIX\ANTHONY	Completed	Medium	5/4/2015 10:06:14 AM	5/4/2015 10:09:58 AM	00:00:03:44
K2 Reporting Training	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/5/2015 9:04:04 AM	5/5/2015 9:05:40 AM	00:00:01:35
Meeting in Chicago	K2:DENALLIX\ANTHONY	Completed	Medium	5/6/2015 11:40:03 AM	5/6/2015 11:42:12 AM	00:00:02:09

The View Flow Report opens. One difference between the Process Overview report and the View Flow report is that we can see at a glance what 'paths' the workflow took by following the **green** lines. We see that the outcome for the Manager Approval task was **Approved** and that after updating the Status and sending an email to the originator, the workflow was complete. (The red lines indicate the workflow outcomes that were *not* followed.) What we don't see, is *who* made the Approved decision, and we'll explore that next.



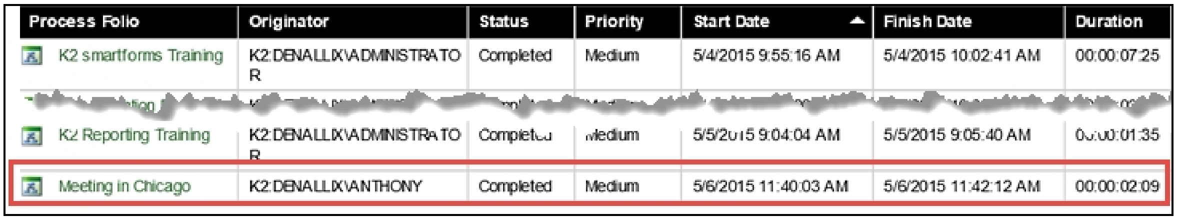
b. Double-click the **Start** button to open the **Process Summary**. (**A** above) Here we find basic details regarding this Process Instance. We can see the Start and Finish Dates and the Duration. We also can see that Anthony was the form originator and we see the Folio name.



Process Summary

Name:	Leave Request Workflow
Description:	
Version:	2
Status:	Completed
Start Date:	5/6/2015 11:40:03 AM
Finish Date:	5/6/2015 11:42:12 AM
Running Duration:	00:00:02:09
Originator:	Anthony
Folio:	Meeting in Chicago

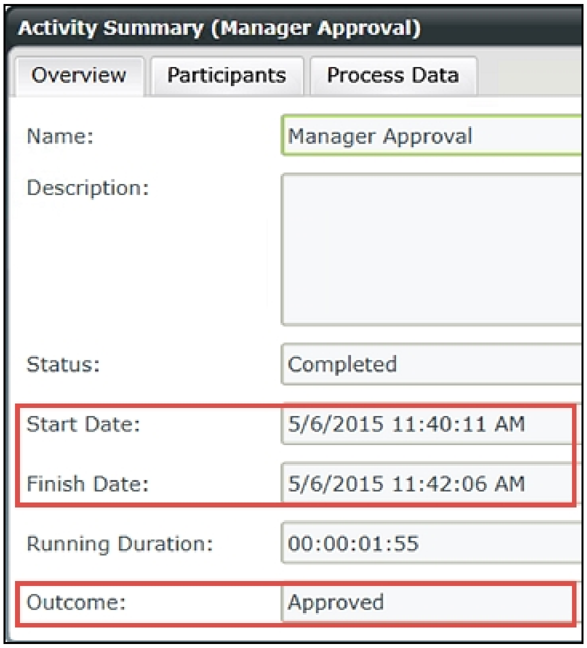
Compare the View Flow details with the Process Overview Report.



Process Folio	Originator	Status	Priority	Start Date	Finish Date	Duration
K2 smartforms Training	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/4/2015 9:55:16 AM	5/4/2015 10:02:41 AM	00:00:07:25
K2 Reporting Training	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/5/2015 9:04:04 AM	5/5/2015 9:05:40 AM	00:00:01:35
Meeting in Chicago	K2:DENALLIX\ANTHONY	Completed	Medium	5/6/2015 11:40:03 AM	5/6/2015 11:42:12 AM	00:00:02:09

Exit the Process Summary screen to continue.

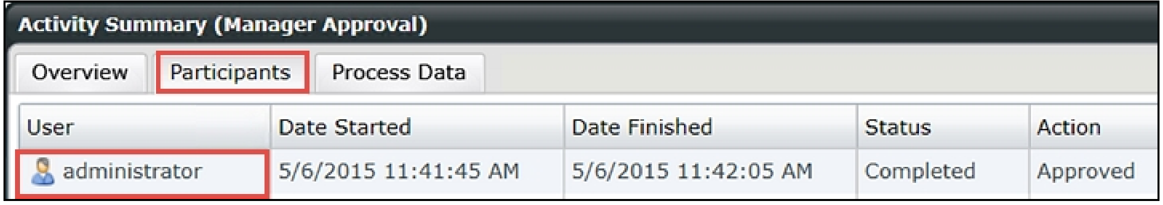
c. Double-click the **Manager Approval** task to open its details. On the **Overview** screen, we have details including the task Start and Finish Dates, as well as the Outcome (Approved).



Activity Summary (Manager Approval)

Name:	Manager Approval
Description:	
Status:	Completed
Start Date:	5/6/2015 11:40:11 AM
Finish Date:	5/6/2015 11:42:06 AM
Running Duration:	00:00:01:55
Outcome:	Approved

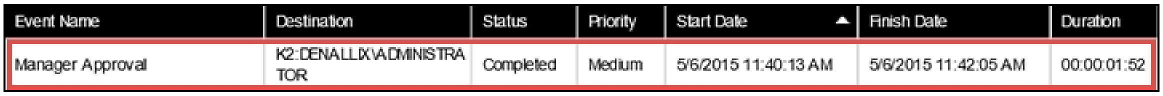
d. Click the **Participants** tab. Now we can see that administrator (this is the same as Denallix Administrator) approved the request.



Activity Summary (Manager Approval)

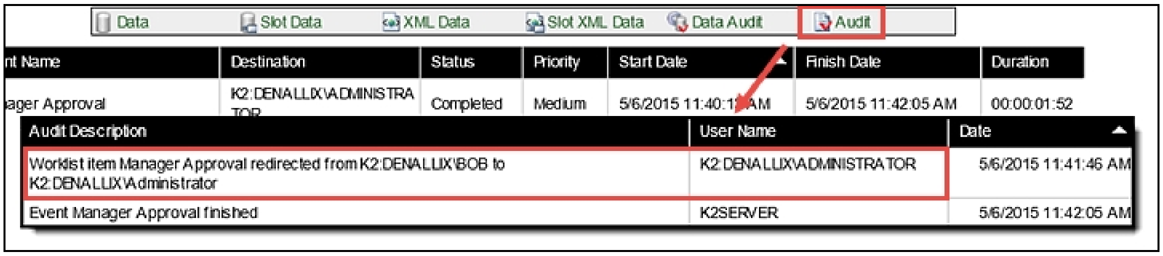
User	Date Started	Date Finished	Status	Action
administrator	5/6/2015 11:41:45 AM	5/6/2015 11:42:05 AM	Completed	Approved

Compare this information with the details found in the Process Overview Report.



Event Name	Destination	Status	Priority	Start Date	Finish Date	Duration
Manager Approval	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/6/2015 11:40:13 AM	5/6/2015 11:42:05 AM	00:00:01:52

Recall that we observed the Audit trail in the Process Overview Report. In View Flow, Audit details are not available.



nt Name	Destination	Status	Priority	Start Date	Finish Date	Duration
ager Approval	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/6/2015 11:40:1 AM	5/6/2015 11:42:05 AM	00:00:01:52

Audit Description	User Name	Date
Worklist item Manager Approval redirected from K2:DENALLIX\BOB to K2:DENALLIX\Administrator	K2:DENALLIX\ADMINISTRATOR	5/6/2015 11:41:46 AM
Event Manager Approval finished	K2SERVER	5/6/2015 11:42:05 AM

e. **Exit** the Activity Summary screen.

STEP 5 REVIEW

The View Flow Report provides the same details as the Process Overview Report with some minor exceptions (for example, the Audit trail). While the details are largely the same, the presentation and navigation is what sets these two reports apart. The View Flow offers a clear vision of the entire Process Instance (start to finish) on one screen. With very few clicks, you have the same Process and Activity Instance details as you do with the Process Overview Report. In the next step, we will see the fundamental difference between the two reports, as we observe the View Flow report update in almost real-time.

Step 6: Observe a Process Instance in real time through the View Flow chart

In this step, we will observe a workflow in progress and how the View Flow Report updates in almost real time. This feature can be very useful for troubleshooting as you are able to view the path the workflow is taking and events as they are occurring. For this step, we will first action a Process Instance that has an Active Status. We will then return to the View Flow chart and confirm the outcome path corresponds with the action taken.

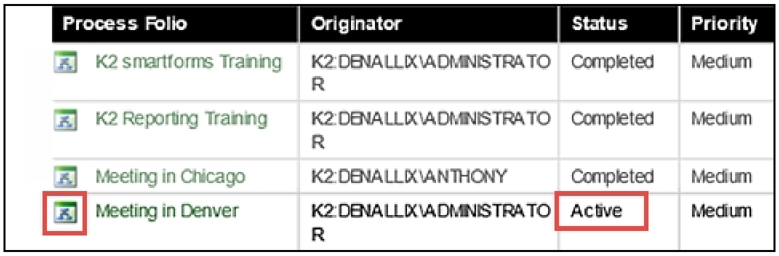
Step 6 Tasks

1. Open a **View Flow** Report for a Process Instance that has an **Active** Status. Observe how the chart has a green line up to the Manager Approval step.
2. Open **Outlook** for the destination user assigned to the **Manager Approval** task.
3. Position **Outlook** and the **View Flow** screen so that they are side-by-side. Using **SmartActions**, action the Leave Request from the manager's Outlook. (SmartActions simply require you to Reply to the task notification email, entering one of the outcome actions in the email message body. K2 will 'read' the action and continue the workflow accordingly.)
4. Observe the **View Flow** chart and how it updates automatically with the action outcome entered.

Step 6 Walkthrough

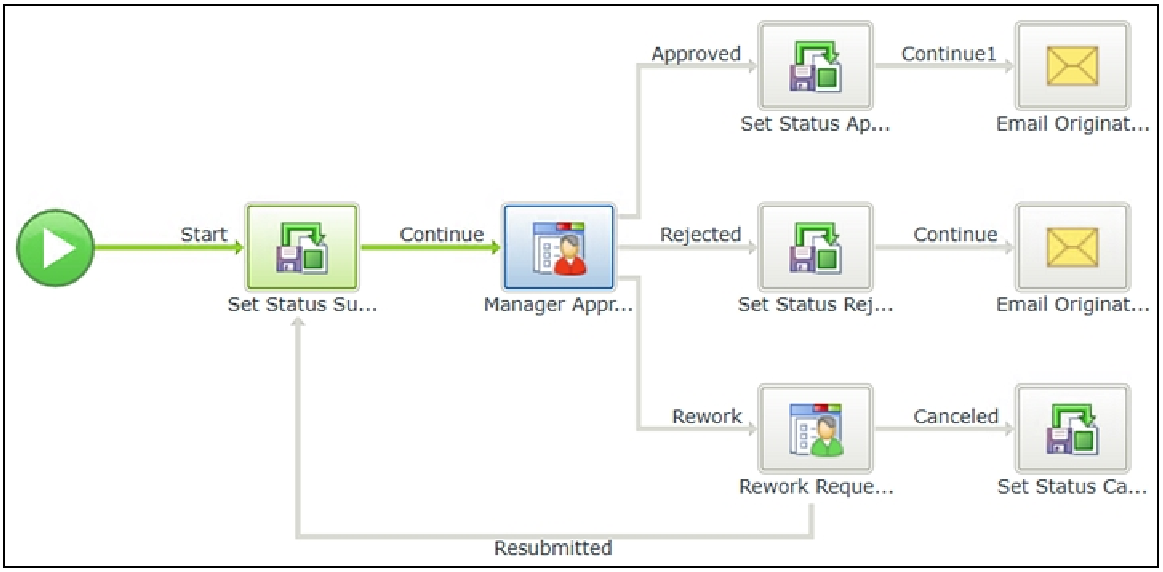
a. Open the **Process Overview** Reports, then the **Leave Request Workflow** Process. Click the **View Flow** icon for a Process Instance that has an **Active** Status. (If you do not have any Process

Instances that are active, you will need to submit a new Leave Request Form to create a new, active Process Instance.)

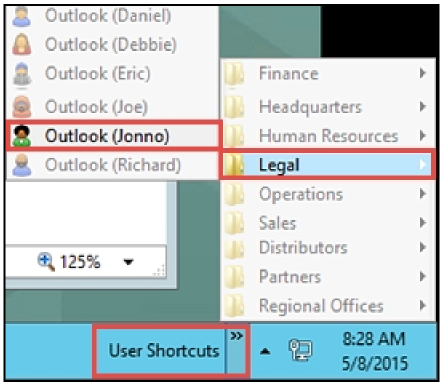


Process Folio	Originator	Status	Priority
K2 smartforms Training	K2:DENALLIX\ADMINISTRATOR	Completed	Medium
K2 Reporting Training	K2:DENALLIX\ADMINISTRATOR	Completed	Medium
Meeting in Chicago	K2:DENALLIX\ANTHONY	Completed	Medium
Meeting in Denver	K2:DENALLIX\ADMINISTRATOR	Active	Medium

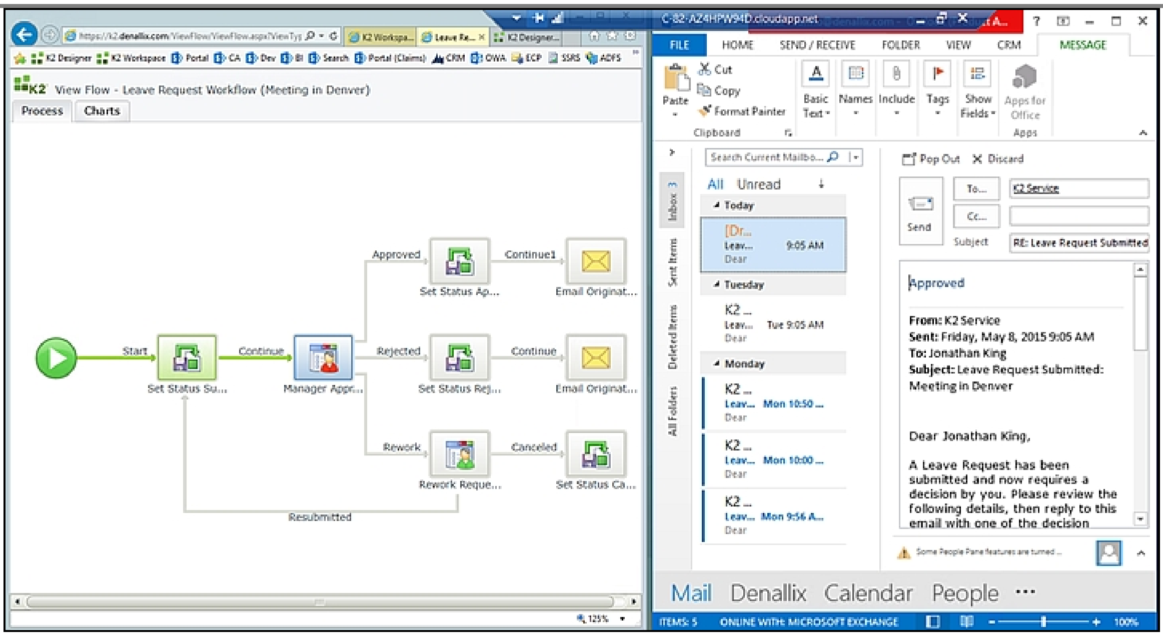
Your View Flow Report should look similar to the image below. Notice the green line flows through the system task (Set Status), then stops at the Manager Approval task. This is telling you that the workflow is currently waiting for the Manager Approval task to be actioned. In the next step, we are going to action this request, then observe the View Flow Report update itself.



b. Since the Leave Request Workflows were submitted by us (Denallix Administrator), we need to open Administrator's manager's Outlook (who is Jonno). Click the **User Shortcuts** link in the lower right corner of your screen. Expand the **Legal** folder, then click on **Outlook (Jonno)**. (Jonno's Outlook will launch. This may take a minute or two. If you get a dialog about CRM syncing, let it complete, then click OK. If you get a dialog about licensing, click Close to continue.)



Before you action the request, adjust your View Flow Report screen and Outlook screen so that they are side-by-side. The View Flow Report can update very quickly (depending on your environment) after we action the request in Outlook. With the screens side-by-side, you will be able to see the View Flow update.



Pop Out ✕ Discard

To... K2 Service

Cc...

Subject RE: Leave Request Submitted

Approved

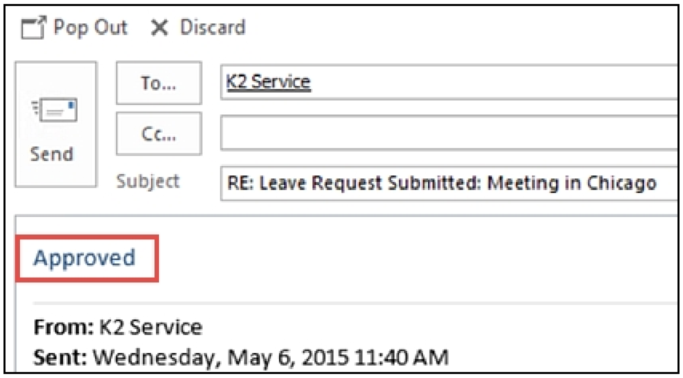
From: K2 Service
Sent: Friday, May 8, 2015 9:05 AM
To: Jonathan King
Subject: Leave Request Submitted: Meeting in Denver

Dear Jonathan King,

A Leave Request has been submitted and now requires a decision by you. Please review the following details, then reply to this email with one of the decision

We are going to action the Leave Request using a technology called SmartActions. SmartActions allow the user to action a task by simply replying to the email, then entering one of the action options as the message body. K2 will 'read' the action and move the workflow along accordingly.

c. **Reply** to the email with either **Approved** or **Rejected** as the message body. Click **Send**, then observe the **View Flow** Report update. (The timing for the update will vary, depending on your environment.)



Pop Out ✕ Discard

Send

To... K2 Service

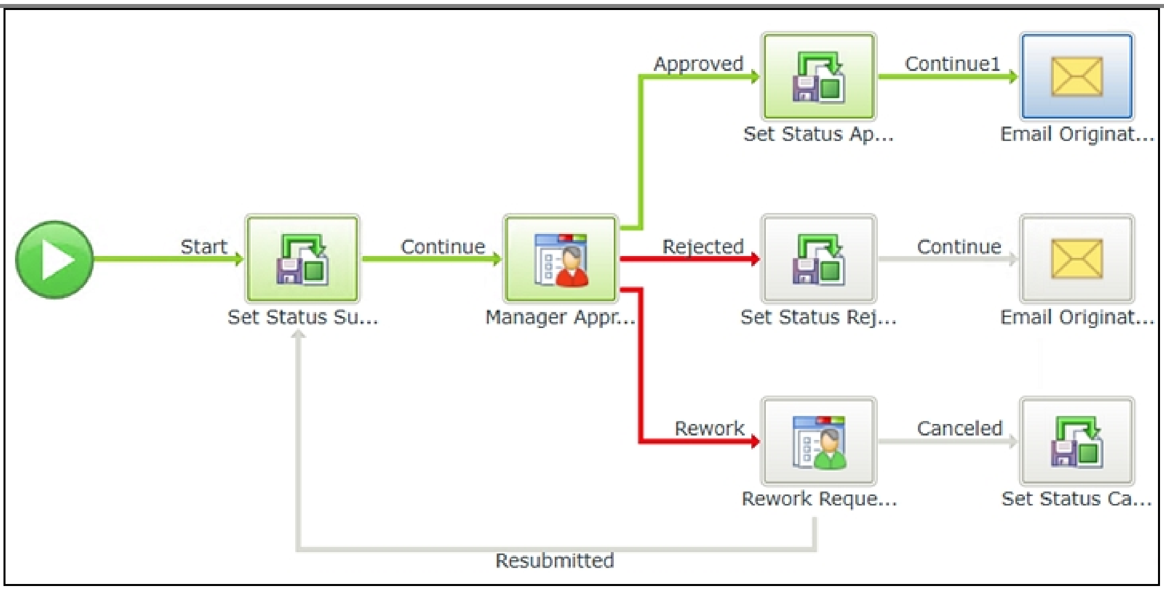
Cc...

Subject RE: Leave Request Submitted: Meeting in Chicago

Approved

From: K2 Service
Sent: Wednesday, May 6, 2015 11:40 AM

The View Flow Report after it has updated.

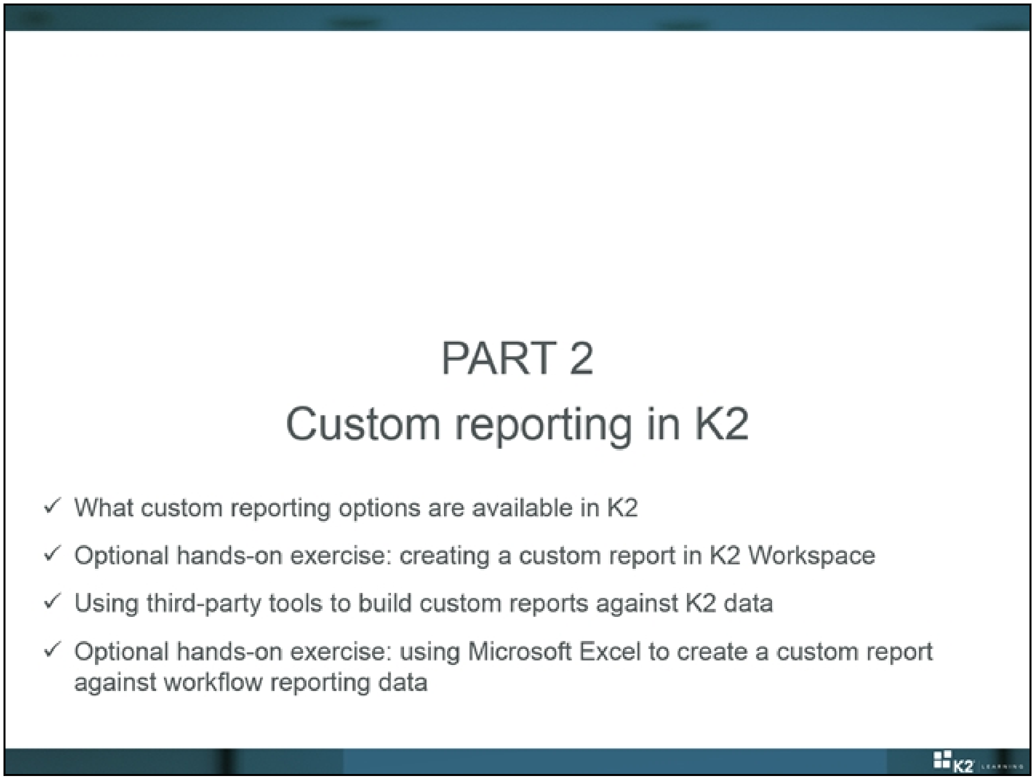


STEP 6 REVIEW

In this step, we observed in real-time how the View Flow Report updates after a task has been processed. In our example, we actioned a Leave Request using SmartActions. This report can be useful for troubleshooting as you can observe the system and user tasks as they are processed, giving you immediate feedback as to your workflow's efficiency.

This concludes the View Flow Reports tutorial.

Part 2: Custom reporting in K2



PART 2

Custom reporting in K2

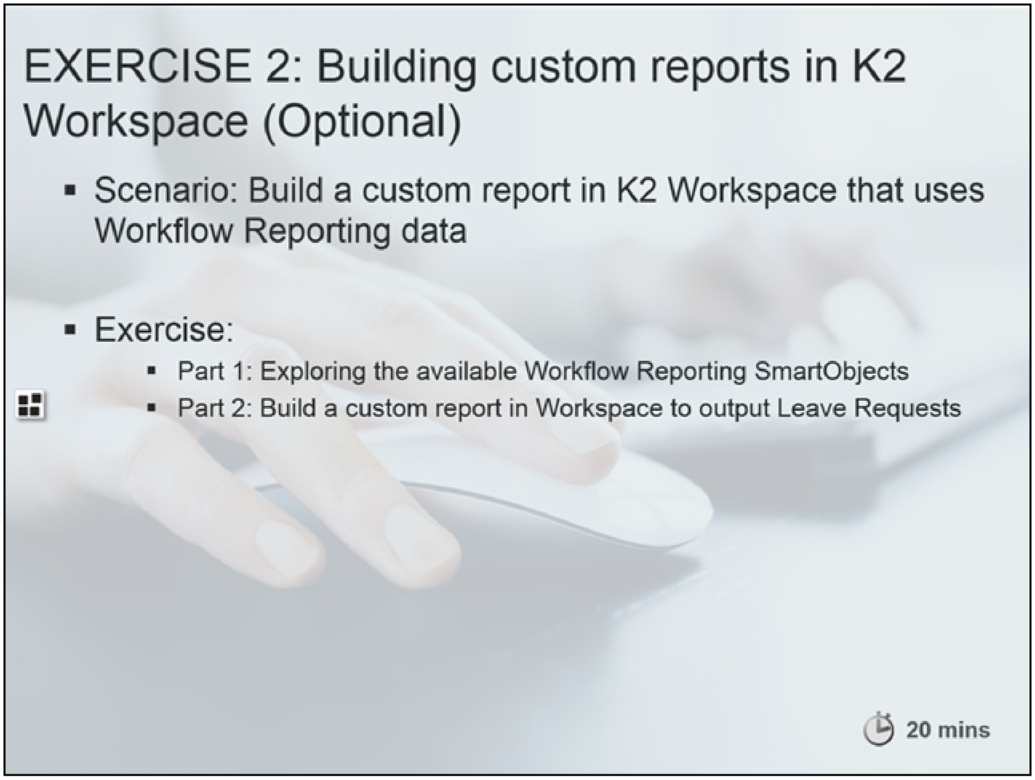
- ✓ What custom reporting options are available in K2
- ✓ Optional hands-on exercise: creating a custom report in K2 Workspace
- ✓ Using third-party tools to build custom reports against K2 data
- ✓ Optional hands-on exercise: using Microsoft Excel to create a custom report against workflow reporting data

Video

In Part 2 we will look at the custom reporting options that are available in K2: what tools K2 provides for building custom reports (followed by an optional exercise to build a custom report in K2 Workspace) and how to use third-party tools to build custom reports against K2 data (followed by an optional exercise to build a custom report with Microsoft Excel).

Note
If your organization will not be utilizing any custom reporting on K2, you may skip Part 2 of this course module.

EXERCISE 2: Building custom reports in K2 Workspace (Optional)



EXERCISE 2: Building custom reports in K2 Workspace (Optional)

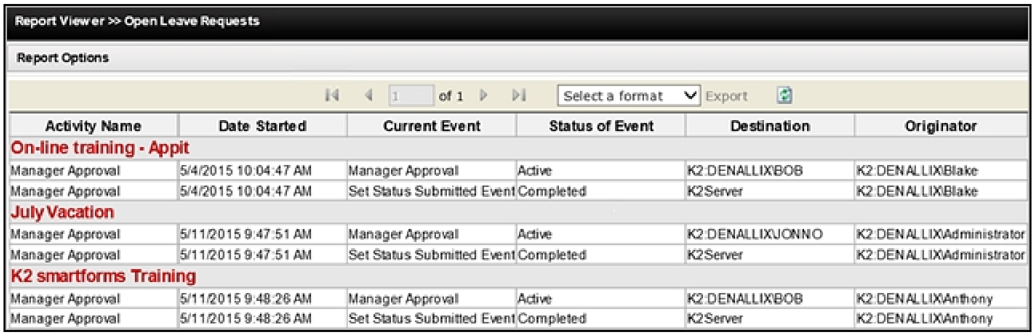
- Scenario: Build a custom report in K2 Workspace that uses Workflow Reporting data
- Exercise:
  - Part 1: Exploring the available Workflow Reporting SmartObjects
  - Part 2: Build a custom report in Workspace to output Leave Requests

20 mins

In this exercise, you will use the Report Designer in K2 Workspace to build a simple tabular report of workflow reporting data. There are two parts to this exercise. In Part 1, you will use the SmartObject Service Tester utility to explore the available workflow reporting SmartObjects. In Part 2, you will use the Report Designer in K2 Workspace to build a custom tabular report of workflow reporting data to output all of the activities for each instance of the Leave Request workflow.

> **Note**
> This exercise assumes that you have completed the Leave Request Approval application from the *100BHX: Introduction to K2 Applications with K2 Designer* learning module. If you have not, you can follow the same steps to build custom reports against another workflow in your environment, but the report data and screenshots will look different.

Ultimately, we will be producing a report like the one below. This report lists the instances of the Leave Request Approval workflow and the activities in those instances. Although the data in this report may not be that compelling, the principles of how you build a custom report are the same regardless of the SmartObject being used.



Report Viewer >> Open Leave Requests

Report Options

Activity Name	Date Started	Current Event	Status of Event	Destination	Originator
On-line training - Appit					
Manager Approval	5/4/2015 10:04:47 AM	Manager Approval	Active	K2:DENALLIX\BOB	K2:DENALLIX\Blake
Manager Approval	5/4/2015 10:04:47 AM	Set Status Submitted Event	Completed	K2Server	K2:DENALLIX\Blake
July Vacation					
Manager Approval	5/11/2015 9:47:51 AM	Manager Approval	Active	K2:DENALLIX\JONNO	K2:DENALLIX\Administrator
Manager Approval	5/11/2015 9:47:51 AM	Set Status Submitted Event	Completed	K2Server	K2:DENALLIX\Administrator
K2 smartforms Training					
Manager Approval	5/11/2015 9:48:26 AM	Manager Approval	Active	K2:DENALLIX\BOB	K2:DENALLIX\Anthony
Manager Approval	5/11/2015 9:48:26 AM	Set Status Submitted Event	Completed	K2Server	K2:DENALLIX\Anthony

When you are ready, continue on to the Exploring the Workflow Reporting SmartObjects exercise to learn what workflow reporting SmartObjects are available, and then continue with the Building custom reports in K2 Workspace exercise to build a custom report with K2 Workspace.

Exploring the Workflow Reporting SmartObjects

K2 Reports are exposed as Workflow Reporting SmartObjects that are installed by default in K2 blackpearl installations. Because they are SmartObjects, they can be used in multiple places, including Standard and Custom Reports. The same permissions required for Standard Reports are also required for Custom Reports. At a minimum, users require Start and View rights for Processes they wish to report on. Users with K2 Server Admin rights have full-control over reporting tools.

> **Note**
> The steps and screenshots in this tutorial are based on the Leave Approval (Extended Version) application. You can, however, use these same steps and instructions to report on other workflows in your K2 environment

Step 1: View the Workflow Reporting SmartObjects using SmartObject Tester

In this step, we are going to open the SmartObject Tester and observe the structure of the Workflow Reporting SmartObjects.

Step 1 Tasks

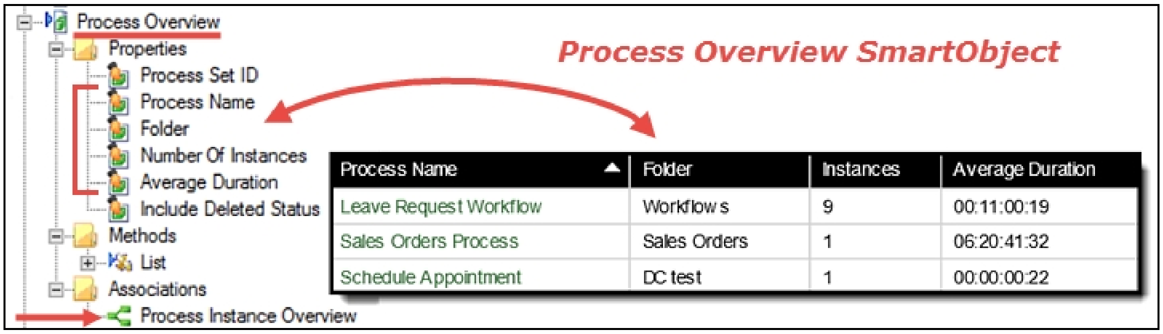
1. Launch the **SmartObject Tester** tool.
   ("C:\Program Files (x86)\K2 blackpearl\Bin\SmartObject Service Tester.exe")
2. Navigate to the **Process Overview** SmartObject found under the Workflow Reports > Workflow General nodes. **Execute** the **Process Overview SmartObject** and observe the results returned. (Make a note of one of the **Process Names** that are returned.)
3. **Execute** the **Process Instance SmartObject** using the Process Name noted above as the Process Name parameter. Observe the results returned.

Step 1 Walkthrough

a. Launch the **SmartObject Tester** tool.
   ("C:\Program Files (x86)\K2 blackpearl\Bin\SmartObject Service Tester.exe")

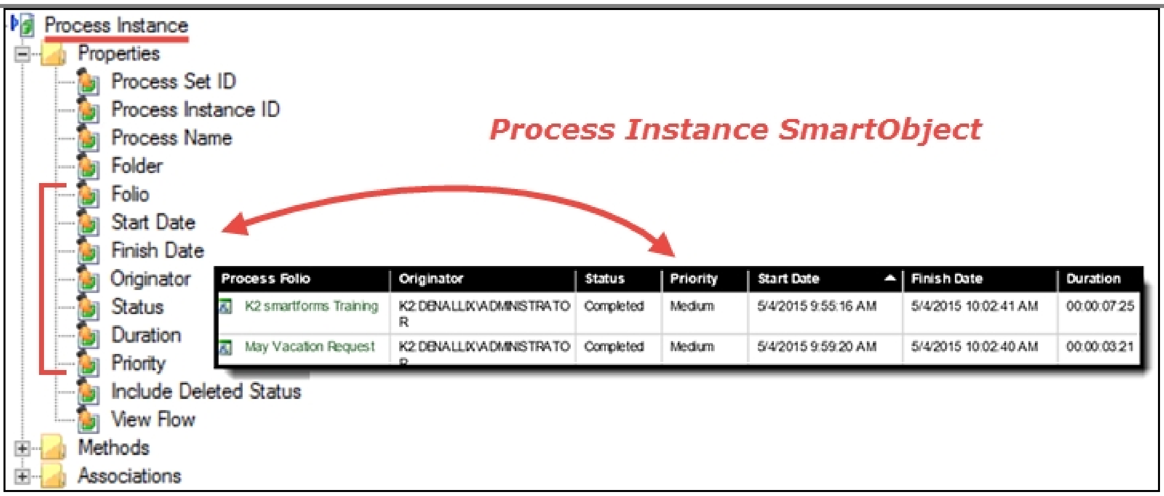
b. Expand **SmartObject Explorer** > **Workflow Reports** > **Workflow General** > **Process Overview**.

The Process Overview SmartObject is the basis for the Process Overview Report we worked with in the Standard Reports tutorial. Notice the SmartObject Properties that are the Report columns. The Process Overview Report is simply a representation of the objects found in its SmartObject. Notice too, this SmartObject has an Association called *Process Instance Overview*. If we look at the Association properties, we see that the referenced SmartObject is Process Instance.



Process Name	Folder	Instances	Average Duration
Leave Request Workflow	Workflow s	9	00:11:00:19
Sales Orders Process	Sales Orders	1	06:20:41:32
Schedule Appointment	DC test	1	00:00:00:22

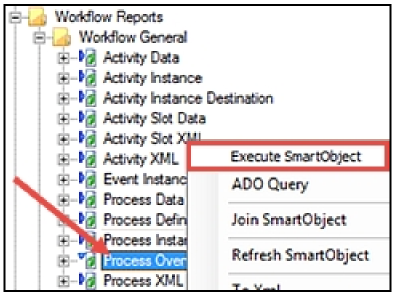
Expand the **Process Instance** SmartObject. In the Process Overview Report > Process Instances view, we now see the SmartObject Properties as object (columns) for this report.



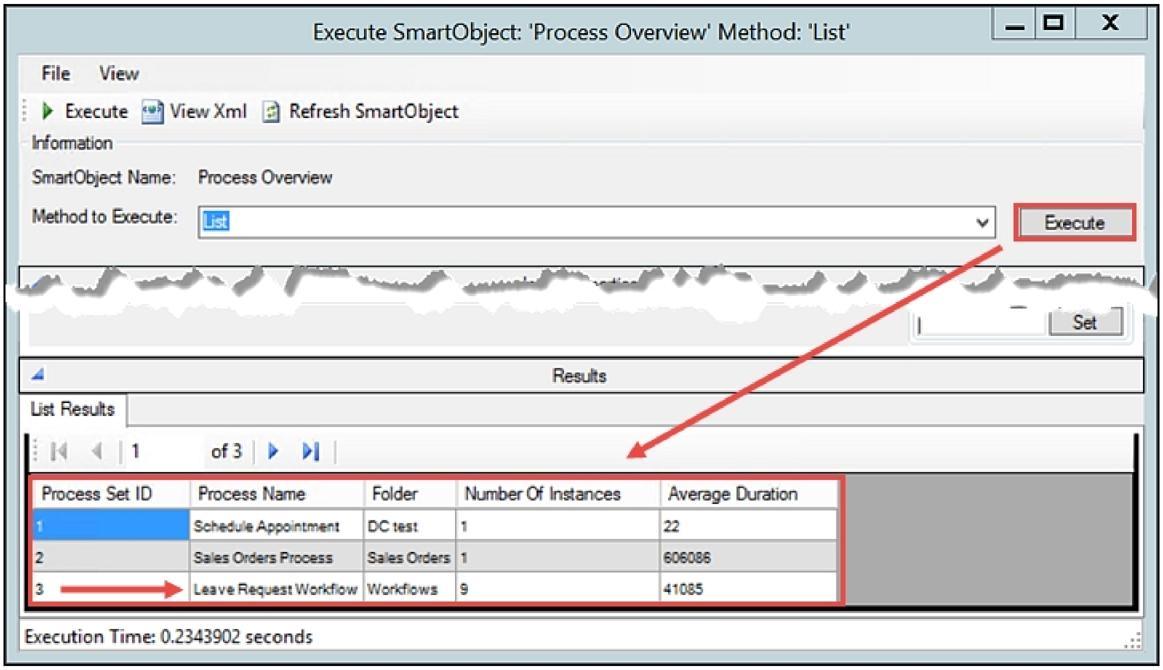
Process Folio	Originator	Status	Priority	Start Date	Finish Date	Duration
K2 smartforms Training	K2 DENALLIX\ADMINISTRATOR	Completed	Medium	5/4/2015 9:55:16 AM	5/4/2015 10:02:41 AM	00:00:07:25
May Vacation Request	K2 DENALLIX\ADMINISTRATOR	Completed	Medium	5/4/2015 9:59:20 AM	5/4/2015 10:02:40 AM	00:00:03:21

Now we will demonstrate the SmartObject actually retrieving report data from the K2 database.

c. Right-click the **Process Overview** SmartObject and select **Execute SmartObject**.



d. Click the **Execute** button. Notice the Results that are returned at the bottom of the screen. All of the deployed Processes (workflows) are listed. Notice the **Leave Request Workflow** Process and in particular, the **Process Set ID (3)** for the Leave Request Workflow. In the next step, we will Execute the *Process Instance* SmartObject, with the Leave Request Workflow as a parameter. **Exit** the Execute SmartObject screen.



SmartObject Name: Process Overview

Method to Execute: List

Process Set ID	Process Name	Folder	Number Of Instances	Average Duration
1	Schedule Appointment	DC test	1	22
2	Sales Orders Process	Sales Orders	1	606086
3	Leave Request Workflow	Workflows	9	41085

Execution Time: 0.2343902 seconds

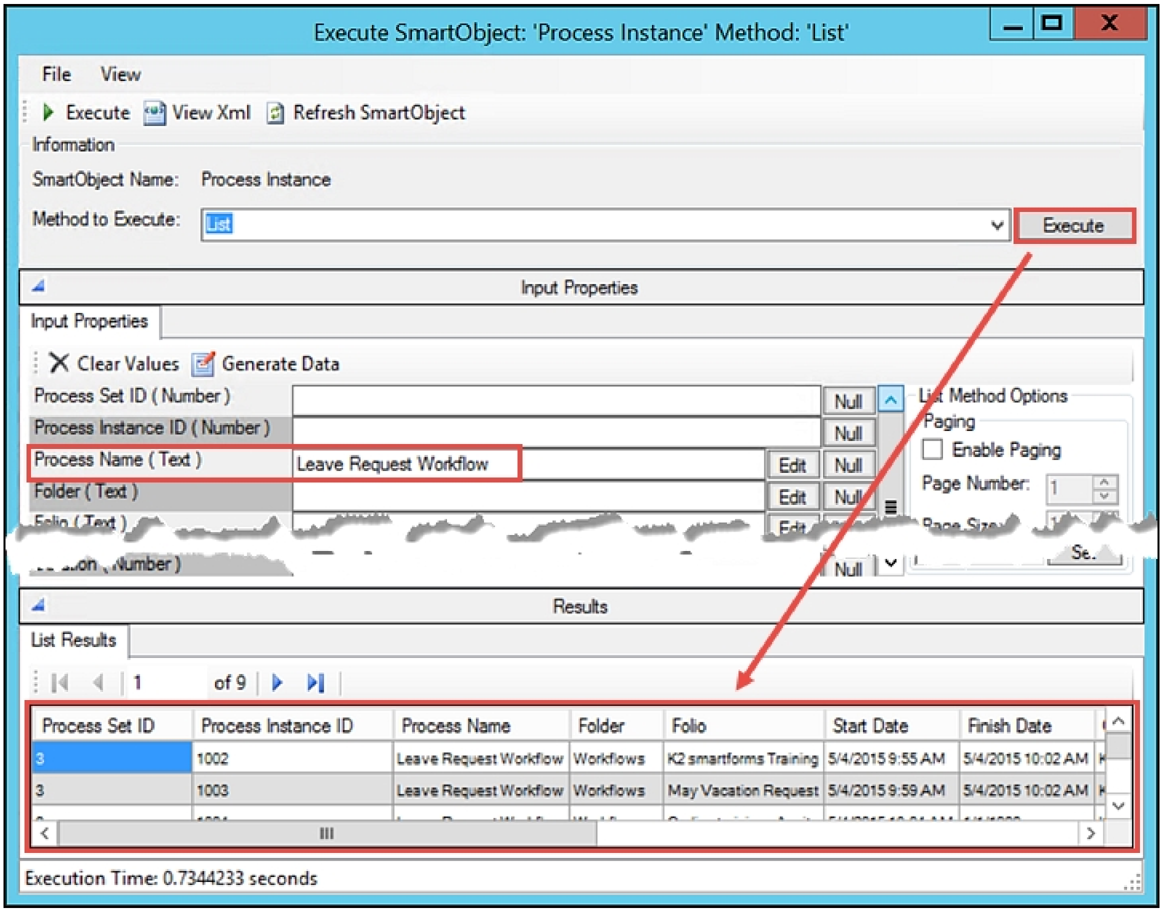
e. Right-click the **Process Instance** SmartObject and select **Execute SmartObject**. Enter the following parameter for the **Process Name**:

Leave Request Workflow

(Or, any of the other Process Names that you observed in the previous step.) Click **Execute**.

Now the results display all of the Process Instances for the Leave Request Workflow Process. Notice again, the **Process Set ID** is the same for each record, and for the Process Overview we executed in the previous step. The Process Set ID is the common factor between both sets of records and is how they are associated with each other.

If you are familiar with queries and JOIN statements, the associated SmartObject concept should be relatively straightforward. If you are not familiar, just know that K2 reports are actually the contents of the Workflow Reporting SmartObjects that are associated with each other to create a parent/child set of records.



SmartObject Name: Process Instance

Method to Execute: List

Process Name (Text): Leave Request Workflow

Process Set ID	Process Instance ID	Process Name	Folder	Folio	Start Date	Finish Date
3	1002	Leave Request Workflow	Workflows	K2 smartforms Training	5/4/2015 9:55 AM	5/4/2015 10:02 AM
3	1003	Leave Request Workflow	Workflows	May Vacation Request	5/4/2015 9:59 AM	5/4/2015 10:02 AM

Execution Time: 0.7344233 seconds

f. **Exit** the Execute SmartObject screen. **Exit** the SmartObject Tester.

STEP 1 REVIEW

In this step we demonstrated how K2 reports are actually just an execution of Workflow Reporting SmartObjects. The SmartObject Tester allows us to see the properties, methods and associations for the SmartObjects. The parent/child relationships that are formed from the associations translate into the Standard Report drill-down functionality. For Custom Reports, we will pull properties from several associated SmartObjects to build our report.

This concludes exploring the Reporting SmartObjects tutorial. .

Building custom reports in K2 Workspace

In the Standard Reports tutorial, we explored the out-of-the-box reports available with K2 blackpearl installations. In this tutorial, we will look at the Report Designer in K2 Workspace and build our own custom report. This report will be based off of the Activity Instance SmartObject (you can refer back to the Exploring the Workflow Reporting SmartObjects to learn more about the available reporting SmartObjects). This SmartObject has a number of associations that allow for a report that's flexible with a wide variety of reporting options.

Note
The steps and screenshots in this tutorial are based on the Leave Approval (Extended Version) application. You can, however, use these same steps and instructions to report on other workflows in your K2 environment

Step 1: Create a Custom Report with Report Designer

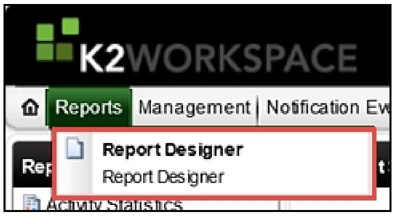
In this step, we will create a custom report, choosing our fields from the Reporting SmartObjects. We will customize the report layout, rename the column titles, add two filters and change the font style.

Step 1 Tasks

1. Using **Report Designer** in K2 Workspace, create a custom **Summary** report and name it *Open Leave Requests*.
2. Select **Workflow Reports** > **Workflow General** > **Activity Instance** as the Primary Data Source.
3. After the Activity Instance SmartObject has been added, adjust the **Related Data** properties as follows:
    - For the **Activity Instance** SmartObject, CHECK (only) the following properties:
        - **Activity Name**, **Start Date**, **Status**
    - Locate the **Event Instance** SmartObject and CHECK the following properties:
        - **Destination**, **Event Name**, **Status**
    - Locate the **Process Instance** SmartObject and CHECK the following properties:
        - **Folio**, **Originator**, **Process Name**
4. Add the following data source properties for the report **Columns**:
    - **Activity Instance** > **Activity Name**, **Activity Instance** > **Start Date**
    - **Event Instance** > **Event Name**, **Event Instance** > **Status**, **Event Instance** > **Destination**
    - **Process Instance** > **Originator**
5. Rename each column with user-friendly names. Use the names below as an option.
    - **Activity Name** > **Date Started** > **Current Event** > **Status of Event** > **Destination** > **Originator**
6. Group the report content by the **Folio**.
7. Add two **Filters**. The first should be the **Process Name (Equals)**
   Leave Request Workflow
   then the **Activity Instance Status (Equals)**
   Active
8. Edit the **'Group By' style** so that the Font is larger, bold and a different color.
9. View the Report, then **Export** the report content to **Excel**.

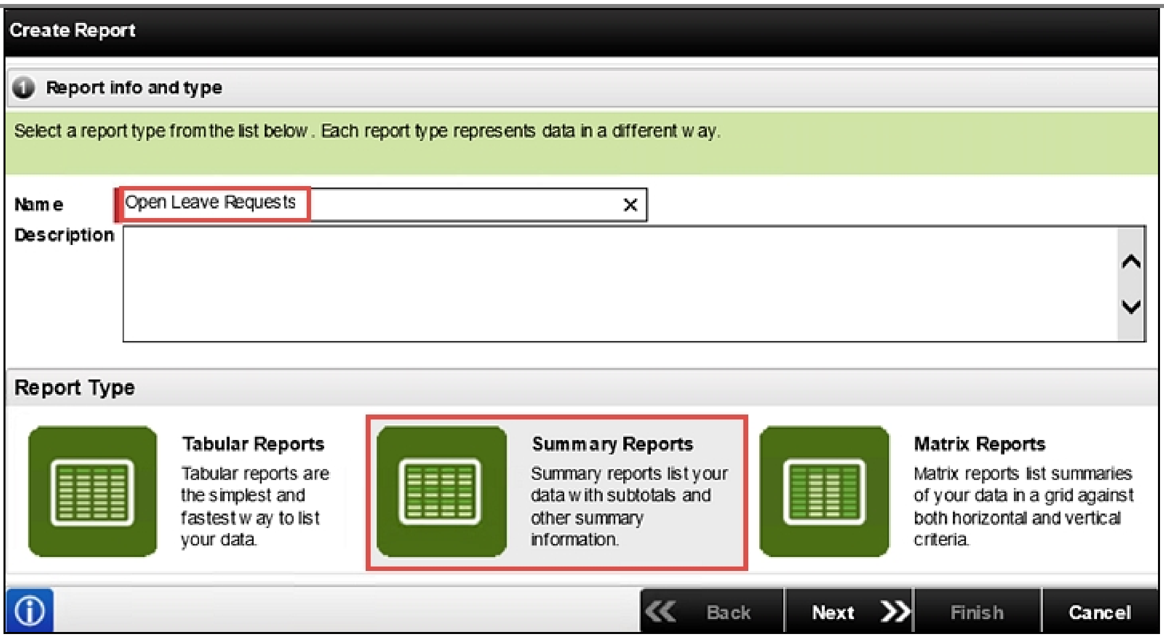
Step 1 Walkthrough

a. Launch **K2 Workspace** (**Start** > **All Programs** > **K2 blackpearl** > **K2 blackpearl Workspace**). Open the **Report Designer**.



b. When the Report Designer loads, click the **Create Report** button found in the lower right corner of the screen.

c. On the **Report info and type** screen, **Name** the report
Open Leave Requests
then select the **Summary Reports** option. Click **Next**.



Create Report

Report info and type

Select a report type from the list below. Each report type represents data in a different w ay.

Nam e: Open Leave Requests

Description:

Report Type

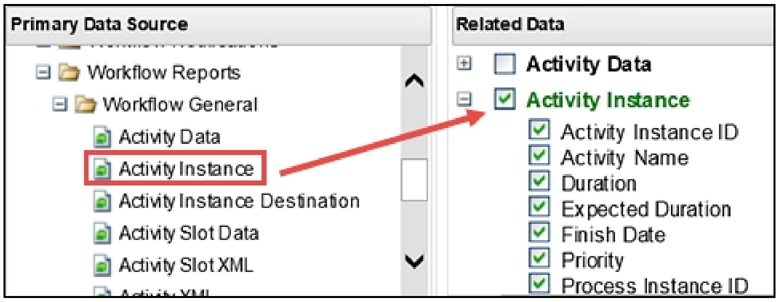
Tabular Reports – Tabular reports are the simplest and fastest w ay to list your data.

Summary Reports – Summary reports list your data w ith subtotals and other summary information.

Matrix Reports – Matrix reports list summaries of your data in a grid against both horizontal and vertical criteria.

Reports are built from SmartObjects, or data sources. Notice the left column that shows all of the Primary Data Source options available. We are going to build our report from the Activity Instance data source, but you could build a report based off of the Leave Request SmartObject, found under the K2 Learning category if you wanted to. Notice too, that the reporting data source options are identical to what we observed with the SmartObject Tester.

d. Scroll to the bottom of the data source options and expand the **Workflow Reports** category, then **Workflow General**. Click the **Activity Instance** SmartObject to add it to the Related Data pane.



Scroll through the Related Data pane and notice how K2 has added a number of SmartObjects in addition to the Activity Instance SmartObject. These are all *associated* SmartObjects of Activity Instance. These associations allow you a great deal of flexibility in building your custom report. Notice too, how K2 has automatically selected every property in the Activity Instance SmartObject by default. We are going to adjust which properties we want, plus we are going to add a few other properties found in associated SmartObjects. Behind the scenes, K2 takes care of the SmartObject associations, allowing us to choose any property found in the Related Data SmartObjects for our report.

e. For the **Activity Instance** SmartObject, CHECK (only) the following properties:

- **Activity Name**, **Start Date**, **Status**

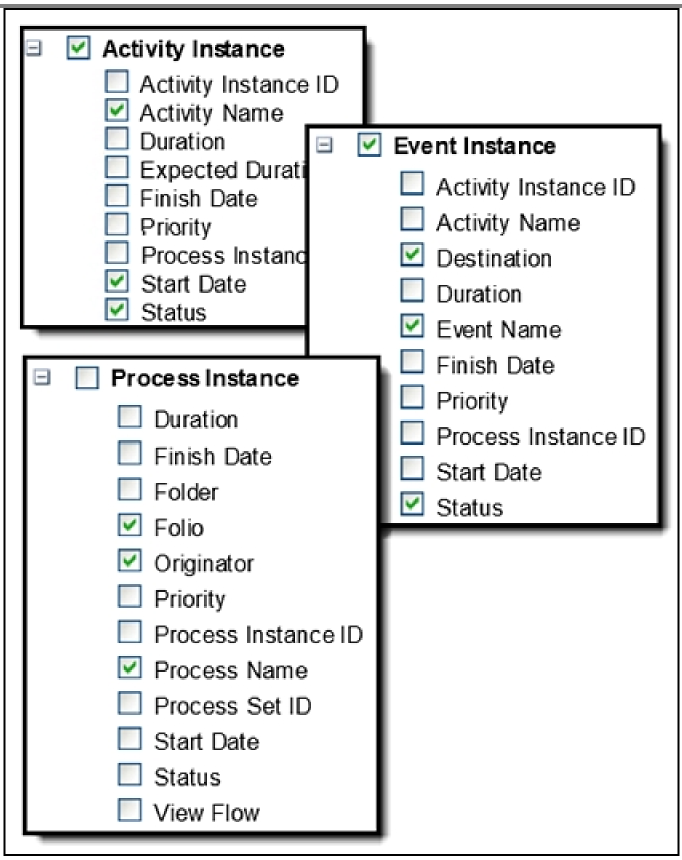
f. Locate the **Event Instance** SmartObject and CHECK the following properties:

- **Destination**, **Event Name**, **Status**

g. Locate the **Process Instance** SmartObject and CHECK the following properties:

- **Folio**, **Originator**, **Process Name**

Your Related Data selections should reflect the image below. Click **Next** to continue.



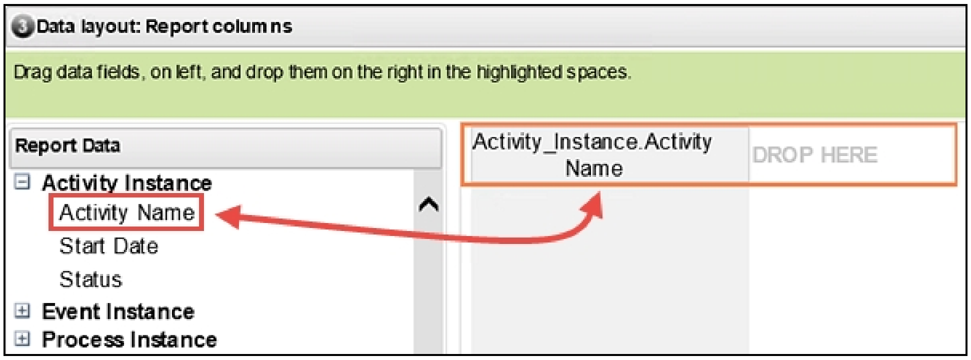
⊟ ☑ **Activity Instance**
- ☐ Activity Instance ID
- ☑ Activity Name
- ☐ Duration
- ☐ Expected Durati
- ☐ Finish Date
- ☐ Priority
- ☐ Process Instanc
- ☑ Start Date
- ☑ Status

⊟ ☑ **Event Instance**
- ☐ Activity Instance ID
- ☐ Activity Name
- ☑ Destination
- ☐ Duration
- ☑ Event Name
- ☐ Finish Date
- ☐ Priority
- ☐ Process Instance ID
- ☐ Start Date
- ☑ Status

⊟ ☐ **Process Instance**
- ☐ Duration
- ☐ Finish Date
- ☐ Folder
- ☑ Folio
- ☑ Originator
- ☐ Priority
- ☐ Process Instance ID
- ☑ Process Name
- ☐ Process Set ID
- ☐ Start Date
- ☐ Status
- ☐ View Flow

On the **Data layout: Report columns** screen, we are going to add the SmartObject properties that we want for our report columns. We will simply drag and drop the properties onto the design canvas, then rename the column titles so that they are user friendly.

h. Expand the **Activity Instance** SmartObject, then drag the **Activity Name** into the first column (**DROP HERE**) box. After you drop the property into its place, notice that new column box appears.



Data layout: Report columns

Drag data fields, on left, and drop them on the right in the highlighted spaces.

Report Data

⊟ **Activity Instance**
- Activity Name
- Start Date
- Status

⊞ **Event Instance**

⊞ **Process Instance**

Activity_Instance.Activity Name	DROP HERE

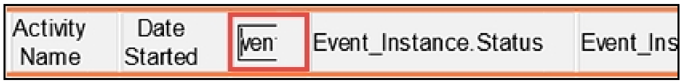
i. Continue dragging properties until you have all of the following properties represented as columns in your report:

- **Activity Instance** > **Activity Name**, **Activity Instance** > **Start Date**
- **Event Instance** > **Event Name**, **Event Instance** > **Status**, **Event Instance** > **Destination**
- **Process Instance** > **Originator**

Notice the column names are not user friendly. In this next step, we will take a minute to edit each column name so that they are in a format that is easy to read and understand.

j. Double-click the first column name (**Activity_Instance.Activity Name**) and using your keyboard **Delete** button, delete the default name. When the name is deleted, enter

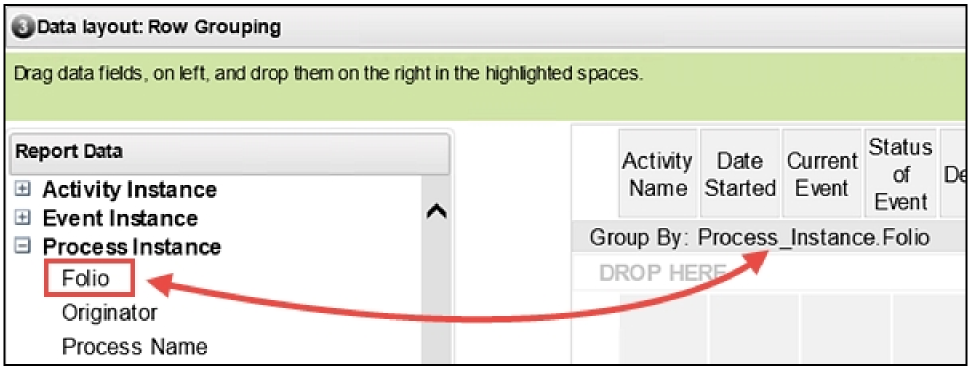
Activity Name
in its place.



k. Repeat this process for each of the column titles until you have the following column titles for your report:
Activity Name > **Date Started** > **Current Event** > **Status of Event** > **Destination** > **Originator**
Click **Next** to continue.

Now, we are going to group our report results by the Folio. Recall that the Folio is the Leave Request Title (that the form originator entered). This will group all of the Activity Instance results under their Folio name.

l. On the **Data layout: Row Grouping** screen, drag the **Process Instance** > **Folio** property into the **DROP HERE** box on the design canvas. Click **Next** to continue.

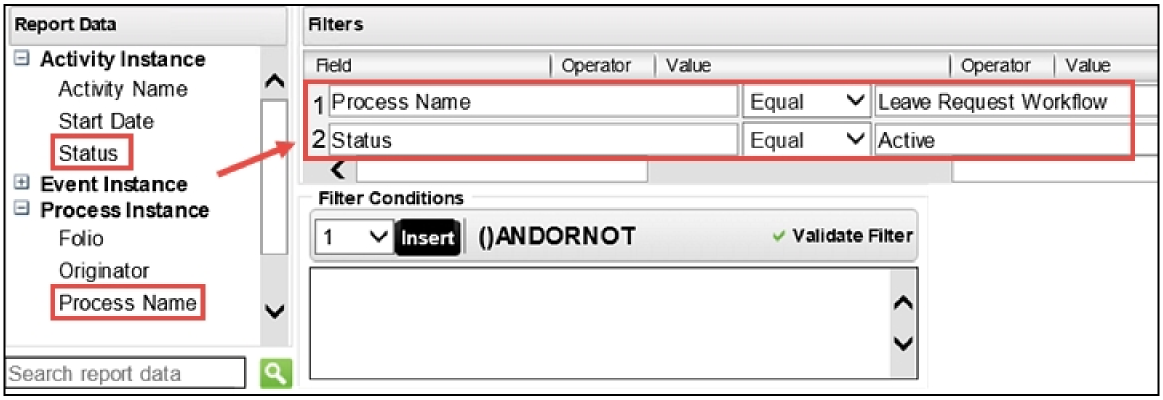


m. We will not have any **Summary Filters** for this report, so click **Next**.

At this point, our report will return all Activity Instance data that has been saved to the K2 database. We just want to see the data from the Leave Request Workflow Activity Instances, and we only want to see the data from Activity Instances that are still Active. To achieve this, we will apply two filters to our report.

n. On the **Filter the data** screen, expand the **Process Instance** category and drag the **Process Name** property into the **DROP HERE** Field box. Leave the Operator with its default Equal value, then enter
Leave Request Workflow
for the Value. (You may have to scroll to the right depending on your screen size.)

o. Add a second filter by dragging the **Activity Instance > Status** property into the second line. Keep the default (Equal) Operator and for the **Value**, enter
Active
Click **Next** to continue.

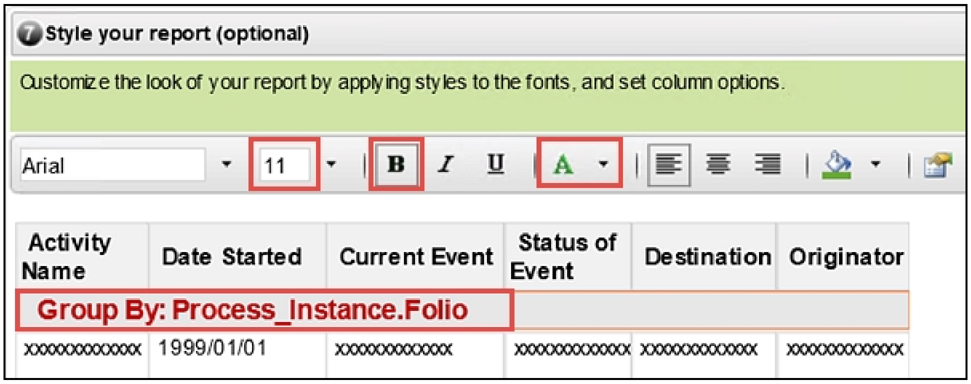


	Field	Operator	Value
1	Process Name	Equal	Leave Request Workflow
2	Status	Equal	Active

p. There are no **Parameters** for this report, so click **Next** to continue.

The final adjustment we want to make for our report is to edit the Group By: Process_Instance Folio font so that it stands out and is easier to read.

q. On the **Style your report** screen, click the **Group By** title to highlight it. Then adjust the **Font size to 11pt**, make the **Font weight bold**, then change the **Font color to dark red**. Click **Next** to continue.

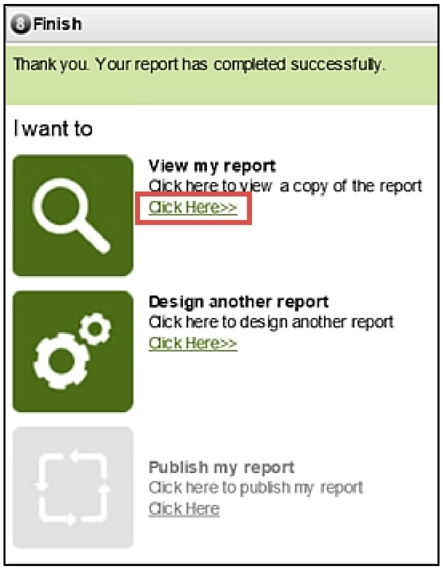


7 Style your report (optional)

Customize the look of your report by applying styles to the fonts, and set column options.

Activity Name	Date Started	Current Event	Status of Event	Destination	Originator
Group By: Process_Instance.Folio					
xxxxxxxxxxxxx	1999/01/01	xxxxxxxxxxxxx	xxxxxxxxxxxxx	xxxxxxxxxxxxx	xxxxxxxxxxxxx

r. We are now ready to view our Custom Report. On the **Finish** screen, click the **View my report** link to run the report.



8 Finish

Thank you. Your report has completed successfully.

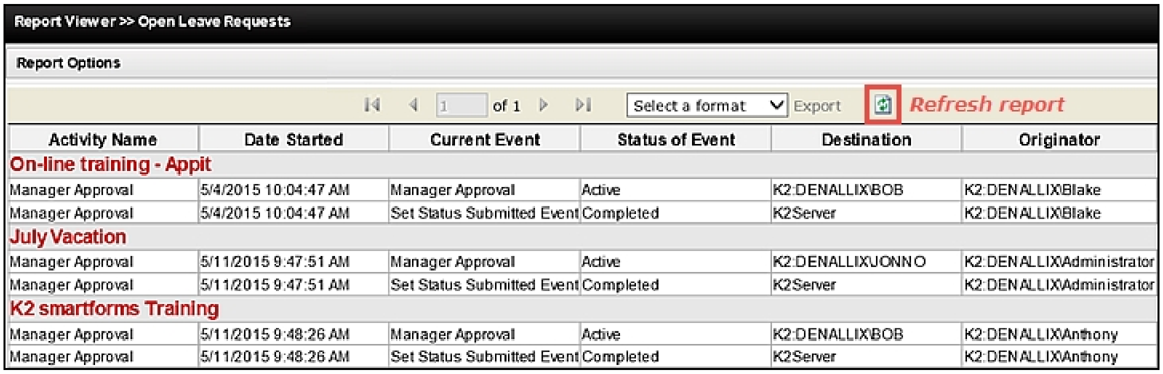
I want to

View my report
Click here to view a copy of the report
Click Here>>

Design another report
Click here to design another report
Click Here>>

Publish my report
Click here to publish my report
Click Here

After your custom report is generated, you should see results similar to the image below. (If you do not have any results, or very few results, consider submitting a few more Leave Request Forms (without taking any action on them), then refresh the report by clicking the Refresh icon just right of the 'Select a format' Export option.)

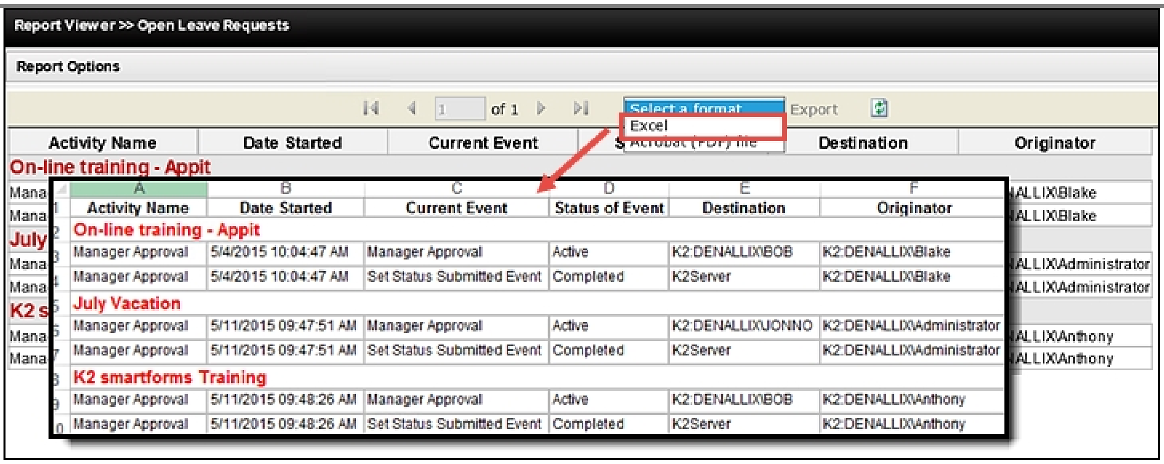


Report Viewer >> Open Leave Requests

Report Options

Activity Name	Date Started	Current Event	Status of Event	Destination	Originator
On-line training - Appit					
Manager Approval	5/4/2015 10:04:47 AM	Manager Approval	Active	K2:DENALLIX\BOB	K2:DENALLIX\Blake
Manager Approval	5/4/2015 10:04:47 AM	Set Status Submitted Event	Completed	K2Server	K2:DENALLIX\Blake
July Vacation					
Manager Approval	5/11/2015 9:47:51 AM	Manager Approval	Active	K2:DENALLIX\JONNO	K2:DENALLIX\Administrator
Manager Approval	5/11/2015 9:47:51 AM	Set Status Submitted Event	Completed	K2Server	K2:DENALLIX\Administrator
K2 smartforms Training					
Manager Approval	5/11/2015 9:48:26 AM	Manager Approval	Active	K2:DENALLIX\BOB	K2:DENALLIX\Anthony
Manager Approval	5/11/2015 9:48:26 AM	Set Status Submitted Event	Completed	K2Server	K2:DENALLIX\Anthony

s. In the Standard Reports tutorial, we exported the Process Overview Report to Excel. You can export custom reports as well to Excel or in a PDF file format. This allows you to combine reporting data over time, or share reports with your users. From the **Report Viewer** > **Open Leave Requests** screen, **Export** this report to **Excel**.

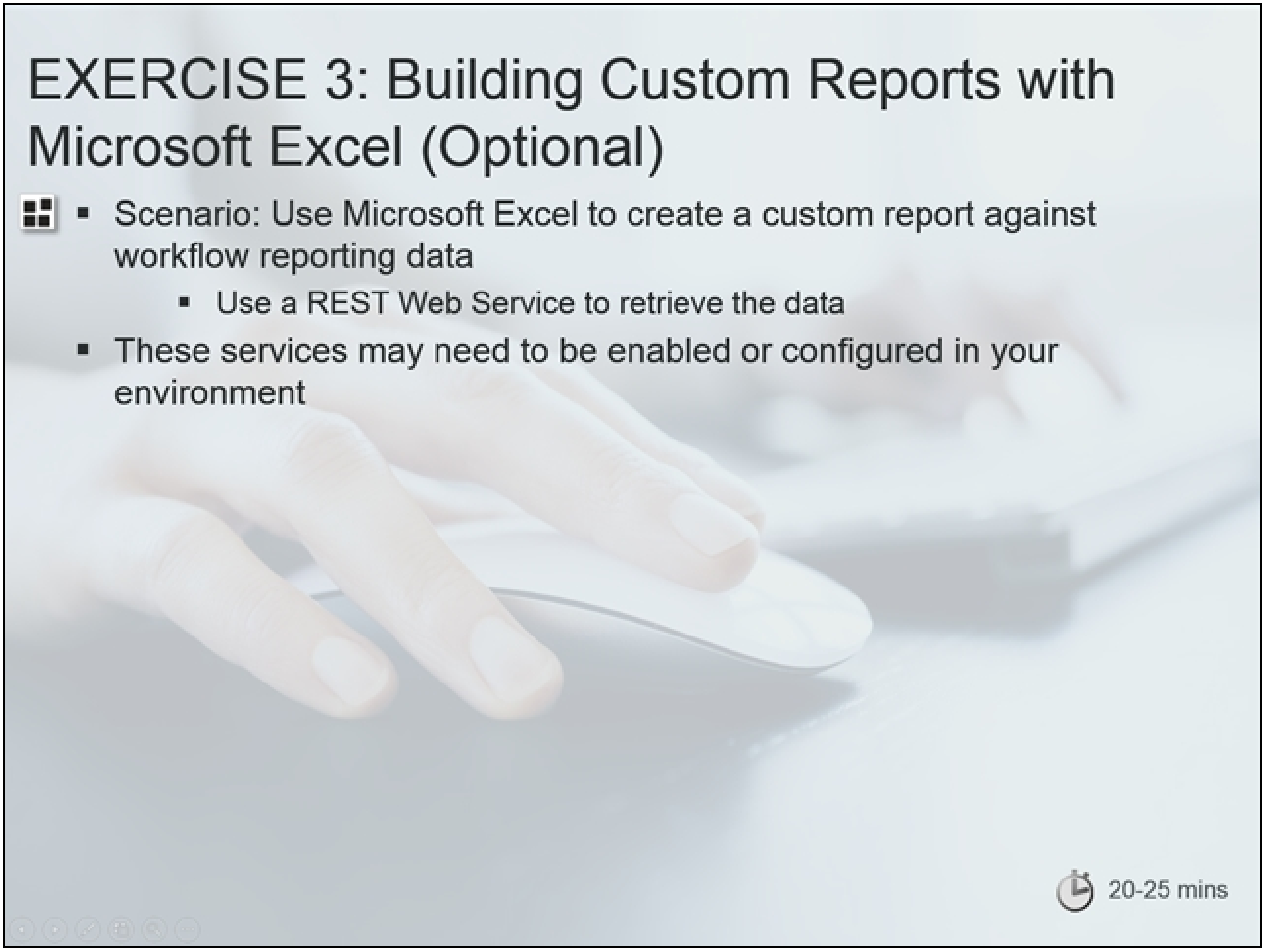


Activity Name	Date Started	Current Event	Status of Event	Destination	Originator
On-line training - Appit					
Manager Approval	5/4/2015 10:04:47 AM	Manager Approval	Active	K2:DENALLIX\BOB	K2:DENALLIX\Blake
Manager Approval	5/4/2015 10:04:47 AM	Set Status Submitted Event	Completed	K2Server	K2:DENALLIX\Blake
July Vacation					
Manager Approval	5/11/2015 09:47:51 AM	Manager Approval	Active	K2:DENALLIX\JONNO	K2:DENALLIX\Administrator
Manager Approval	5/11/2015 09:47:51 AM	Set Status Submitted Event	Completed	K2Server	K2:DENALLIX\Administrator
K2 smartforms Training					
Manager Approval	5/11/2015 09:48:26 AM	Manager Approval	Active	K2:DENALLIX\BOB	K2:DENALLIX\Anthony
Manager Approval	5/11/2015 09:48:26 AM	Set Status Submitted Event	Completed	K2Server	K2:DENALLIX\Anthony

STEP 1 REVIEW

In this step, we used Report Designer to create and style a custom K2 report. We added the Activity Instance SmartObject as the primary data source for our report, then observed how K2 automatically added any SmartObjects that have associations with Activity Instance. Behind the scenes, K2 maintains the SmartObject associations which allow us to simply pick from multiple SmartObject properties to add to our report. We included an option to group our results, using the Folio as the grouping property. Adding Filters allowed us to narrow our reporting results to just one Process and finally, we included some simple Font formatting to make our report stand out. As you build reports in your own environment, you can save them for reuse later.

EXERCISE 3: Building Custom Reports with Microsoft Excel (Optional)



EXERCISE 3: Building Custom Reports with Microsoft Excel (Optional)

- Scenario: Use Microsoft Excel to create a custom report against workflow reporting data
  - Use a REST Web Service to retrieve the data
- These services may need to be enabled or configured in your environment

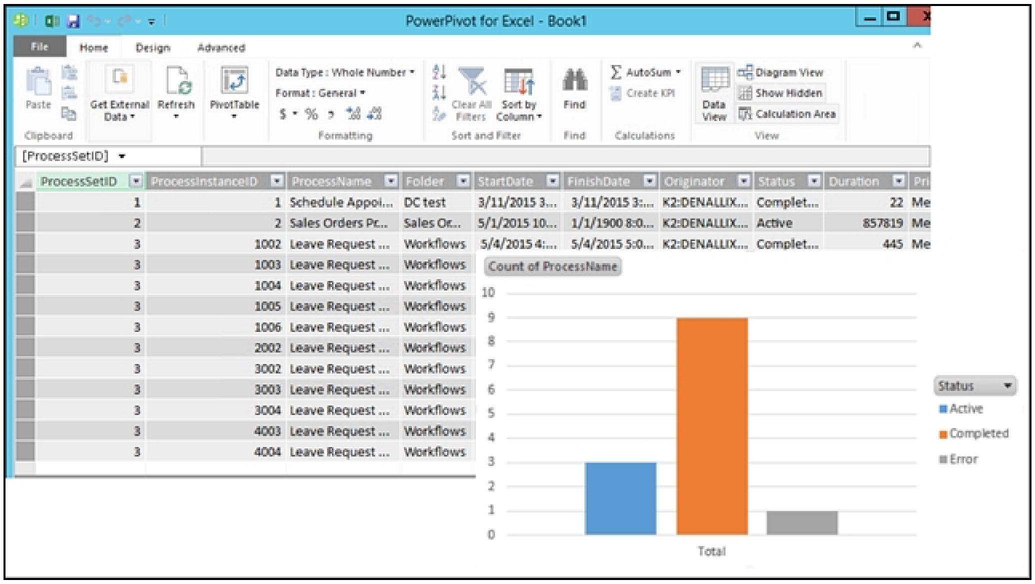
20-25 mins

Video

In this exercise, you will build a custom K2 report in Excel using PowerPivot and a Pivot Chart. (This exercise is optional and not required if you will not use Microsoft Excel or similar third-party reporting tools.)

We will be using the SmartObject REST service endpoints to retrieve the workflow reporting data. As part of the exercise you will configure (by editing a K2 configuration file) and test the service before using it in Microsoft Excel.

The resulting report will show the status for all workflows in the environment, and below is an example of the report we will create. As with the previous exercise, even though this data may not be particularly interesting, the concepts and principles apply equally to any other SmartObject you may wish to use as the source of reporting data.



ProcessSetID	ProcessInstanceID	ProcessName	Folder	StartDate	FinishDate	Originator	Status	Duration	Pri
1	1	Schedule Appoi...	DC test	3/11/2015 3...	3/11/2015 3:...	K2:DENALLIX...	Complet...	22	Me
2	2	Sales Orders Pr...	Sales Or...	5/1/2015 10...	1/1/1900 8:0...	K2:DENALLIX...	Active	857819	Me
3	1002	Leave Request ...	Workflows	5/4/2015 4:...	5/4/2015 5:0...	K2:DENALLIX...	Complet...	445	Me
3	1003	Leave Request ...	Workflows						
3	1004	Leave Request ...	Workflows						
3	1005	Leave Request ...	Workflows						
3	1006	Leave Request ...	Workflows						
3	2002	Leave Request ...	Workflows						
3	3002	Leave Request ...	Workflows						
3	3003	Leave Request ...	Workflows						
3	3004	Leave Request ...	Workflows						
3	4003	Leave Request ...	Workflows						
3	4004	Leave Request ...	Workflows						

When you are ready to continue, please move on to the Building a custom report in Excel with a Data Feed, PowerPivot and a PivotChart exercise to start building the report.

Building a custom report in Microsoft Excel with a REST Endpoint data feed, PowerPivot table and a PivotChart

Exporting report data into Excel can be accomplished by simply choosing the Export option in either reporting format (Standard or Custom). The exported data is static, however. To view the most current data, a new export is necessary. You can however, import a data feed into Excel for use in a PowerPivot table and PivotChart. This allows you to expose current K2 data continuously.

> **Note**
> The steps and screenshots in this tutorial are based on the K2-provided virtual server image. The list of workflows may be different in your environment. You can, however, use these same steps and instructions to report on other workflows in your K2 environment.

> **Caution**
> To work through the steps in this tutorial, you must have a deployed process with two or more active process instances. If you do not have any process instances started on your K2 server, you can download a sample process, then deploy it using the K2 Package and Deployment tool. After you have deployed the process, start at least three instances of the process from the K2 Management site. Download the sample process

- For help with downloading and deploying the sample process package, see the Administering K2: Download and Deploy a Simple Package topic. Return to this tutorial after you have deployed the package.
- For help with starting process instances, see the Administering K2: Managing Processes and Process Instances topic. Return to this tutorial after you have started several process instances.

Step 1: Import a Data Feed into Excel

In this step, we will generate a data feed using K2's Endpoint Service, then import that data feed into an Excel PowerPivot table. From the table, we will create a PivotChart. We will then update one Leave Request Process and view the updated content in Excel.

> *Step 1 Tasks*
>
> 1. Use the K2 services endpoints listing web page, located at *http://k2.denallix.com:8888/SmartObjectServices/endpoints/endpoints.xml*
>   to determine the URL for the **Process Instance** SmartObject. It should be something like:
>
>   *http://k2.denallix.com:8888/SmartObjectServices/rest/Workflow Reports/Workflow General/Process*

Instance/List?$format={format}&amp;$top={top}&amp;$skip={skip}&amp;filterXml={filterXml}&amp;ProcessInstanceID={ProcessInstanceID}

NOTE: If you get an error when trying to connect to the K2 services endpoints listing web page, try to access the page using a secure (SSL) connection. (*HTTPS://k2.denallix.com:8888/SmartObjectServices/endpoints/endpoints.xml*)

> **Note**
> Due to possible configuration differences between K2-provided virtual environments, the URL you need to open could be either HTTP or HTTPS.

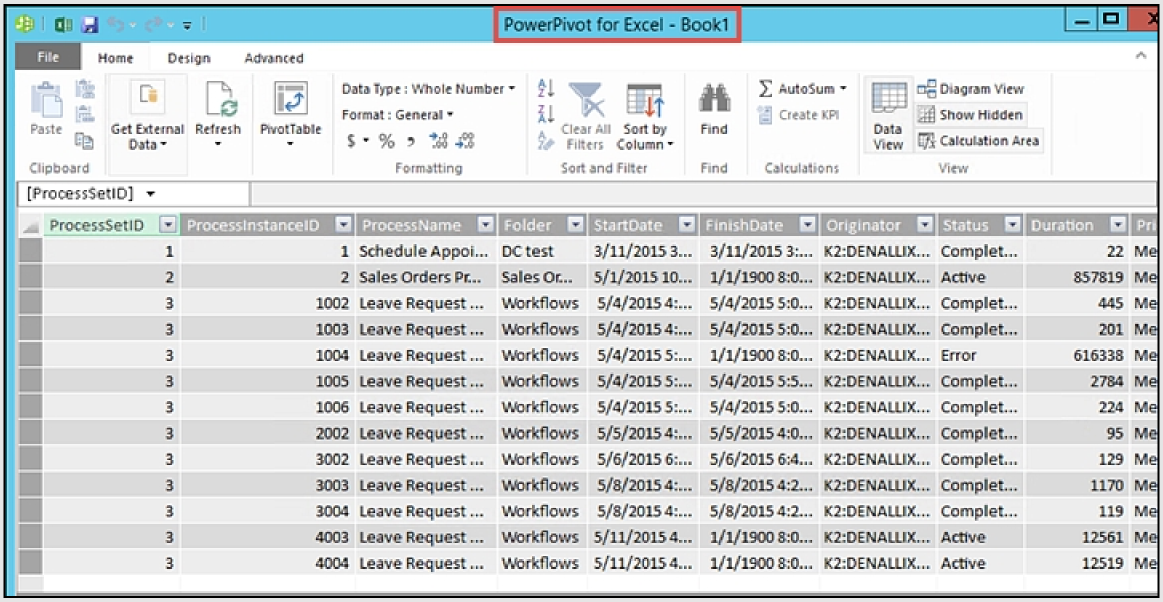
> **Caution**
> The name of the server used in this sample (k2.denallix.com) is based on the standard K2-provided virtual server image. If you are attempting this exercise in another environment, you will probably need to adjust the server name appropriately; check with your K2 administrator what the URL is for your K2 environment.

> **Note**
> If the URL is not found, follow the steps described in the detailed walkthrough below to enable the Rest endpoint.

2. Change the URL parameters to retrieve all items in ATOM format, and then test the URL in Internet Explorer. (Remember you may need to use HTTP or HTTPS.)

   https://k2.denallix.com:8888/SmartObjectServices/rest/Workflow Reports/Workflow General/Process Instance/List?$format=ATOM

3. Use this URL as the data feed for a **PowerPivot** table in Excel that displays the SmartObject properties for Process Instances.



ProcessSetID	ProcessInstanceID	ProcessName	Folder	StartDate	FinishDate	Originator	Status	Duration	Pri
1	1	Schedule Appoi...	DC test	3/11/2015 3...	3/11/2015 3:...	K2:DENALLIX...	Complet...	22	Me
2	2	Sales Orders Pr...	Sales Or...	5/1/2015 10...	1/1/1900 8:0...	K2:DENALLIX...	Active	857819	Me
3	1002	Leave Request ...	Workflows	5/4/2015 4:...	5/4/2015 5:0...	K2:DENALLIX...	Complet...	445	Me
3	1003	Leave Request ...	Workflows	5/4/2015 4:...	5/4/2015 5:0...	K2:DENALLIX...	Complet...	201	Me
3	1004	Leave Request ...	Workflows	5/4/2015 5:...	1/1/1900 8:0...	K2:DENALLIX...	Error	616338	Me
3	1005	Leave Request ...	Workflows	5/4/2015 5:...	5/4/2015 5:5...	K2:DENALLIX...	Complet...	2784	Me
3	1006	Leave Request ...	Workflows	5/4/2015 5:...	5/4/2015 5:0...	K2:DENALLIX...	Complet...	224	Me
3	2002	Leave Request ...	Workflows	5/5/2015 4:...	5/5/2015 4:0...	K2:DENALLIX...	Complet...	95	Me
3	3002	Leave Request ...	Workflows	5/6/2015 6:...	5/6/2015 6:4...	K2:DENALLIX...	Complet...	129	Me
3	3003	Leave Request ...	Workflows	5/8/2015 4:...	5/8/2015 4:2...	K2:DENALLIX...	Complet...	1170	Me
3	3004	Leave Request ...	Workflows	5/8/2015 4:...	5/8/2015 4:2...	K2:DENALLIX...	Complet...	119	Me
3	4003	Leave Request ...	Workflows	5/11/2015 4...	1/1/1900 8:0...	K2:DENALLIX...	Active	12561	Me
3	4004	Leave Request ...	Workflows	5/11/2015 4...	1/1/1900 8:0...	K2:DENALLIX...	Active	12519	Me

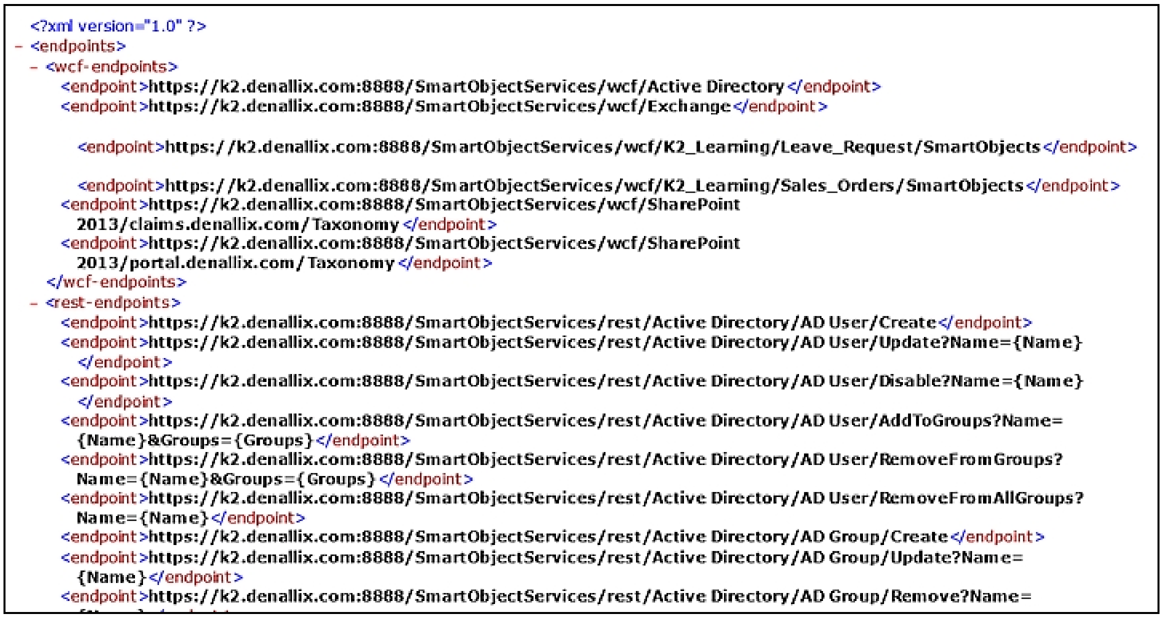
Step 1 Walkthrough

a. The first step we want to take is to test the **SmartObject Services URL** in a browser to confirm we can connect to and read, K2 Endpoint Services. We also need to confirm that all of the Endpoint services are being returned. From your **browser**, open a new tab and **enter** the following web service URL:
http://k2.denallix.com:8888/SmartObjectServices/endpoints/endpoints.xml

NOTE: If you get an error when trying to connect to the K2 services endpoints listing web page, try to access the page using a secure (SSL) connection. (*HTTPS://k2.denallix.com:8888/SmartObjectServices/endpoints/endpoints.xml*)

Note
Due to possible configuration differences between K2-provided virtual environments, the URL you need to open could be either HTTP or HTTPS.

Your screen should look similar to the image below.

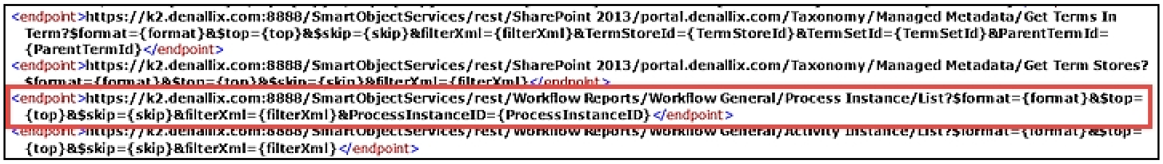


```
  <?xml version="1.0" ?>
- <endpoints>
  - <wcf-endpoints>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/wcf/Active Directory</endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/wcf/Exchange</endpoint>

        <endpoint>https://k2.denallix.com:8888/SmartObjectServices/wcf/K2_Learning/Leave_Request/SmartObjects</endpoint>

        <endpoint>https://k2.denallix.com:8888/SmartObjectServices/wcf/K2_Learning/Sales_Orders/SmartObjects</endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/wcf/SharePoint
        2013/claims.denallix.com/Taxonomy</endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/wcf/SharePoint
        2013/portal.denallix.com/Taxonomy</endpoint>
    </wcf-endpoints>
  - <rest-endpoints>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Active Directory/AD User/Create</endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Active Directory/AD User/Update?Name={Name}
        </endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Active Directory/AD User/Disable?Name={Name}
        </endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Active Directory/AD User/AddToGroups?Name=
        {Name}&Groups={Groups}</endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Active Directory/AD User/RemoveFromGroups?
        Name={Name}&Groups={Groups}</endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Active Directory/AD User/RemoveFromAllGroups?
        Name={Name}</endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Active Directory/AD Group/Create</endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Active Directory/AD Group/Update?Name=
        {Name}</endpoint>
      <endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Active Directory/AD Group/Remove?Name=
```

Caution
The name of the server used in this sample (k2.denallix.com) is based on the standard K2-provided virtual server image. If you are attempting this exercise in another environment, you will probably need to adjust the server name appropriately; check with your K2 administrator what the URL is for your K2 environment

Tip
Consider viewing the browser content in NotePad. Copy (CTRL-C) the browser contents, then paste (CTRL-V) them into NotePad. By turning off Word Wrap, there will be one result per line. This can be easier to read.

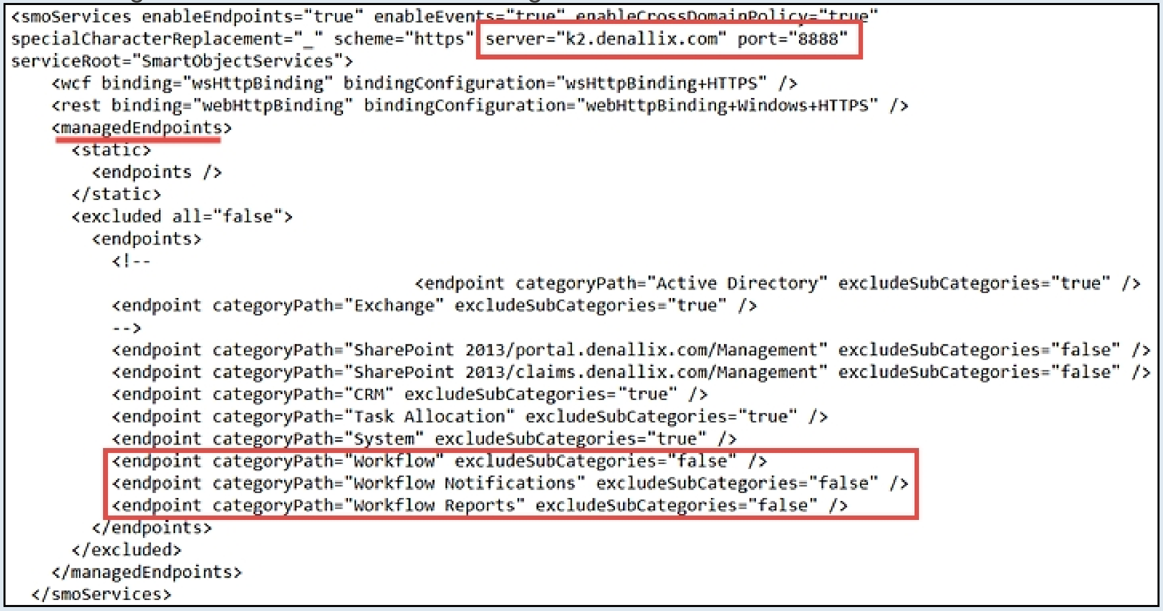
b. Scroll through the results returned and confirm the URLs contain entries for the **Workflow Reports/Workflow General** processes. We are specifically looking for a URL that looks like the URL below (remember the URL could be either HTTP or HTTPS):
https://k2.denallix.com:8888/SmartObjectServices/rest/Workflow Reports/Workflow General/Process Instance/List?$format={format}&$top={top}&$skip={skip}&filterXml={filterXml}&ProcessInstanceID={ProcessInstanceID}



```
<endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/SharePoint 2013/portal.denallix.com/Taxonomy/Managed Metadata/Get Terms In
  Term?$format={format}&$top={top}&$skip={skip}&filterXml={filterXml}&TermStoreId={TermStoreId}&TermSetId={TermSetId}&ParentTermId=
  {ParentTermId}</endpoint>
<endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/SharePoint 2013/portal.denallix.com/Taxonomy/Managed Metadata/Get Term Stores?
  $format={format}&$top={top}&$skip={skip}&filterXml={filterXml}</endpoint>
<endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Workflow Reports/Workflow General/Process Instance/List?$format={format}&$top=
  {top}&$skip={skip}&filterXml={filterXml}&ProcessInstanceID={ProcessInstanceID}</endpoint>
<endpoint>https://k2.denallix.com:8888/SmartObjectServices/rest/Workflow Reports/Workflow General/Activity Instance/List?$format={format}&$top=
  {top}&$skip={skip}&filterXml={filterXml}</endpoint>
```

Note
If the URL is missing
If there is not a URL that matches the above, you will need to edit a K2 configuration file to enable the service. The steps below explain how to edit this configuration file in the K2-provided virtual server environment. If you are attempting this tutorial in a different environment, check with your K2 administrator or ask them to enable the services for you.

- Navigate to the **K2HostService.exe.config** file located at:
  "C:\Program Files (x86)\K2 blackpearl\Host Server\Bin\ K2HostServer.exe.Config"
- Back up the file prior to making any changes!
- **Open** the file using **NotePad** then search for "**managedEndpoints**"
- Locate the element called <smoServices> just above managedEndoints element. Scroll to the end of the line and confirm that the server reflects "**k2.denallix.com**" if you are using a K2-provided VM (or your server name if working within your own environment). Confirm the port is "**8888**".
- Change the **Workflow** exclude attributes (there will be three of them) to "**false**". (We want show the categories, not exclude them.)
  Your configuration should look like the image below.



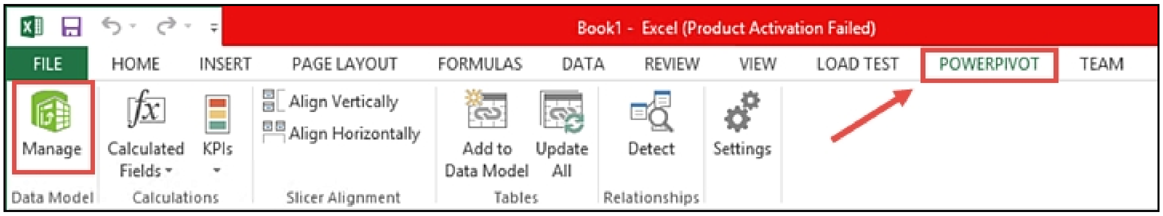
```
<smoServices enableEndpoints="true" enableEvents="true" enableCrossDomainPolicy="true"
specialCharacterReplacement="_" scheme="https" server="k2.denallix.com" port="8888"
serviceRoot="SmartObjectServices">
    <wcf binding="wsHttpBinding" bindingConfiguration="wsHttpBinding+HTTPS" />
    <rest binding="webHttpBinding" bindingConfiguration="webHttpBinding+Windows+HTTPS" />
    <managedEndpoints>
      <static>
        <endpoints />
      </static>
      <excluded all="false">
        <endpoints>
          <!--
                                <endpoint categoryPath="Active Directory" excludeSubCategories="true" />
          <endpoint categoryPath="Exchange" excludeSubCategories="true" />
          -->
          <endpoint categoryPath="SharePoint 2013/portal.denallix.com/Management" excludeSubCategories="false" />
          <endpoint categoryPath="SharePoint 2013/claims.denallix.com/Management" excludeSubCategories="false" />
          <endpoint categoryPath="CRM" excludeSubCategories="true" />
          <endpoint categoryPath="Task Allocation" excludeSubCategories="true" />
          <endpoint categoryPath="System" excludeSubCategories="true" />
          <endpoint categoryPath="Workflow" excludeSubCategories="false" />
          <endpoint categoryPath="Workflow Notifications" excludeSubCategories="false" />
          <endpoint categoryPath="Workflow Reports" excludeSubCategories="false" />
        </endpoints>
      </excluded>
    </managedEndpoints>
  </smoServices>
```

- **Restart** the **K2 blackpearl** Service. (**Start** > **Administration Panel** > **Services**)

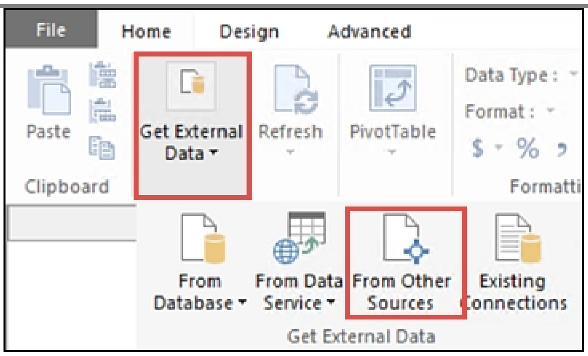
Once we have confirmed the Process Instance Endpoint URL exists, we are ready to move on to the next step. This URL is the basis for our data feed URL, with a minor edit.

c. Edit the URL so that it matches the example below.
https://k2.denallix.com:8888/SmartObjectServices/rest/Workflow Reports/Workflow General/Process Instance/List?**$format=ATOM** (and remove the remaining string).

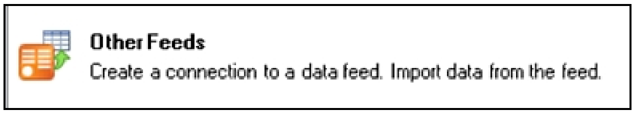
d. Launch **Excel**. Create a new Workbook. Click the **PowerPivot** tab, then click **Manage**.



e. When the Manage PowerPivot window opens, select **Get External Data**, then **From Other Sources**.

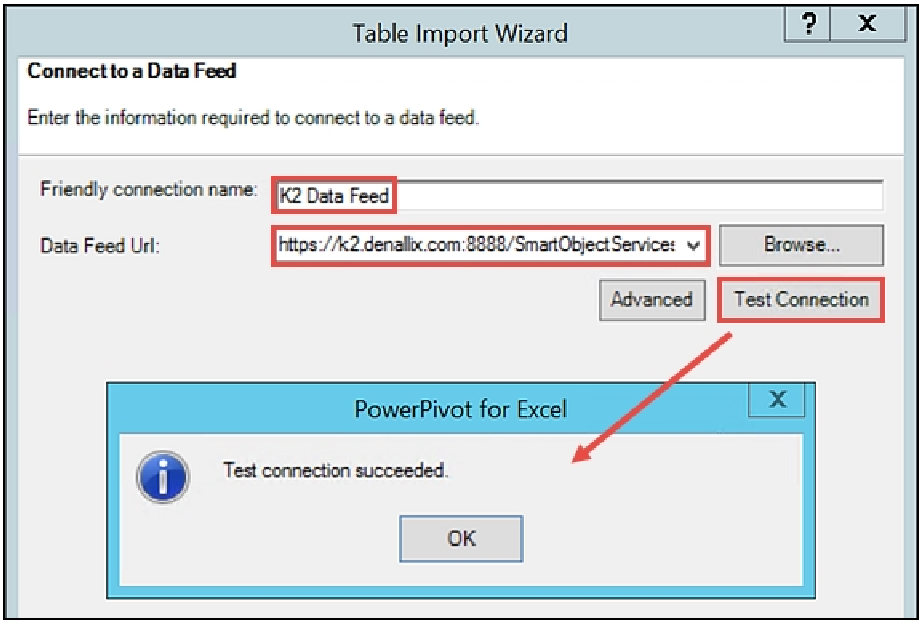


f. The **Connect to a Data Source** Import Wizard opens. Select **Other Feeds** then click **Next**.



Other Feeds
Create a connection to a data feed. Import data from the feed.

g. For the **Friendly connection name**, enter
K2 Data Feed
then **copy the URL** we configured in the previous steps into the **Data Feed Url** field. **Test** the connection.



Connect to a Data Feed

Enter the information required to connect to a data feed.

Friendly connection name: K2 Data Feed

Data Feed Url: https://k2.denallix.com:8888/SmartObjectServices

Test connection succeeded.

Tip
Connection Failed
If the connect fails, click the **Advanced** button. Change the **Include Atom Elements** from Auto to **True**, then test again. The connection should be successful.

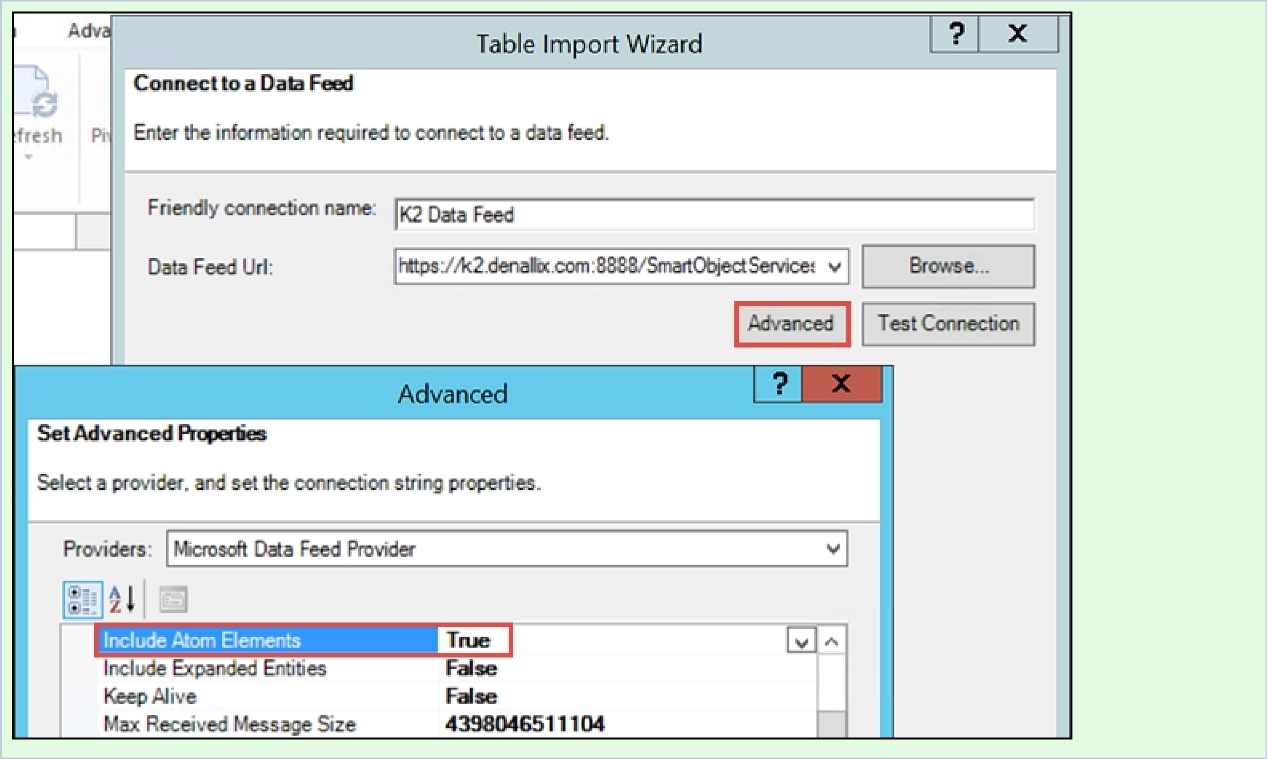


Table Import Wizard

Connect to a Data Feed

Enter the information required to connect to a data feed.

Friendly connection name: K2 Data Feed

Data Feed Url: https://k2.denallix.com:8888/SmartObjectServices

Advanced

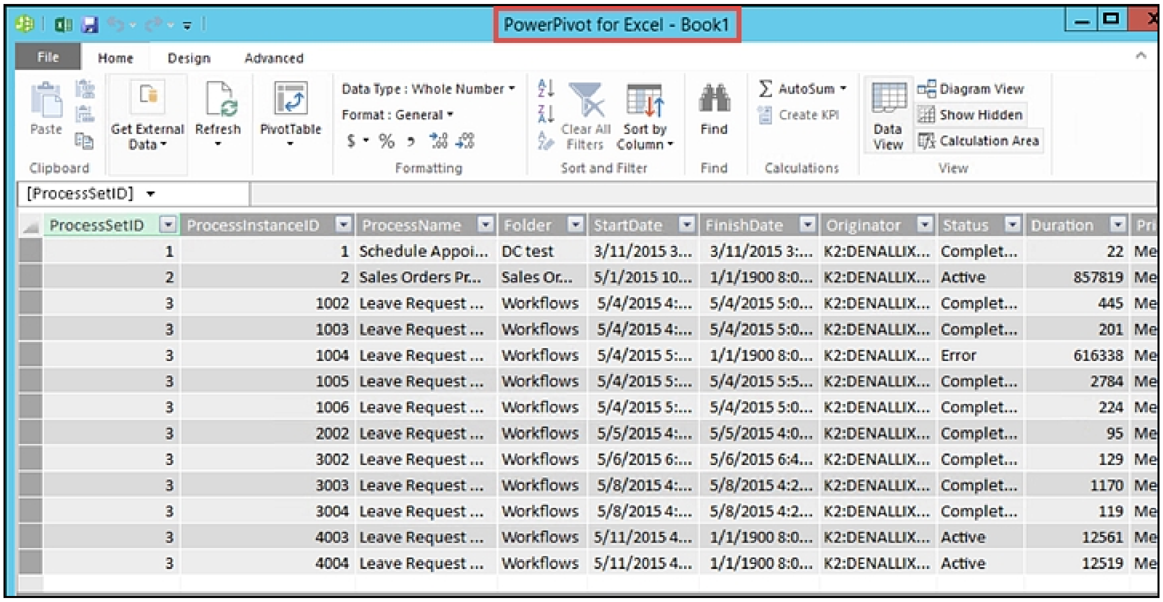
Set Advanced Properties

Select a provider, and set the connection string properties.

Providers: Microsoft Data Feed Provider

Include Atom Elements	True
Include Expanded Entities	False
Keep Alive	False
Max Received Message Size	4398046511104

Click **Next** then **Finish** to complete the data feed configuration. **Close** the Import Wizard. You should now see your PowerPivot table with K2 reporting data.



ProcessSetID	ProcessInstanceID	ProcessName	Folder	StartDate	FinishDate	Originator	Status	Duration	Pri
1	1	Schedule Appoi...	DC test	3/11/2015 3:...	3/11/2015 3:...	K2:DENALLIX...	Complet...	22	Me
2	2	Sales Orders Pr...	Sales Or...	5/1/2015 10:...	1/1/1900 8:0...	K2:DENALLIX...	Active	857819	Me
3	1002	Leave Request ...	Workflows	5/4/2015 4:...	5/4/2015 5:0...	K2:DENALLIX...	Complet...	445	Me
3	1003	Leave Request ...	Workflows	5/4/2015 4:...	5/4/2015 5:0...	K2:DENALLIX...	Complet...	201	Me
3	1004	Leave Request ...	Workflows	5/4/2015 5:...	1/1/1900 8:0...	K2:DENALLIX...	Error	616338	Me
3	1005	Leave Request ...	Workflows	5/4/2015 5:...	5/4/2015 5:5...	K2:DENALLIX...	Complet...	2784	Me
3	1006	Leave Request ...	Workflows	5/4/2015 5:...	5/4/2015 5:0...	K2:DENALLIX...	Complet...	224	Me
3	2002	Leave Request ...	Workflows	5/5/2015 4:...	5/5/2015 4:0...	K2:DENALLIX...	Complet...	95	Me
3	3002	Leave Request ...	Workflows	5/6/2015 6:...	5/6/2015 6:4...	K2:DENALLIX...	Complet...	129	Me
3	3003	Leave Request ...	Workflows	5/8/2015 4:...	5/8/2015 4:2...	K2:DENALLIX...	Complet...	1170	Me
3	3004	Leave Request ...	Workflows	5/8/2015 4:...	5/8/2015 4:2...	K2:DENALLIX...	Complet...	119	Me
3	4003	Leave Request ...	Workflows	5/11/2015 4...	1/1/1900 8:0...	K2:DENALLIX...	Active	12561	Me
3	4004	Leave Request ...	Workflows	5/11/2015 4...	1/1/1900 8:0...	K2:DENALLIX...	Active	12519	Me

STEP 1 REVIEW

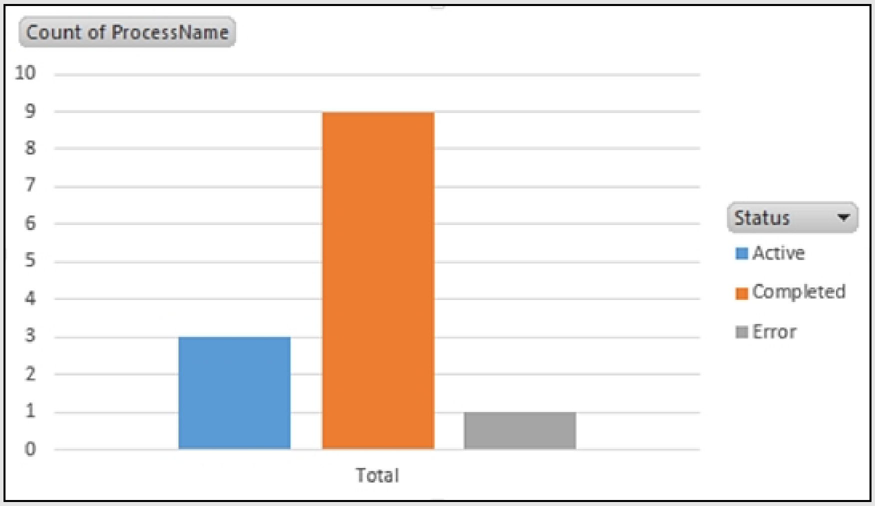
Using a data feed allows you to create external reports and charts using K2 data. The data feed is live, meaning that you can simply refresh your Excel tables and charts to view current reporting content. As we configured the data feed, we first confirmed the connection by testing the URL in a browser. Recall the data returned with the SmartObject Services endpoint web service. Depending on the environment you are working in, K2 provides data connections to many different types of endpoint services, including SharePoint, CRM, Workflow Reports, etc. This ability to connect to K2 data sources provides a wide range of options for retrieving and using K2 data in external applications.

Step 2: Create a PivotChart

In this step, we will create a PivotChart using the PowerPivot data returned from our data feed. This chart will display a count of current status values for Process Instances (Active, Completed, Error, etc.) We will return to K2 Workspace and action one of the (Active) Leave Request Workflow Process Instances, then refresh our PivotChart to confirm the action is displayed correctly.

Step 2 Tasks

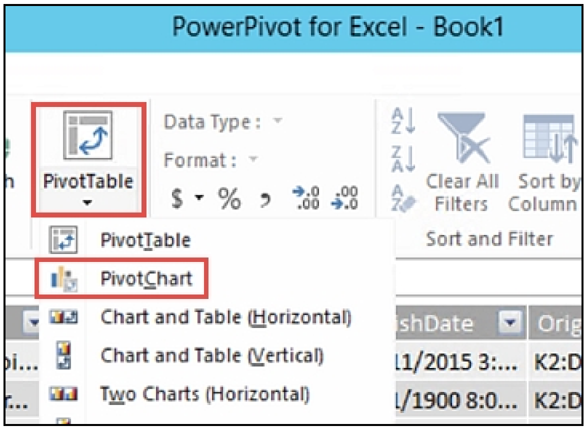
1. Create a **PivotChart** from the PowerPivot table created in the previous step.
2. Configure the **Status** property to be the chart **Legend** and the **ProcessName** to be the chart **Sum**. Make a note of the current Active and Completed values.



3. In K2 Workspace, action one of the **(Active) Leave Request Workflow** Process Instances. You can either redirect a Worklist Item to yourself and action it directly from K2 Workspace, or you can open Outlook for the Destination user and action the request using SmartActions.
4. Return to the **PivotChart** and **Refresh** the chart **Data**. Confirm the Active and Complete counts are updated to reflect the action in the previous step.

Step 2 Walkthrough

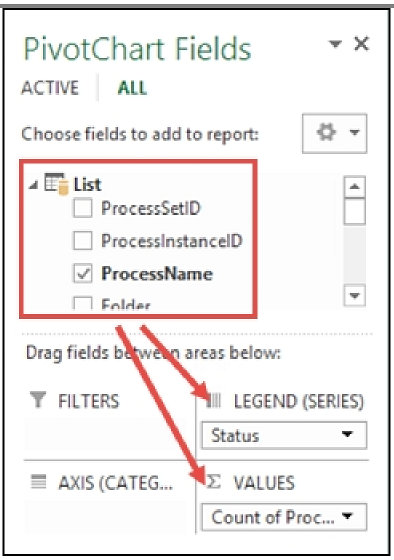
a. Still in the **PowerPivot** table view, click the **PivotTable** button and select **PivotChart**.



b. Select **New Worksheet**, then click **OK**.

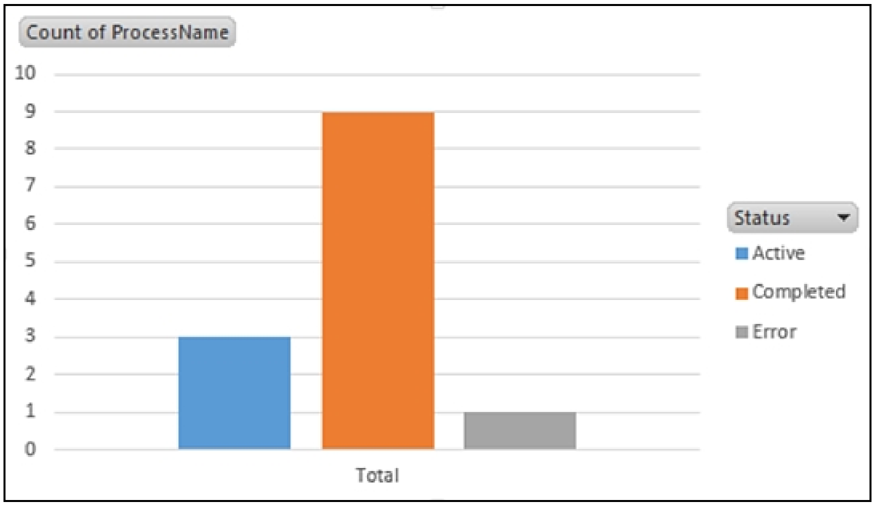
The PivotChart we are going to configure will show the status values for all Processes. We want to see how many Processes have Completed, how many are Active and how many are in an Error state.

c. In the PivotChart Fields pane, expand the List options. Drag the **Status** into the **Legend** box and the **ProcessName** into the **Sum** box.



Your PivotChart should look like the image below. This data is coming directly from the PowerPivot table, which is a data feed from K2. In the next step, we will update one of the Leave Request Process Instances, then return to the PivotChart and refresh it. Because this is a live data feed, the change will be reflected immediately.

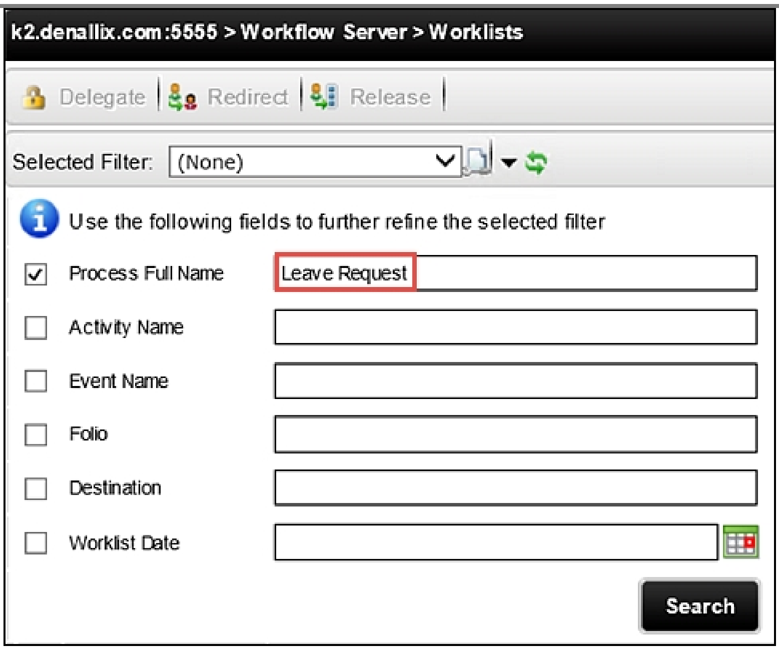
Make a note of number of Active Process Instances. In the chart below, this value is **3**. (Your chart values are likely to be different.) Do not exit Excel at this time, we will be returning to the PivotChart shortly.



To save on time, we will redirect one of the Manager Approval Worklist Items back to ourselves (Administrator if you are using a K2-provided VM), then action the request from K2 Workspace. If you prefer, you can still open Outlook for the Destination User and action the request from there using SmartActions.

d. Open **K2 Workspace** (if it is not already open). Navigate to the **Worklists** category. (**Management** > **Management Console** > **K2 Server** > **Workflow Server** > **Worklists**)

e. When you click on Worklists, the **Worklist Filter** pane will open. Enter
Leave Request
for the **Process Full Name** and click **Search**.



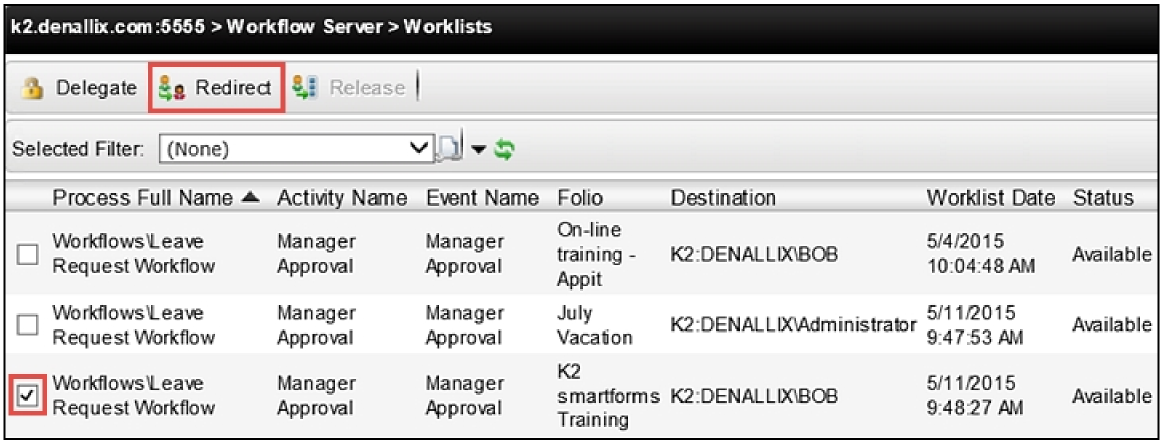
k2.denallix.com:5555 > Workflow Server > Worklists

Selected Filter: (None)

Use the following fields to further refine the selected filter

☑	Process Full Name	Leave Request
□	Activity Name	
□	Event Name	
□	Folio	
□	Destination	
□	Worklist Date	

f. A list of active Leave Request Worklist Items is displayed. CHECK the box to the left of one of the **Active** Workflist Items to select it, then click **Redirect**. When the **Search for Users** screen opens, enter *Administrator* then click the **Search** icon. After you see **Administrator** in the results pane, CHECK the box to the left of the **Administrator** name to select it, and click **OK**. This Worklist Item has now been redirected to Administrator (presumably you), where you can now action it directly from K2 Workspace.

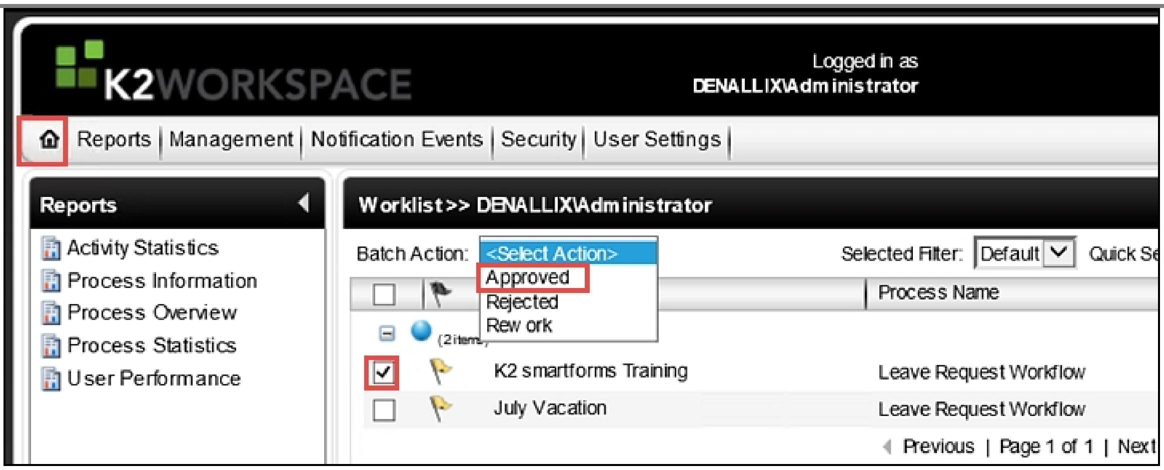


k2.denallix.com:5555 > Workflow Server > Worklists

Selected Filter: (None)

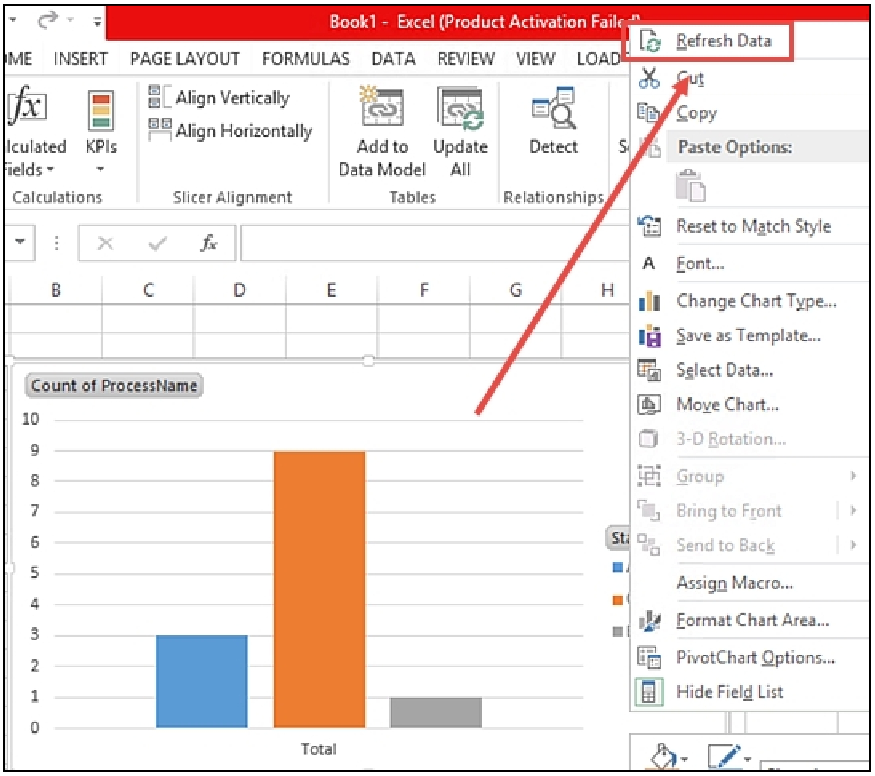
	Process Full Name ▲	Activity Name	Event Name	Folio	Destination	Worklist Date	Status
□	Workflows\Leave Request Workflow	Manager Approval	Manager Approval	On-line training - Appit	K2:DENALLIX\BOB	5/4/2015 10:04:48 AM	Available
□	Workflows\Leave Request Workflow	Manager Approval	Manager Approval	July Vacation	K2:DENALLIX\Administrator	5/11/2015 9:47:53 AM	Available
☑	Workflows\Leave Request Workflow	Manager Approval	Manager Approval	K2 smartforms Training	K2:DENALLIX\BOB	5/11/2015 9:48:27 AM	Available

g. Navigate back to the K2 Workspace landing page by clicking the **Home** button in the navigation bar. You should now see the Worklist Item for Administrator displayed. CHECK the box to the left of the **Folio**, then action the request using an option from the **Batch Action** drop-down list. Click **OK** twice to complete this action.

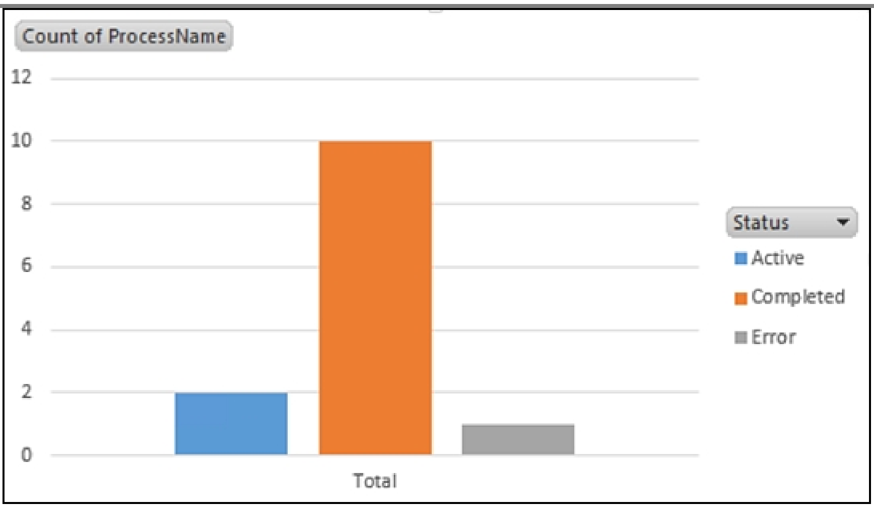


Now that we have actioned the request, the Process Instance will have completed. We will return to our PivotChart and refresh it. We should see the Count for Completed instances go up by one, and the Count for Active instances, go down by one. This will demonstrate the live data feed configuration.

h. Click on **Excel** to display the **PivotChart**. Click once inside of the chart, then **right-click**. Select **Refresh Data**.



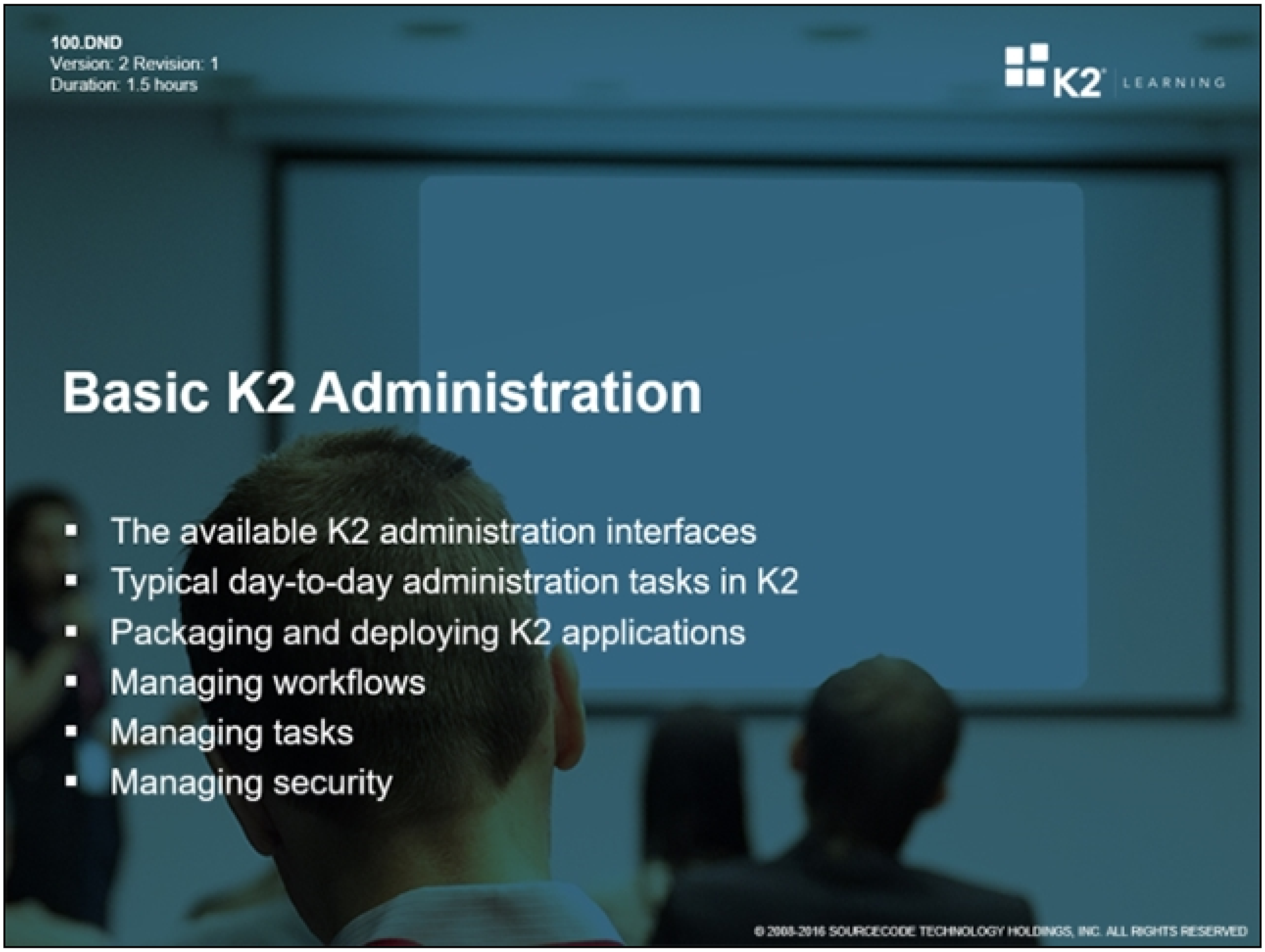
The chart is refreshed with the new counts for Active and Completed Process Instances.



STEP 2 REVIEW

In this step, we created a PivotChart from the Process Instance PowerPivot table. The PowerPivot table displays live data from K2 via a data feed. We demonstrated the live feed by updating a Leave Request Worklist Item in K2 Workspace, then refreshing the chart data.

100.DND: Basic K2 Administration



100.DND
Version: 2 Revision: 1
Duration: 1.5 hours

Basic K2 Administration

- The available K2 administration interfaces
- Typical day-to-day administration tasks in K2
- Packaging and deploying K2 applications
- Managing workflows
- Managing tasks
- Managing security

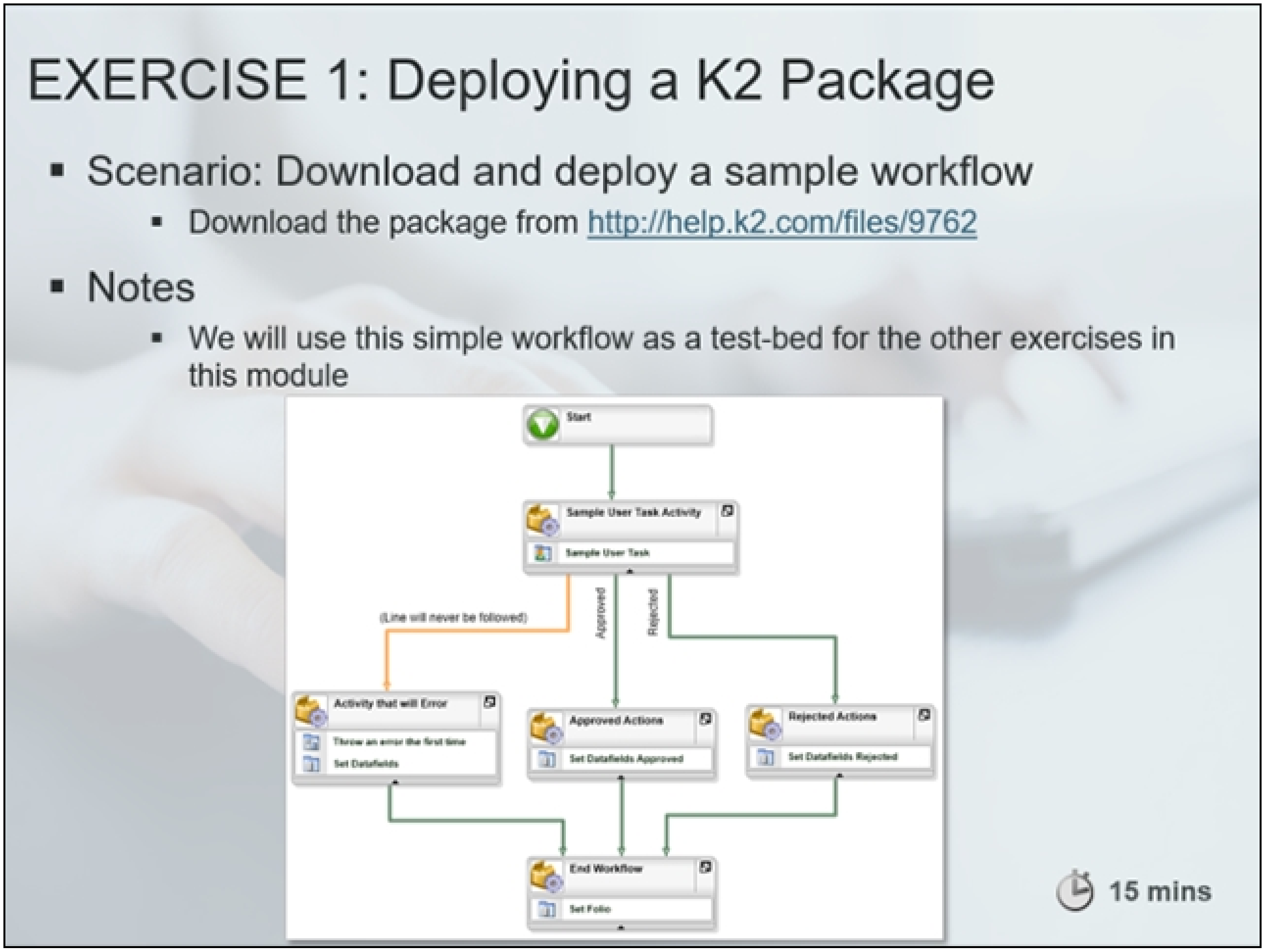
K2 LEARNING

Video

The *100.DND: Basic K2 Administration* training module is focused on common, day-to-day administration tasks in K2. As part of this module you will learn:

- The available interfaces used to administer a K2 environment
- Packaging and deploying K2 applications
- Typical workflow administration tasks in K2 like managing workflows and worklists
- Other common administrative tasks like managing service instances and permissions

EXERCISE 1: Deploying a K2 Package



EXERCISE 1: Deploying a K2 Package

- Scenario: Download and deploy a sample workflow
  - Download the package from http://help.k2.com/files/9762
- Notes
  - We will use this simple workflow as a test-bed for the other exercises in this module

15 mins

Video

In this exercise we will download a package and then use the K2 Package and Deployment tool to deploy a pre-created K2 workflow to our exercise environment. We will be using this sample workflow throughout this module to practice administrative tasks, so you must complete this exercise before attempting to complete the later exercises in this module.

You can download the package from K2's help site at: http://help.k2.com/files/9762

The workflow is a simple 1-step approval workflow which we will use for the other exercises in this module as a "test-bed". You can delete this workflow from your environment after this module, if you prefer.

When you are ready, continue on to K2 Administration: Download and deploy a simple K2 Package to download and deploy the sample workflow.

Download and Deploy a K2 Package

In this tutorial, you will download and deploy a sample workflow using the K2 Package and Deployment tool. Once the workflow has been deployed, you will start several instances of the workflow to use throughout the exercises.

> **Note**
> This tutorial includes steps using the K2 Management site. You must have K2 blackpearl 4.7 or later to complete this tutorial as the steps and screen shots are specific to the K2 Management site.

Step 1: Download and deploy the workflow package

The first step you need to take is to download a K2 project file that you will then deploy to the K2 server. This project file contains a basic workflow that was created in K2 Studio. Once deployed, you will start a number of process instances to use in the tutorial steps. To deploy this solution, you will use the K2 Package and Deployment tool.

In a real-world environment, a K2 package could be created by a developer from your development K2 environment, or perhaps you may have downloaded the package from the K2 community site or another resource. In this case, K2 is providing a simple workflow in a K2 package so you can learn how to deploy a simple K2 package.

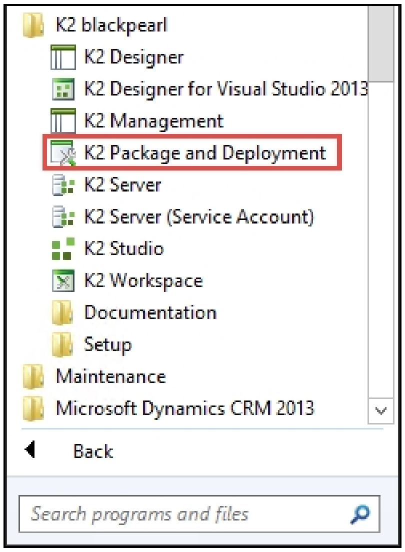
Step 1 Tasks

1. Download the K2 project file located at http://help.k2.com/files/9762.
2. Use the **K2 Package and Deployment** tool to deploy the project file. (**Start** > **All Programs** > **K2 blackpearl** > **K2 Package and Deployment**)
3. Confirm the **Workflow Administration Sample Process** is shown in the **Workflow Server** > **Workflows** central pane in the K2 Management site. (**Start** > **All Programs** > **K2 blackpearl** > **K2 Management**)

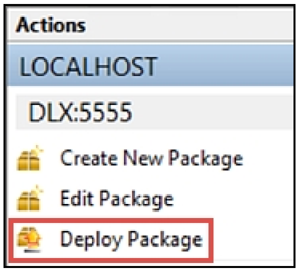
Step 1 Walkthrough

a. **Download** the K2 project file located at http://help.k2.com/files/9762 and **Save As** to your desktop. (To download the file, enter the file URL in your browser. You should then see a message with the options to either Open the file, Save or Save As. Select Save As and save the file to your desktop.)

b. Launch the **K2 Package and Deployment** tool.
(**Start** > **All Programs** > **K2 blackpearl** > **K2 Package and Deployment**)

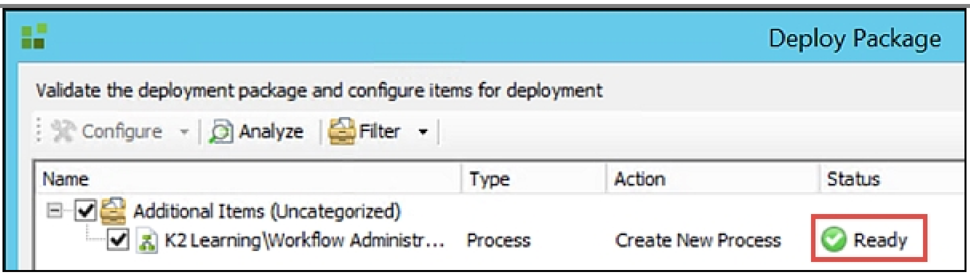

K2 blackpearl
K2 Designer
K2 Designer for Visual Studio 2013
K2 Management
K2 Package and Deployment
K2 Server
K2 Server (Service Account)
K2 Studio
K2 Workspace
Documentation
Setup
Maintenance
Microsoft Dynamics CRM 2013
Back
Search programs and files

c. From the right-side **Actions** column, click **Deploy Package**.



d. For the **File Name**, **Browse** to the K2 project file that you previously saved to your desktop. (The file name will be: *K2Learning-100.DND_BasicK2Administration.kspx*.) Keep the default values for the remaining deployment options and click **Next**.

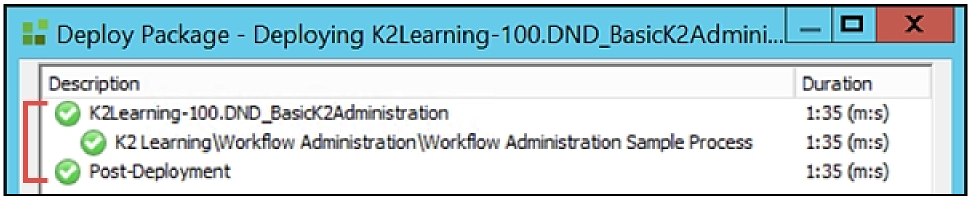
e. On the **Validate the deployment package** screen, wait a few seconds until you see the **Ready** indicator in the **Status** column. Click **Next** to continue.



Validate the deployment package and configure items for deployment

Name	Type	Action	Status
☑ Additional Items (Uncategorized)			
☑ K2 Learning\Workflow Administr…	Process	Create New Process	✓ Ready

f. Allow a minute or two for the package deployment to process and complete. When you see the green 'success' indicators next to the deployment descriptions, click **Finish**. **Exit** the K2 Package and Deployment tool.

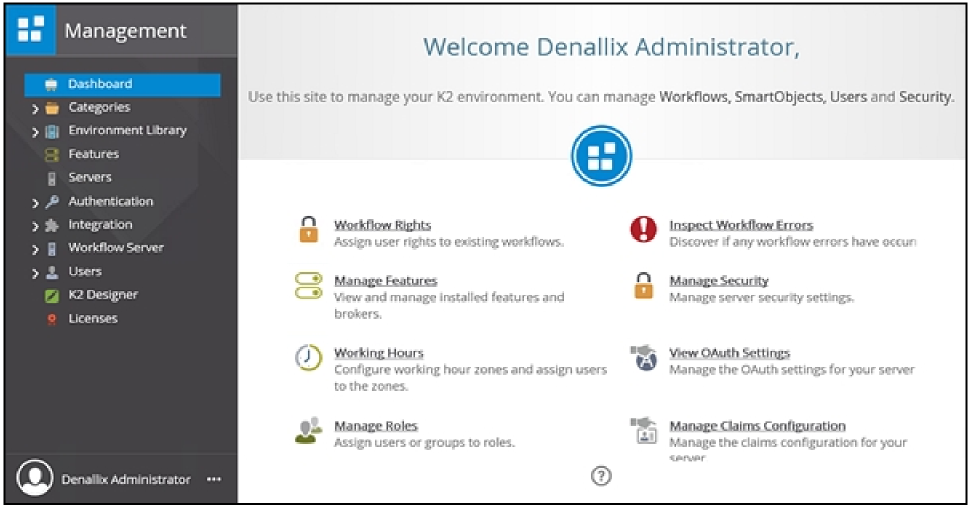

Deploy Package - Deploying K2Learning-100.DND_BasicK2Admini...

Description	Duration
✓ K2Learning-100.DND_BasicK2Administration	1:35 (m:s)
✓ K2 Learning\Workflow Administration\Workflow Administration Sample Process	1:35 (m:s)
✓ Post-Deployment	1:35 (m:s)

Now you will confirm the workflow appears in the K2 Management site. In later steps, you will start a number of process instances to use throughout the tutorial.

g. Launch the **K2 Management** site.
(**Start** > **All Programs** > **K2 blackpearl** > **K2 Management**)
The site opens with the default Dashboard view. Your dashboard may or may not contain all of the features available, depending on your K2 server rights.

The K2 administrator's Dashboard view with admin rights



Welcome Denallix Administrator,

Use this site to manage your K2 environment. You can manage Workflows, SmartObjects, Users and Security.

Workflow Rights
Assign user rights to existing workflows.

Inspect Workflow Errors
Discover if any workflow errors have occurr

Manage Features
View and manage installed features and brokers.

Manage Security
Manage server security settings.

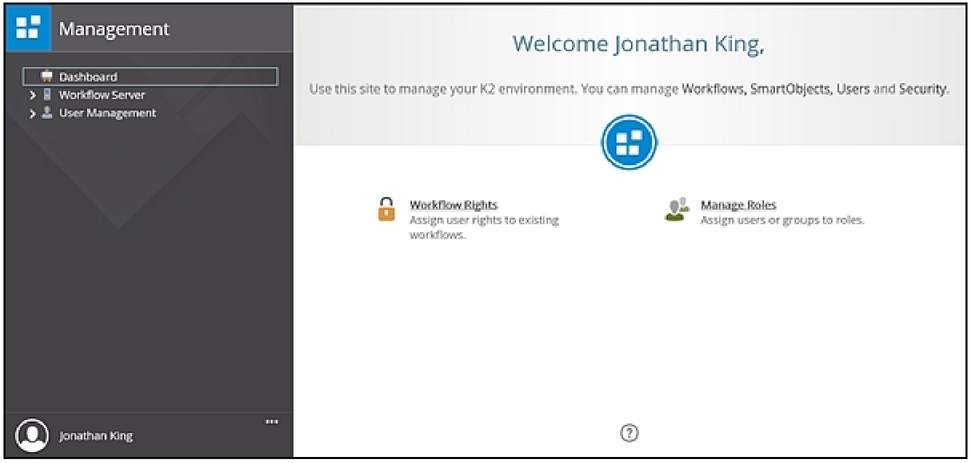
Working Hours
Configure working hour zones and assign users to the zones.

View OAuth Settings
Manage the OAuth settings for your server

Manage Roles
Assign users or groups to roles.

Manage Claims Configuration
Manage the claims configuration for your server

An end-user's Dashboard view with limited rights



Welcome Jonathan King,

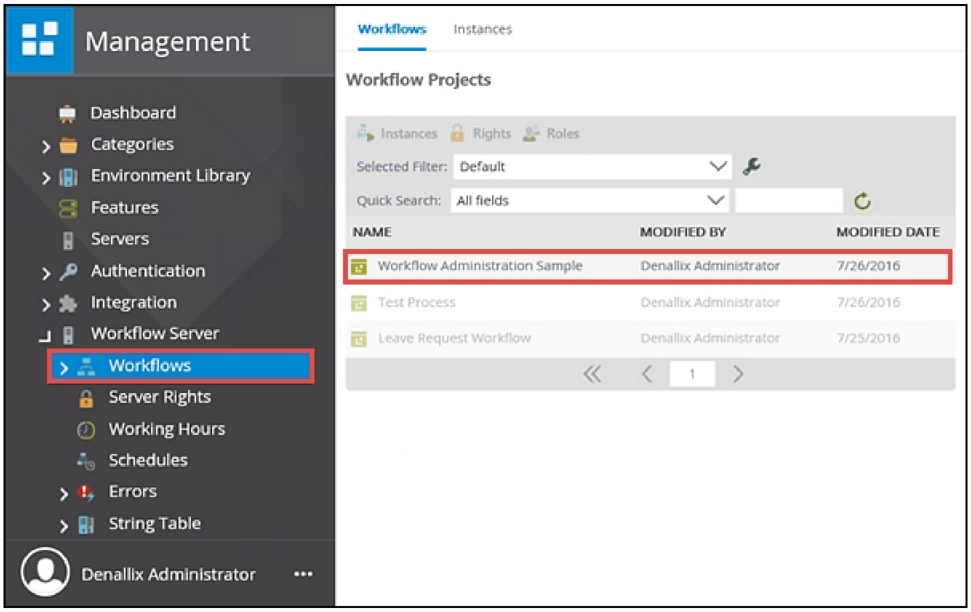
Use this site to manage your K2 environment. You can manage Workflows, SmartObjects, Users and Security.

Workflow Rights
Assign user rights to existing workflows.

Manage Roles
Assign users or groups to roles.

Note
Many of the features found in the K2 Workspace can be found in the new K2 Management site, plus several additional features, such as those found in the SmartObjects Services Tester utility. You will not explore all of the features available in the new K2 Management site (in this tutorial), but if time allows, take a few minute to explore the menu and dashboard lists.

h. Expand the **Workflow Server** node, then click the **Workflows** node to expose the current Workflow Projects (also known as processes) in the central pane. Confirm the **Workflow Administration Sample** process is shown. This is the process that you just deployed. (Your list of processes will not be the same as the image below, outside of the Workflow Administration Sample.)



Workflow Projects

Selected Filter: Default

Quick Search: All fields

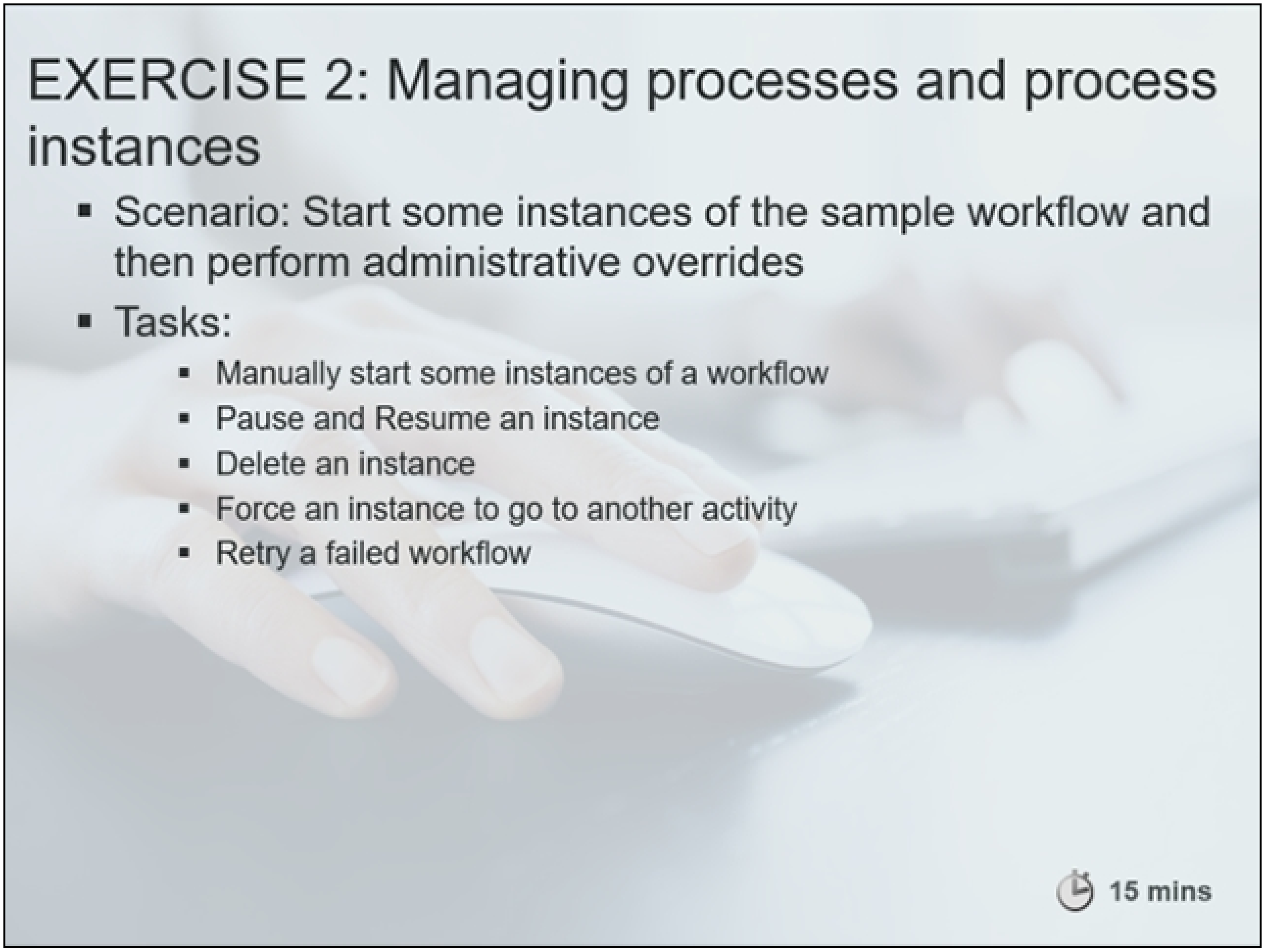
NAME	MODIFIED BY	MODIFIED DATE
Workflow Administration Sample	Denallix Administrator	7/26/2016
Test Process	Denallix Administrator	7/26/2016
Leave Request Workflow	Denallix Administrator	7/25/2016

STEP 1 REVIEW

In this step, you downloaded and deployed a sample process to use in later tutorials. The deployment package was actually a K2 project file, which could have been deployed from K2 Studio or K2 for Visual Studio as well. You used the K2 Package and Deployment tool as an option, to give you experience for using this tool in your own environment to deploy projects between environments.

Now that you have successfully deployed your project and workflow, you can move on to the next section, Managing Processes and Process Instances.

EXERCISE 2: Managing processes and process instances



EXERCISE 2: Managing processes and process instances

- Scenario: Start some instances of the sample workflow and then perform administrative overrides
- Tasks:
  - Manually start some instances of a workflow
  - Pause and Resume an instance
  - Delete an instance
  - Force an instance to go to another activity
  - Retry a failed workflow

15 mins

Video

In this exercise we will explore how to administer workflow instances. You will manually start some instances of the workflow deployed in Exercise 1, then perform some administrative functions on those instances, including:

- Pause and Resume an instance
- Delete an instance
- Force an instance to go to another activity
- Repair a failed workflow

This exercise requires that the sample workflow from Exercise 1 be deployed. When you are ready, continue on to Administering K2: Managing Processes and Process Instances to start this exercise.

Managing Processes and Process Instances

Recall that workflows are known as processes in K2 and process instances are a single occurrence of a process. Throughout this tutorial, both sets of terms will be used. Just know that workflows are processes, and workflow instances are process instances.

The most common tasks for the K2 administrator are starting and stopping (pausing) process instances, managing task lists and assigning rights, or permissions. When managing task lists, the administrator might have a request to redirect a task from one user to another, or manage Out of Office settings. In this part, you will explore these tasks using the features in the K2 Management site.

> **Note**
> This tutorial includes steps using the K2 Management site. You must have K2 blackpearl 4.7 or later to complete this tutorial as the steps and screen shots are specific to the K2 Management site.

Note
This tutorial requires that you have deployed the Workflow Administration Sample as described in the Download and deploy a simple K2 package section.

Step 1: Start new instances of the Workflow Administration Sample process

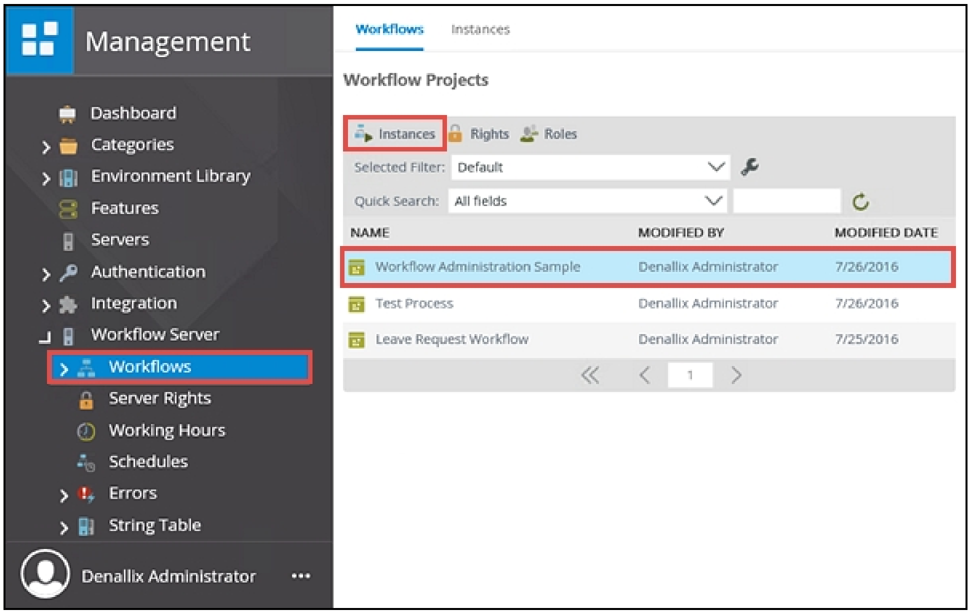
As a precursor to your tutorial steps, you need to generate several process instances of the Workflow Administration Sample process. In the real world, new process instances are most likely created when a user submits a form (and starts the workflow). Because you are not working with forms, you will start process instances from the K2 Management site. You could also use this tool for testing and troubleshooting your workflows without having to open and submit forms.

Step 1 Tasks

1. **Start** six instances of the **Workflow Administration Sample** process, using the following entries for each **Folio** parameter.
   - *TaskRedirect*
   - *GoToActivity*
   - *StartStop*
   - *Deleted*
   - *Normal Execution 1*
   - *Normal Execution 2*

Step 1 Walkthrough

a. In the K2 Management site, expand the **Workflow Server** node, then click the **Workflows** node (if it is not already expanded from the previous exercise). Click to highlight the **Workflow Administration Sample** process, then click the **Instances** button.

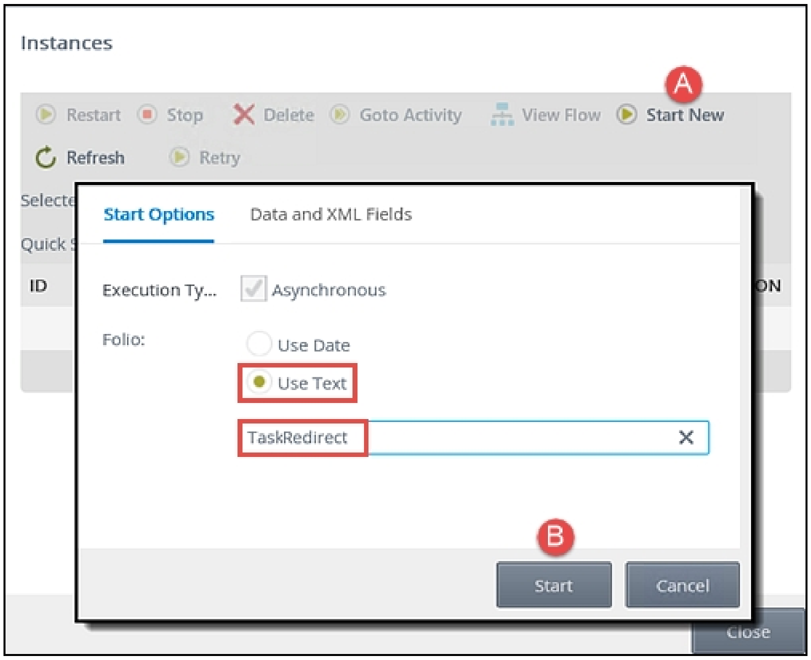


NAME	MODIFIED BY	MODIFIED DATE
Workflow Administration Sample	Denallix Administrator	7/26/2016
Test Process	Denallix Administrator	7/26/2016
Leave Request Workflow	Denallix Administrator	7/25/2016

The Instances screen opens in the central pane. Notice there are no instances shown. This is telling you that this process (Workflow Administration Sample) does not have any *active* process instances at this time. It could be a newly deployed process, or simply mean that any process instances for this workflow have completed or have been removed.

b. Click the **Start New** button found in the menu bar. For the **Folio** value, click the **Use Text** option. In the text box that is exposed, enter
TaskRedirect
then click **Start**. Click **OK** when you see the success confirmation dialogue.

> **Note**
> The Folio is a unique identifier that allows the administrator to differentiate one process instance from another. By default, K2 assigns the date for the Folio when starting process instances from the K2 Management site. You are going to enter your own Folio properties, which will allow you to pick and choose different process instances for the tutorial steps.

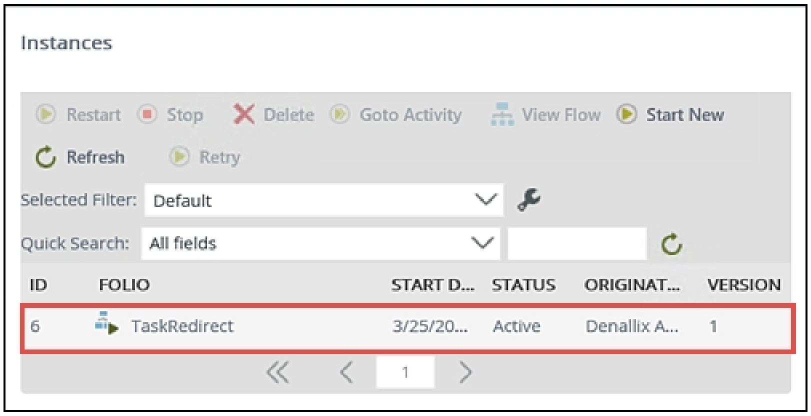


Execution Ty... ☑ Asynchronous

Folio: ○ Use Date
● Use Text

TaskRedirect

Notice now, you have one active process instance. Notice too, the **Folio** value is the TaskRedirect property you just entered. The **Active** status indicates that K2 is actively processing this workflow and the workflow is waiting for input of some kind before proceeding.



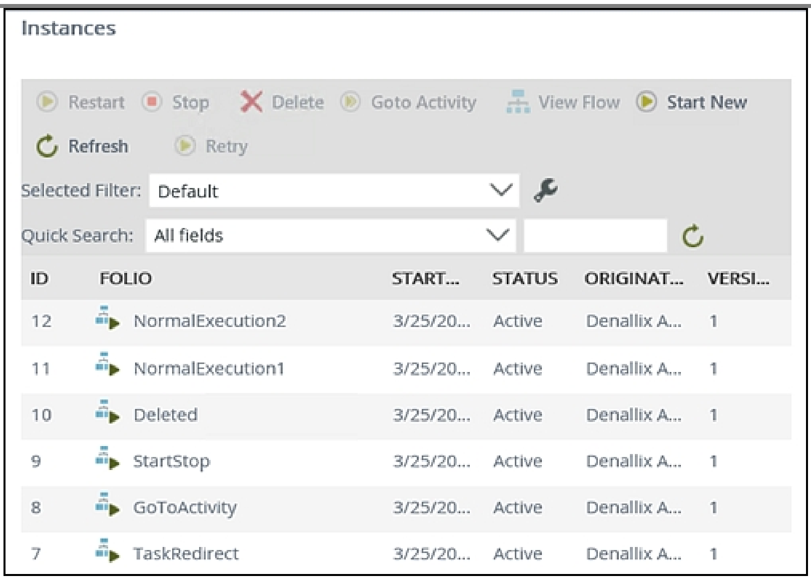
Selected Filter: Default

Quick Search: All fields

ID	FOLIO	START D...	STATUS	ORIGINAT...	VERSION
6	TaskRedirect	3/25/20...	Active	Denallix A...	1

c. You need to create several more process instances for future tutorial steps. Using the same steps as above, **Start New** process instances using the following entries for each **Folio** property:
   - *GoToActivity*
   - *StartStop*
   - *Deleted*
   - *Normal Execution1*
   - *Normal Execution2*

Your screen should now look like the image below. Each process instance shows a status of **Active**, wherein the workflow is now *waiting* for the next workflow step to take place. The Instances screen also shows us who started the workflow (Originator) and when it was started.



Selected Filter: Default

Quick Search: All fields

ID	FOLIO	START...	STATUS	ORIGINAT...	VERSI...
12	NormalExecution2	3/25/20...	Active	Denallix A...	1
11	NormalExecution1	3/25/20...	Active	Denallix A...	1
10	Deleted	3/25/20...	Active	Denallix A...	1
9	StartStop	3/25/20...	Active	Denallix A...	1
8	GoToActivity	3/25/20...	Active	Denallix A...	1
7	TaskRedirect	3/25/20...	Active	Denallix A...	1

STEP 1 REVIEW

In this step, you started a number of process instances of the Workflow Administration Sample process. You set the Folio property for your process instances using manually entered values. In the real world, you will likely assign a form field value as the Folio during the workflow build (such as the form originator's name or any other key value from the form or workflow context). The Folio is not a required value and there are some instances where you will not assign any value (leaving it blank). Because it helps make each process instance unique, you will find that using the Folio provides useful information later on when managing your process instances.

Step 2: Stop, start and remove a process instance

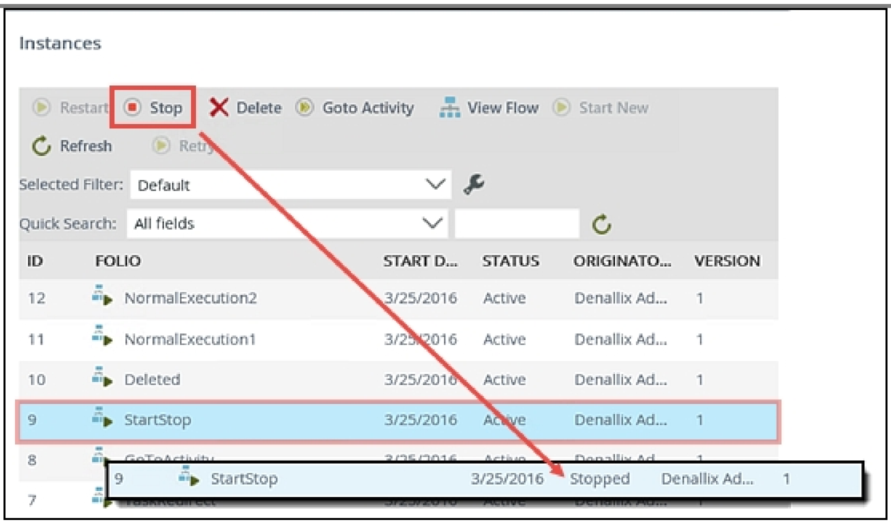
Now, you will use the Instances tools to stop (pause), then start (resume) a process instance. There are number of reasons why you might want to stop, or pause a workflow. One example might be if the workflow is currently waiting for a user to action a task of some kind. Suppose that user is unavailable and will be for a long period of time. If it's not feasible to redirect the task to another user, you might just stop it, then restart it when the user is once again available. Or perhaps you have workflows that need to query some external system that will be offline for a while. To prevent errors, you can pause these workflow instances until that system is back online and then resume the workflows. Stopping workflows suspends all workflow tasks (system and user), then immediately starts those tasks up again when the process instance is restarted. In this step, you will also remove a process instance.

Step 2 Tasks

1. **Stop**, then **Restart**, the **StartStop** process instance, observing the Status values after each action.
2. Open the **Process Overview** Report (in K2 Workspace) and confirm you can see the **Deleted** process instance for the **Workflow Administration Sample** process.
3. Returning to the **Instances** screen, **delete** the **Deleted** process instance.
4. Open the **Process Overview** Report, then confirm that the **Deleted** process instance is no longer shown.

Step 2 Walkthrough

a. In the **Instances** screen, click to highlight the **StartStop** process instance. Click the **Stop** button found in the navigation bar. Click **OK** for the confirmation message. Notice the **Status** value after you have stopped the instance.



Selected Filter: Default

Quick Search: All fields

ID	FOLIO	START D...	STATUS	ORIGINATO...	VERSION
12	NormalExecution2	3/25/2016	Active	Denallix Ad...	1
11	NormalExecution1	3/25/2016	Active	Denallix Ad...	1
10	Deleted	3/25/2016	Active	Denallix Ad...	1
9	StartStop	3/25/2016	Active	Denallix Ad...	1
8	GoToActivity	3/25/2016	Active	Denallix Ad...	1
7	TaskRedirect	3/25/2016	Active	Denallix Ad...	1

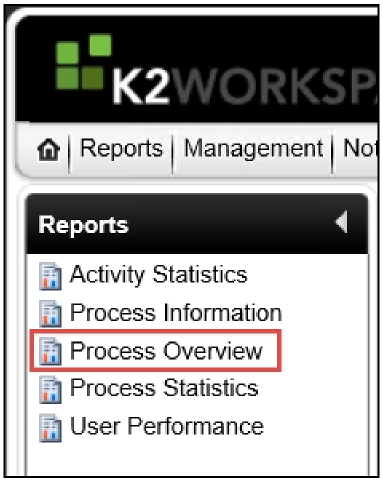
9	StartStop	3/25/2016	Stopped	Denallix Ad...	1

b. Select the **StartStop** process instance once again and **Restart** it. Notice the **Status** value.

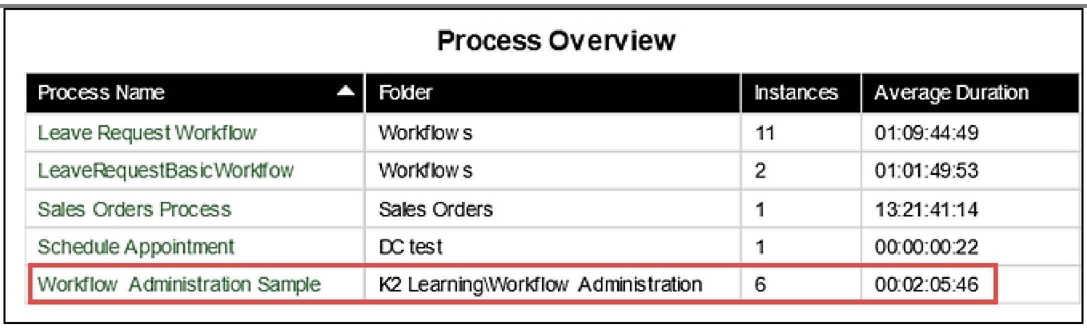
Now you are going to delete a process instance, which will permanently remove it from K2. (You cannot retrieve a process instance if you have deleted it.) To demonstrate this, you will first open the Process Overview report and observe the *Deleted* process instance's current workflow step. The Process Overview report contains current, as well as historical data for process instances. (To learn more about the Process Overview report, access the Reporting in K2 tutorials.)

c. Launch **K2 Workspace**.
(**Start** > **All Programs** > **K2 blackpearl** > **K2 Workspace**) If you do not see the link in your start menu, contact your K2 Administrator to obtain the URL of the K2 Workspace in your environment.

d. From the landing page, click the **Process Overview** Report link found in the Reports pane. (It may take a while for the report to open depending on your environment.)



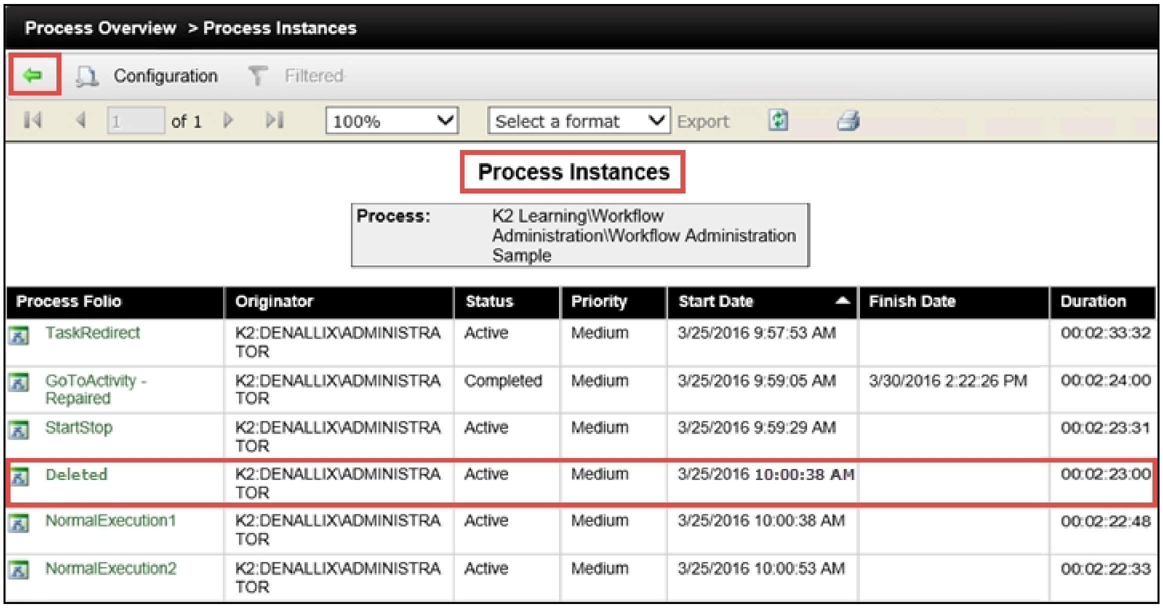
e. When the Process Overview Report opens, you will first see a list of all processes that have been deployed, along with the number of process instances that have been started for each process. (Your screen will not match the image below exactly.) Notice the **Workflow Administration Sample** Process indicates the 6 instances you started in the previous steps. Click on the **Workflow Administration Sample** process to drill-down into its Process Instances report.



Process Overview

Process Name	Folder	Instances	Average Duration
Leave Request Workflow	Workflows	11	01:09:44:49
LeaveRequestBasicWorkflow	Workflows	2	01:01:49:53
Sales Orders Process	Sales Orders	1	13:21:41:14
Schedule Appointment	DC test	1	00:00:00:22
Workflow Administration Sample	K2 Learning\Workflow Administration	6	00:02:05:46

f. On the **Process Instances** screen you will see a list of all the instances for this process. This list will include instances that are currently active, as well as instances that historical (completed). Confirm that the **Deleted** process instance is shown and is currently **Active**. Click the **green back arrow** in the Process Instances navigation bar to return to the Process Overview page.



Process Instances

Process: K2 Learning\Workflow Administration\Workflow Administration Sample

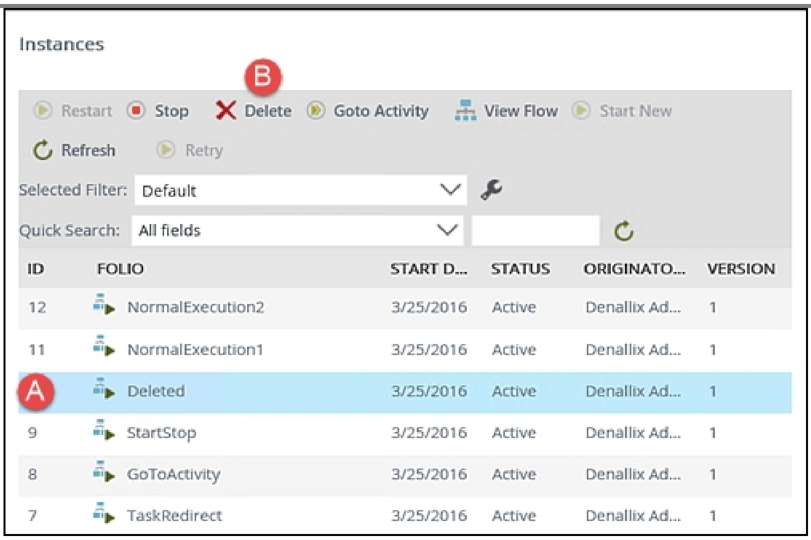
Process Folio	Originator	Status	Priority	Start Date	Finish Date	Duration
TaskRedirect	K2:DENALLIX\ADMINISTRATOR	Active	Medium	3/25/2016 9:57:53 AM		00:02:33:32
GoToActivity - Repaired	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	3/25/2016 9:59:05 AM	3/30/2016 2:22:26 PM	00:02:24:00
StartStop	K2:DENALLIX\ADMINISTRATOR	Active	Medium	3/25/2016 9:59:29 AM		00:02:23:31
Deleted	K2:DENALLIX\ADMINISTRATOR	Active	Medium	3/25/2016 10:00:38 AM		00:02:23:00
NormalExecution1	K2:DENALLIX\ADMINISTRATOR	Active	Medium	3/25/2016 10:00:38 AM		00:02:22:48
NormalExecution2	K2:DENALLIX\ADMINISTRATOR	Active	Medium	3/25/2016 10:00:53 AM		00:02:22:33

Now you will return to the K2 Management site and remove the Deleted process instance from the Instances screen. Once deleted, you will return to the Process Overview report in K2 Workspace and confirm the instance is no longer shown.

Here is an example of why you might want to remove a process instance. Perhaps you have a user that has submitted a workflow and the workflow is currently waiting approval from some destination user. The originator decides at some point they want to cancel the workflow (and hopefully after having informed the destination user they are canceling), so they ask you if you can delete the workflow. You can accomplish this by removing the process instance. This is also another example of how the Folio might be very helpful in locating the correct process instance to remove, if there are many.

g. Return to the **K2 Management** site. (This should just be a matter of clicking the Management tab in your browser if you still have the site open.) If the Instances screen isn't still open, click the **Workflow Server** > **Workflows** node to expose the active processes. Click to highlight the **Workflow Administration Sample** process, then click the **Instances** button to open the Instances screen.

h. Click to highlight the **Deleted** process instance, then click **Delete**. Click **OK** when you see the confirmation dialogue.



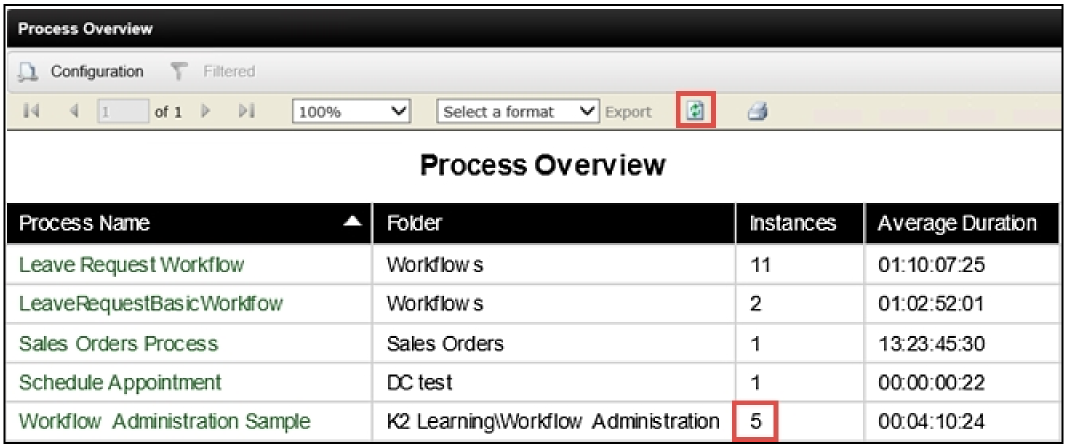
Selected Filter: Default

Quick Search: All fields

ID	FOLIO	START D...	STATUS	ORIGINATO...	VERSION
12	NormalExecution2	3/25/2016	Active	Denallix Ad...	1
11	NormalExecution1	3/25/2016	Active	Denallix Ad...	1
	Deleted	3/25/2016	Active	Denallix Ad...	1
9	StartStop	3/25/2016	Active	Denallix Ad...	1
8	GoToActivity	3/25/2016	Active	Denallix Ad...	1
7	TaskRedirect	3/25/2016	Active	Denallix Ad...	1

The Deleted process instance should no longer be shown on the Instances screen. To confirm the process instance has been completely removed from K2, you will return to the Process Overview reports in K2 Workspace.

i. Return to **K2 Workspace** (it should be a tab in your browser). You should be on the Process Overview page. Notice there are now five (5) process instances indicated for the **Workflow Administration Sample** process. (Click the refresh icon in the navigation bar if necessary.)



Process Overview

Process Name	Folder	Instances	Average Duration
Leave Request Workflow	Workflow s	11	01:10:07:25
LeaveRequestBasicWorkflow	Workflow s	2	01:02:52:01
Sales Orders Process	Sales Orders	1	13:23:45:30
Schedule Appointment	DC test	1	00:00:00:22
Workflow Administration Sample	K2 Learning\Workflow Administration	5	00:04:10:24

j. Click the **Workflow Administration Sample** process to drill down into the process instances. Confirm the **Deleted** process instance is no longer there.

Leave K2 Workspace open (on the Process Instances page).

STEP 2 REVIEW

In this step, you stopped, or paused, the StartStop process instance. Stopping a process instance does not remove it, it merely puts it 'on hold' until the instance is either restarted or deleted. If there are any tasks that have been fired off when the instance was stopped, they will immediately fire off again when the instance is restarted. To permanently delete a process instance from K2, choose the Delete option. Once you delete a process instance, there is no way to retrieve it.

Step 3: Force a workflow to go to another activity

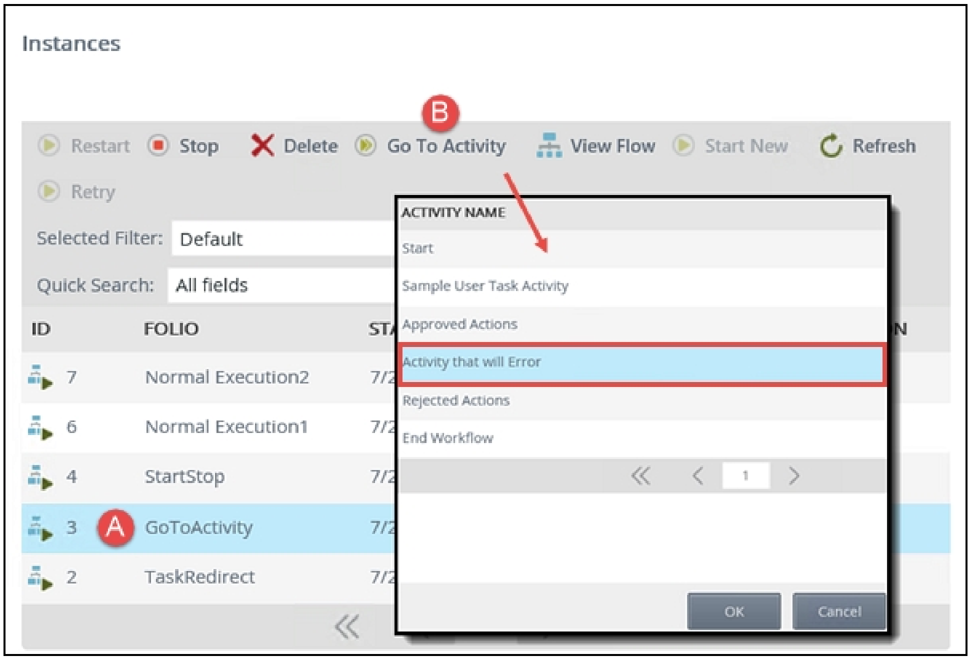
In this step, you will choose the Go To Activity option to send a process instance to another workflow step. For example, suppose a user has submitted a workflow. They contact you at some point to say they need to make a change to their form entry and want to know if there is any possibility of retrieving the form/workflow. This can be accomplished by using the Go To Activity option. The Go To Activity allows you to send the process instance to any of the steps in the workflow.

Step 3 Tasks

1. Return to the K2 Management site. Using the **Go To Activity** option, redirect the **GoToActivity** process instance to the **Activity that will Error** step.

Step 3 Walkthrough

a. Return once again to the K2 Management site. You should still be on the **Instances** screen for the **Workflow Administration Sample** process. (If you are not, click **Workflow Server** > **Workflows**, then click the **Workflow Administration Sample** process from the central pane, then the **Instances** button in the navigation bar.) Click to highlight the **GoToActivity** process instance, then click the **Go To Activity** button in the navigation bar. Select the **Activity that will Error** activity. Click **OK**, then click **OK** when you see the confirmation dialogue. Click **OK** once again to close the success dialogue.



Selected Filter: Default

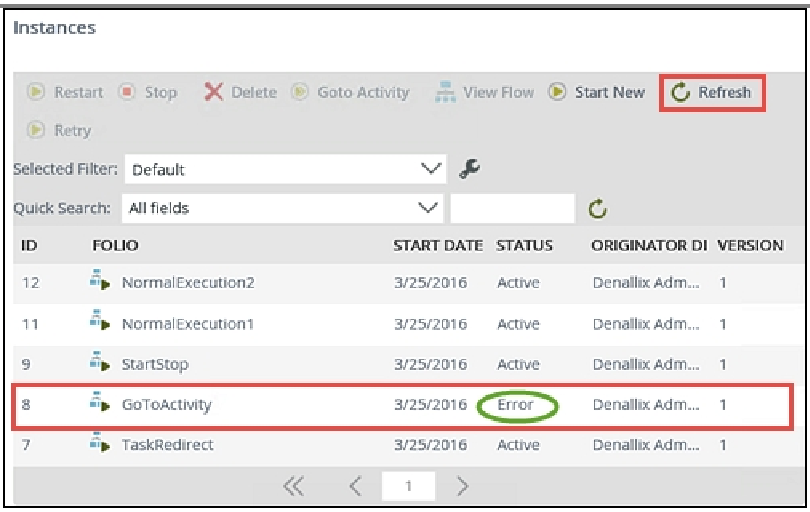
Quick Search: All fields

ID	FOLIO	STA
7	Normal Execution2	7/2
6	Normal Execution1	7/2
4	StartStop	7/2
3	GoToActivity	7/2
2	TaskRedirect	7/2

ACTIVITY NAME
Start
Sample User Task Activity
Approved Actions
Activity that will Error
Rejected Actions
End Workflow

b. **Refresh** the **Instances** screen by clicking the **Refresh** button in the navigation bar. (If you use the refresh button on your browser, you will be redirected back to the Workflow Projects screen. If this happens, simply navigate back to the Instances screen by highlighting the Workflow Administration Sample process, then click the Instances button.)

Notice the **Status** is now showing **Error**. (You deliberately sent this process instance to a step that will throw an error so that you can see how to repair a failed workflow in a future step.) The Instances screen is one area in the K2 Management site that indicates if a process instance is in an error state. You can also view process instances that are in an error state by accessing the **Errors** option also located under the Workflow Server node.



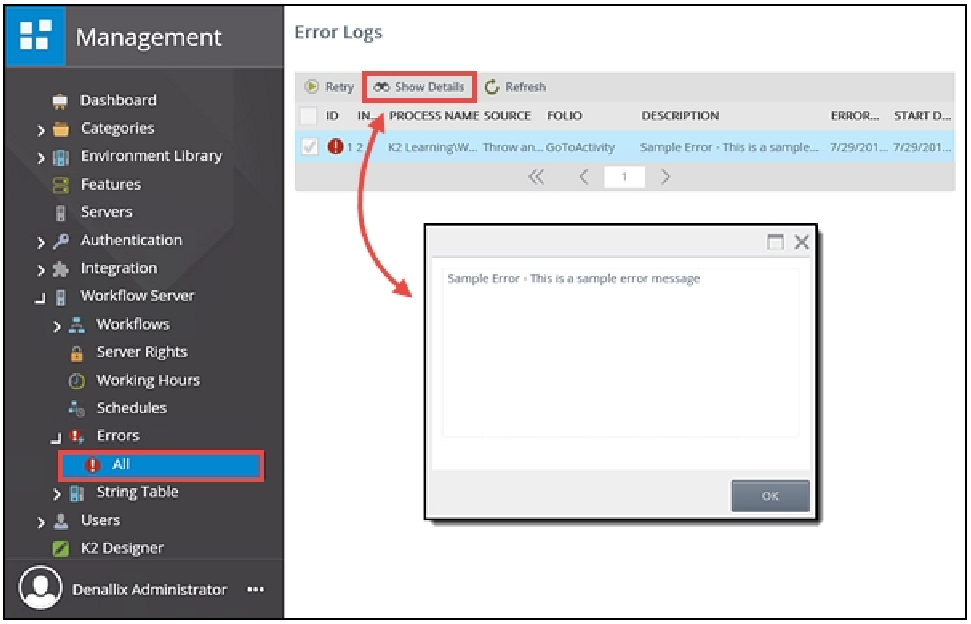
Instances

Selected Filter: Default

Quick Search: All fields

ID	FOLIO	START DATE	STATUS	ORIGINATOR DI	VERSION
12	NormalExecution2	3/25/2016	Active	Denallix Adm...	1
11	NormalExecution1	3/25/2016	Active	Denallix Adm...	1
9	StartStop	3/25/2016	Active	Denallix Adm...	1
8	GoToActivity	3/25/2016	Error	Denallix Adm...	1
7	TaskRedirect	3/25/2016	Active	Denallix Adm...	1

c. In the left column, still under the Workflow Server node, expand the **Errors** option, then click **All**. In this screen, you can view all of the process instances (for all processes) that are in an error state. (Remember, this applies to those processes for which you have permissions.) By selecting the error, then clicking **Show Details**, you will be able to see the full error description. This will give you a starting point for troubleshooting the issue.



Error Logs

ID	IN...	PROCESS NAME	SOURCE	FOLIO	DESCRIPTION	ERROR...	START D...
1	2	K2 Learning\W...	Throw an...	GoToActivity	Sample Error - This is a sample...	7/29/201...	7/29/201...

Sample Error - This is a sample error message

STEP 3 REVIEW

In this step, you used the Go To Activity option to redirect a process instance to a different step in the workflow. You will find this tool very useful for sending workflows back to originators or any other users that need to make a change or addition to their form input. You purposely sent this process instance to a step that will throw an error to demonstrate how the Status column updates to indicate the process instance is in an error state. The Instances screen is one area in the K2 Management site that indicates if a process instance is in an error state. You can also view all process instances that are in an Error state by accessing the Errors screen. Here you will find details about the error that you can use as a starting point in troubleshooting the issue.

Step 4: Repair a workflow

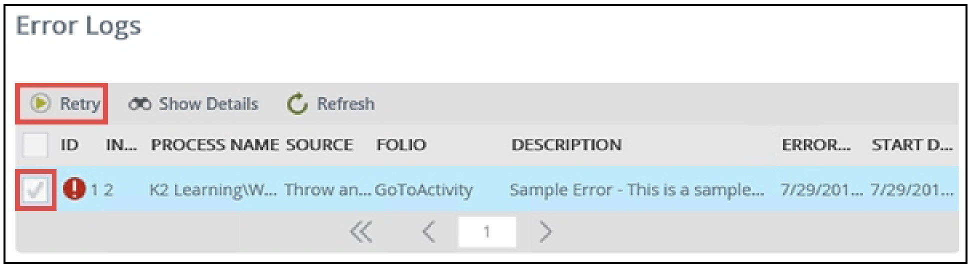
Still on the Errors (All) screen, you will now retry the instance to see if you can repair it. For this tutorial, the Error step has been specifically coded to repair itself the second time it is run. A real-world example however, might be if a task was assigned to a new employee, who did not have an email account set up yet. The error might read something like 'destination user not found'. In this case, you could confirm the email account has been set up, then retry the step. K2 would 'see' the user and move the workflow along.

Step 4 Tasks

1. **Retry** the **K2 Learning\Workflow Administration\Workflow Administration Sample** > **GoToActivity** process instance.
2. Access the **Process Overview** report and confirm the **GoToActivity** process has been repaired and has completed.

Step 4 Walkthrough

a. On the **Error Logs** screen, CHECK the box to the left of the **K2 Learning\Workflow Administration\Workflow Administration Sample** process > **GoToActivity** process instance, then click the **Retry** button. (Depending on your screen size, you may not be able to see the entire process name, but you should be able to see enough of the name to know which process to select. This is also assuming there is more than one process listed.)

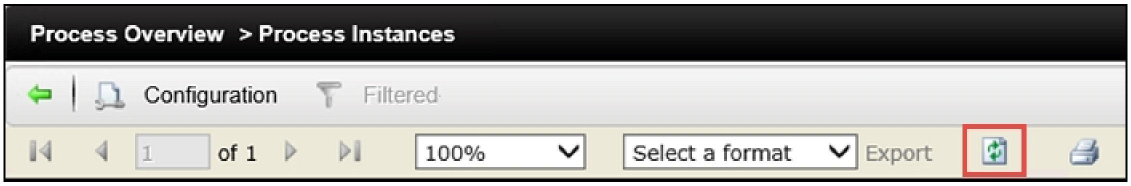


Error Logs

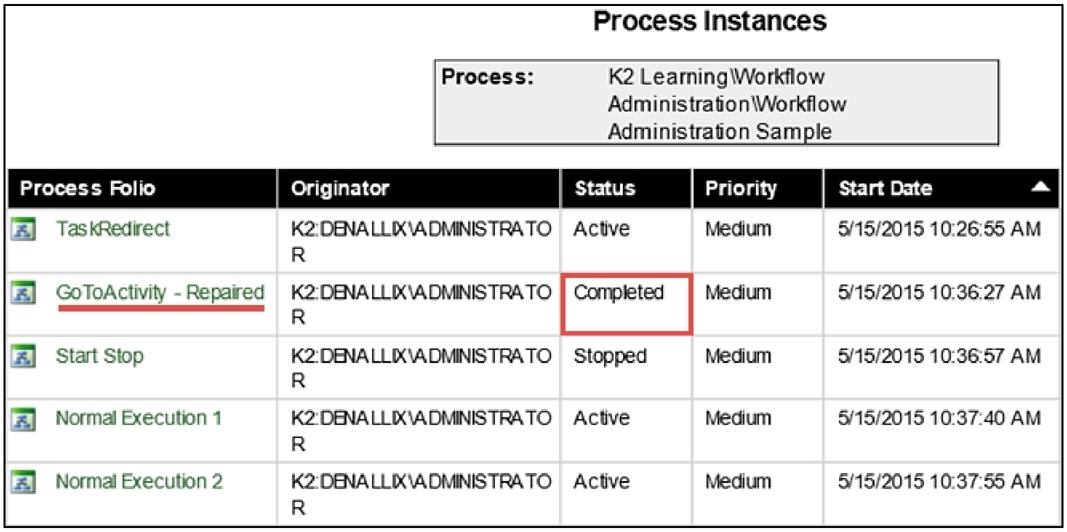
	ID	IN...	PROCESS NAME	SOURCE	FOLIO	DESCRIPTION	ERROR...	START D...
☑	1	2	K2 Learning\W...	Throw an...	GoToActivity	Sample Error - This is a sample...	7/29/201...	7/29/201...

The process should no longer be shown on the Error Logs screen. Now you want to confirm the process has completed. You will do that by returning to the Process Instances report to view the details for this instance.

b. Switch back to **K2 Workspace**. On the Process Instances page, click the **Refresh** icon.



c. Note that the **GoToActivity** process instance is shown as repaired and completed. (Depending on your screen size, you may not be able to see the entire name or status.)



Process Instances

Process: K2 Learning\Workflow Administration\Workflow Administration Sample

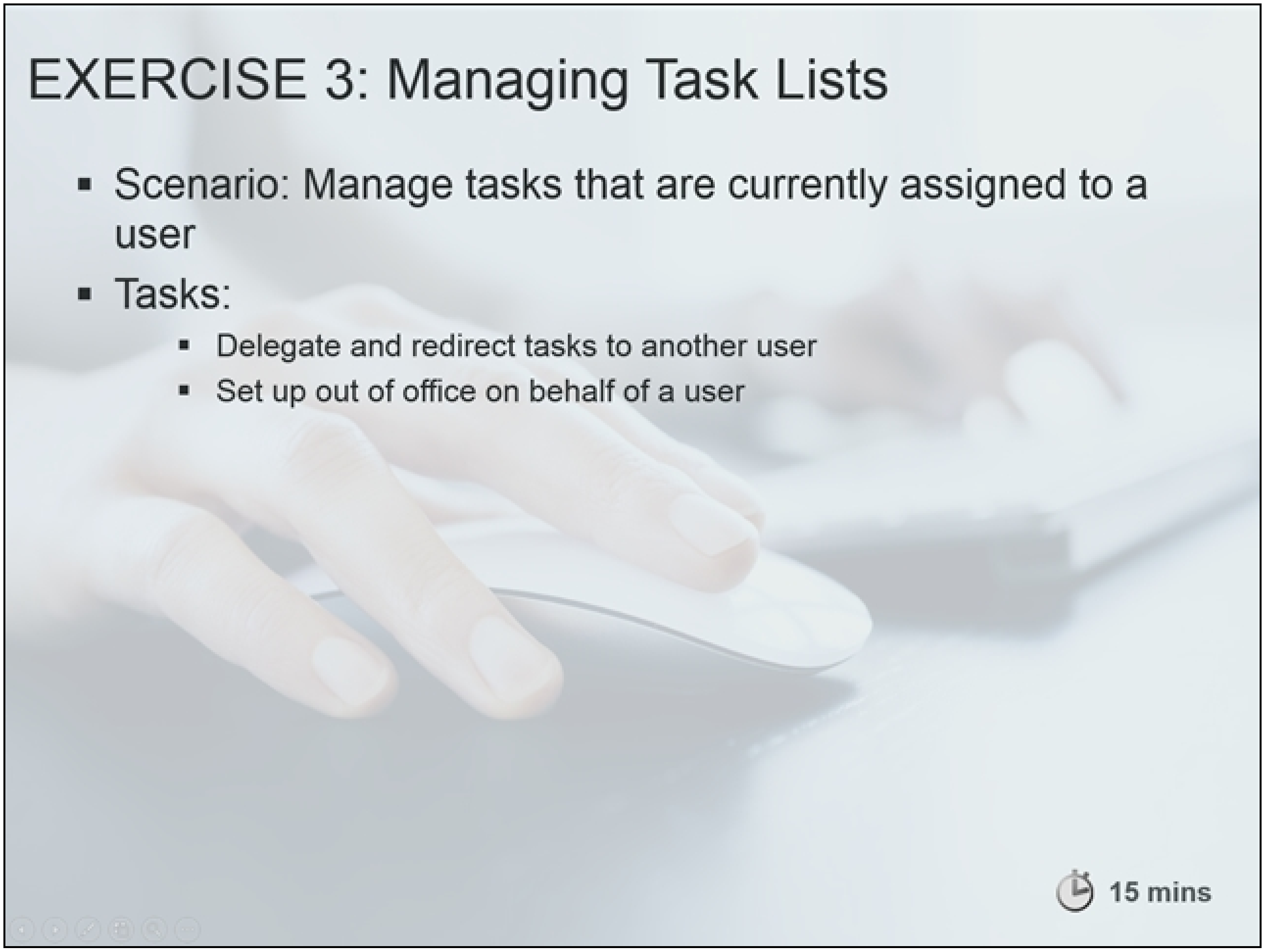
Process Folio	Originator	Status	Priority	Start Date
TaskRedirect	K2:DENALLIX\ADMINISTRATOR	Active	Medium	5/15/2015 10:26:55 AM
GoToActivity - Repaired	K2:DENALLIX\ADMINISTRATOR	Completed	Medium	5/15/2015 10:36:27 AM
Start Stop	K2:DENALLIX\ADMINISTRATOR	Stopped	Medium	5/15/2015 10:36:57 AM
Normal Execution 1	K2:DENALLIX\ADMINISTRATOR	Active	Medium	5/15/2015 10:37:40 AM
Normal Execution 2	K2:DENALLIX\ADMINISTRATOR	Active	Medium	5/15/2015 10:37:55 AM

STEP 4 REVIEW

In this step, you repaired a process instance that was in an error state by using the retry option found on the Error Logs central pane. Retrying the error will execute the same event where the error occurred. Use caution when retrying events as you may have unexpected, and possibly duplicate, results based on what the event was designed to do.

Continue on to the next section, Managing Task Lists, when you are ready.

EXERCISE 3: Managing Task Lists



EXERCISE 3: Managing Task Lists

- Scenario: Manage tasks that are currently assigned to a user
- Tasks:
  - Delegate and redirect tasks to another user
  - Set up out of office on behalf of a user

15 mins

Video

In this exercise, we will explore how to administer K2 tasks that appear on users' task lists. You will learn how to delegate (create a copy of a task) and redirect (move a task) between users. You will also learn how to define an Out-of-Office rule for a user and then view the resulting task assignment.

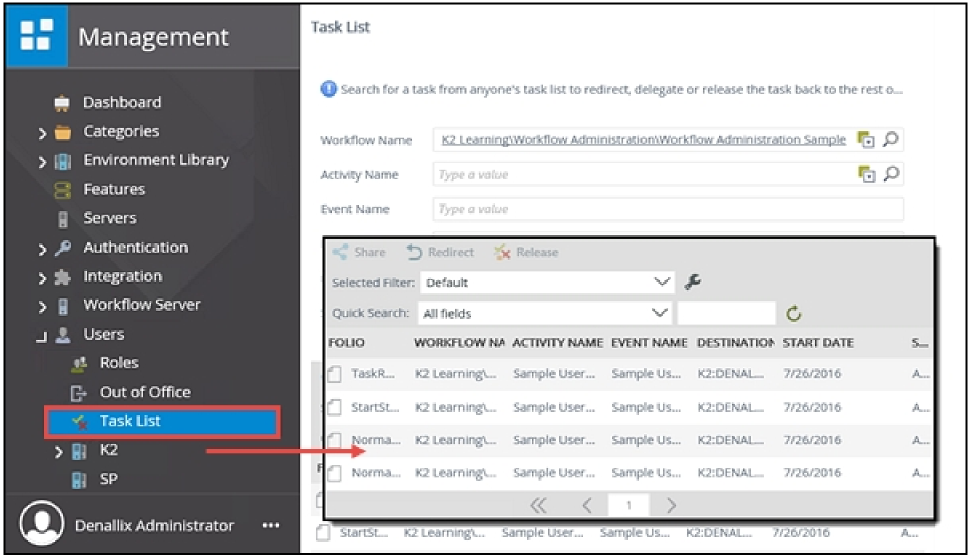
This exercise requires the that the sample workflow from Exercise 1 be deployed and that you started some instances of the workflow as described in Exercise 2. When you are ready, continue on to K2 Administration: Managing Worklists to start this exercise.

Managing Task Lists

One of the most common administrative tasks in K2 is redirecting a process instance from one user to another. There are a number of reasons you might do this, one being the destination user (or the user assigned a task) is not available and the originator wants to move the workflow along prior to them returning. At other times, a destination user might contact you and ask that the workflow be redirected to another user for making the task decision. When you redirect a user task, the task is removed from the original destination user's task list and placed in the target user's task list.

The K2 administrator can see all user tasks currently active, while users can only see the tasks assigned to them. Administrating tasks can be performed from the K2 Management site.

K2 administrator's Task List view from the K2 Management site



Search for a task from anyone's task list to redirect, delegate or release the task back to the rest o...

Workflow Name: K2 Learning\Workflow Administration\Workflow Administration Sample

Activity Name: Type a value

Event Name: Type a value

Selected Filter: Default

Quick Search: All fields

FOLIO	WORKFLOW NA	ACTIVITY NAME	EVENT NAME	DESTINATION	START DATE	S...
TaskR...	K2 Learning\...	Sample User...	Sample Us...	K2:DENAL...	7/26/2016	A...
StartSt...	K2 Learning\...	Sample User...	Sample Us...	K2:DENAL...	7/26/2016	A...
Norma...	K2 Learning\...	Sample User...	Sample Us...	K2:DENAL...	7/26/2016	A...
Norma...	K2 Learning\...	Sample User...	Sample Us...	K2:DENAL...	7/26/2016	A...
StartSt...	K2 Learning\...	Sample User...	Sample Us...	K2:DENAL...	7/26/2016	A...

Note
This tutorial includes steps using the K2 Management site. You must have K2 blackpearl 4.7 or later to complete this tutorial as the steps and screen shots are specific to the K2 Management site.

Note
This tutorial requires that you have deployed the Workflow Administration Sample as described in the Download and deploy a simple K2 package tutorial and have started instances of this workflow as described in the Managing Processes and Process Instances tutorial.

Step 1: Redirect a task from one user to another

In this step, you will redirect a task list item from one user to another. Prior to redirecting the task list item, you will open Internet Explorer for both users and observe their task lists. You want to observe how the task list item is completely removed from the original destination user's task list and subsequently added to the new destination user's task list.

Note
The following exercises are described using a K2-provided virtual machine (VM). If you are working through this tutorial in your own environment, you will need access to each destination user's computer, so that you can observe the results as that user.

Step 1 Tasks

1. In the **K2 Management** site, navigate to the **Task List** option found under the User Management node.
2. Search for the workflow named
   workflow administration sample
   then observe the task list items and who they are assigned to.
3. Open a second instance of **Internet Explorer** for **Jonno** using the **User Shortcuts** > **Legal** > **IE (Jonno)** links. With Jonno's IE the active browser, open another instance of **K2 Workspace**. Observe that Jonno does not have any Workflow Administration Sample tasks assigned to him.
4. Return to the **Administrator's Task List** screen. **Redirect** the task with the **Folio** named **TaskRedirect** to Jonno. Confirm the **Destination** now displays Jonno.
5. Return to **Jonno's Task List** and refresh the page. Confirm Jonno now has a task assigned to him.

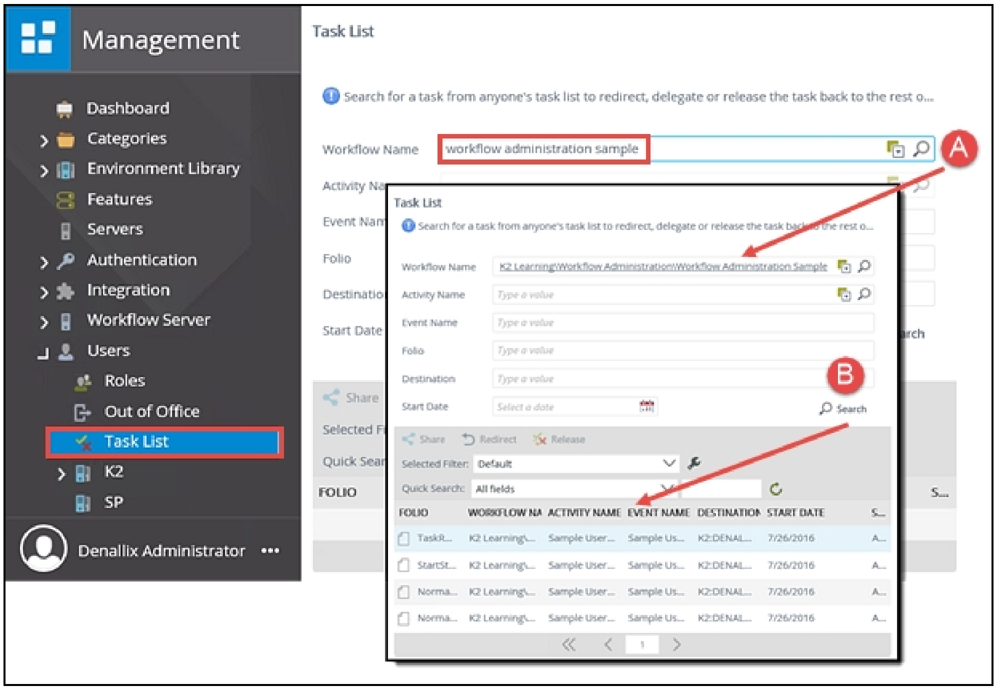
Step 1 Walkthrough

Note
Be sure you are logged in to your computer with a K2 Admin user account. (If you are using a K2-provided VM, you are logged in correctly as Administrator.)

a. Open the **K2 Management** site if it is not already open. Navigate to the **Task List** option found under the **Users** node.

b. Click **Task List** to expose the central pane. In the **Workflow Name** field, enter
workflow administration sample
then click the **search spyglass** to the right of the field. After K2 has found the workflow and replaced your search entry with the system name, click the search spyglass found at the bottom of the search fields (roughly half-way down the screen, on the right-hand side.)

Note
To return a list of *all* active tasks, simply click the Task List spyglass (about half-way down the screen on the right-hand side) without entering any search criteria.



Task List

Search for a task from anyone's task list to redirect, delegate or release the task back to the rest o...

Workflow Name: workflow administration sample

Workflow Name: K2 Learning\Workflow Administration\Workflow Administration Sample

Activity Name: Type a value

Event Name: Type a value

Folio: Type a value

Destination: Type a value

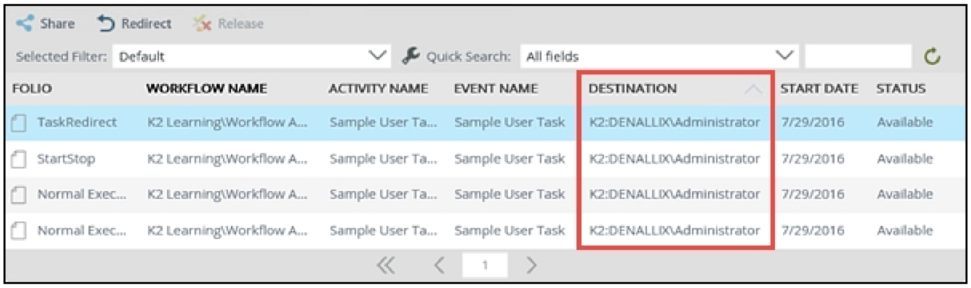
Start Date: Select a date

Selected Filter: Default

Quick Search: All fields

FOLIO	WORKFLOW NA	ACTIVITY NAME	EVENT NAME	DESTINATION	START DATE	S...
TaskR...	K2 Learning\...	Sample User...	Sample Us...	K2:DENAL...	7/26/2016	A...
StartSt...	K2 Learning\...	Sample User...	Sample Us...	K2:DENAL...	7/26/2016	A...
Norma...	K2 Learning\...	Sample User...	Sample Us...	K2:DENAL...	7/26/2016	A...
Norma...	K2 Learning\...	Sample User...	Sample Us...	K2:DENAL...	7/26/2016	A...

You should now see all of the tasks currently assigned for the Workflow Administration Sample process instances. Notice the destination users are all the same, because the same user started the workflow (in our sample, "Administrator" started the process instances).



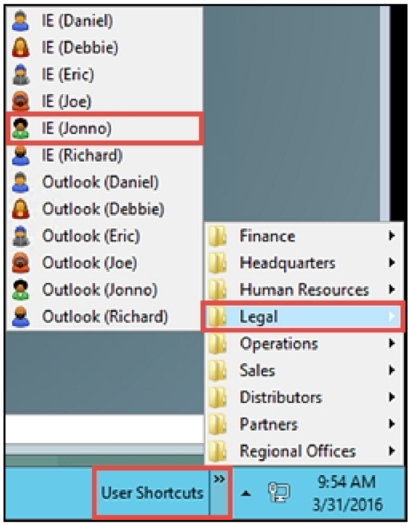
Selected Filter: Default

Quick Search: All fields

FOLIO	WORKFLOW NAME	ACTIVITY NAME	EVENT NAME	DESTINATION	START DATE	STATUS
TaskRedirect	K2 Learning\Workflow A...	Sample User Ta...	Sample User Task	K2:DENALLIX\Administrator	7/29/2016	Available
StartStop	K2 Learning\Workflow A...	Sample User Ta...	Sample User Task	K2:DENALLIX\Administrator	7/29/2016	Available
Normal Exec...	K2 Learning\Workflow A...	Sample User Ta...	Sample User Task	K2:DENALLIX\Administrator	7/29/2016	Available
Normal Exec...	K2 Learning\Workflow A...	Sample User Ta...	Sample User Task	K2:DENALLIX\Administrator	7/29/2016	Available

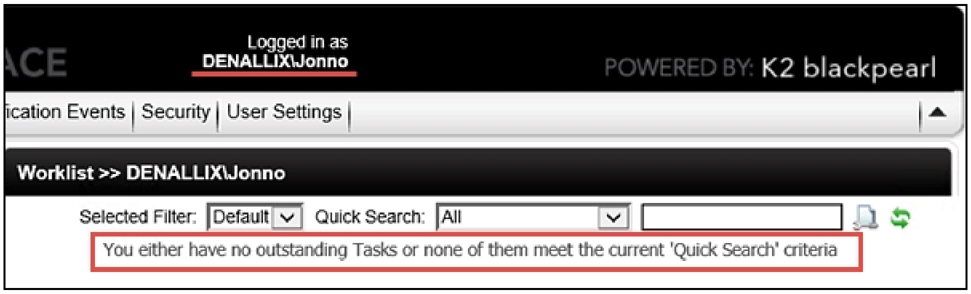
c. You will now open a second instance of **Internet Explorer**, logged in as Jonno. Click on the **User Shortcuts** link found in the lower right corner of your screen. Then expand the **Legal** folder and finally, click on **IE (Jonno)**. A second instance of IE should open.

Note
If you are working within your own environment, you will not have the user shortcuts and will need to access the target user's computer for the following steps.



d. With Jonno's IE the active browser, launch another instance of **K2 Workspace** so that it opens in Jonno's browser. (**Start** > **All Programs** > **K2 blackpearl** > **K2 Workspace**)

Notice that Jonno does not have any Workflow Administration Sample tasks.



Logged in as
DENALLIX\Jonno

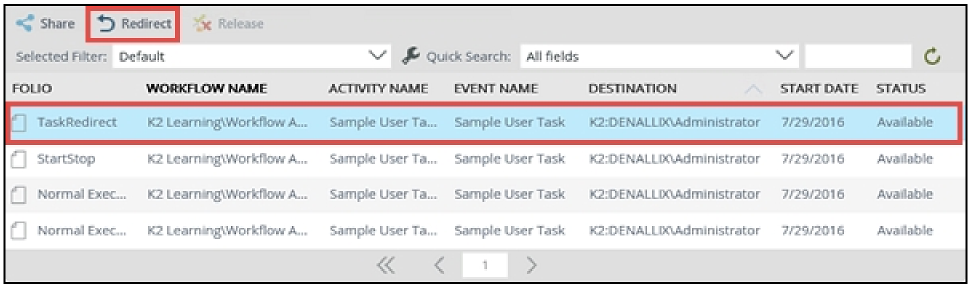
Worklist >> DENALLIX\Jonno

Selected Filter: Default

Quick Search: All

You either have no outstanding Tasks or none of them meet the current 'Quick Search' criteria

e. Return to **Administrator's K2 Management** site. Navigate to the **Workflow Administration Sample Task List**. Click to highlight the **TaskRedirect** process instance, then click the **Redirect** button in the navigation pane.



Selected Filter: Default

Quick Search: All fields

FOLIO	WORKFLOW NAME	ACTIVITY NAME	EVENT NAME	DESTINATION	START DATE	STATUS
TaskRedirect	K2 Learning\Workflow A...	Sample User Ta...	Sample User Task	K2:DENALLIX\Administrator	7/29/2016	Available
StartStop	K2 Learning\Workflow A...	Sample User Ta...	Sample User Task	K2:DENALLIX\Administrator	7/29/2016	Available
Normal Exec...	K2 Learning\Workflow A...	Sample User Ta...	Sample User Task	K2:DENALLIX\Administrator	7/29/2016	Available
Normal Exec...	K2 Learning\Workflow A...	Sample User Ta...	Sample User Task	K2:DENALLIX\Administrator	7/29/2016	Available

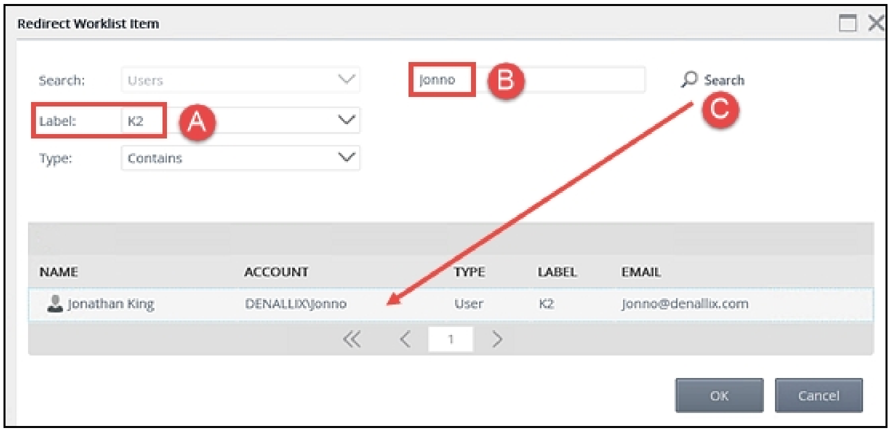
f. The Redirect Worklist Item screen opens. Here, you have a number of options to use in searching for the target user. Begin by confirming the **Label** is set to **K2**.

Note
When configured, K2 can interact with a number of third-party products, such as CRM and SharePoint. You can narrow your search to one of these data sources if desired. You can also change the search type to Starts with, Equal to or Ends with to help further define the search criteria.

You will be searching for Jonno, so in the **Search** field, enter

Jonno

then click the **Search** button or spyglass. When Jonno's name shows up in the results pane, highlight his name, then click **OK**. Click **OK** once again to confirm the action.



Redirect Worklist Item

Search: Users

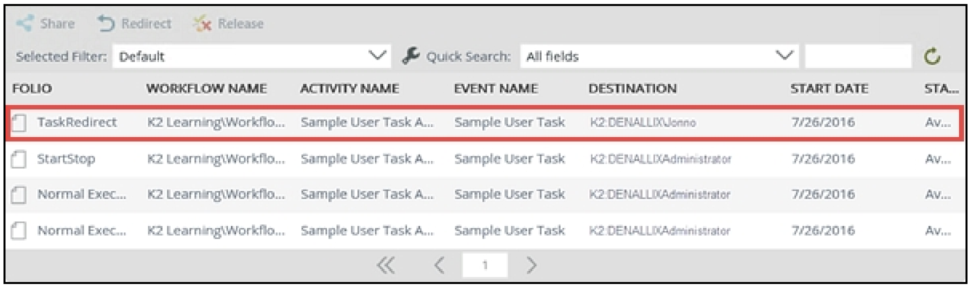
Label: K2

Type: Contains

Jonno

NAME	ACCOUNT	TYPE	LABEL	EMAIL
Jonathan King	DENALLIX\Jonno	User	K2	Jonno@denallix.com

Notice the Task List now shows Jonno as the destination for the **TaskRedirect** process instance.



Selected Filter: Default

Quick Search: All fields

FOLIO	WORKFLOW NAME	ACTIVITY NAME	EVENT NAME	DESTINATION	START DATE	STA...
TaskRedirect	K2 Learning\Workflo...	Sample User Task A...	Sample User Task	K2:DENALLIX\Jonno	7/26/2016	Av...
StartStop	K2 Learning\Workflo...	Sample User Task A...	Sample User Task	K2:DENALLIX\Administrator	7/26/2016	Av...
Normal Exec...	K2 Learning\Workflo...	Sample User Task A...	Sample User Task	K2:DENALLIX\Administrator	7/26/2016	Av...
Normal Exec...	K2 Learning\Workflo...	Sample User Task A...	Sample User Task	K2:DENALLIX\Administrator	7/26/2016	Av...

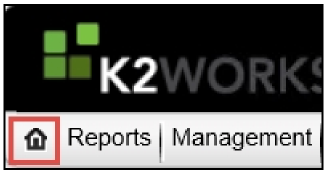
Before you move on to viewing Jonno's task list, let's confirm that the TaskRedirect task is no longer in Administrator's task list.

g. Switch to the **Administrator's K2 Workspace**.

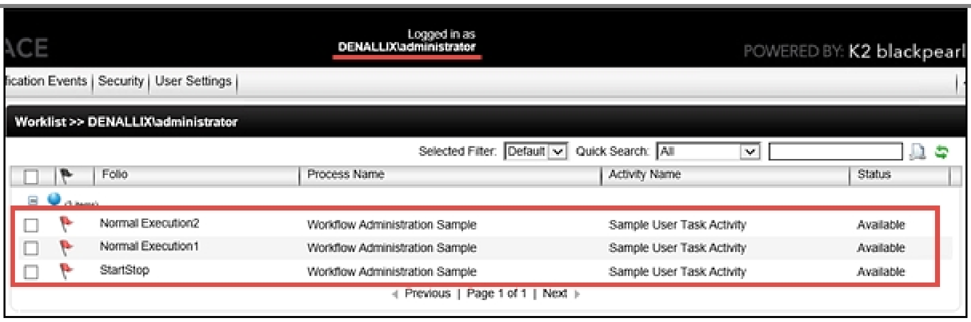
Tip

This should just be a matter of clicking the Workspace tab in your browser. If you closed K2 Workspace after the Process Instances tutorial, launch K2 Workspace once again from the start menu. Make sure that Administrator's IE is the active browser before opening Workspace.

h. If Administrator's K2 Workspace is still open from the last exercise and you are still on the Process Instances screen, click the **home** icon to return to the landing page.



On the K2 Workspace landing page for Administrator, notice the three tasks that remain as part of the Workflow Administration Sample process instances. The redirected task is no longer on Administrator's task list.



ACE

Logged in as DENALLIX\administrator

POWERED BY: K2 blackpearl

ication Events | Security | User Settings |

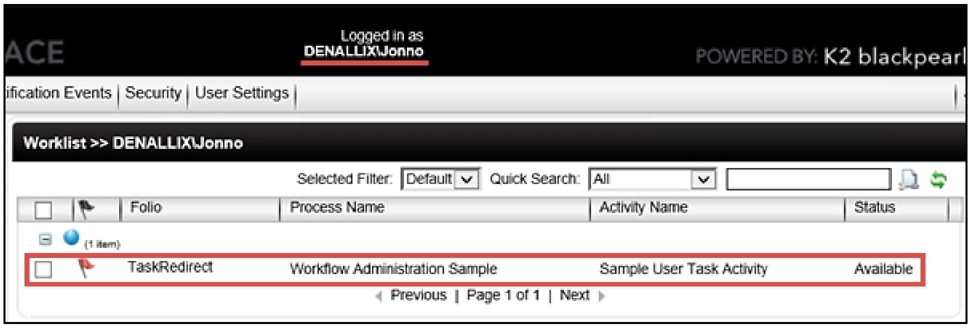
Worklist >> DENALLIX\administrator

Selected Filter: Default Quick Search: All

□		Folio	Process Name	Activity Name	Status
□		Normal Execution2	Workflow Administration Sample	Sample User Task Activity	Available
□		Normal Execution1	Workflow Administration Sample	Sample User Task Activity	Available
□		StartStop	Workflow Administration Sample	Sample User Task Activity	Available

Previous | Page 1 of 1 | Next

i. Return to **Jonno's** K2 Workspace. On the landing page, there should be one task listed, **TaskRedirect**. If you do not see the task listed, refresh the browser.



ACE

Logged in as DENALLIX\Jonno

POWERED BY: K2 blackpearl

ification Events | Security | User Settings |

Worklist >> DENALLIX\Jonno

Selected Filter: Default Quick Search: All

(1 item)

□		Folio	Process Name	Activity Name	Status
□		TaskRedirect	Workflow Administration Sample	Sample User Task Activity	Available

Previous | Page 1 of 1 | Next

STEP 1 REVIEW

In this step, you redirected a task from Administrator to Jonno. Redirecting tasks is a very common request made to the K2 administrator. There are many reasons why a user might want to redirect a task to another user. This is a simple process for the K2 administrator. Redirecting a task completely removes it from the original destination user's task list. The new destination user will receive any notifications that were configured in the workflow. (The workflow simply restarts the event using the new destination user.)

Step 2: Set up Out of Office

Out of Office allows a user to share their tasks with another user. Sharing, or delegating, a task differs from redirecting tasks in that the task is available to *both* users. It is not removed from the original user's task list. Aside from being unavailable, another very common use of the Out of Office feature is for managers or executives, who may want to share their task actions with an assistant. In many cases, the destination user assigned in the workflow is the "manager" and this would include top-level executives as well. By setting up the Out of Office feature (once), they can effectively reassign all of their workflow tasks automatically.

K2 administrators can manage Out of Office settings for all users from the K2 Management site. Users can also manage their own Out of Office settings from their K2 Workspace. In this exercise, you will configure Out of Office settings for Administrator, with Jonno as the task recipient. You will configure the Out of Office settings from the K2 Management site. Then, you will configure Jonno's Out of Office settings, using his K2 Workspace.

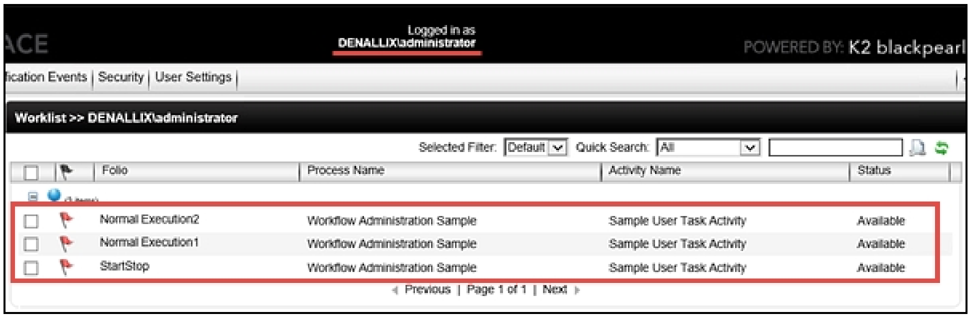
Step 2 Tasks

1. Make a note of the current tasks waiting in **Administrator’s Task List**.
2. Using the K2 Management site, configure Administrator’s **Out of Office** so that all tasks are forwarded to **Jonno**. (You do not need to apply any exceptions.) Be sure to make Administrator currently 'Out of the Office'.
3. Open **Jonno's K2 Workspace** and confirm Administrator's tasks now appear in Jonno's task list. (You may need to refresh the browser to see the changes.)
4. Access the **Settings** for Jonno and configure his **Out of Office** so that his tasks are assigned to **Bob**.
5. Open **Bob's K2 Workspace** and confirm Jonno's tasks now appear in Bob's task list.

6. In the K2 Management site, add an Out of Office **Exception** to Jonno's settings so that the **Sample User Task Activity** from the **Workflow Administration Sample** is shared with **Anthony**.

Step 2 Walkthrough

a. Return to the **K2 Workspace** landing page for **Administrator**. Make a note of the current tasks available.



Logged in as DENALLIX\administrator

POWERED BY: K2 blackpearl

Worklist >> DENALLIX\administrator

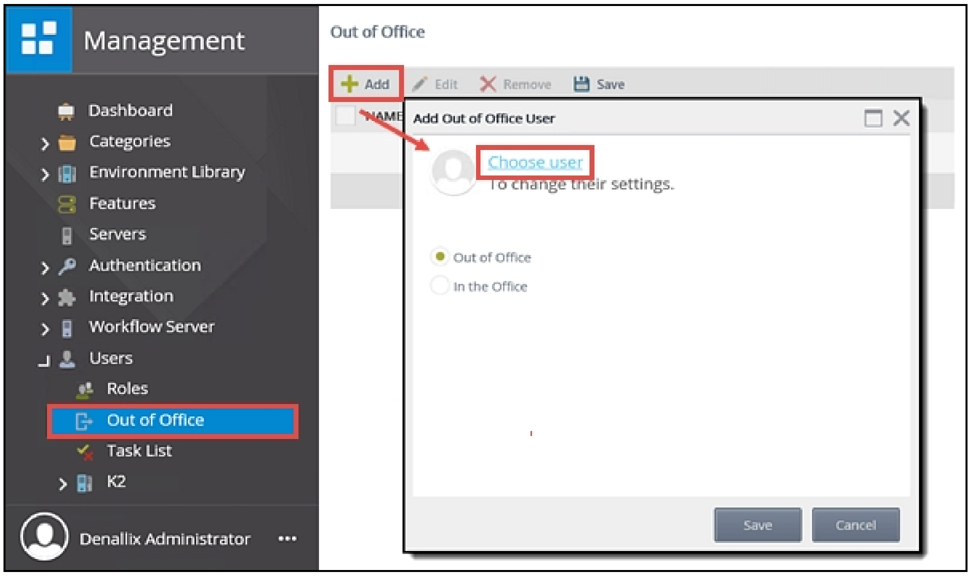
Selected Filter: Default Quick Search: All

		Folio	Process Name	Activity Name	Status
□		Normal Execution2	Workflow Administration Sample	Sample User Task Activity	Available
□		Normal Execution1	Workflow Administration Sample	Sample User Task Activity	Available
□		StartStop	Workflow Administration Sample	Sample User Task Activity	Available

ACE
ication Events | Security | User Settings |
Previous | Page 1 of 1 | Next

Now you will configure Administrator’s Out of Office to forward any tasks to Jonno. There are three steps involved. First, you will select the user you are setting up the out of office for (Administrator), second, you will select the user to forward tasks to (Jonno) and third, you will change Administrator's current status from *In the Office* to *Out of Office*.

b. Returning to the K2 Management site, expand the **Users** node, then click **Out of Office**. The Out of Office details are displayed in the central pane. Click **Add**. On the Add Out of Office User screen, click the **Choose user** link.

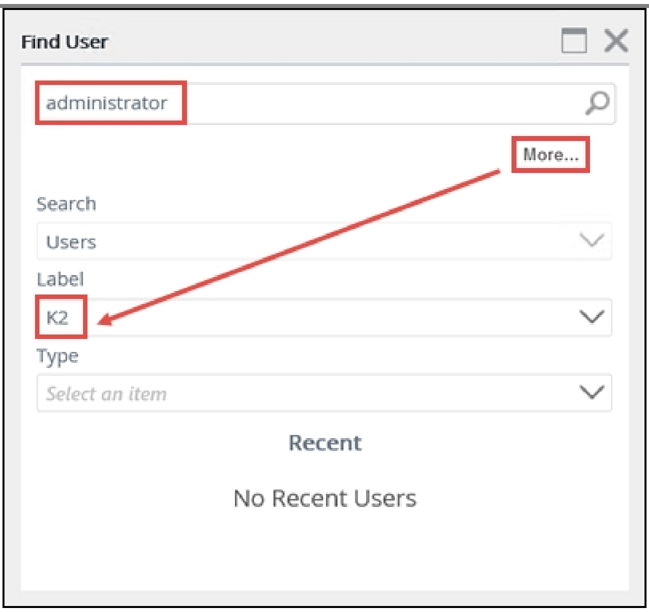


Add Out of Office User

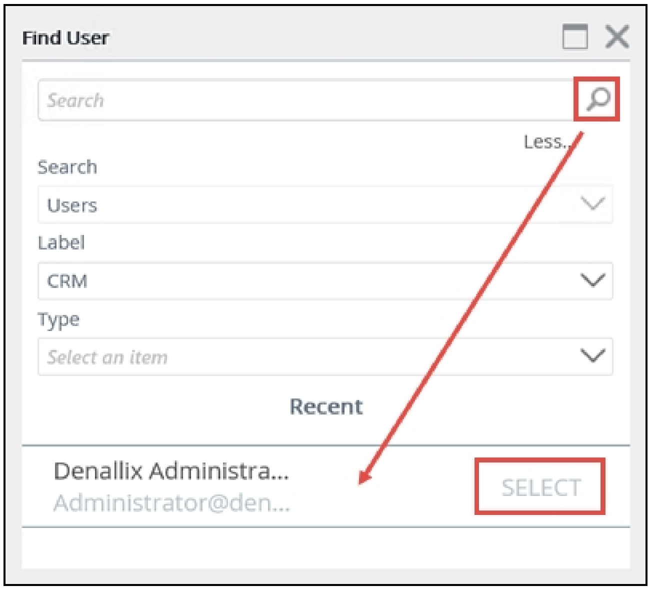
Choose user
To change their settings.

● Out of Office
○ In the Office

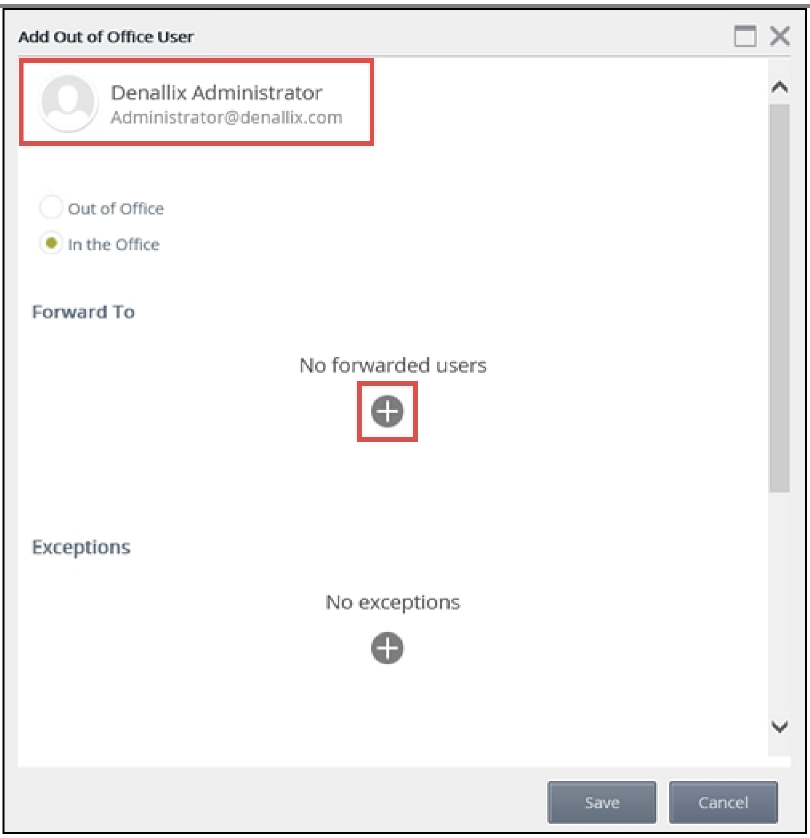
c. On the Find User screen, type
administrator
into the text box. Click the **More...** link and confirm the **Label** is set to **K2**.



d. Click the **spyglass**. You should now see the search results just below the search criteria properties. Click **SELECT** to add Denallix Administrator.



Back on the Add Out of Office User screen, notice now that Administrator is displayed as the user you are creating settings for. Now you will add Jonno as the recipient of Administrator's redirected tasks. Notice that the Forward To and Exceptions panes are now displayed.



Add Out of Office User

Denallix Administrator
Administrator@denallix.com

○ Out of Office
● In the Office

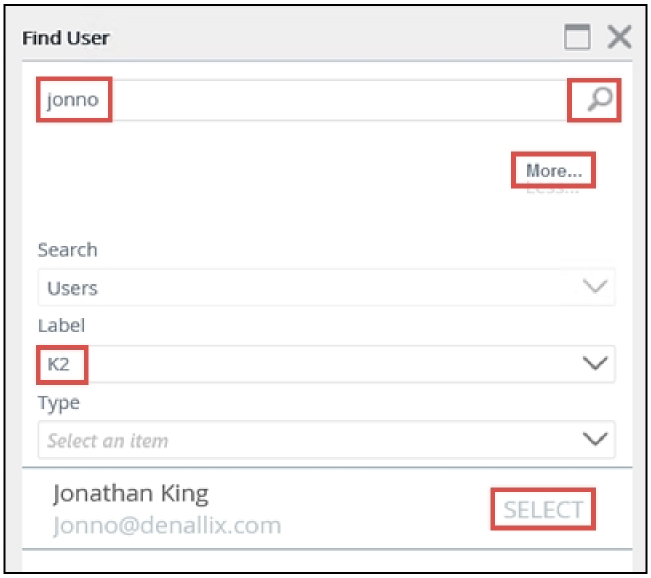
Forward To

No forwarded users

Exceptions

No exceptions

e. Click the **plus (+)** sign in the **Forward To** section. On the Find User screen, enter *jonno* into the text box, then click the **More...** link. Confirm the **Label** is set to **K2**, then click the **spyglass**. After Jonathan King's name is displayed, click **SELECT**.



Find User

jonno

Search: Users

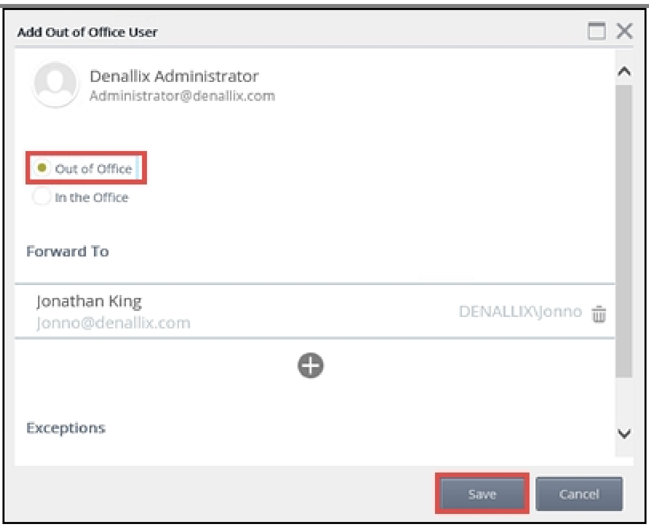
Label: K2

Type: Select an item

Jonathan King
Jonno@denallix.com

f. The final step is to set the current status for Administrator from *In the Office* to *Out of Office*.

g. On the Add Out of Office User screen, change Administrator's status to **Out of Office**, then click **Save**. You will not be configuring any exceptions for this user.



Add Out of Office User

Denallix Administrator
Administrator@denallix.com

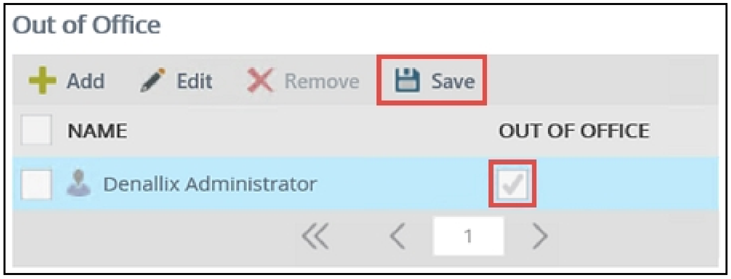
● Out of Office
○ In the Office

Forward To

Jonathan King
Jonno@denallix.com DENALLIX\Jonno

Exceptions

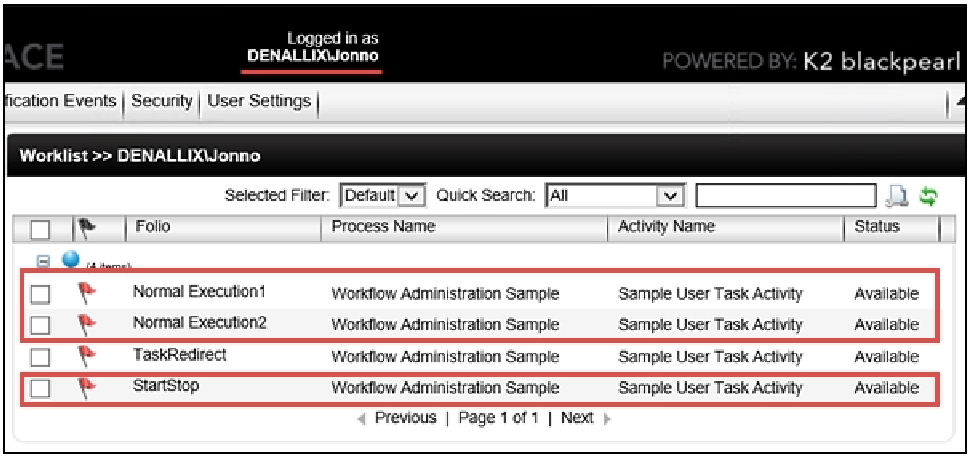
h. You are returned to the Out of Office screen. Confirm that the OUT OF OFFICE box is CHECKED for Administrator. If it is not, CHECK the **OUT OF OFFICE** box, then click **Save** once again.



Out of Office

	NAME	OUT OF OFFICE
□	Denallix Administrator	☑

i. At this point, all of Administrator's tasks should be shared with Jonno. Open Jonno's **K2 Workspace** and confirm the tasks from Administrator are assigned to Jonno as well. (You may need to refresh the browser.)



ACE
Logged in as DENALLIX\Jonno
POWERED BY: K2 blackpearl
fication Events | Security | User Settings |

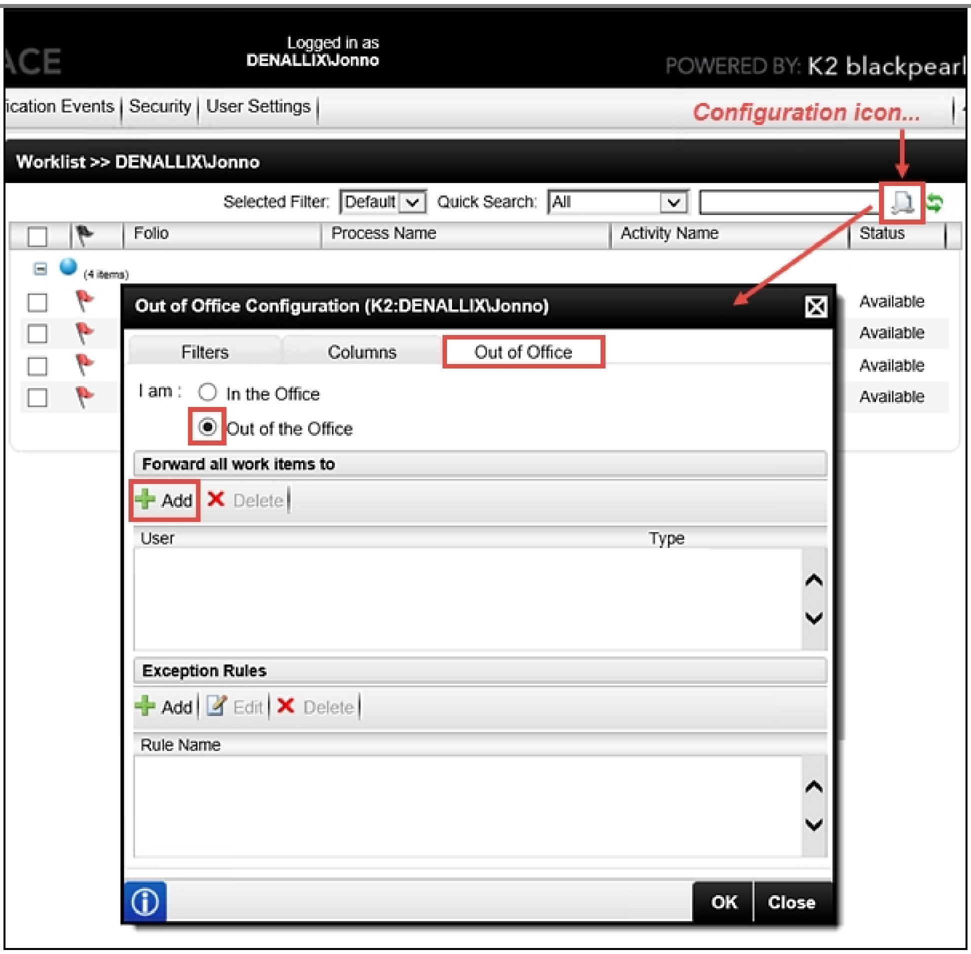
Worklist >> DENALLIX\Jonno

Selected Filter: Default Quick Search: All

□		Folio	Process Name	Activity Name	Status
		(4 items)			
□		Normal Execution1	Workflow Administration Sample	Sample User Task Activity	Available
□		Normal Execution2	Workflow Administration Sample	Sample User Task Activity	Available
□		TaskRedirect	Workflow Administration Sample	Sample User Task Activity	Available
□		StartStop	Workflow Administration Sample	Sample User Task Activity	Available

Now you will configure Out of Office for Jonno. In the image above, notice that in addition to Administrator's tasks, Jonno also has a task of his own, TaskRedirect. By configuring Jonno's Out of Office, you will see how Jonno's task is shared, while Administrator's tasks are not.

j. Click the **Configuration** icon in Jonno's K2 Workspace. Click the **Out of Office** tab. Change Jonno's status to **Out of Office**, then click the **Add** button in the **Forward all work items to** pane.



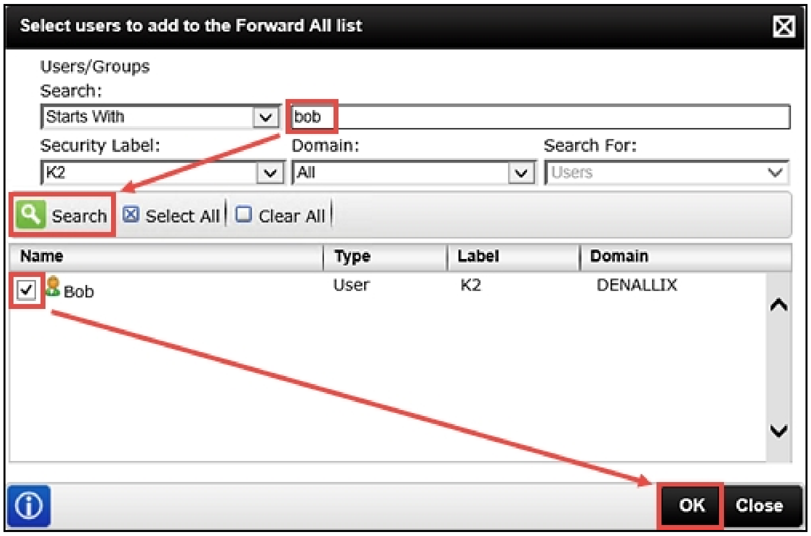
Selected Filter: Default

Quick Search: All

I am : ○ In the Office

● Out of the Office

k. Enter
bob
into the search text box and click the **Search** button. When you see Bob's name displayed, CHECK the box to the left of his name and click **OK**. If you see a confirmation dialog, click **OK**.



Users/Groups Search: Starts With | bob

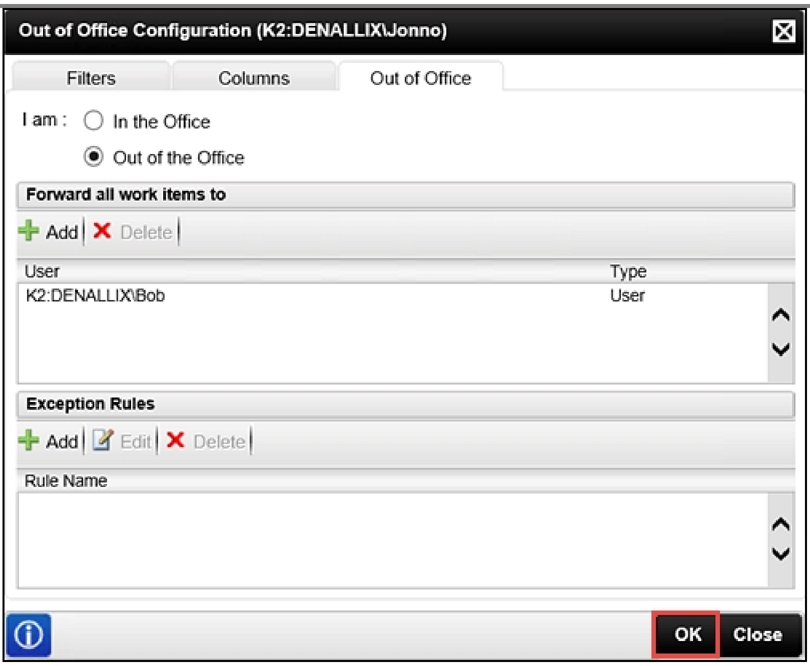
Security Label: K2

Domain: All

Search For: Users

Name	Type	Label	Domain
☑ Bob	User	K2	DENALLIX

l. On the Out of Office Configuration page, click **OK** twice to commit the setting.



Out of Office Configuration (K2:DENALLIX\Jonno)

Filters | Columns | Out of Office

I am : ○ In the Office

● Out of the Office

Forward all work items to

Add | Delete

User	Type
K2:DENALLIX\Bob	User

Exception Rules

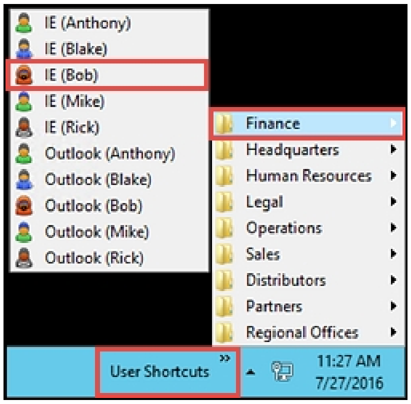
Add | Edit | Delete

Rule Name

OK | Close

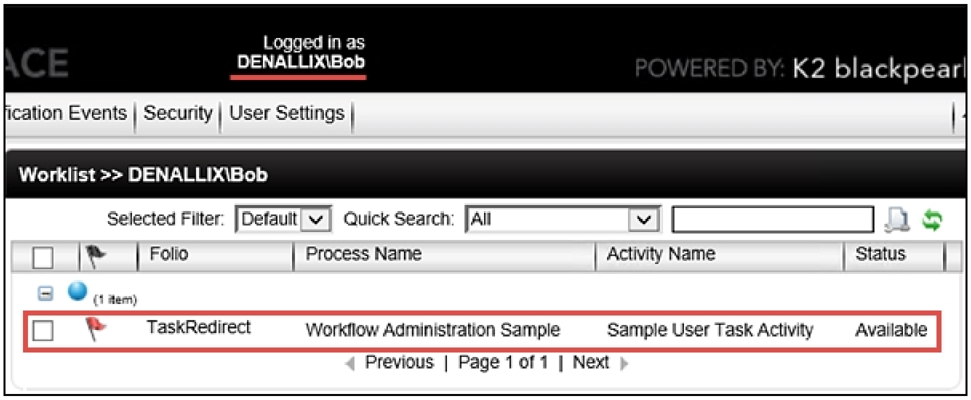
Now you will open Bob's K2 Workspace and confirm that Jonno's task is now shared with him.

m. Use the **User Shortcuts** to open **Bob's IE**. Bob is a member of the Finance group.



n. With Bob's IE the active browser, launch another instance of **K2 Workspace**. (Start > All Programs > K2 blackpearl > K2 Workspace)

Notice that Jonno's **TaskRedirect** task is now shared with Bob.



Logged in as DENALLIX\Bob

POWERED BY: K2 blackpearl

ication Events | Security | User Settings |

Worklist >> DENALLIX\Bob

Selected Filter: Default Quick Search: All

☐		Folio	Process Name	Activity Name	Status
☐		TaskRedirect	Workflow Administration Sample	Sample User Task Activity	Available

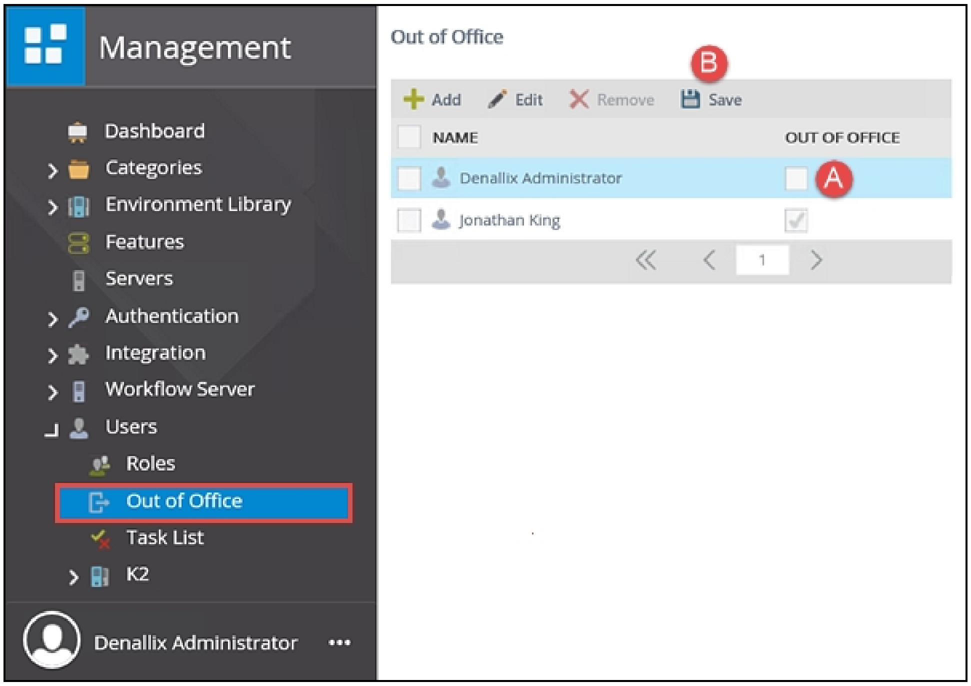
(1 item)

Previous | Page 1 of 1 | Next

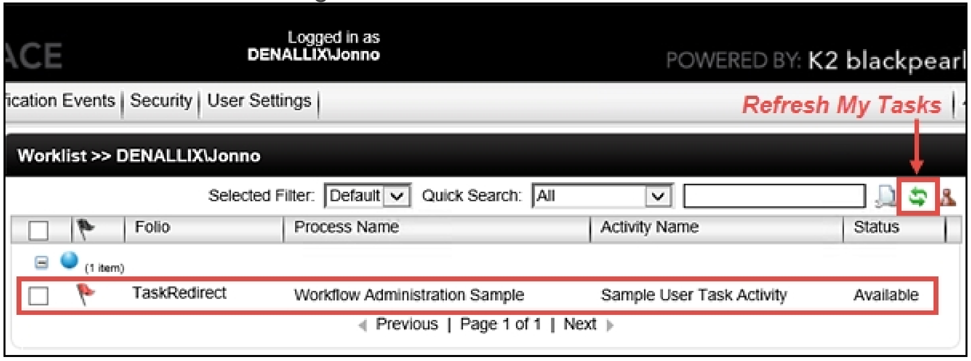
You've learned how a user can share their tasks by configuring their Out of Office settings. As a K2 administrator, you can manage Out of Office settings for other users, in addition to your own. You can add, edit and remove Out of Office settings from the K2 Management site.

Note
If you have a user that frequently switches from In Office to Out of Office, one option for quickly managing their settings is to not remove them (when In Office), but to simply UNCHECK the Out of Office check-box (then Save) to toggle back and forth. This will save a great deal of time if your users do not manage their own Out of Office settings.

o. Switch to Administrator's browser and access the **K2 Management** site. Expand the **Users** node, then click the **Out of Office** option. Notice in the central pane, there are two users listed, Denallix Administrator and Jonathan King (Jonno). (You may need to refresh the browser.) Each time a user configures their own Out of Office settings, the Administrator will see (and be able to manage) their configuration from the K2 Management site. UNCHECK the Out of Office box for Administrator. Click **Save**.



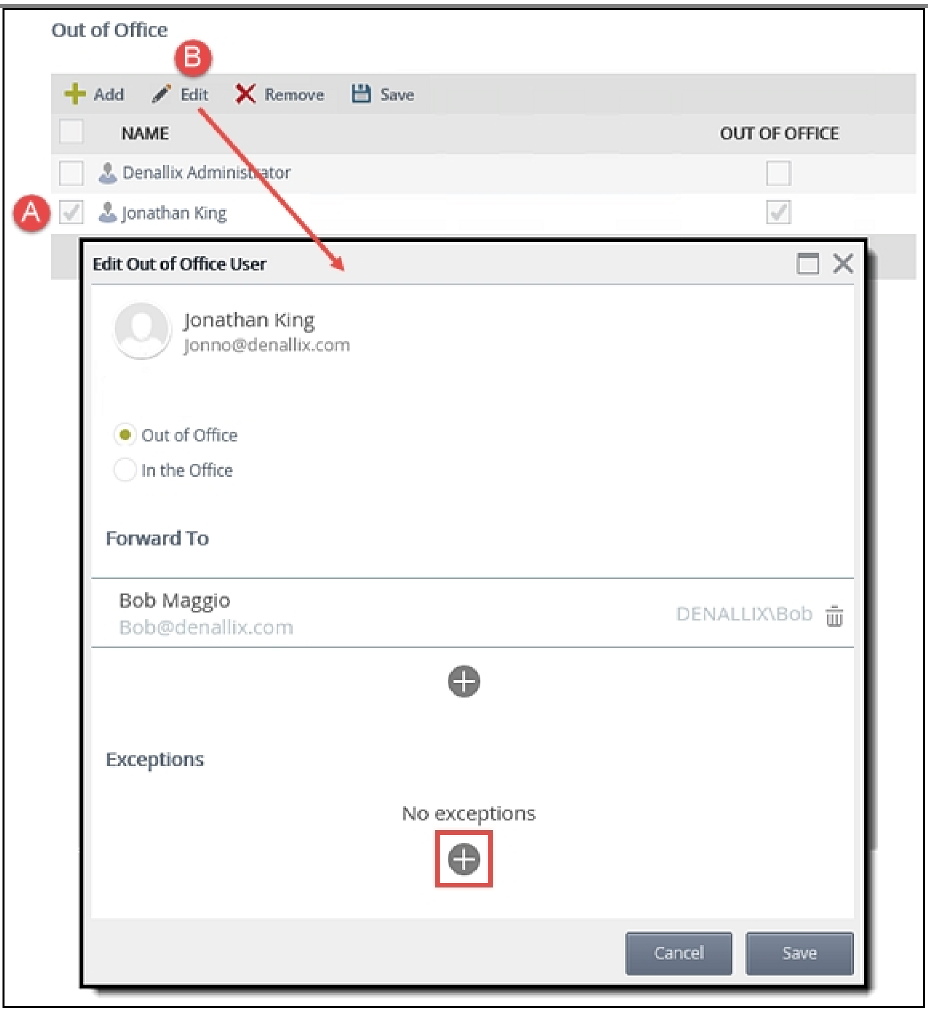
p. Once again, switch to **Jonno's IE** and **K2 Workspace**. Click the **Refresh My Tasks** icon. Notice now, Administrator's tasks are no longer shared with Jonno.


ACE
Logged in as
DENALLIX\Jonno
POWERED BY: K2 blackpearl
ication Events | Security | User Settings |
Refresh My Tasks
Worklist >> DENALLIX\Jonno
Selected Filter: Default
Quick Search: All
(1 item)
Previous | Page 1 of 1 | Next

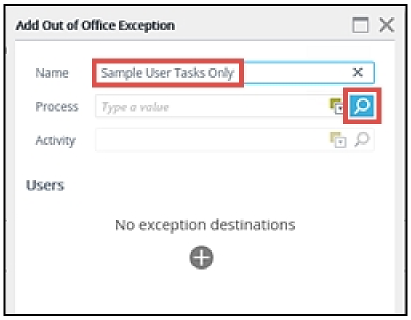
□		Folio	Process Name	Activity Name	Status
□		TaskRedirect	Workflow Administration Sample	Sample User Task Activity	Available

One last concept to explore when configuring Out of Office settings is exceptions. By default, all processes are shared with the target user when configuring Out of Office. Exceptions allow users or administrators to specify individual processes or process activities to be shared, rather than having all processes shared.

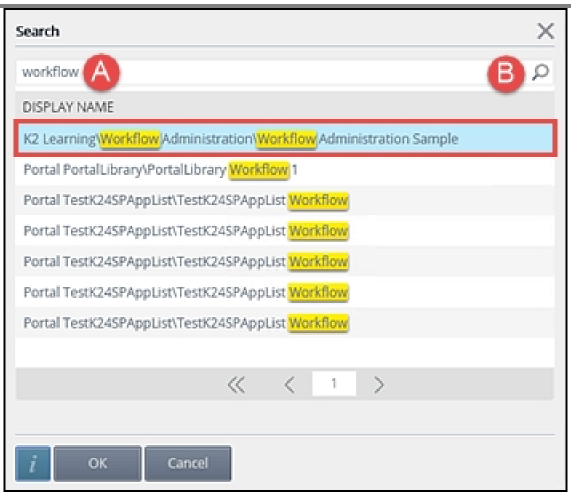
q. Switch back to **Administrator's K2 Management** site. You should still be on the Out of Office screen, but if you are not, expand the **Users** node, then click **Out of Office**. In the central pane, CHECK the box to the left of **Jonathan King** (Jonno), then click **Edit**. On the Edit Out of Office User screen, locate the **Exceptions** section (on the lower half of the screen) and click the **plus (+)** icon.



r. The Add Out of Office Exception screen is now displayed. For the **Name**, enter *Sample User Tasks Only* then click the spyglass to the right of the **Process** text box.



s. You will now search for the correct process. In the **Search** text box, enter *workflow* then click the spyglass to the right of the field. Click to highlight the **Workflow Administration Sample**, then click **OK**. (Your screen may or may not have additional processes displayed.)



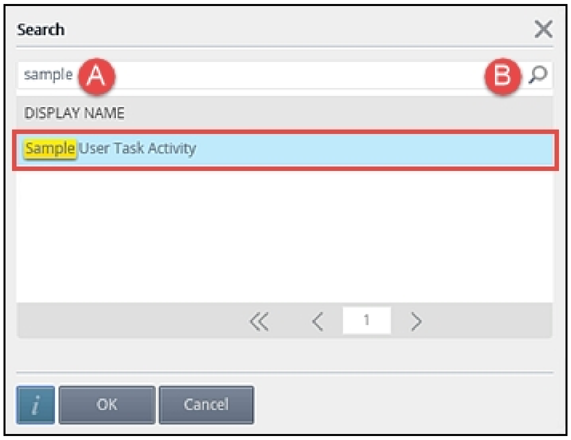
workflow

DISPLAY NAME

K2 Learning\Workflow\Administration\Workflow\Administration Sample
Portal PortalLibrary\PortalLibrary Workflow 1
Portal TestK24SPAppList\TestK24SPAppList Workflow
Portal TestK24SPAppList\TestK24SPAppList Workflow
Portal TestK24SPAppList\TestK24SPAppList Workflow
Portal TestK24SPAppList\TestK24SPAppList Workflow
Portal TestK24SPAppList\TestK24SPAppList Workflow

t. On the Add Out of Office Exception screen, click the spyglass to the right of the **Activity** text box.

u. On the **Search** screen, enter
sample
for the search criteria, then click the spyglass. Click to highlight the **Sample User Task Activity**, then click **OK**.



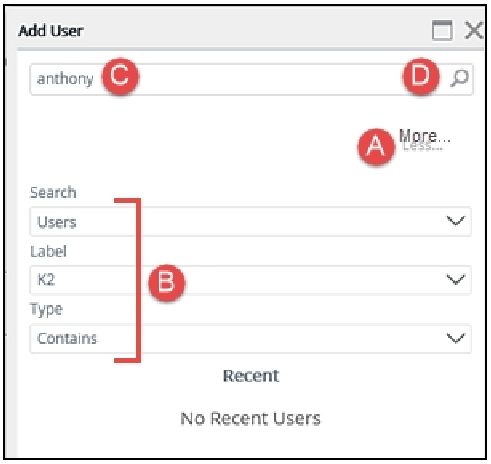
sample

DISPLAY NAME

Sample User Task Activity

v. The last setting is to assign the target user for Jonno's Out of Office, Sample User Tasks Only, exception. In the **Users** section, click the **plus (+)** icon.

w. On the Add User screen, click the '**More...**' link. For the **Search** value, select or confirm **Users**. Confirm to change the **Label** to **K2**. Confirm or change the **Type** to **Contains**. Enter
anthony
in the search text box, then click the spyglass.

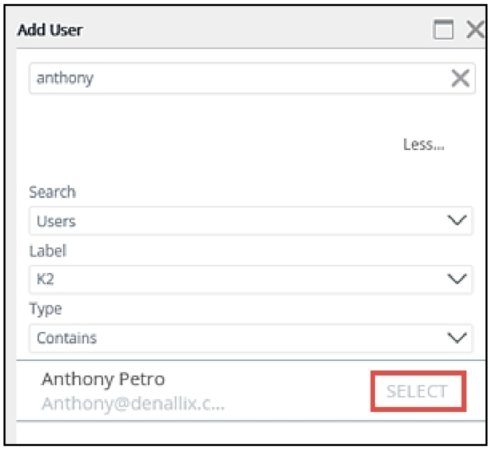


anthony

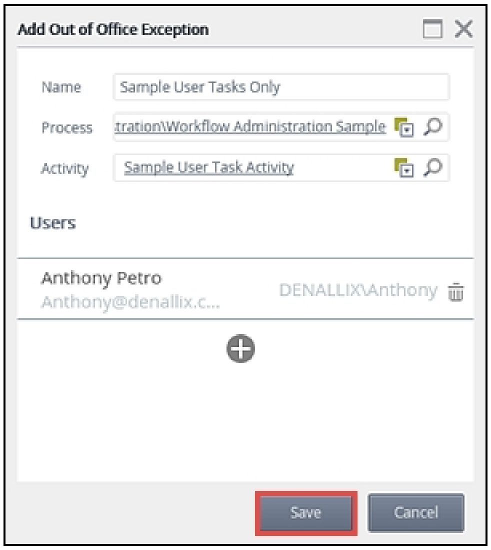
Search: Users
Label: K2
Type: Contains

No Recent Users

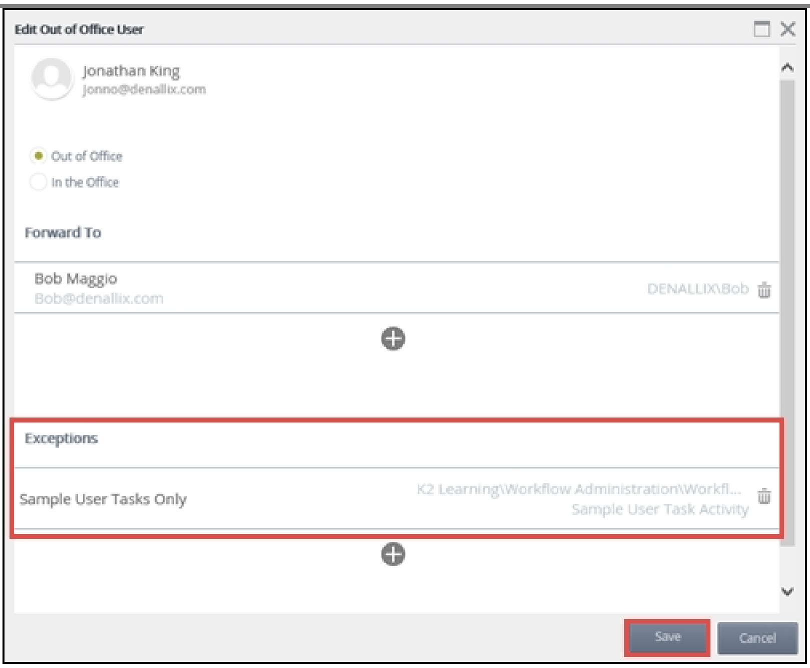
x. Anthony's account is displayed just below the search settings. Click **SELECT** to add Anthony as the target user for Jonno's Out of Office exception.



y. On the Add Out of Office Exception screen, review the Name, Process, Activity and User information, then click **Save**.



z. On the Edit Out of Office User screen, click **Save** once again to commit the exception addition.



Edit Out of Office User

Jonathan King
Jonno@denallix.com

● Out of Office
○ In the Office

Forward To

Bob Maggio
Bob@denallix.com
DENALLIX\Bob

Exceptions

Sample User Tasks Only
K2 Learning\Workflow Administration\Workfl...
Sample User Task Activity

STEP 2 REVIEW

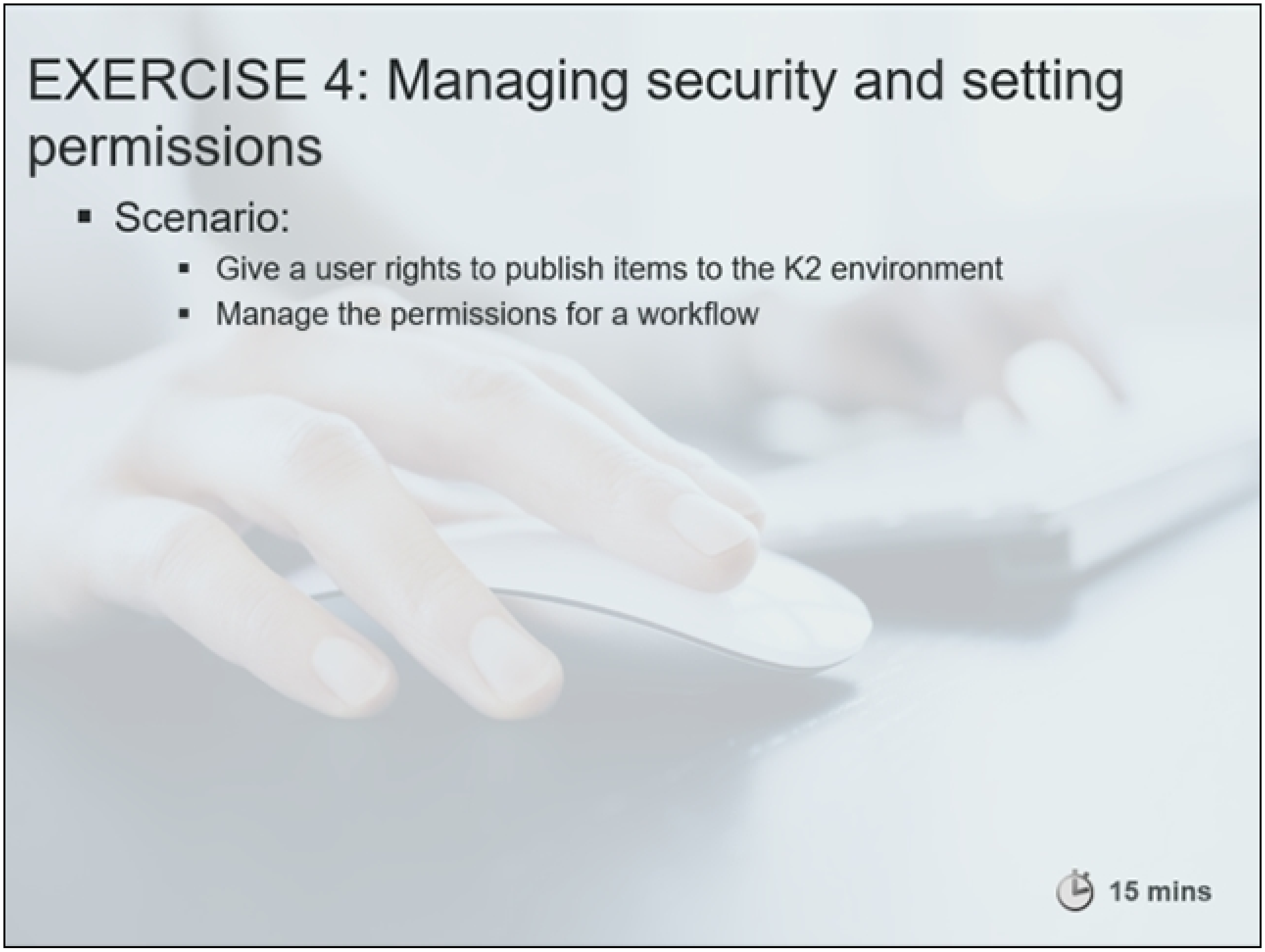
In this step, you worked with the Out of Office feature and learned how users can manage their own Out of Office settings through K2 Workspace. Configuring Out of Office essentially shares the task(s) with the target user. One tip is to simply toggle the Out of Office setting (from in-office to out-of-office), instead of removing it. This is helpful if your users need to activate/deactivate this feature often. By default, K2 shares all of the processes with the target user if Out of Office is activated. Exceptions can be added to filter which processes and activities are shared.

The K2 administrator can manage Out of Office settings for all users, while users can manage just their own settings. The K2 administrator manages Out of Office through the K2 Management site.

The Out of Office feature helps to ensure that workflows continue to move along, even if the destination user is unavailable. Another common use for this feature is to assign Out of Office sharing for executives that want another user (an assistant perhaps), to manage their workflow tasks.

In the next section, you will take a look at permissions in K2. If you are ready to continue, click the Managing Permissions link to get started. (If you are continuing on, leave the IE instances open for Anthony and Bob, as you will use them in the next section.)

EXERCISE 4: Managing security and setting permissions



EXERCISE 4: Managing security and setting permissions

- Scenario:
  - Give a user rights to publish items to the K2 environment
  - Manage the permissions for a workflow

15 mins

Video

This exercise describes how to manage some permissions and security in K2. You will learn:

- Setting K2 permissions to restrict who may do what in your K2 environment
- Giving a user publishing permissions in our environment so that they can deploy workflow definitions to the K2 environment
- Managing the permissions for a sample workflow to control who may start, administer and report on the workflow

When you are ready, continue with K2 Administration: Managing Permissions to start the exercise.

Managing Permissions

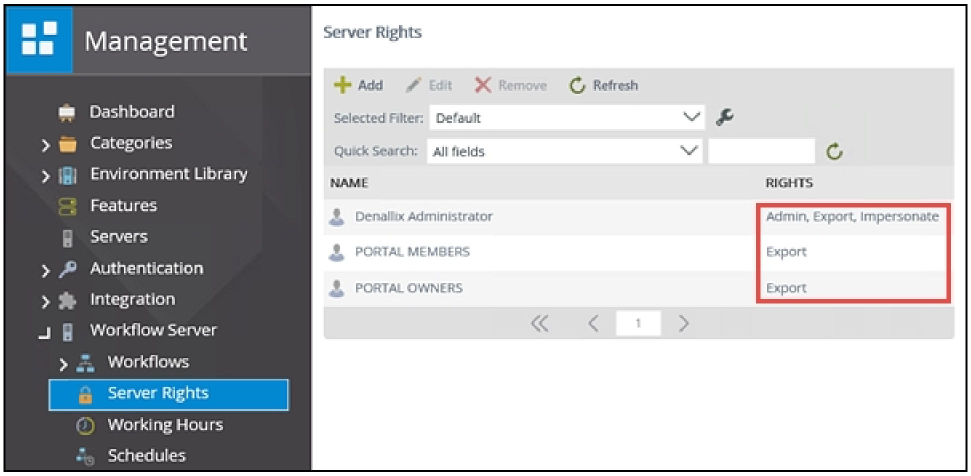
As a K2 administrator, you will manage server rights and workflow rights. In this section, you will take a look at the more common permissions you are likely to assign, and the level of security those permissions grant. Most permissions can be assigned to individual users or groups.

There are three levels of permissions that you (as the K2 administrator) will work with. The following is a high-level overview:

- **K2 Administrator**: This user has full-control over the K2 environment. This user will likely interact with other system, network and database resources as well. If this user does not have access to these other resources, they will need to work with the individuals that do. Typically, there are very few K2 administrators within the organization (one or two perhaps). The K2 administrator assigns server rights and often workflow rights to other users. K2 administrators manage the entire K2 environment.
- **Server Rights**: There are three options for Server Rights permissions.
  - Admin: Full control; this is the K2 administrator. Again, usually only one or two people are full K2 Admins.
  - Export: Users with Export rights can publish (deploy) workflows. Assign this right to users that will be building workflows and will need to deploy them, or users that need to use the K2 Package and Deployment

tool to deploy packages. Some organizations prefer to limit the number of users that actually build and deploy workflows as a quality control measure, while others grant export rights to all of their users.

- Impersonate: This is a system right that allows an account to impersonate another user after the initial connection is established. This is an advanced use case that is not covered in this tutorial.



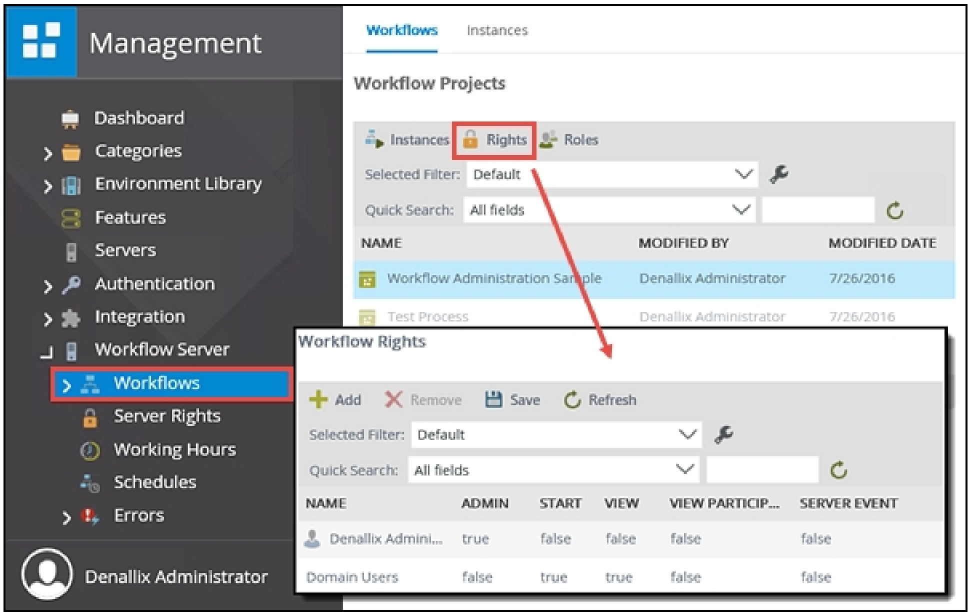
Server Rights

Selected Filter: Default

Quick Search: All fields

NAME	RIGHTS
Denallix Administrator	Admin, Export, Impersonate
PORTAL MEMBERS	Export
PORTAL OWNERS	Export

- **Workflow Rights**: Workflow Rights are permissions that are assigned to each individual process. At some point in time, you may build a workflow for a specific target group of users. To prevent other users from submitting the workflow, you can control the workflow access with workflow rights. Workflow rights consist of the following:
  - Admin: Required to view, add and edit the rights for a deployed process, as well as administer active workflow instances.
  - Start: Users with start rights can start process instances.
  - View: Users with view rights can report on all instances of a workflow.
  - View Participate: Users with view participate rights can report only on those workflow instances where they are the originator (submitted the original form), or where they actioned a user task.
  - Server Event: This is a special type of permission used for asynchronous server tasks, where an external system completes a K2 server event. This is an advanced use case that is not covered in this tutorial.



Workflow Projects

Selected Filter: Default

Quick Search: All fields

NAME	MODIFIED BY	MODIFIED DATE
Workflow Administration Sample	Denallix Administrator	7/26/2016
Test Process	Denallix Administrator	7/26/2016

Workflow Rights

Selected Filter: Default

Quick Search: All fields

NAME	ADMIN	START	VIEW	VIEW PARTICIP...	SERVER EVENT
Denallix Admini...	true	false	false	false	false
Domain Users	false	true	true	false	false

Note
This tutorial requires that you have deployed the Workflow Administration Sample as described in the Download and deploy a simple K2 package tutorial.

Note
This tutorial includes steps using the K2 Management site. You must have K2 blackpearl 4.7 or later to complete this tutorial as the steps and screen shots are specific to the K2 Management site.

Step 1: Give a user export rights so that they can deploy (publish) workflows

In this Step, you will grant Anthony (export) rights so that he can deploy (or publish) workflows.

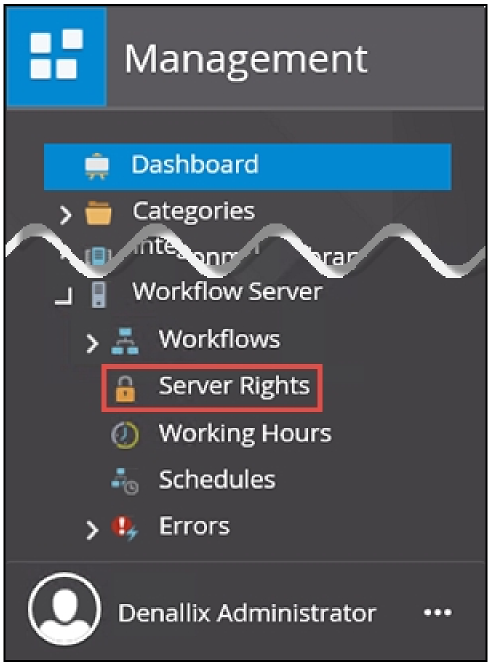
Step 1 Tasks

1. Grant **Anthony (export)** rights.

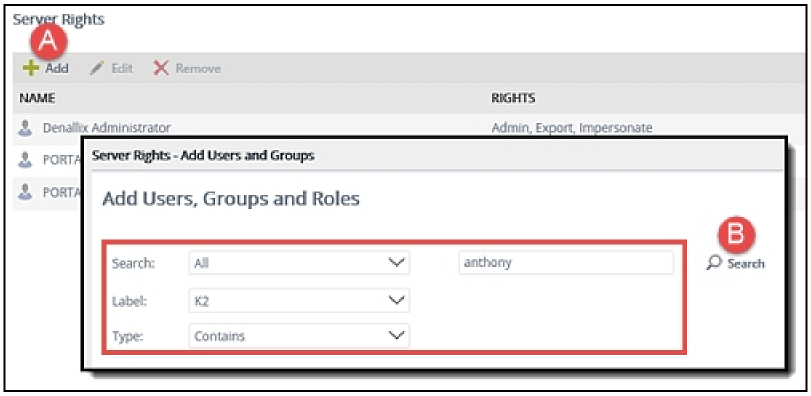
Step 1 Walkthrough

a. Launch the **K2 Management** site for Administrator (if it is not already). (**Start** > **All Programs** > **K2 blackpearl** > **K2 Management**)

b. Navigate to the **Server Rights** option found in the **Workflow Server** > **Workflows** node.



c. When the Server Rights screen opens in the central pane, click **Add**. Confirm the **Label**is set to K2, then enter
anthony
into the search text box. (Leave the other default values as is.) Click the spyglass or **Search** button.



NAME	RIGHTS
Denallix Administrator	Admin, Export, Impersonate
PORTA	
PORTA	

Server Rights - Add Users and Groups

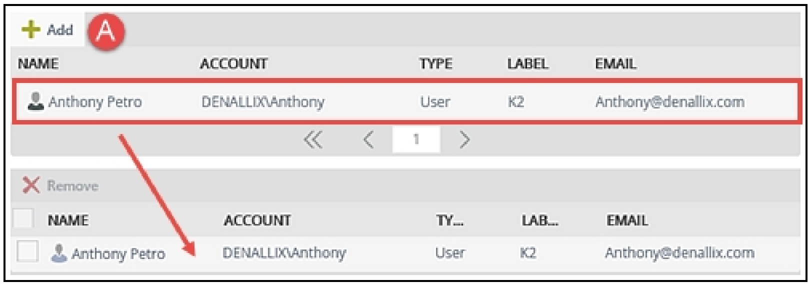
Add Users, Groups and Roles

Search: All anthony

Label: K2

Type: Contains

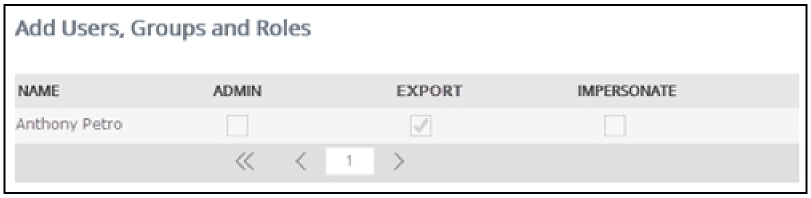
d. Anthony's full name should appear in the search results pane. Click to highlight Anthony's name, then click **Add**. Click **Next**. (While this may appear to be a bit tedious, the reason behind the multiple steps is so you can search for, then add, several users at the same time, then click Next to assign the server rights all at once.)



NAME	ACCOUNT	TYPE	LABEL	EMAIL
Anthony Petro	DENALLIX\Anthony	User	K2	Anthony@denallix.com

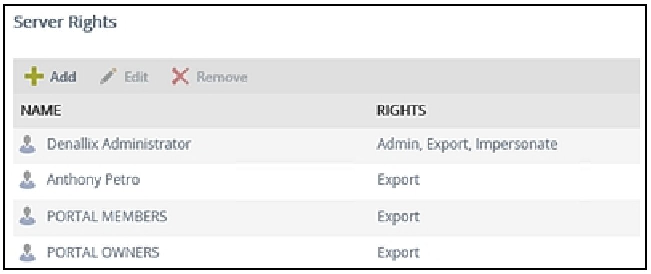
NAME	ACCOUNT	TY...	LAB...	EMAIL
Anthony Petro	DENALLIX\Anthony	User	K2	Anthony@denallix.com

e. You should now see Anthony's name listed with the Server Rights options. CHECK the box in the **Export** column, then click **Finish**.


Add Users, Groups and Roles

NAME	ADMIN	EXPORT	IMPERSONATE
Anthony Petro	□	☑	□

Anthony now has server rights to publish, or deploy, workflows.


Server Rights

NAME	RIGHTS
Denallix Administrator	Admin, Export, Impersonate
Anthony Petro	Export
PORTAL MEMBERS	Export
PORTAL OWNERS	Export

STEP 1 REVIEW

In this step, you granted Anthony (export) rights so that he can publish (deploy) workflows. In the image above, notice that all Portal Members and Owners also have export rights. Typically, you would not grant individual users rights if you have already granted all users export rights. (The assumption is that all users are included in the Portal Members group.) Some organizations may grant export rights to a handful of users as a quality control measure. Users in this scenario are most likely application designers and have had more advanced training than the typical user. Other organizations grant all users export rights and allow everyone to build and deploy workflows.

Step 2: Give one user admin rights and another start rights

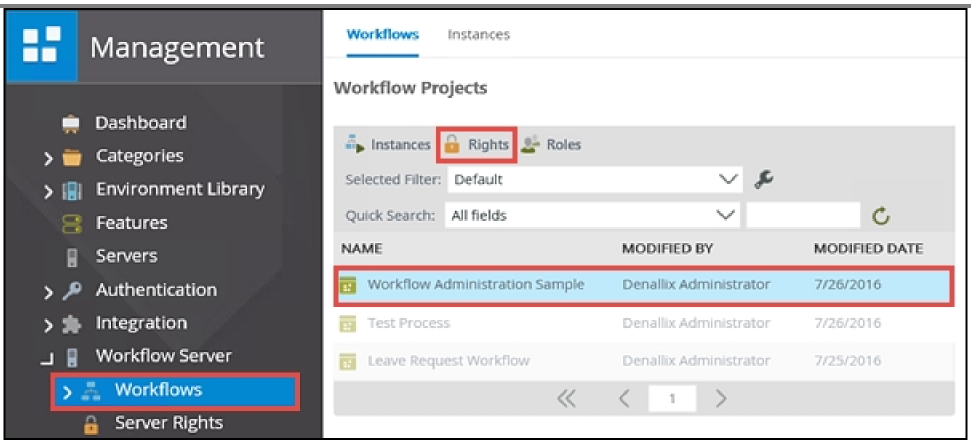
Now you are going to work with Workflow Rights, or permissions assigned to a single process. In the following steps, you will grant Bob (admin) rights to the Workflow Administration Sample process. You want Bob to be able to manage the permissions for this process. You will then grant all other users (start) rights so that everyone can start this workflow.

Step 2 Tasks

1. Grant **Bob (admin)** rights for the **Workflow Administration Sample** process.
2. Grant **Domain Users (start)** rights on the same process.

Step 2 Walkthrough

a. Still in the K2 Management site, expand the **Workflow Server** node (if it isn't already), then click **Workflows**. A list of workflow projects are displayed in the central pane. Click to highlight the **Workflow Administration Sample** process, then click the **Rights** button located in the navigation pane.



Workflow Projects

Selected Filter: Default

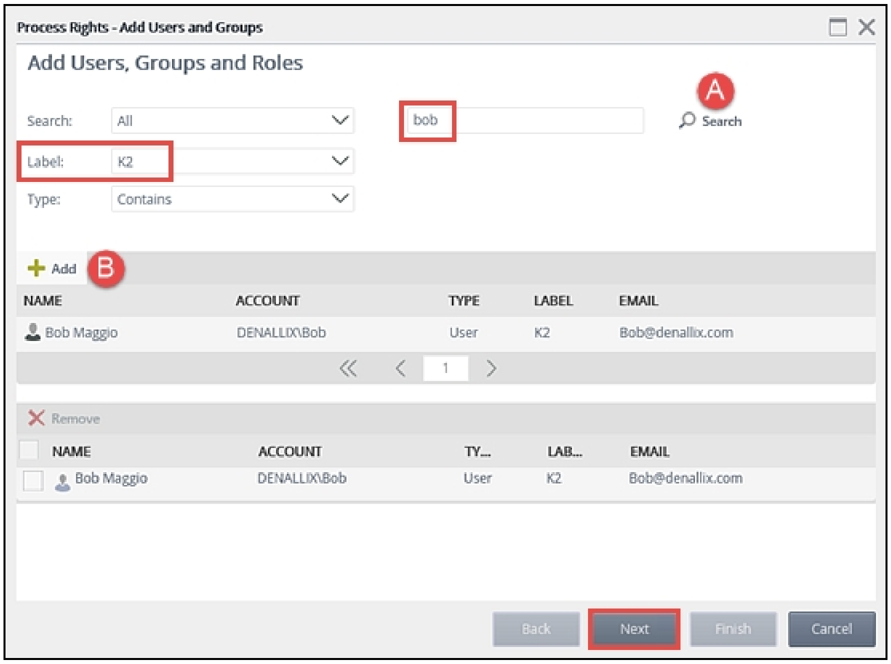
Quick Search: All fields

NAME	MODIFIED BY	MODIFIED DATE
Workflow Administration Sample	Denallix Administrator	7/26/2016
Test Process	Denallix Administrator	7/26/2016
Leave Request Workflow	Denallix Administrator	7/25/2016

At this point, you should see that Administrator is the only user with workflow rights. (If on a K2-provided VM.) Now you are going to add Bob and give him admin rights. This will allow Bob to grant permissions to other users for this single workflow process as he needs to.

b. In the **Workflow Rights** screen, click **Add**.

c. Confirm the **Label** is set to **K2**, then **search** for and **Add Bob**. Click **Next**.



Process Rights - Add Users and Groups

Add Users, Groups and Roles

Search: All

bob

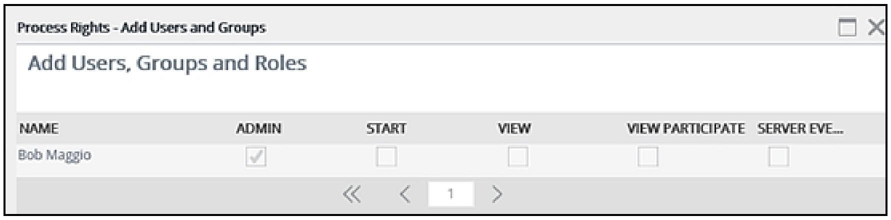
Label: K2

Type: Contains

NAME	ACCOUNT	TYPE	LABEL	EMAIL
Bob Maggio	DENALLIX\Bob	User	K2	Bob@denallix.com

	NAME	ACCOUNT	TY...	LAB...	EMAIL
□	Bob Maggio	DENALLIX\Bob	User	K2	Bob@denallix.com

d. You should now see a screen displaying Bob's name and the Workflow Rights options. CHECK the box for **Admin**, then click **Finish**.

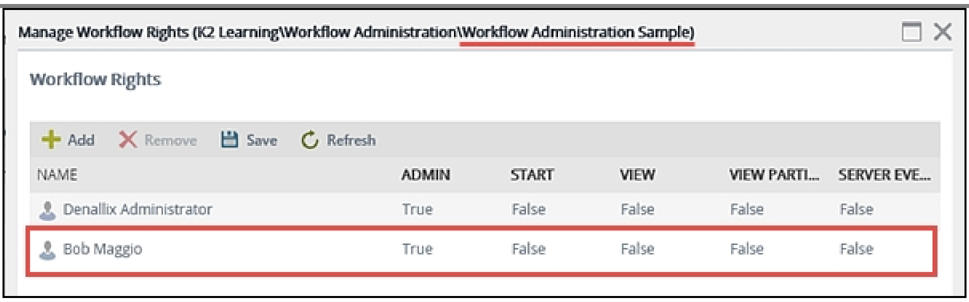


Process Rights - Add Users and Groups

Add Users, Groups and Roles

NAME	ADMIN	START	VIEW	VIEW PARTICIPATE	SERVER EVE...
Bob Maggio	☑	□	□	□	□

On the Workflow Rights screen, you should see that Bob has been added with admin rights to the Workflow Administration Sample process. Bob is now able to grant permissions to other users for this process.



Manage Workflow Rights (K2 Learning\Workflow Administration\Workflow Administration Sample)

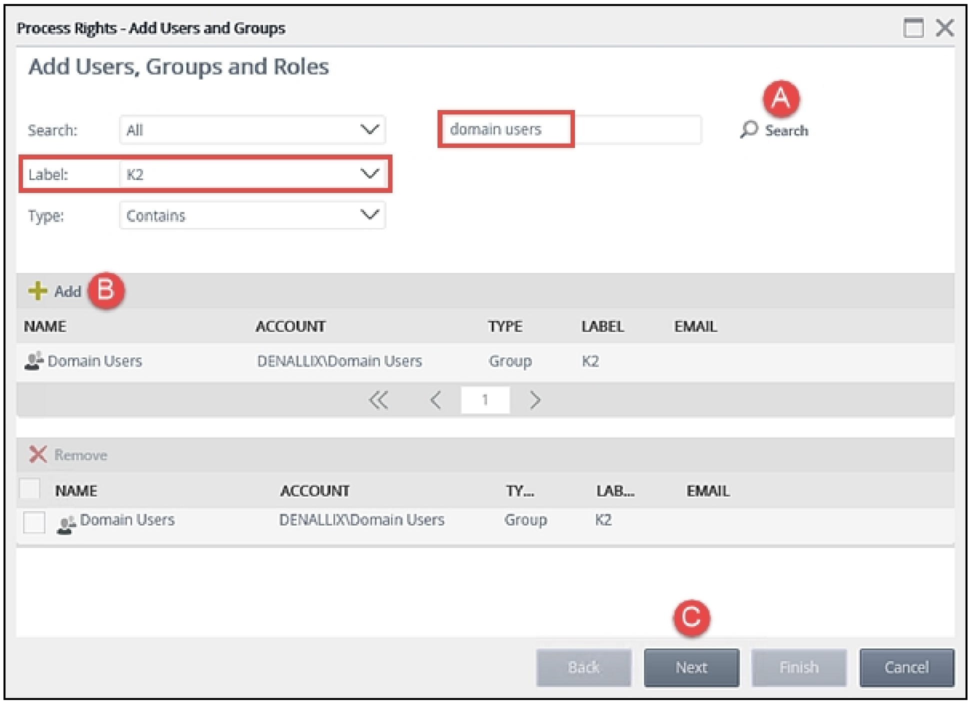
Workflow Rights

NAME	ADMIN	START	VIEW	VIEW PARTI...	SERVER EVE...
Denallix Administrator	True	False	False	False	False
Bob Maggio	True	False	False	False	False

Now you will grant start rights to all users. This is allow anyone to submit the Workflow Administration Sample process.

e. Click **Add** once again.

f. On the Add User, Groups and Roles screen, confirm the **Label** is set to **K2**, then search for *domain users*
then click **Add** when you see Domain Users in the results pane. Click **Next**.



Process Rights - Add Users and Groups

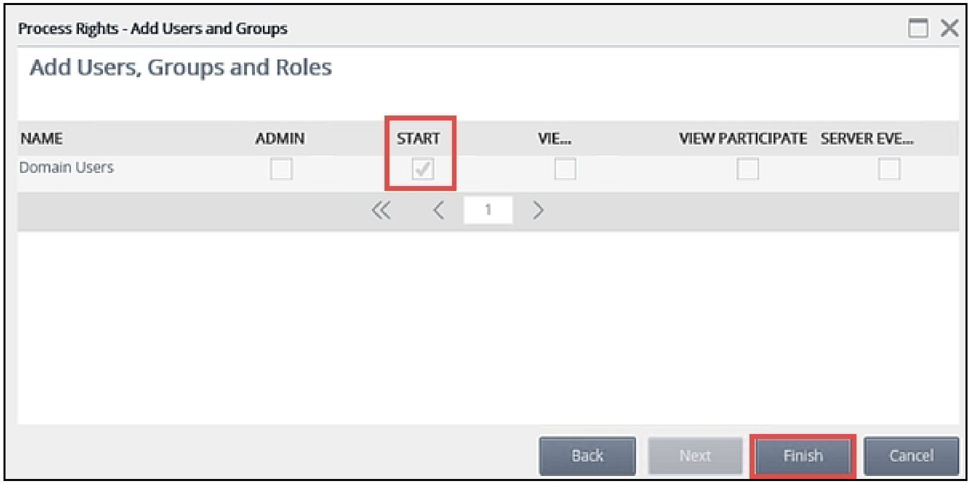
Add Users, Groups and Roles

Search: All
domain users
Label: K2
Type: Contains

NAME	ACCOUNT	TYPE	LABEL	EMAIL
Domain Users	DENALLIX\Domain Users	Group	K2	

	NAME	ACCOUNT	TY...	LAB...	EMAIL
□	Domain Users	DENALLIX\Domain Users	Group	K2	

g. CHECK the box in the **Start** column, then click **Finish**.

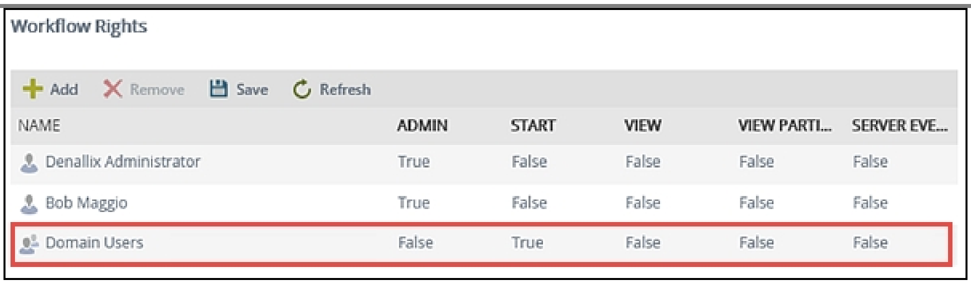


Process Rights - Add Users and Groups

Add Users, Groups and Roles

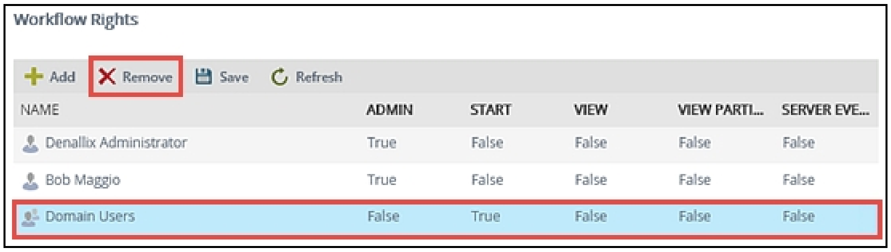
NAME	ADMIN	START	VIE...	VIEW PARTICIPATE	SERVER EVE...
Domain Users	□	☑	□	□	□

All domain users now have the necessary permissions to start the Workflow Administration Sample process.



NAME	ADMIN	START	VIEW	VIEW PARTI...	SERVER EVE...
Denallix Administrator	True	False	False	False	False
Bob Maggio	True	False	False	False	False
Domain Users	False	True	False	False	False

Workflow rights can be assigned to both individual users and groups of users. To delete the workflow rights for a user or group, click to highlight the name, then click Remove. Close the Workflow Rights screen.



NAME	ADMIN	START	VIEW	VIEW PARTI...	SERVER EVE...
Denallix Administrator	True	False	False	False	False
Bob Maggio	True	False	False	False	False
Domain Users	False	True	False	False	False

STEP 2 REVIEW

In this final step, you granted permissions, or workflow rights, to the Workflow Administration Sample process. With admin level rights, Bob can now assign permissions for this process as he needs to. All domain users now have the ability to start this workflow. In this step, you observed how you can assign rights to both individuals and groups. To delete the rights for a user or group, simply highlight the name, then click Remove.

You have now completed the Administering K2 tutorial. If you would like to review the topics covered, or work through some challenge exercises, continue on to the Summary and Challenge Steps.

Administering K2 - Summary and Challenge Steps

The K2 Management site contains tools for administering the K2 environment, server permissions, Out of Office settings, errors and process management, to name a few. This site will typically be accessed by users that are K2 administrators. The K2 Workspace is designed to give each user a platform for managing workflows tasks assigned to them, starting new process instances, configuring their Out of Office settings and viewing their workflow reports.

K2 Management Site

The K2 Management site is the primary location for administering deployed processes and managing environmental settings. The following is a summary of the *Administering K2* tutorial:

- **Package and Deployment**: In this section, you downloaded and deployed the Workflow Administration Sample process using the K2 Package and Deployment tool. The K2 Package and Deployment tool is primarily used to package applications from one environment and deploy them to another environment. For example, you might build an application on a development server, then package and deploy it to a production server. You are able to deploy forms, views, SmartObjects, workflows and other related K2 artifacts using this tool.
- **Managing Processes and Process Instances**: In this section, you started a number process instances directly from the K2 Management site. This allows you to test and troubleshoot workflows without having to open and submit forms. You also started, stopped and deleted instances. Stopping an instance pauses, or suspends it, until it is restarted. Deleting an instance will completely remove the process instance from K2. In this section, you also used the Go To Activity option to send a workflow to a different step. This is quite useful for sending workflows *back* in time. For example, if a user needs adjust their form entry, you can send the workflow back to the step they were working on. K2 will have saved the form content and the user can simply open the form and resume.
- **Managing Task Lists**: A Task List is simply a list of tasks assigned to a user. The task itself is referred to as a task list item. You can redirect task list items from one user to another. A common scenario is the destination user assigned a task is not available and the form originator would like to redirect the task to another user so that the workflow can keep moving along. With the Out of Office feature, both K2 administrators and users

themselves can set up and manage alternative users to share their task list items with. Out of Office does not remove the task from the original destination user, instead it makes a copy for the alternative user. Aside from users being literally out of the office, another common use for this feature is for executives to forward their task list items to an assistant, again to keep the workflow process moving.

- **Managing Permissions**: Typically speaking, the K2 administrator will manage server rights and workflow rights. The most common permission setting in Server Rights is the export option, which allows users to publish (or deploy) workflows. Some organizations limit the number of users that can deploy workflows as a quality control measure. Other organizations allow all of their users to build and deploy workflows as they need to. You learned that you must grant users at least start rights for them to be able to start a workflow. View rights are also a common option and allow the user to run reports against the workflow.

If you would like to further test the knowledge you gained while working through this tutorial, continue on to the following challenge exercises.

Note
The challenge exercises require that you have deployed the Workflow Administration Sample as described in the Download and deploy a simple K2 package tutorial and started instances of this workflow as described in the Managing Processes and Process Instances tutorial.

Challenge 1: Filter task list items

Challenge 1 Task

1. Filter the task list items so that you are viewing only **Jonno's** tasks.
2. Filter the task list items so that you are viewing only the '**Normal Execution**' process instances.
3. Do not filter the task list items and instead, view all task list items.

Challenge 1 Solution

Filter the task list items so that you are viewing only Jonno's tasks.

a. In the K2 Management site, click the **Task List** option found under the **Users** node.
b. In the central pane, locate the **Destination** filter and enter *jonno*. Click the spyglass or **Search** button located near the middle of the screen on the right-hand side. There should be one task displayed in the results pane (TaskRedirect).

Filter the task list items so that you are viewing tasks that contain 'Normal Execution' in the process instance name.

a. Highlight and delete the entry you just made in the destination filter. In the **Folio** filter, enter *normal*, then click the **Search** button. There should be two tasks displayed in the results pane (Normal Execution 1 and Normal Execution 2)

Do not filter the task list items and instead, view all of the task list items.

a. Delete the entry you just made in the Folio filter. Click the **Search** button without entering any filters. There should be at least four tasks displayed in the results pane. (TaskRedirect, StartStop, Normal Execution1 and Normal Execution2)

Challenge 2: Remove permissions for domain users and add start rights for the Finance group

Challenge 2 Task

1. **Remove** the permissions for the **Domain Users** in the Workflow Administration Sample process. Add **start** permissions for the **Finance** group.

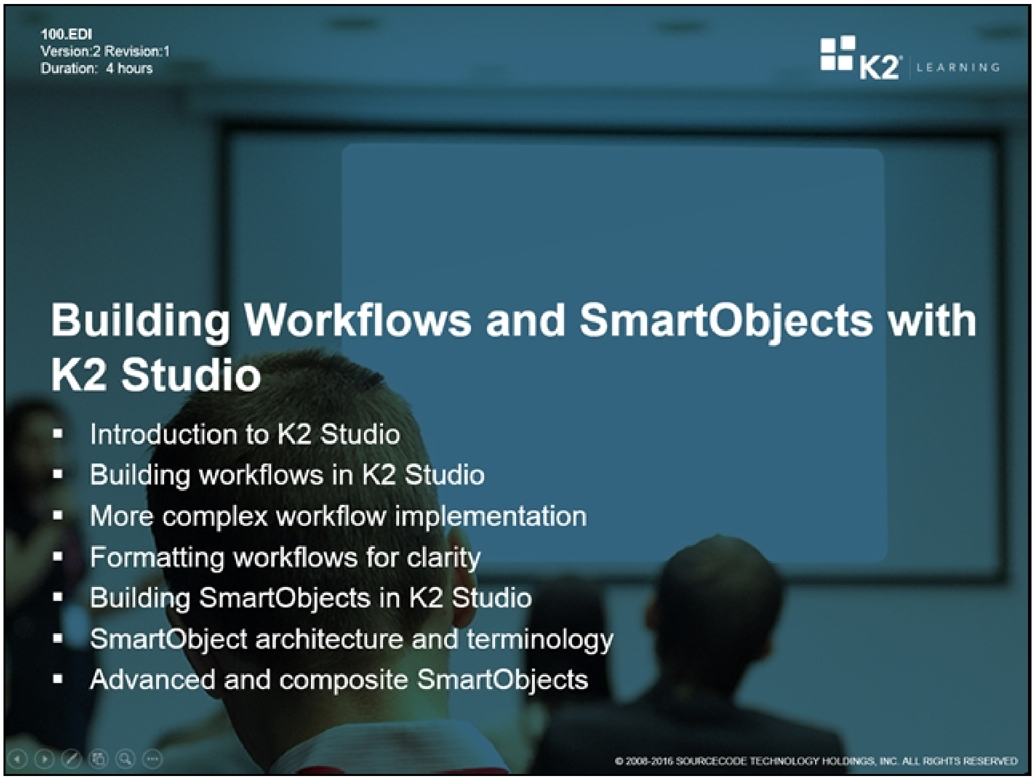
Challenge 2 Solution

a. Navigate to the **Workflow Projects** screen. (Workflow Server > Workflows) Click to highlight the **Workflow Administration Sample** process, then click the **Rights** button.
b. Click to highlight the **Domain Users** group, then click **Remove**.

c. Click **Add**. Confirm the **Label** is set to **K2**, then change the **Search** scope from 'All' to **Groups**. Search for *finance*.

d. Assign the **Finance** group **start** rights.

100.EDI: Building Workflows and SmartObjects with K2 Studio



100.EDI
Version:2 Revision:1
Duration: 4 hours

K2 LEARNING

Building Workflows and SmartObjects with K2 Studio

- Introduction to K2 Studio
- Building workflows in K2 Studio
- More complex workflow implementation
- Formatting workflows for clarity
- Building SmartObjects in K2 Studio
- SmartObject architecture and terminology
- Advanced and composite SmartObjects

▶ Video

The *100.EDI: Building Workflows and SmartObjects with K2 Studio* training module is focused on using K2 Studio to build more advanced workflows and more advanced SmartObjects. This module assumes that you have some basic knowledge of K2, such as having completed the *100ABZ: Introduction to K2* and *100BHX: Introduction to K2 Applications with K2 Designer* learning modules.

This module covers the following concepts:

- Working with K2 Studio
- Building basic workflows in K2 Studio
- How to implement advanced workflow concepts like conditional outcomes, parallel paths, loops and escalations
- A deeper look at SmartObject architecture and authentication
- Building SmartObjects that integrate with other systems and building composite SmartObjects that combine data from different systems
- Some recommended design patterns and practices when building K2 applications

Note
Many of the design concepts and user interfaces described in this learning module apply equally to K2 for Visual Studio. Some of the advanced workflow and SmartObjects concepts also apply to the browser-based K2 Designer tool.

Part 1: Workflows in K2 Studio

PART 1

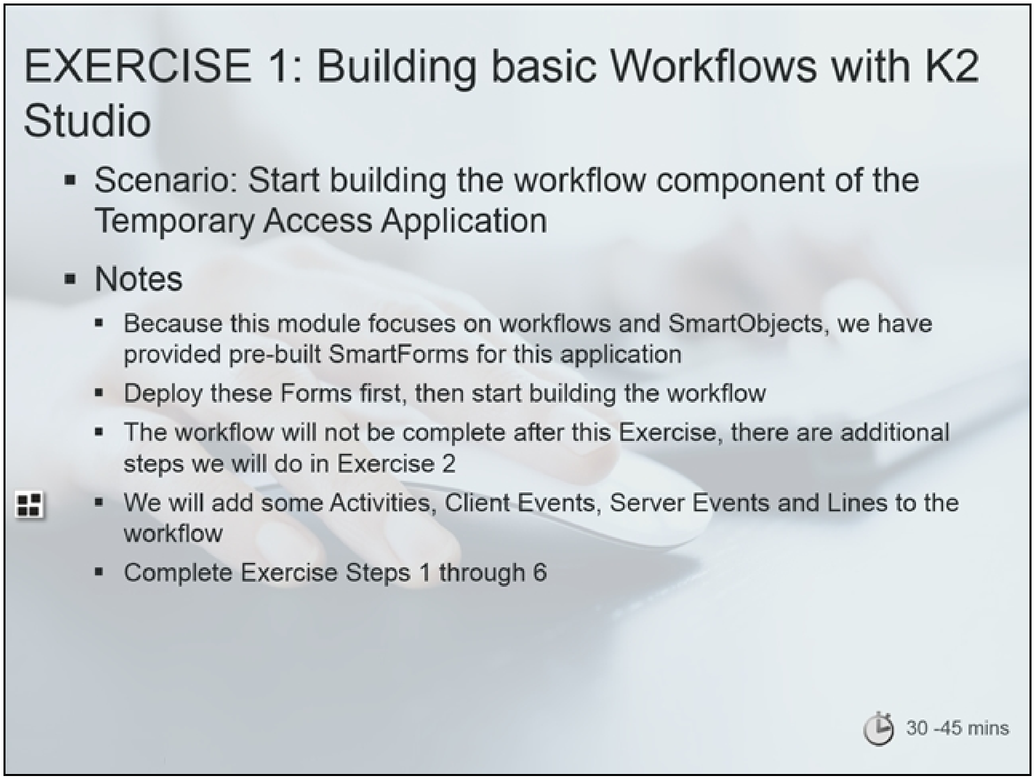
Workflows in K2 Studio

Video

In Part 1 we will look at building workflows with K2 Studio, and also at some more advanced things you can do with K2 workflows such as loops, escalations, parallel paths, conditional paths and more. There are two exercises in this module, the first exercise will look at basic workflow components in K2 Studio and the second will look at more advanced workflow concepts.

Note
The workflow we will build in Part 1 will be used by the SmartObject we will create in Part 2.

EXERCISE 1: Building basic Workflows with K2 Studio



EXERCISE 1: Building basic Workflows with K2 Studio

- Scenario: Start building the workflow component of the Temporary Access Application
- Notes
  - Because this module focuses on workflows and SmartObjects, we have provided pre-built SmartForms for this application
  - Deploy these Forms first, then start building the workflow
  - The workflow will not be complete after this Exercise, there are additional steps we will do in Exercise 2
  - We will add some Activities, Client Events, Server Events and Lines to the workflow
  - Complete Exercise Steps 1 through 6

30 -45 mins

Video

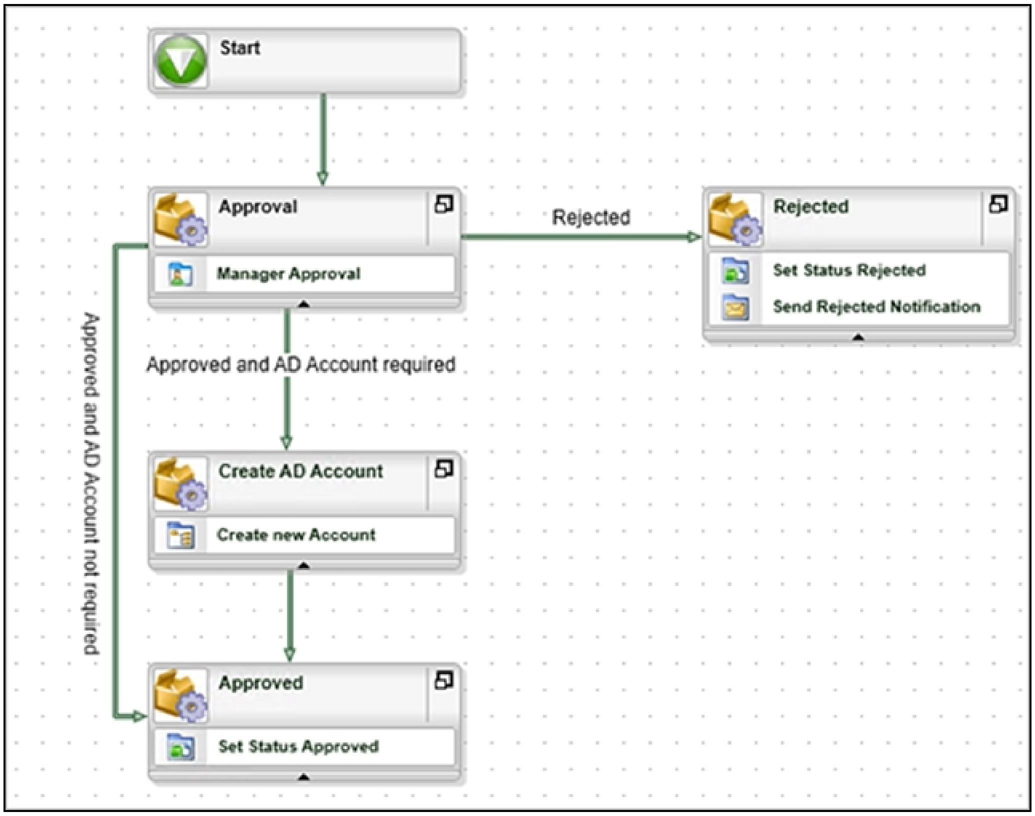
In this exercise, we will start building the workflow component of the Temporary Access Application. You will learn how to create a project in K2 Studio, add a workflow to the project and then learn how to work with the K2 Studio design canvas to add and configure Activities, Events (Server and Client Events) and Lines.

Because this module focuses on workflows and SmartObjects, we have provided pre-built SmartForms and two SmartObjects for this application. You will deploy these items first, before building the workflow.

Note
At the end of this exercise, the workflow will NOT be complete yet. You have to complete the exercise in the next part as well before you can deploy the workflow.

The screenshot below shows what the workflow will look like at the end of this exercise.

The workflow as it will look at the end of Exercise 1



When you are ready to begin, start by reviewing the design of the Temporary Access Application, and then you can move on to the exercise Temporary Access Application: Part 1 to start building the application.

Temporary Access Application (K2 Studio)

This tutorial will teach you how to build workflows and SmartObjects with K2 Studio. The sample scenario for this tutorial is the Temporary Access Application, where employees can request temporary access to the premises and systems of an organization. Think of contractors or consultants that come on-site and need some kind of system access or security access for the duration of their visit. It is a type of on-boarding and off-boarding application, where systems and humans work together to make sure that resources are in place before someone starts, and that the necessary steps are performed once those users have left to revoke access. The primary driver for this application is security, so that any access to the organization is approved and so that it is easy for anyone in the organization to see who is allowed to be on the premises. As part of this application, automatic actions are applied by the workflow to terminate access at specific dates, if no user overrides have occurred.

Note
Users familiar with K2 Workspace will notice that, from K2 4.7 onward, many of the workspace features have been moved to the new **K2 Management** site. The K2 Management site contains the tools commonly used to administer the K2 environment, plus several new additions. K2 administrators (full control) will use the K2 Management site to manage task lists (worklists), server and process rights, out-of-office settings, errors, etc. New additions to the K2 Management site include the ability to register new service types and service instances, generate and execute SmartObjects and view categories (both K2 Designer and system).
This tutorial can be completed in any environment that has K2 blackpearl 4.7 (or later) or K2 Appit 1.5 (or later) installed. The screen shots and users in the tutorial reflect the standard K2-provided virtual environment, commonly used for K2 blackpearl training events. You can, however, complete this tutorial in any other environment as long as the necessary K2 components are installed and operational, though your screens and users may be different. You must also have the necessary rights and permissions to create the K2 artifacts in the target environment. Be aware that it may be difficult to follow the steps in this tutorial if you are working in a K2 blackpearl version prior to 4.7.

Some tutorials require that you have internet access to be able to connect to an external SQL database used for retrieving data. Those tutorials will contain additional information on internet connection requirements.

Since this tutorial focuses on workflows and SmartObjects built with K2 Studio, you will be provided with a starter project that contains the basic SmartObjects and SmartForms used for the application.

Tutorial Overview

The sections in this tutorial are designed to help you become familiar with K2 Studio and how to build workflows and SmartObjects with K2 Studio.

In Part 1, you will build the workflow component of the Temporary Access Application. You will be using K2 Studio to build the workflow, and you will learn about some more advanced workflow concepts like conditional lines, evaluating user data and business data in a line, escalations, loop-back lines and more. You will also become familiar with the K2 Studio workflow design environment and how to format workflows for clarity.

In Part 4, you will create a composite SmartObject in K2 Studio that will combine data from the workflow reporting service with data from the K2 SmartBox service. In effect, you are going to combine workflow reporting data with business data to build a report that is relevant to your users, so that users can easily see which temporary access authorizations are in effect and what the workflow status is.

Before building your application, it is important to understand what you want to achieve by designing the necessary pieces of the application. Let's start with the workflow component.

Workflow Design

The diagram below shows the conceptual flow of the Temporary Access Application workflow. You've laid out your flowchart in a swim-lane format, where you have separated the user tasks and system tasks into separate lanes. (The user tasks have been highlighted in a light blue color.)

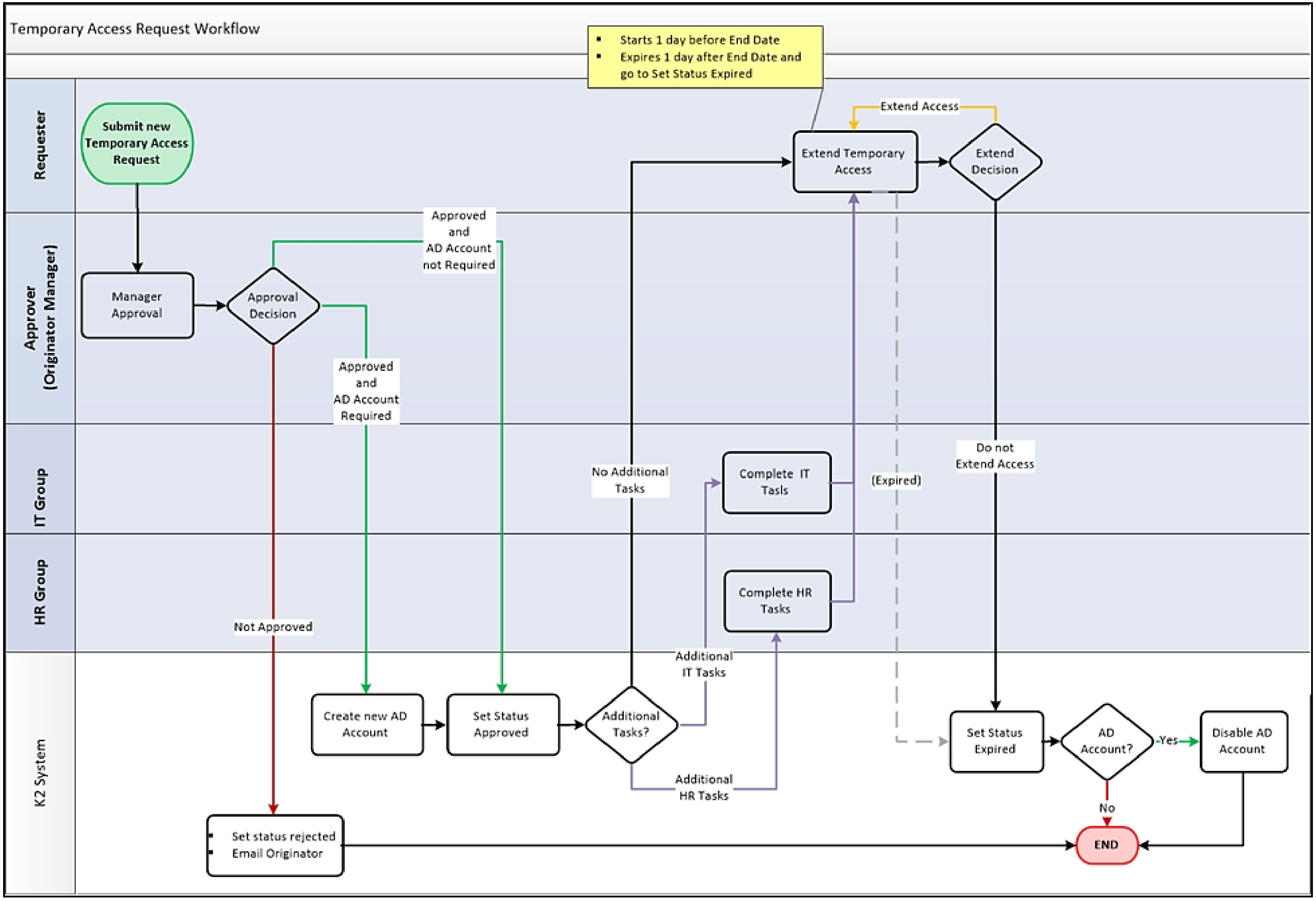
In your workflow, a user submits a new temporary access request which is routed to their manager for approval. The access request includes information like the temporary user's name and company, the start date and end date of the request, whether an Active Directory (AD) account should be created for the user and whether there are additional IT and HR tasks for the request. If the request is rejected, the system will set the access request status to rejected and email the originator. If the request is approved, the system will determine whether a new AD user account was requested for the temporary user.

If an account was requested, the system will automatically create a new AD account for the temporary user. Once this is complete (or if no account was requested), the workflow will mark the status of the request as approved. Then, K2 will determine whether there were additional IT tasks and/or HR tasks and route the workflow accordingly. Note that if both IT and HR tasks were selected, the workflow will split into a parallel execution mode, since there is no reason for IT to wait on HR or vice-versa. Once the required tasks are complete (or if there were no additional HR or IT tasks), the workflow will wait until the day before the temporary access end date. Then, the requester must decide whether or not to extend the access for the temporary user. If access is extended, the workflow will loop back to the extend access step and wait until the day before the new end date.

If the originator selects the not to extend the access, the system will set the access request status to expired. If an AD account was created for the temporary user, the account will be automatically disabled.

As a security measure, you will also configure the workflow to expire the extended access step one day after the end date, in case the originator does not complete their task. In this case, the workflow will automatically set the access request status to expired and if necessary, disable the AD user account for the temporary user.

The Temporary Access workflow conceptual design



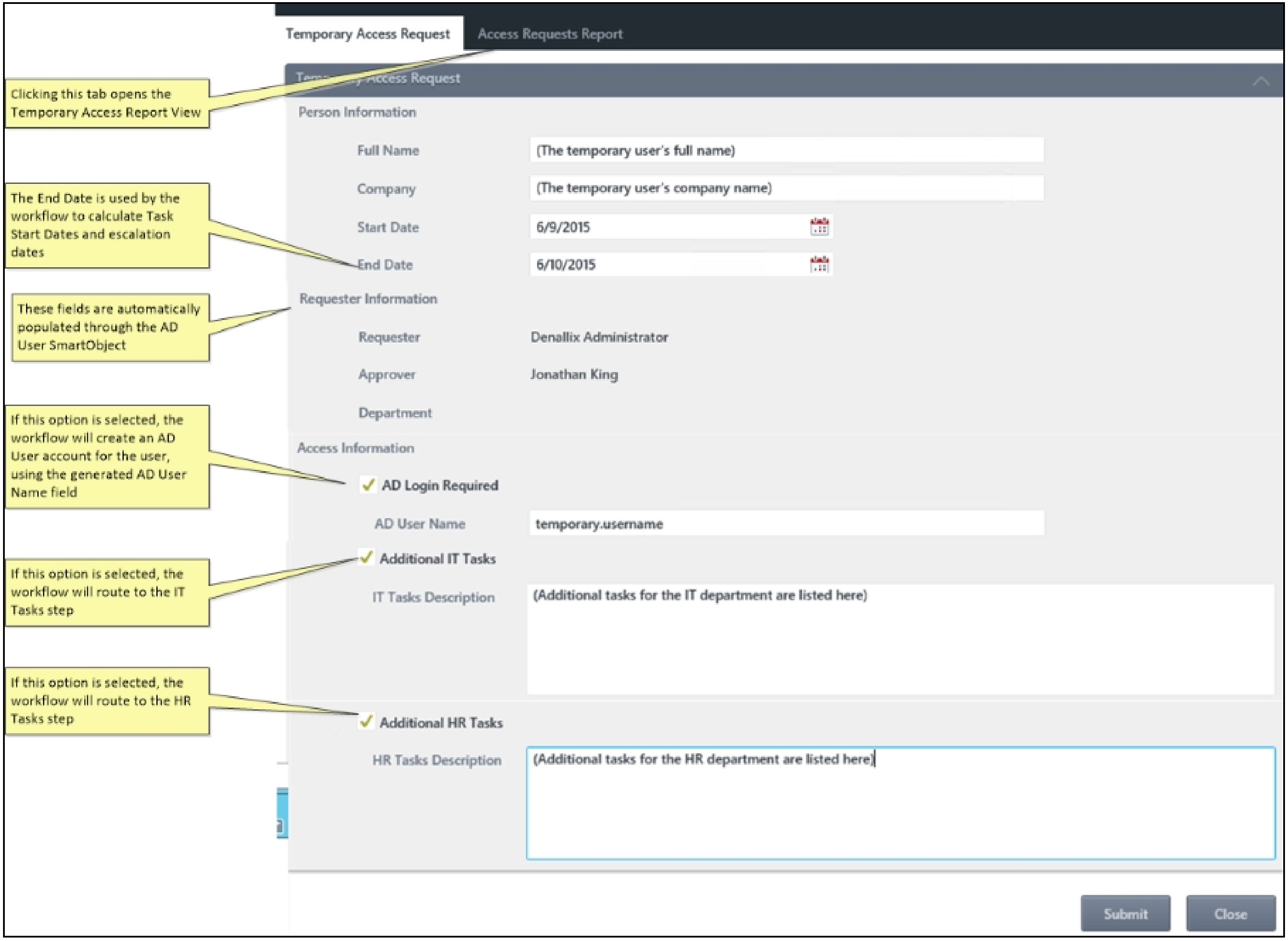
From a learning perspective, this workflow contains a number of interesting paths and escalations, including conditional lines, evaluation of user decisions along with business data, activities that loop back on themselves, parallel execution, escalations and pausing the start of certain activities until a particular date.

Now that you understand the basic flow of the workflow, let's look at the forms used in the application.

Forms Design

This application will consist of a single K2 SmartForms form that is used to capture and display the temporary access request information. Behind the scenes, this form has some logic that will check the current workflow step and selectively disable and hide certain sections of the form that are not applicable for the current user. Since this tutorial focuses mainly on workflows and SmartObjects, do not be too concerned with the behavior and rules behind this form, just understand the data that is being captured and displayed on the form, and that the workflow will behave differently based on the data captured on the form.

The Temporary Access request form



Clicking this tab opens the Temporary Access Report View

The End Date is used by the workflow to calculate Task Start Dates and escalation dates

These fields are automatically populated through the AD User SmartObject

If this option is selected, the workflow will create an AD User account for the user, using the generated AD User Name field

If this option is selected, the workflow will route to the IT Tasks step

If this option is selected, the workflow will route to the HR Tasks step

Person Information

Full Name	(The temporary user's full name)
Company	(The temporary user's company name)
Start Date	6/9/2015
End Date	6/10/2015

Requester Information

Requester	Denallix Administrator
Approver	Jonathan King
Department	

Access Information

✓ AD Login Required

AD User Name	temporary.username

✓ Additional IT Tasks

IT Tasks Description	(Additional tasks for the IT department are listed here)

✓ Additional HR Tasks

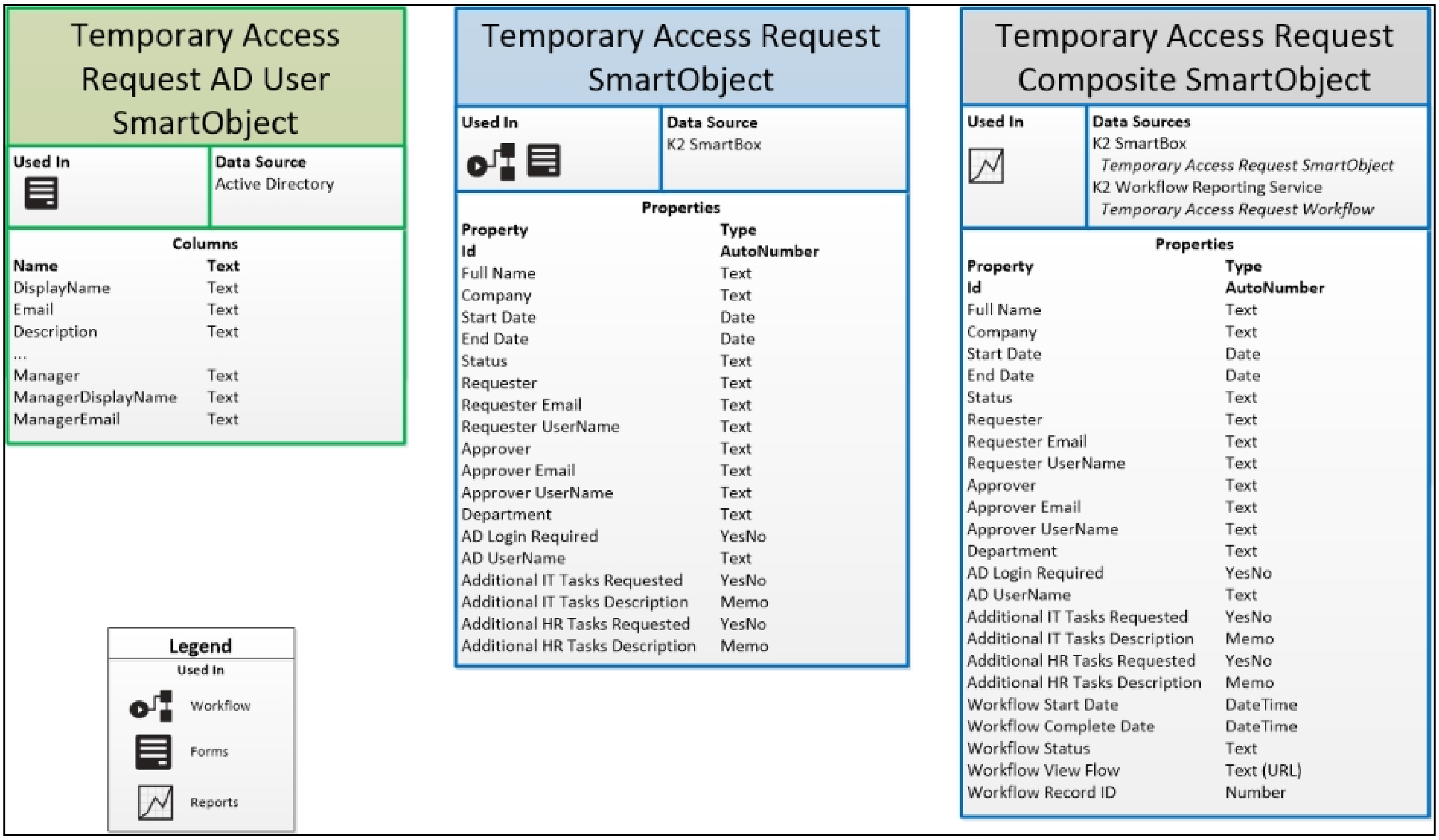
HR Tasks Description	(Additional tasks for the HR department are listed here)

Data Design

This application requires three SmartObjects for its data components. The *Temporary Access Request AD User SmartObject* is used to automatically populate fields on the form with user information, as well as to save the requester and approver details into the Temporary Access Request SmartObject.

The *Temporary Access Request SmartObject* is the primary SmartObject for this application and stores all of the data captured for a temporary access request. This data is displayed on the forms and is used by the workflow.

Finally, the *Temporary Access Request Composite SmartObject* will be used to build a report of request data along with workflow reporting data - see the Reports section for more information on this SmartObject.



Temporary Access Request AD User SmartObject

Used In	Data Source
	Active Directory

Columns

Name	Text
DisplayName	Text
Email	Text
Description	Text
...	
Manager	Text
ManagerDisplayName	Text
ManagerEmail	Text

Temporary Access Request SmartObject

Used In	Data Source
	K2 SmartBox

Properties

Property	Type
Id	AutoNumber
Full Name	Text
Company	Text
Start Date	Date
End Date	Date
Status	Text
Requester	Text
Requester Email	Text
Requester UserName	Text
Approver	Text
Approver Email	Text
Approver UserName	Text
Department	Text
AD Login Required	YesNo
AD UserName	Text
Additional IT Tasks Requested	YesNo
Additional IT Tasks Description	Memo
Additional HR Tasks Requested	YesNo
Additional HR Tasks Description	Memo

Temporary Access Request Composite SmartObject

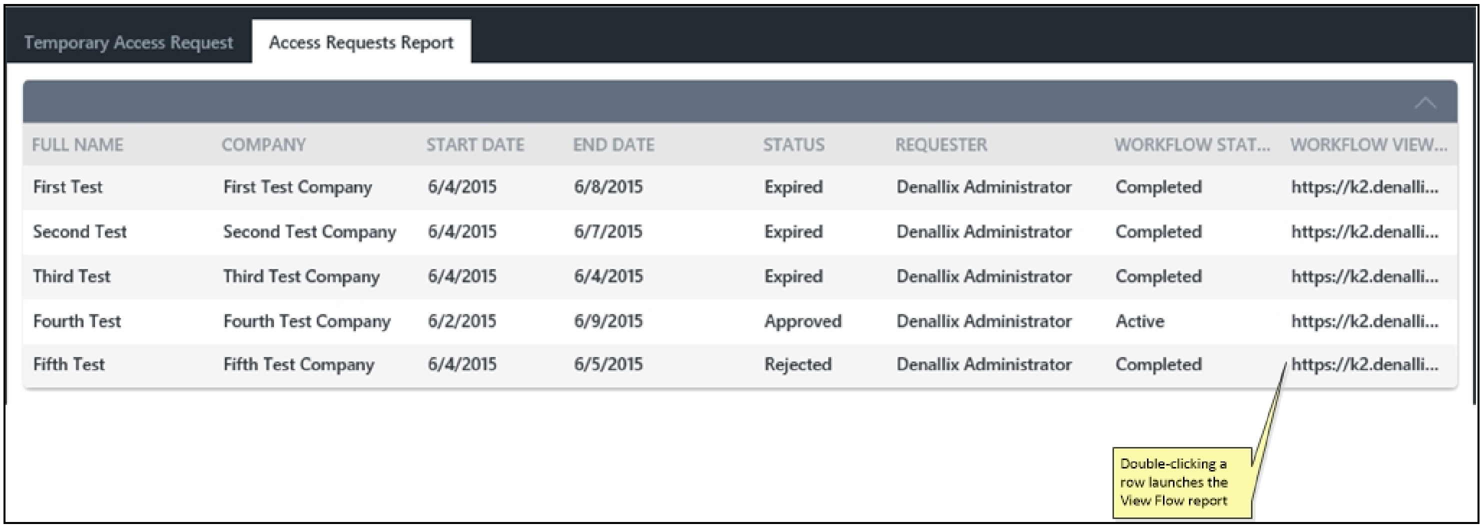
Used In	Data Sources
	K2 SmartBox *Temporary Access Request SmartObject* K2 Workflow Reporting Service *Temporary Access Request Workflow*

Properties

Property	Type
Id	AutoNumber
Full Name	Text
Company	Text
Start Date	Date
End Date	Date
Status	Text
Requester	Text
Requester Email	Text
Requester UserName	Text
Approver	Text
Approver Email	Text
Approver UserName	Text
Department	Text
AD Login Required	YesNo
AD UserName	Text
Additional IT Tasks Requested	YesNo
Additional IT Tasks Description	Memo
Additional HR Tasks Requested	YesNo
Additional HR Tasks Description	Memo
Workflow Start Date	DateTime
Workflow Complete Date	DateTime
Workflow Status	Text
Workflow View Flow	Text (URL)
Workflow Record ID	Number

Reports Design

In this application, you are going to create a single report which lists all the temporary access request records. What makes this report interesting is that you will create a composite SmartObject which will combine data from the K2 Workflow Reporting service with data from the K2 SmartBox SmartObject service.

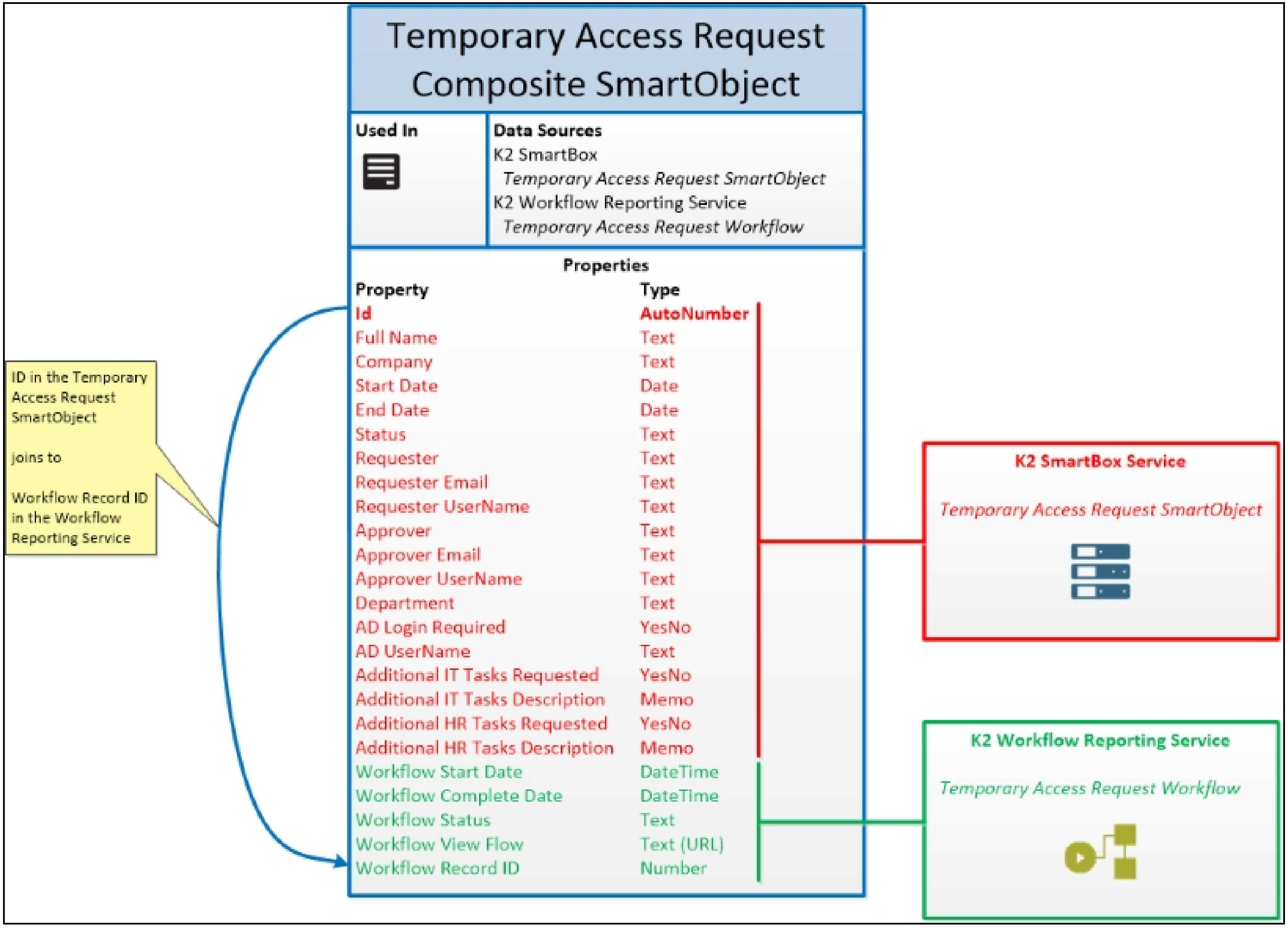


FULL NAME	COMPANY	START DATE	END DATE	STATUS	REQUESTER	WORKFLOW STAT...	WORKFLOW VIEW...
First Test	First Test Company	6/4/2015	6/8/2015	Expired	Denallix Administrator	Completed	https://k2.denalli...
Second Test	Second Test Company	6/4/2015	6/7/2015	Expired	Denallix Administrator	Completed	https://k2.denalli...
Third Test	Third Test Company	6/4/2015	6/4/2015	Expired	Denallix Administrator	Completed	https://k2.denalli...
Fourth Test	Fourth Test Company	6/2/2015	6/9/2015	Approved	Denallix Administrator	Active	https://k2.denalli...
Fifth Test	Fifth Test Company	6/4/2015	6/5/2015	Rejected	Denallix Administrator	Completed	https://k2.denalli...

Double-clicking a row launches the View Flow report

In effect, you are going to combine workflow reporting data with business data to build a report that is relevant to your users. In this case, you want to create a view of temporary access requests that anyone in the organization can review at any time, along with the workflow status and a link to the workflow. This is a security measure so that it is easy to see who is allowed to access the premises and so that it is easy to track down a workflow if it is paused at a user task. You might also use this report to identify if the workflow failed. To achieve this, you will build a composite SmartObject in K2 Studio, build a view against the SmartObject and then add the view to your Temporary Access request form.

The Temporary Access Request Composite SmartObject conceptual design



Temporary Access Request Composite SmartObject
Used In
Data Sources
K2 SmartBox
Temporary Access Request SmartObject
K2 Workflow Reporting Service
Temporary Access Request Workflow
Properties
ID in the Temporary Access Request SmartObject joins to Workflow Record ID in the Workflow Reporting Service
K2 SmartBox Service
Temporary Access Request SmartObject
K2 Workflow Reporting Service
Temporary Access Request Workflow

Property	Type
Id	AutoNumber
Full Name	Text
Company	Text
Start Date	Date
End Date	Date
Status	Text
Requester	Text
Requester Email	Text
Requester UserName	Text
Approver	Text
Approver Email	Text
Approver UserName	Text
Department	Text
AD Login Required	YesNo
AD UserName	Text
Additional IT Tasks Requested	YesNo
Additional IT Tasks Description	Memo
Additional HR Tasks Requested	YesNo
Additional HR Tasks Description	Memo
Workflow Start Date	DateTime
Workflow Complete Date	DateTime
Workflow Status	Text
Workflow View Flow	Text (URL)
Workflow Record ID	Number

From a learning perspective, the important point in this tutorial is how to build an advanced composite SmartObject that combines data from different services. While you are using K2 services for this example, the principles apply equally to other SmartObjects as well: you can use the same approach to build a SmartObject that, for example, combines data from your HR SQL database with your Active Directory system to build a logical employee details SmartObject.

Building the Application

Now that you are familiar with the designs for each of the components (workflow, data, forms, reports), you are ready to begin building the workflow and report parts of the application. (Again, because this tutorial focuses on workflows and SmartObjects with K2 Studio, you will be provided a starter project with the forms and the majority of the data elements. You will only focus on the workflow and composite SmartObject in this tutorial.)

When you are ready to start building the application, continue on to the Part 1: Building a workflow with K2 Studio.

Temporary Access Application: Part 1 (Building a workflow with K2 Studio)

In Part 1, you will build the basic parts of the workflow component for your Temporary Access application. Using K2 Studio to build the workflow, you will become familiar with the K2 Studio workflow design environment, how to build workflows and use wizards and how to format workflows for clarity.

> Note
> Note that the workflow will not be complete at the end of Part 1. You will add more advanced workflow concepts in Part 2 before deploying the workflow and testing it. You must complete Part 1 before proceeding to Part 2.

Step 1: Download and deploy the Temporary Access starter project

Because this tutorial focuses on using K2 Studio to build workflows and advanced SmartObjects, you are provided with

a starter project that contains the user interfaces (SmartForms) and SmartBox-based SmartObject. (Other tutorials go into more detail on creating basic SmartBox SmartObjects and SmartForms.)
You must deploy this starter project in your environment before beginning the application build.

> Note
> If you are using a virtual environment provided by K2, you may log into your environment using the following credentials:
> User Name: Denallix\Administrator
> Password: K2pass!

Step 1 Tasks

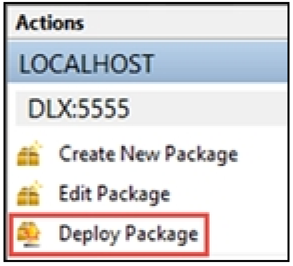
1. Download the K2 deployment package from http://help.k2.com/files/9872.
2. Use the K2 Package and Deployment tool to deploy the project file. (**Start > All Programs > K2 blackpearl > K2 Package and Deployment**)
3. Verify that the forms and SmartObject have been deployed.

Step 1 Walkthrough

a. Download the K2 project file located at http://help.k2.com/files/9872 and save it to your desktop. (To download the file, enter the URL in your browser. You should then see a message with the options to either open the file or save the file. Select **Save As** and save the file to your desktop.)

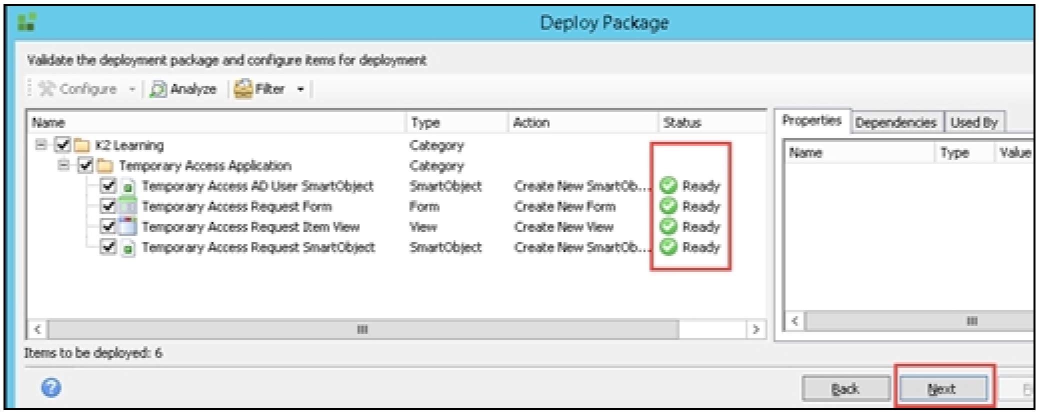
b. Launch the K2 Package and Deployment tool. (**Start > All Programs > K2 blackpearl > K2 Package and Deployment**)

c. Once the package and deployment tool has opened, from the right-side **Actions** column, click **Deploy Package**.



d. For the **File Name**, **Browse** to the K2 project file that you previously saved to your desktop. (The file name will be: **K2Learning-100.EDI_K2StudioIntro.kspx**.) Keep the default values for the remaining deployment options and click **Next**.

e. On the **Validate the deployment package** screen, CHECK the box next to **K2 Learning** to select all of the package elements. Wait a few seconds until you see the **Ready** indicators in the **Status** column. Click **Next** to continue.



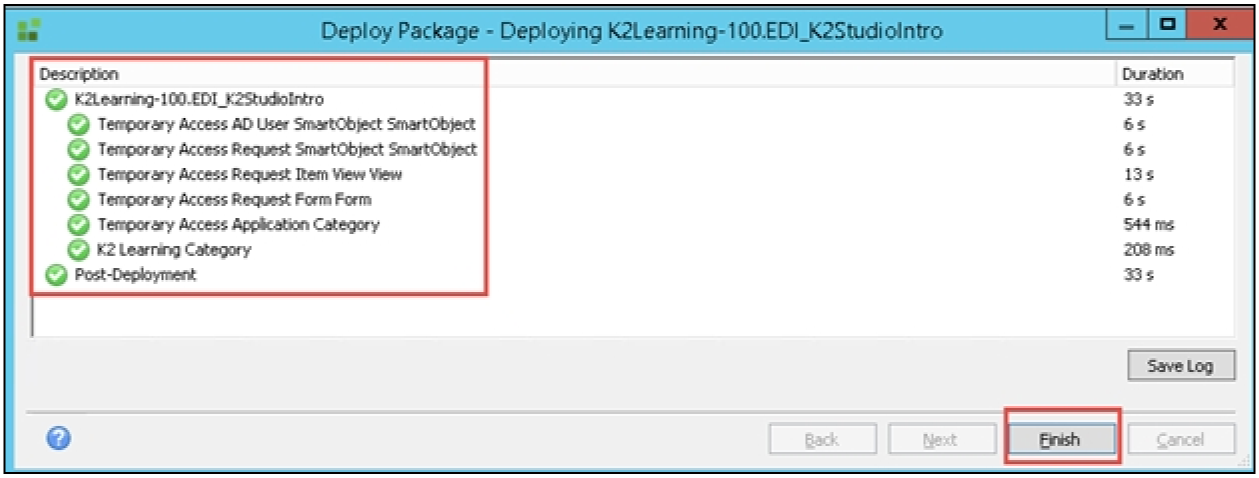
Validate the deployment package and configure items for deployment

Name	Type	Action	Status
☑ K2 Learning	Category		
☑ Temporary Access Application	Category		
☑ Temporary Access AD User SmartObject	SmartObject	Create New SmartOb...	Ready
☑ Temporary Access Request Form	Form	Create New Form	Ready
☑ Temporary Access Request Item View	View	Create New View	Ready
☑ Temporary Access Request SmartObject	SmartObject	Create New SmartOb...	Ready

Items to be deployed: 6

f. Allow a minute or two for the package deployment to process and complete. When you see the green success indicators next to the deployment descriptions, click **Finish**. Close the K2 Package and Deployment tool.

> Note
> The K2 Package and Deployment tool is the preferred tool for moving K2 artifacts between environments. For example, if you build an application within a development environment and then want to deploy it to a production environment. See the on-line help article on Package and Deployment for more information.

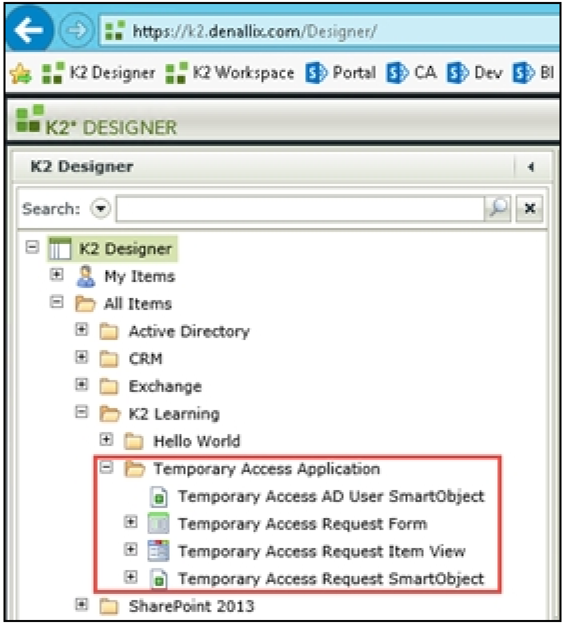


Deploy Package - Deploying K2Learning-100.EDI_K2StudioIntro

Description	Duration
✓ K2Learning-100.EDI_K2StudioIntro	33 s
✓ Temporary Access AD User SmartObject SmartObject	6 s
✓ Temporary Access Request SmartObject SmartObject	6 s
✓ Temporary Access Request Item View View	13 s
✓ Temporary Access Request Form Form	6 s
✓ Temporary Access Application Category	544 ms
✓ K2 Learning Category	208 ms
✓ Post-Deployment	33 s

g. Now you will confirm that the forms and SmartObjects have been deployed.
Launch K2 Designer (**Start > All Programs > K2 blackpearl > K2 Designer**)
(If the menu link is missing, contact your K2 administrator to obtain the URL for K2 Designer.)

h. Once K2 Designer has opened, browse to the following category:
All Items > K2 Learning > Temporary Access Application
You should see the following item view, form and two SmartObjects in this category. If all four artifacts are there, the deployment was successful, continue to Step 2.



- K2 Designer
  - My Items
  - All Items
    - Active Directory
    - CRM
    - Exchange
    - K2 Learning
      - Hello World
      - Temporary Access Application
        - Temporary Access AD User SmartObject
        - Temporary Access Request Form
        - Temporary Access Request Item View
        - Temporary Access Request SmartObject
    - SharePoint 2013

Step 1 Review

In this step you downloaded and deployed a starter project for this tutorial. The starter project contains a view, a form and two SmartObjects. You will be using these items in the workflow and the composite SmartObject steps later on.

Step 2: Create a new K2 Studio project and add a blank workflow

In this step you will create a new project with K2 Studio and add a workflow to the project.

Step 2 Tasks

1. Launch **K2 Studio** and create a new project with the following settings:

Field	Value
K2 Type	**K2 Project**
K2 Template	**K2 Project**
Project Name	*K2 Learning*
Location	(Select a location such as *C:\K2 Learning*)
Target Framework	(use default value)

2. Add a folder called to the K2 Learning project and name it *Temporary Access Application*
3. Add a new workflow to the project and name it *Temporary Access Application Workflow*

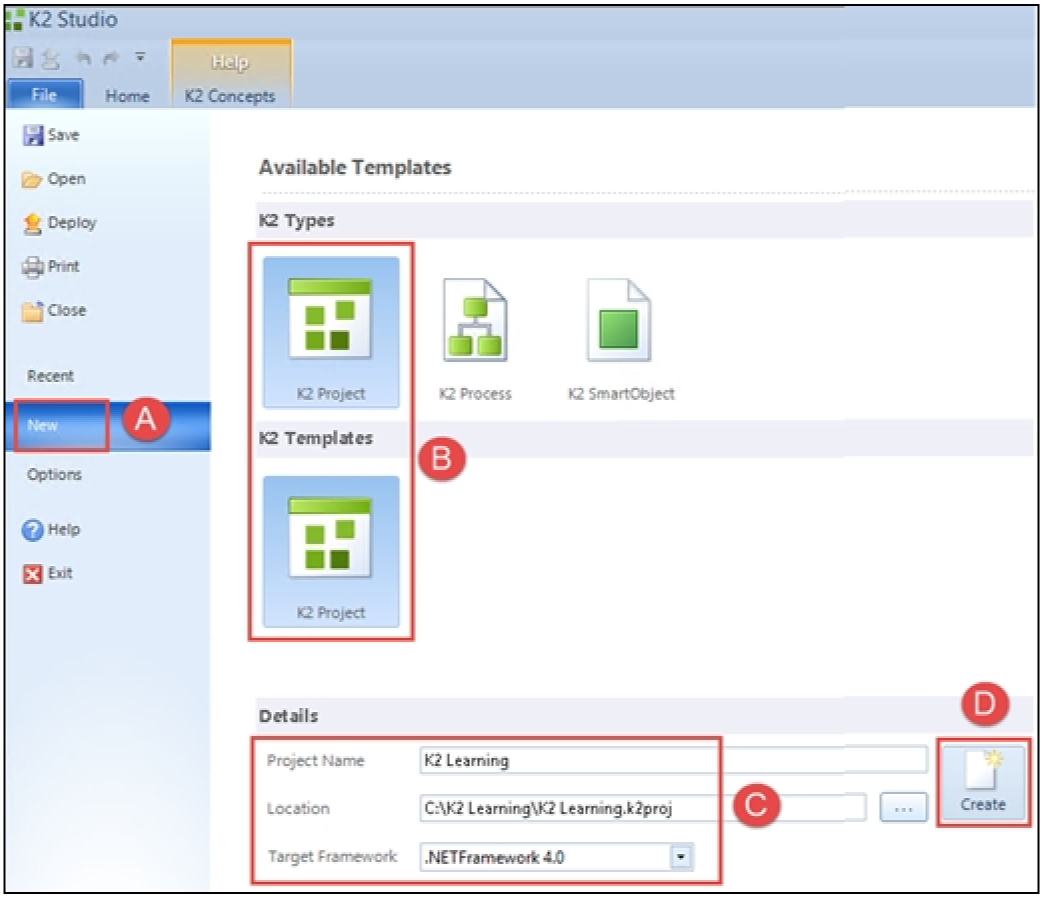
Step 2 Walkthrough

a. Launch K2 Studio (**Start** > **All Programs** > **K2 blackpearl** > **K2 Studio**)

b. Click on **File** > **New**.

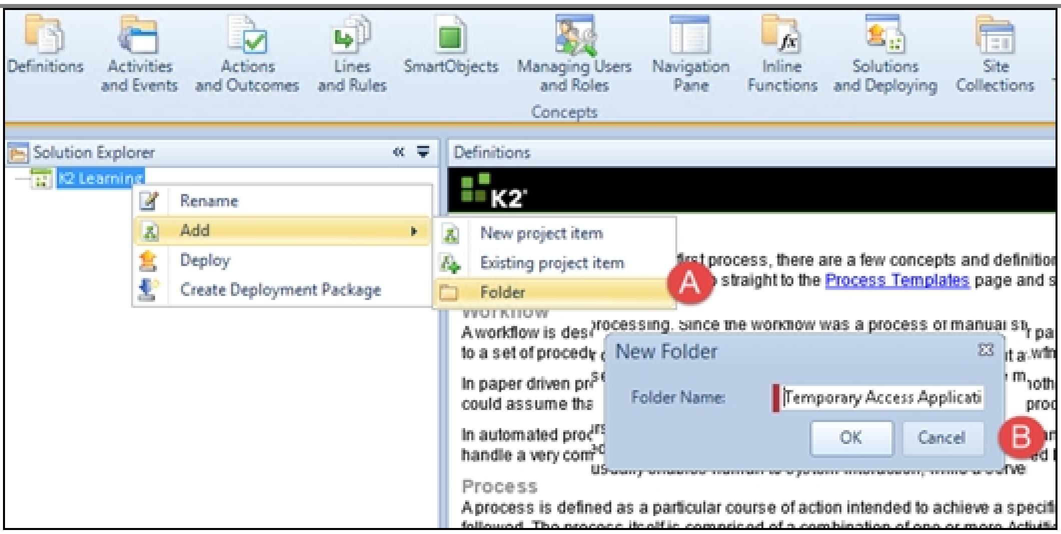
c. Create a new K2 Project with the following settings. Click **Create** when you are ready.

Field	Value
K2 Type	**K2 Project**
K2 Template	**K2 Project**
Project Name	*K2 Learning*
Location	(Select a location such as *C:\K2 Learning*)
Target Framework	(use default value)



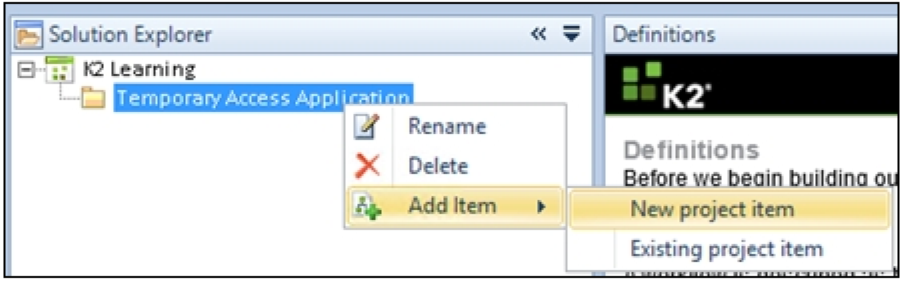
Project Name: K2 Learning
Location: C:\K2 Learning\K2 Learning.k2proj
Target Framework: .NETFramework 4.0

d. Once the new project is created, right-click on the **K2 Learning** node and select **Add** > **Folder**.
Name the new folder
Temporary Access Application

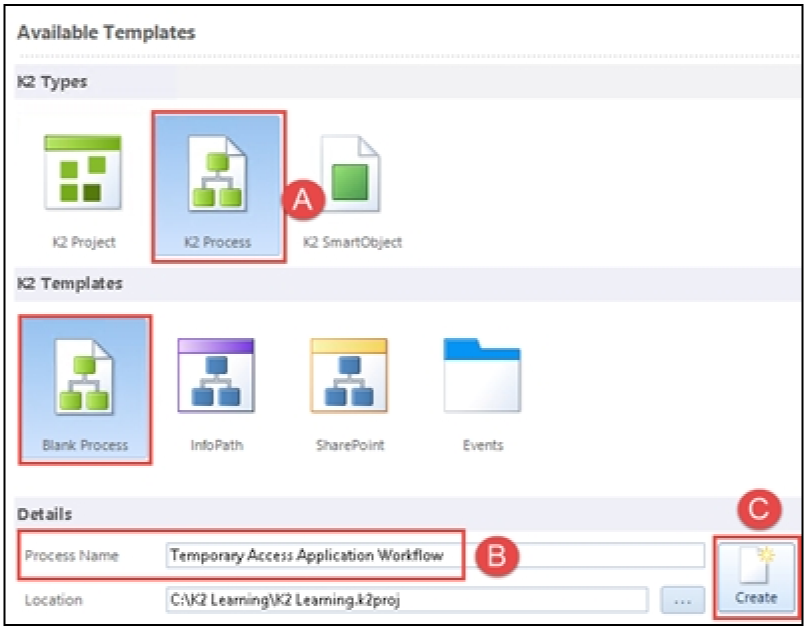


Folder Name: Temporary Access Applicati

e. Right-click on the new folder and click **Add Item** > **New project item**.



f. Notice that **K2 Process** and **Blank Process** are preselected for you. **Name** the new process *Temporary Access Application Workflow* then click **Create**.



Process Name: Temporary Access Application Workflow

Location: C:\K2 Learning\K2 Learning.k2proj

g. Once the workflow is created, click **File** > **Save** to save your project.

Step 2 Review

In this step you created a new K2 Studio project and added a blank workflow to the project. You used a specific folder structure (K2 Learning) so that when the workflow is deployed, it will deploy to the existing (K2 Learning) category that was created for this application when you deployed the starter project.

Step 3: Configure the workflow to start when a SmartForm is submitted

In this step you will add a data field to the workflow, then configure the workflow to start when the Temporary Access Request Form is submitted. The data field is necessary as later on you will use this field to bind the workflow reporting data (for example, number of process instances, average process instance duration, etc.) with the Temporary Access Request (user details) data. You will then create a composite SmartObject that joins these two data sources, essentially allowing you to create a report that combines workflow reporting data with business (user) data.

> Note
> The majority of the wizard configuration is specific to SmartForms because you want a SmartForm to start this particular workflow. If you use a different mechanism to start the workflow, the configuration and wizard screens will be different. Other tutorials illustrate how to set up workflows that start with different forms or different event-based mechanisms. The main thing to understand here is that you are using a wizard to configure how the workflow will be started. SmartForms training dives into more detail about the SmartForms-specific configurations, so for now do not be too concerned if you do not immediately grasp the form-workflow integration.

Step 3 Tasks

1. Add a data field to the workflow
   Name: *RecordID*
   Data Type: **Integer**
2. Use the **SmartForms Integration** wizard to configure the workflow to start when the **Temporary Access Request Form** is submitted. Use the following settings:

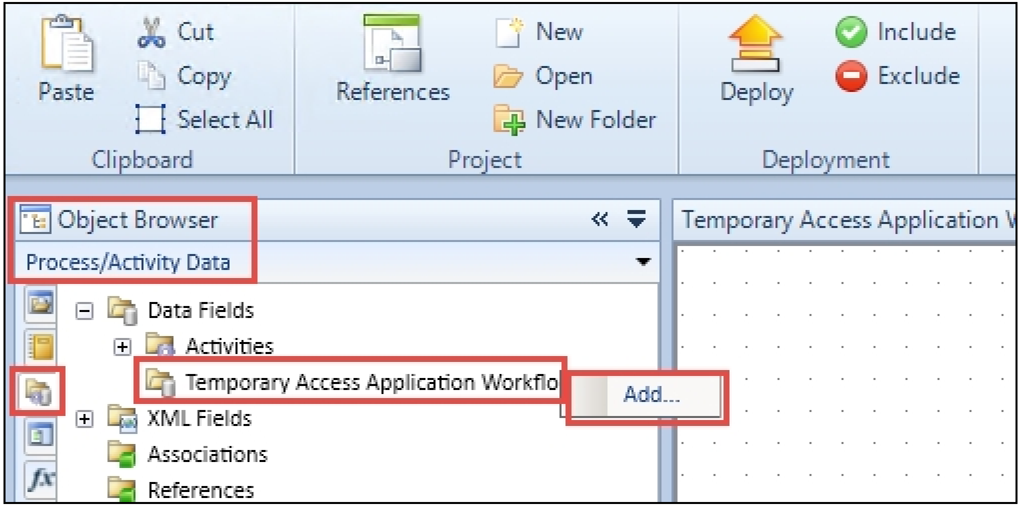
Field	Value
Form	**Temporary Access Request Form**
State	**Use Existing State > NewRequest (Default)**
Select Rule	**When Button Submit is Clicked**. Move the **Start Workflow** action up so that it happens right before the message is shown to the user.
Item Reference	Create an Item Reference called *Temporary Access Request*; browse for and select the **Temporary Access Request SmartObject**.
Data Binding	Link the **RecordID** workflow data field with the **ID** of the **Temporary Access Request SmartObject**
Folio	**FullName - Company** from the **Temporary Access Request SmartObject**

Step 3 Walkthrough

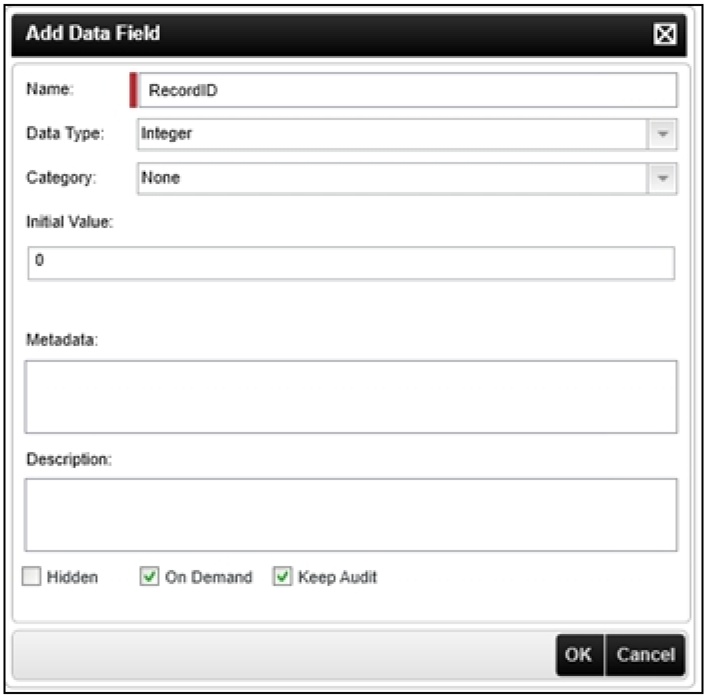
a. First you will add a data field to store the ID of the current process instance (SmartObject) record. Open the **Object Browser** and then the **Process/Activity Data** tab. The goal is to have a relationship between the SmartObject record and the process instance.

> Note
> *Workflows, Processes and Process Instances*. In K2 terms, a process is the equivalent of a workflow (and the two terms are used throughout the learning materials). That is, a process is made up of all the activities, events and lines (and more) that provide some sort of business process automation. A process instance (also referred to as a workflow instance) is a single occurrence of a process. For example, if Neil starts a workflow, a process instance of that workflow is created by K2 on Neil's behalf. If Jason starts a workflow, a process instance of that workflow is created on Jason's behalf.

b. Right-click on the process name in the **Process/Activity Data** tab and select **Add…**



c. Add a data field with the following settings:
Name: *RecordID*
Data Type: **Integer**
Click **OK**



Add Data Field

Name: RecordID

Data Type: Integer

Category: None

Initial Value:

0

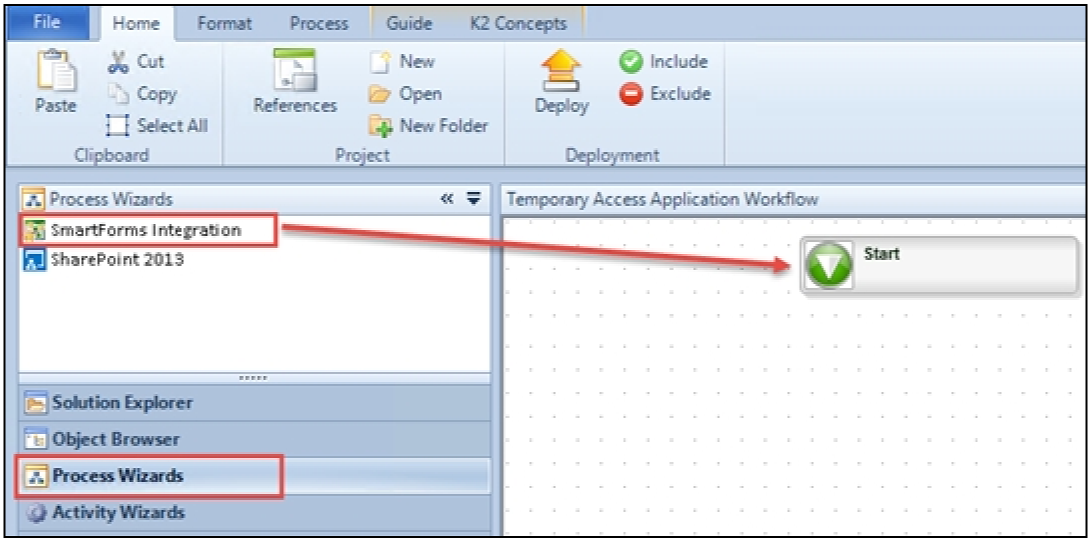
Metadata:

Description:

☐ Hidden ☑ On Demand ☑ Keep Audit

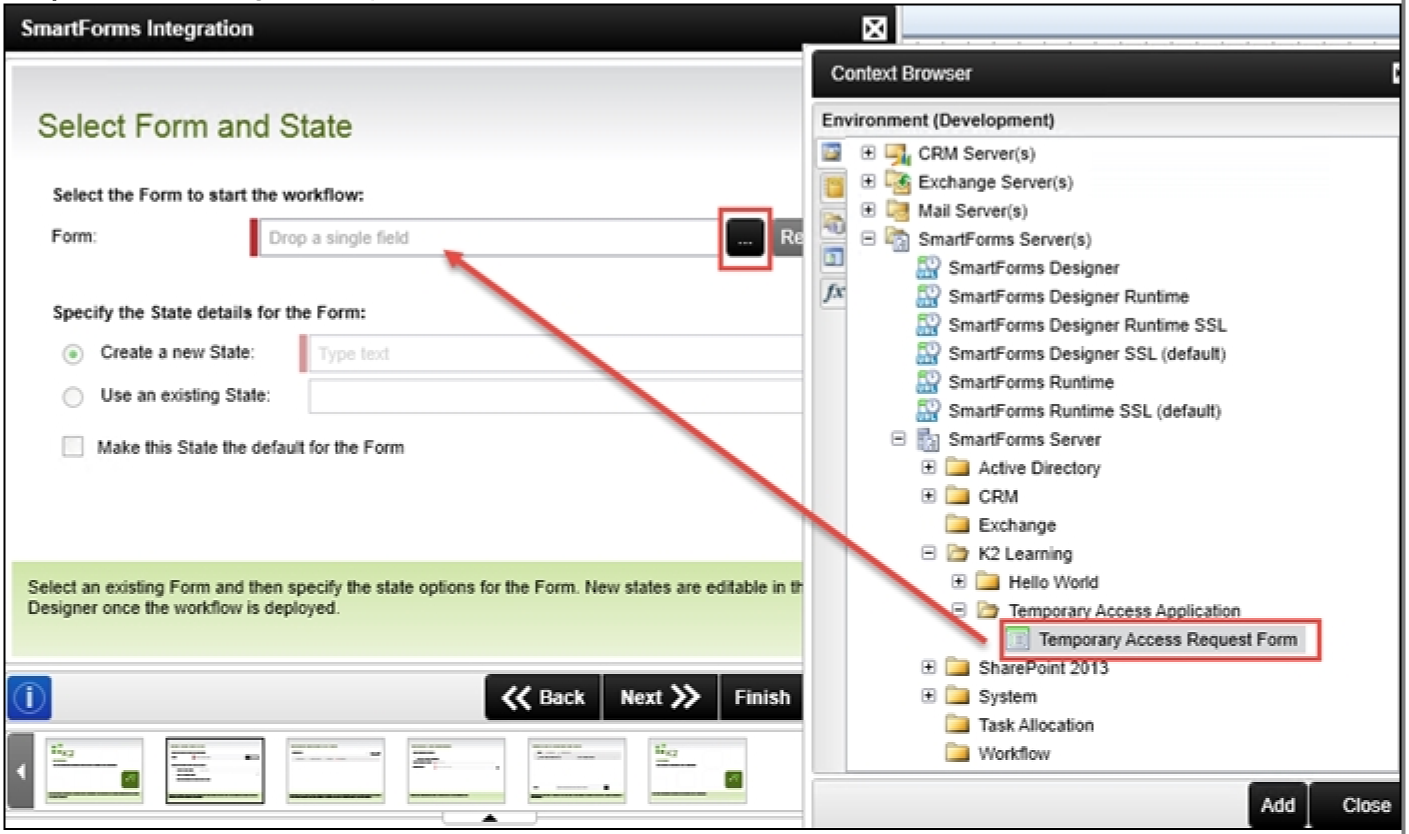
OK Cancel

d. Now you will configure the workflow to start when the Temporary Access Request Form is submitted. Click on **Process Wizards**, then drag and drop the **SmartForms Integration** wizard into the **Start** box.



e. Click **Next** on the wizard **Welcome** page.

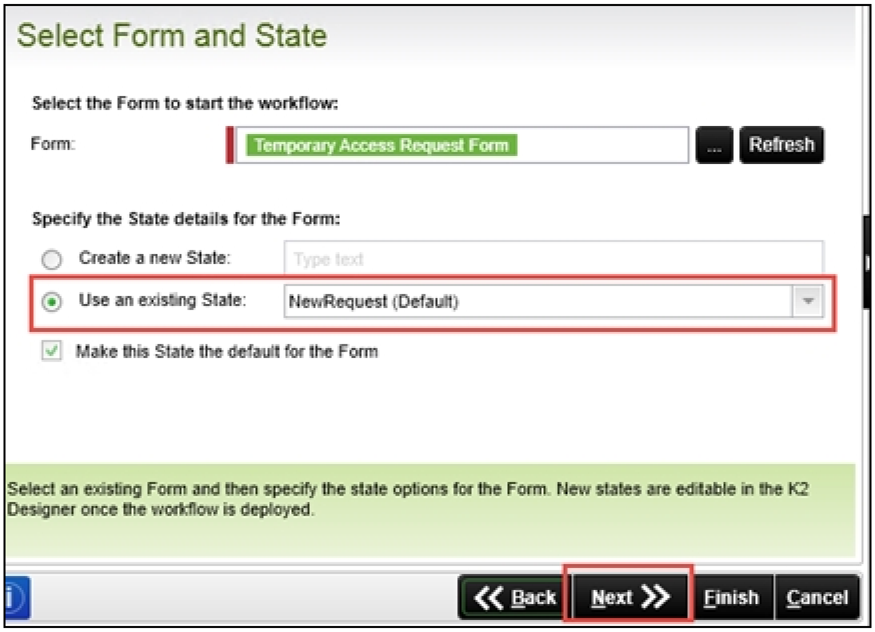
f. On the **Select Form and State** screen, use the browse icon to select the **Temporary Access Request Form**. Drag and drop the form into the **Form** text box.


SmartForms Integration
Select Form and State
Select the Form to start the workflow:
Form:
Drop a single field
...
Re
Specify the State details for the Form:
● Create a new State:
Type text
○ Use an existing State:
□ Make this State the default for the Form
Select an existing Form and then specify the state options for the Form. New states are editable in th
Designer once the workflow is deployed.
<< Back
Next >>
Finish
Context Browser
Environment (Development)
CRM Server(s)
Exchange Server(s)
Mail Server(s)
SmartForms Server(s)
SmartForms Designer
SmartForms Designer Runtime
SmartForms Designer Runtime SSL
SmartForms Designer SSL (default)
SmartForms Runtime
SmartForms Runtime SSL (default)
SmartForms Server
Active Directory
CRM
Exchange
K2 Learning
Hello World
Temporary Access Application
Temporary Access Request Form
SharePoint 2013
System
Task Allocation
Workflow
Add
Close

g. For the **State** of the form, select **Use an Existing State** and pick the **NewRequest (Default)** state. Click **Next** to continue.

Note

A *state* represents the configuration and behavior of a form at a given point in the workflow. States are most commonly configured at the user task event level. For example, you might want the form configuration to contain certain fields for the manager approval task, then contain different fields for the originator rework task. K2 can use the same form and views with adjusted configurations 'assigned' to a specific state. Breaking this down a little further, each user task can be assigned a state of its own. The form designer can then manipulate the form's styling and rules specific to that single state.



Select Form and State

Select the Form to start the workflow:

Form: Temporary Access Request Form

Specify the State details for the Form:

○ Create a new State: Type text

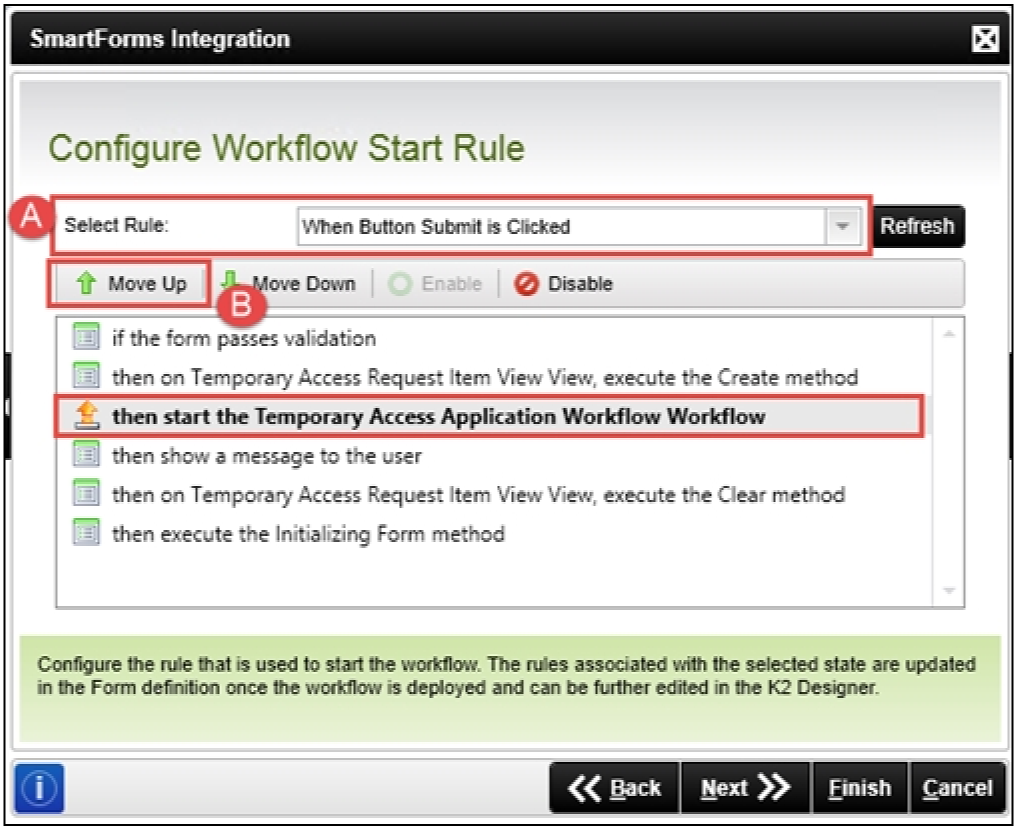
● Use an existing State: NewRequest (Default)

☑ Make this State the default for the Form

Select an existing Form and then specify the state options for the Form. New states are editable in the K2 Designer once the workflow is deployed.

h. On the **Configure Workflow Start Rule** screen, select the **When Button Submit is Clicked** rule from the **Select Rule** drop-down list. If you see a warning message, click **OK** to accept the warning. Use the **Move Up** button to move the **then Start the Temporary Access Application Workflow Workflow** action up so that it happens right before the message is shown to the user.

WARNING: Do not skip this step. Doing so will result in the workflow throwing an error.



SmartForms Integration

Configure Workflow Start Rule

Select Rule: When Button Submit is Clicked

if the form passes validation

then on Temporary Access Request Item View View, execute the Create method

then start the Temporary Access Application Workflow Workflow

then show a message to the user

then on Temporary Access Request Item View View, execute the Clear method

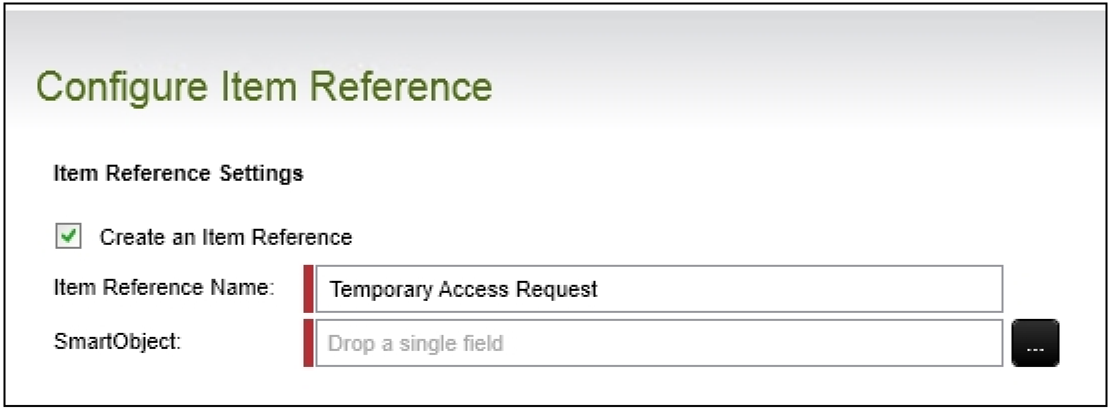
then execute the Initializing Form method

Configure the rule that is used to start the workflow. The rules associated with the selected state are updated in the Form definition once the workflow is deployed and can be further edited in the K2 Designer.

What you are doing in this step is to tell the workflow to start when the user clicks the submit button on the request form. This particular configuration is specific to K2 SmartForms. If you were using a different form technology, you would configure the workflow start rule differently.

i. Click **Next** to continue.

j. On the **Configure Item Reference** screen, select the **Create Item Reference** option.
Name the Item Reference
Temporary Access Request



Configure Item Reference

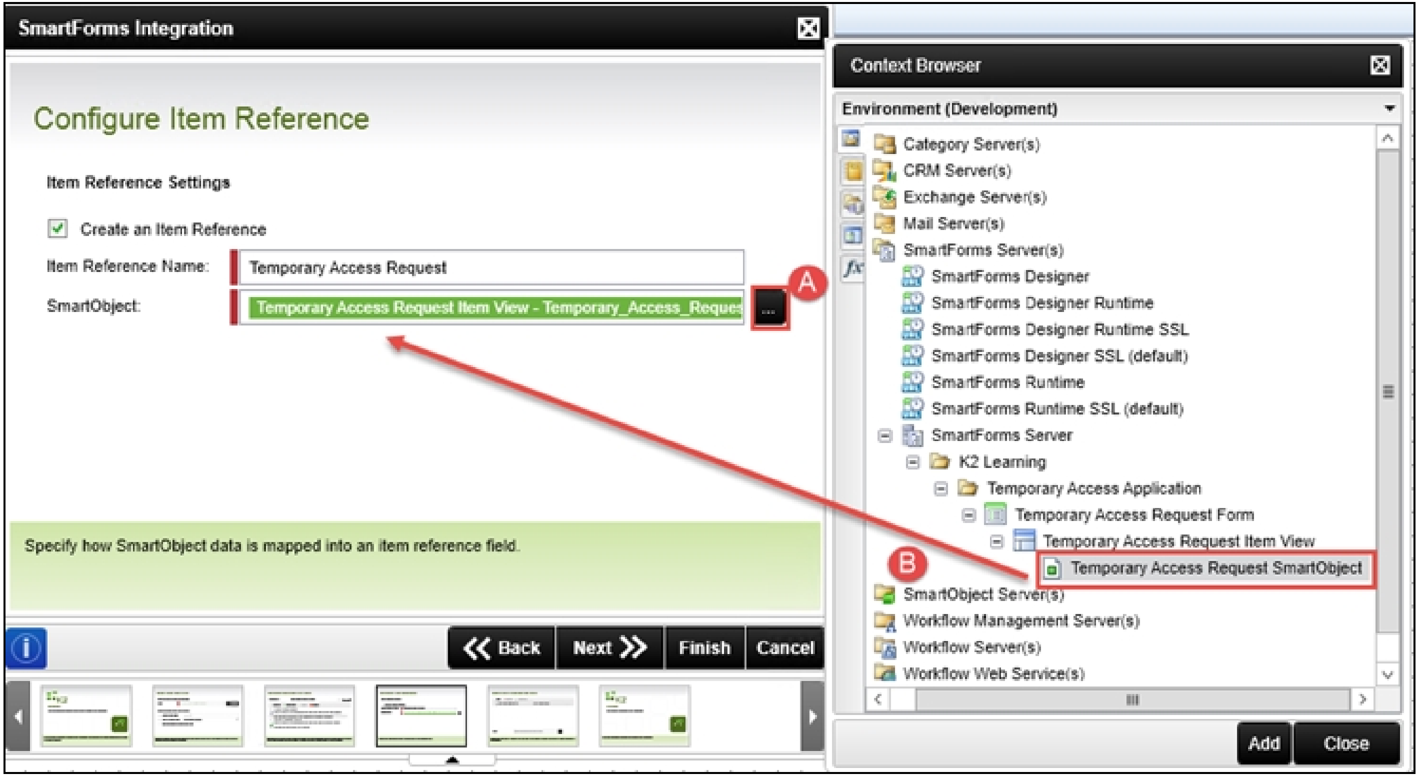
Item Reference Settings

☑ Create an Item Reference

Item Reference Name: Temporary Access Request

SmartObject: Drop a single field

Use the ellipsis button to browse for and select, then drag, the **Temporary Access Request SmartObject** into the **SmartObject** field.



SmartForms Integration

Configure Item Reference

Item Reference Settings

☑ Create an Item Reference

Item Reference Name: Temporary Access Request

SmartObject: Temporary Access Request Item View - Temporary_Access_Reques

Specify how SmartObject data is mapped into an item reference field.

Context Browser

Environment (Development)

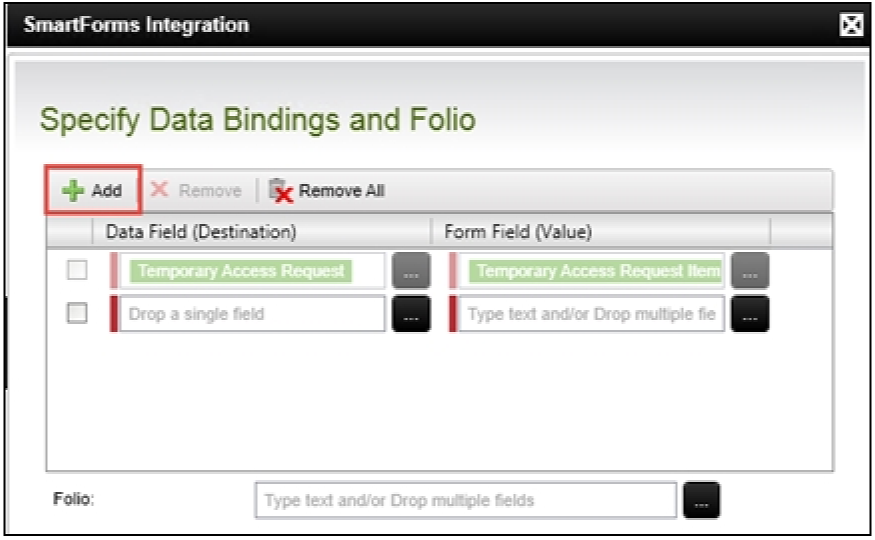
- Category Server(s)
- CRM Server(s)
- Exchange Server(s)
- Mail Server(s)
- SmartForms Server(s)
  - SmartForms Designer
  - SmartForms Designer Runtime
  - SmartForms Designer Runtime SSL
  - SmartForms Designer SSL (default)
  - SmartForms Runtime
  - SmartForms Runtime SSL (default)
  - SmartForms Server
    - K2 Learning
      - Temporary Access Application
        - Temporary Access Request Form
          - Temporary Access Request Item View
            - Temporary Access Request SmartObject
- SmartObject Server(s)
- Workflow Management Server(s)
- Workflow Server(s)
- Workflow Web Service(s)

Item References
K2 keeps track of items that relate to the current process instance, known as Item References. For example, when a workflow is started from a SharePoint list or library, or from a SmartObject, K2 will store a *pointer* to the current record in the Item References properties so that those values can be used later on in the workflow or rules. For now, just think of Item References as a shortcut to the current record values from the SmartObject, or data source.

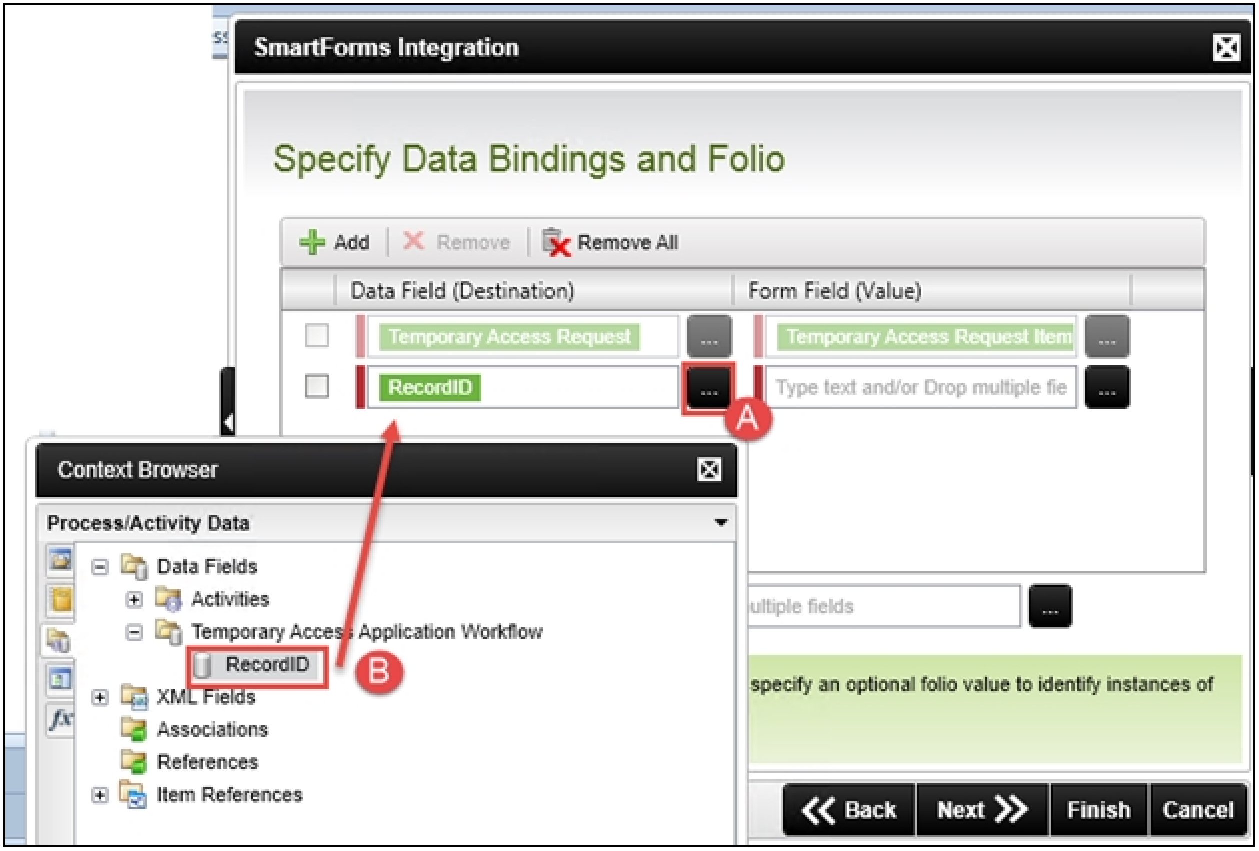
k. **Close** the context browser, then click **Next** to continue.

In a previous step, you created a data field (RecordID) so that you could save the ID for the current SmartObject record that is created as part of the process instance. In the next few steps, you are going to map the RecordID data field to your SmartObject ID property. This will complete the association between the SmartObject record that has been created and its corresponding process instance.

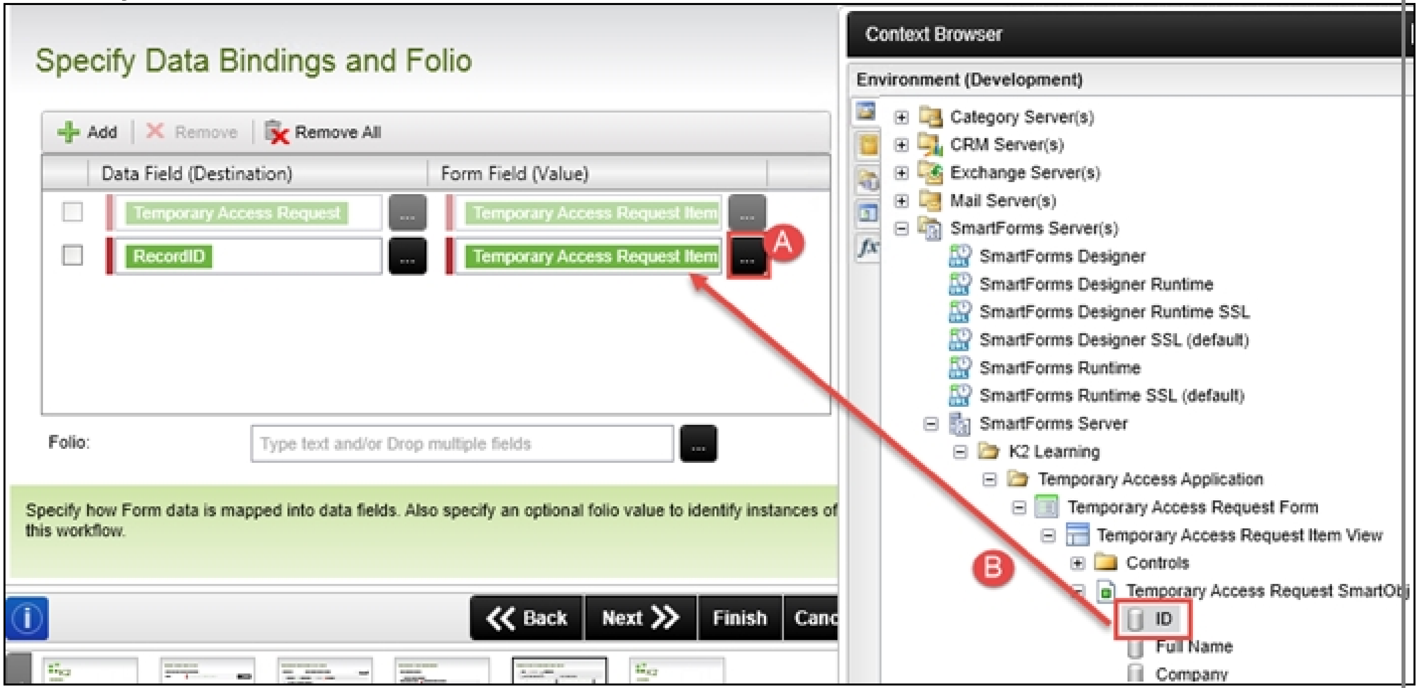
l. On the **Specify Data Bindings and Folio** screen, click on **Add**.



m. Click on the first ellipsis icon to select the **RecordID** data field you created earlier.

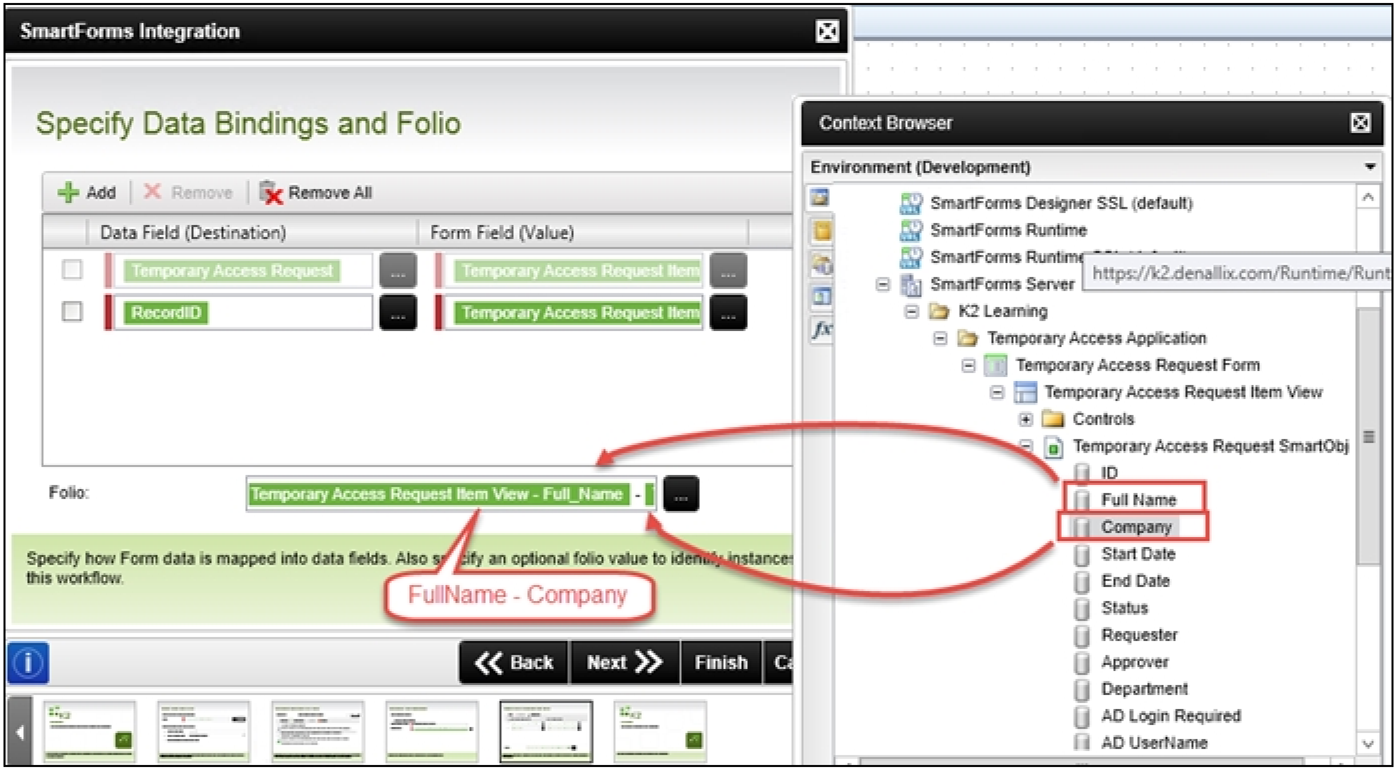


n. Click on the second ellipsis icon and select the **ID** property of the **Temporary Access Request SmartObject**. **Close** the context browser.



Specify Data Bindings and Folio
Add
Remove
Remove All
Data Field (Destination)
Form Field (Value)
Temporary Access Request
Temporary Access Request Item
RecordID
Temporary Access Request Item
A
Folio:
Type text and/or Drop multiple fields
Specify how Form data is mapped into data fields. Also specify an optional folio value to identify instances of this workflow.
Back
Next
Finish
Canc
Context Browser
Environment (Development)
Category Server(s)
CRM Server(s)
Exchange Server(s)
Mail Server(s)
SmartForms Server(s)
SmartForms Designer
SmartForms Designer Runtime
SmartForms Designer Runtime SSL
SmartForms Designer SSL (default)
SmartForms Runtime
SmartForms Runtime SSL (default)
SmartForms Server
K2 Learning
Temporary Access Application
Temporary Access Request Form
Temporary Access Request Item View
Controls
Temporary Access Request SmartObj
B
ID
Full Name
Company

o. For the **Folio**, use the ellipsis button to configure the folio to use the **Full Name** and **Company** of the **Temporary Access Request SmartObject**. (See the note box below the image for a brief explanation of the folio, if necessary.) **Close** the context browser.



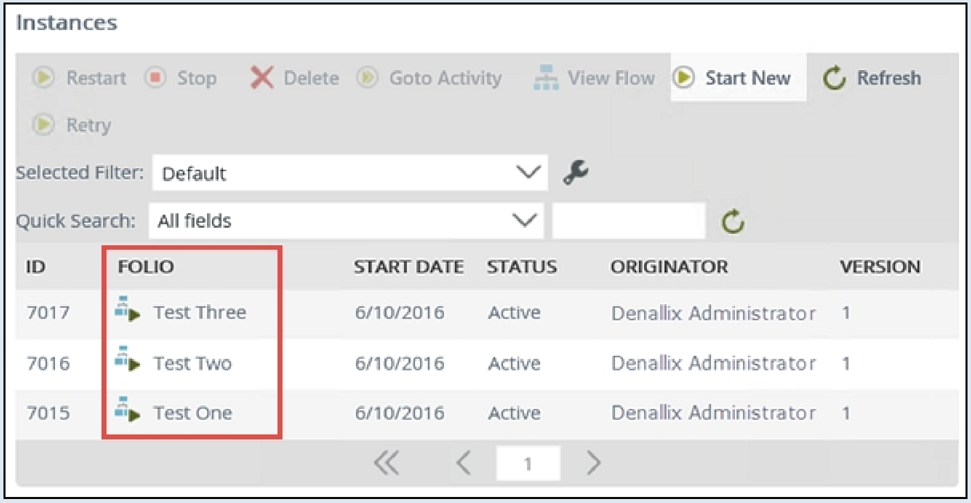
SmartForms Integration
Specify Data Bindings and Folio
Add
Remove
Remove All
Data Field (Destination)
Form Field (Value)
Temporary Access Request
Temporary Access Request Item
RecordID
Temporary Access Request Item
Folio:
Temporary Access Request Item View - Full_Name -
Specify how Form data is mapped into data fields. Also specify an optional folio value to identify instances this workflow.
FullName - Company
Back
Next
Finish
Ca
Context Browser
Environment (Development)
SmartForms Designer SSL (default)
SmartForms Runtime
SmartForms Runtime
https://k2.denallix.com/Runtime/Runt
SmartForms Server
K2 Learning
Temporary Access Application
Temporary Access Request Form
Temporary Access Request Item View
Controls
Temporary Access Request SmartObj
ID
Full Name
Company
Start Date
End Date
Status
Requester
Approver
Department
AD Login Required
AD UserName

Note

What is the Folio? The folio is a text field that is commonly used to distinguish one process instance from another process instance (of the same workflow). The folio is not required, nor does it have to be unique. Since all processes, or workflows, will have the same process name, using a unique folio value will aid in identifying the individual process instances of the workflow. This is especially useful

when viewing workflow reports having many process instances.
To make the folio unique, you can use a combination of fields, such as a customer name combined with an order number. Setting the folio value typically comes from properties found in the context browser. The property values are variables that are replaced at runtime by actual content specific to the process instance.
You can also use the folio as a variable throughout your workflow. For example, if you assign a customer name as the folio, you can use the folio to customize emails so that the customer is referenced by their name.

Three instances of the same workflow showing unique folio values



Selected Filter: Default

Quick Search: All fields

ID	FOLIO	START DATE	STATUS	ORIGINATOR	VERSION
7017	Test Three	6/10/2016	Active	Denallix Administrator	1
7016	Test Two	6/10/2016	Active	Denallix Administrator	1
7015	Test One	6/10/2016	Active	Denallix Administrator	1

To review the steps you just completed, you configured the workflow wizard to pass the ID of the SmartObject record that is created when a new process instance is created, to the RecordID data field. This will allow you to join the workflow reporting data with the user's request data in a later exercise. You also configured the workflow folio so that it will use the full name, along with the company name that is entered by the requester. This folio value will distinguish one process instance from another.

p. **Close** the context browser, then click **Next** and **Finish** to complete the SmartForms wizard.

q. **Save** your workflow.

Step 3 Review

In this step you added a data field to the workflow and configured the workflow to start when the Temporary Access Request Form is submitted. As part of the configuration, you specified when the workflow should be started and opted to create an item reference to the SmartObject record that is created. This will allow you to use values from the SmartObject record throughout your workflow and in form rules. You mapped the SmartObject record ID to the RecordID data field, that essentially binds the record that is created with the process instance. The majority of this configuration is specific to SmartForms because you want a SmartForm to start the workflow.

Step 4: Add a client event to the workflow

In this step you will add a client event (or user task) to the workflow. You will be using the same Temporary Access Request Form and using a specific state of the form for the task. While this approval task will have two actions (approve and reject), there will be three outcomes:

- Rejected (the approver rejects the request)
- Approved and AD Account required (the approver approved the request and the workflow must create an Active Directory account for the temporary user)
- Approved and AD Account not required (the approver approved the request and the workflow does not need to create an Active Directory account for the temporary user)

The workflow will follow one of the three outcomes above, depending on the requester's options selected from the request form.

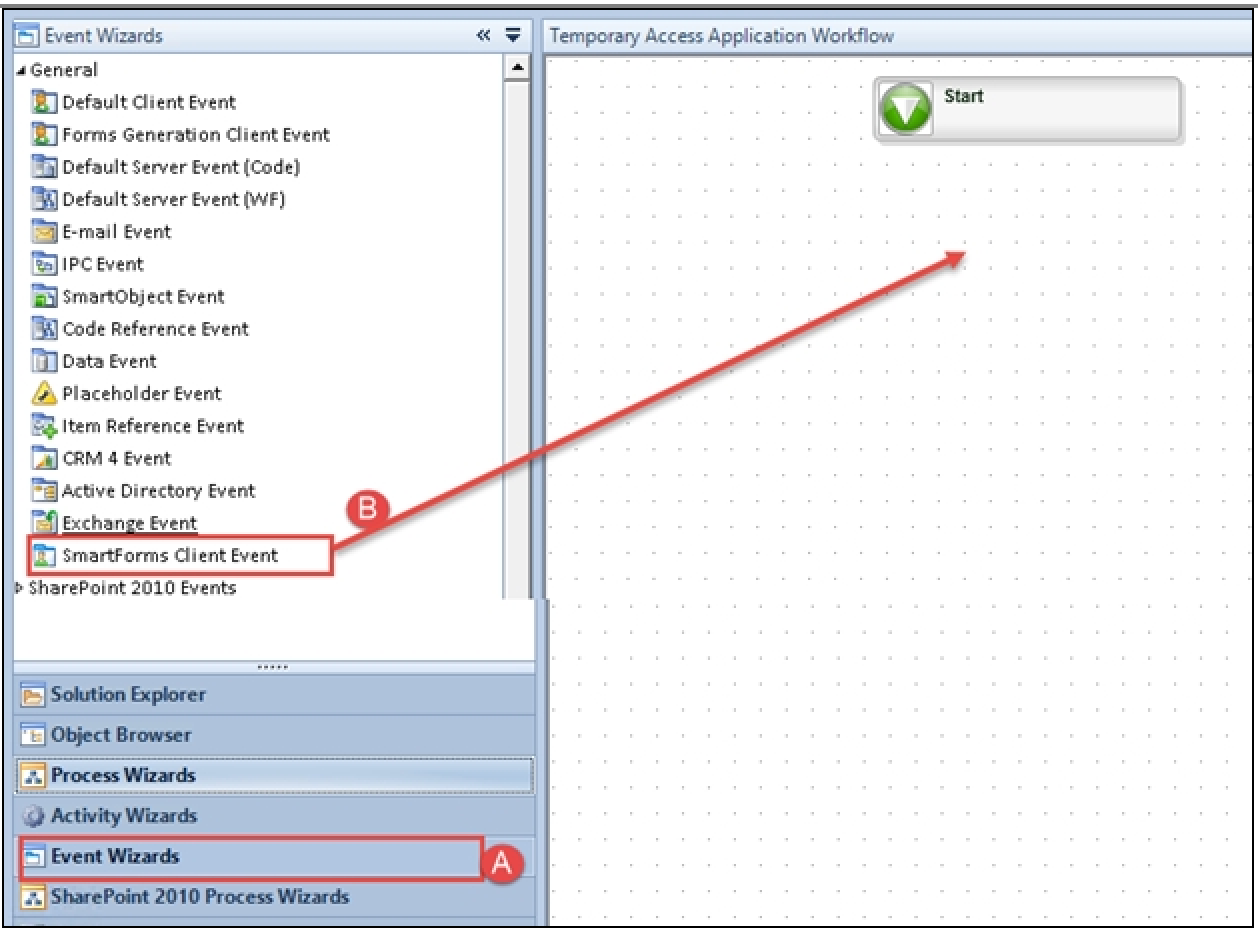
Step 4 Tasks

1. Add a **SmartForms Client Event** to the workflow and configure it as follows:

Field	Value
Name	*Manager Approval*
Form	**Temporary Access Request Form**
State	Use existing **Approval** state
Configure Item Reference	(no changes, use default values)
Specify Form Parameters	(no changes, use default values)
Configure Open Worklist Item Rule	(no changes, use default values)
Transfer Workflow Data to Form	(no changes, use default values)
Actions	*Approved* *Rejected*
Action Display Settings	(no changes, use default values)
Transfer Form Data to Workflow	(no changes, use default values)
Outcomes	Edit the existing **Approval** outcome. Add a condition **And** **Temporary Access Request.AD Login Required** **=** *True* Rename the outcome to *Approved and AD Account required* Add another outcome so that **All Slots = Approved** **And** **Temporary Access Request.AD Login Required =** *False* Rename the outcome to *Approved and AD Account not required* Leave the **Reject** outcome as-is
Destination User	**Temporary Access Request Item Reference > Approver Login Name**
Task Notification	Customize the task notification email and add some contextual information about the temporary access requests, such as the **Full Name**, **Company**, **Start Date**, **End Date** and **Requester**.

Step 4 Walkthrough

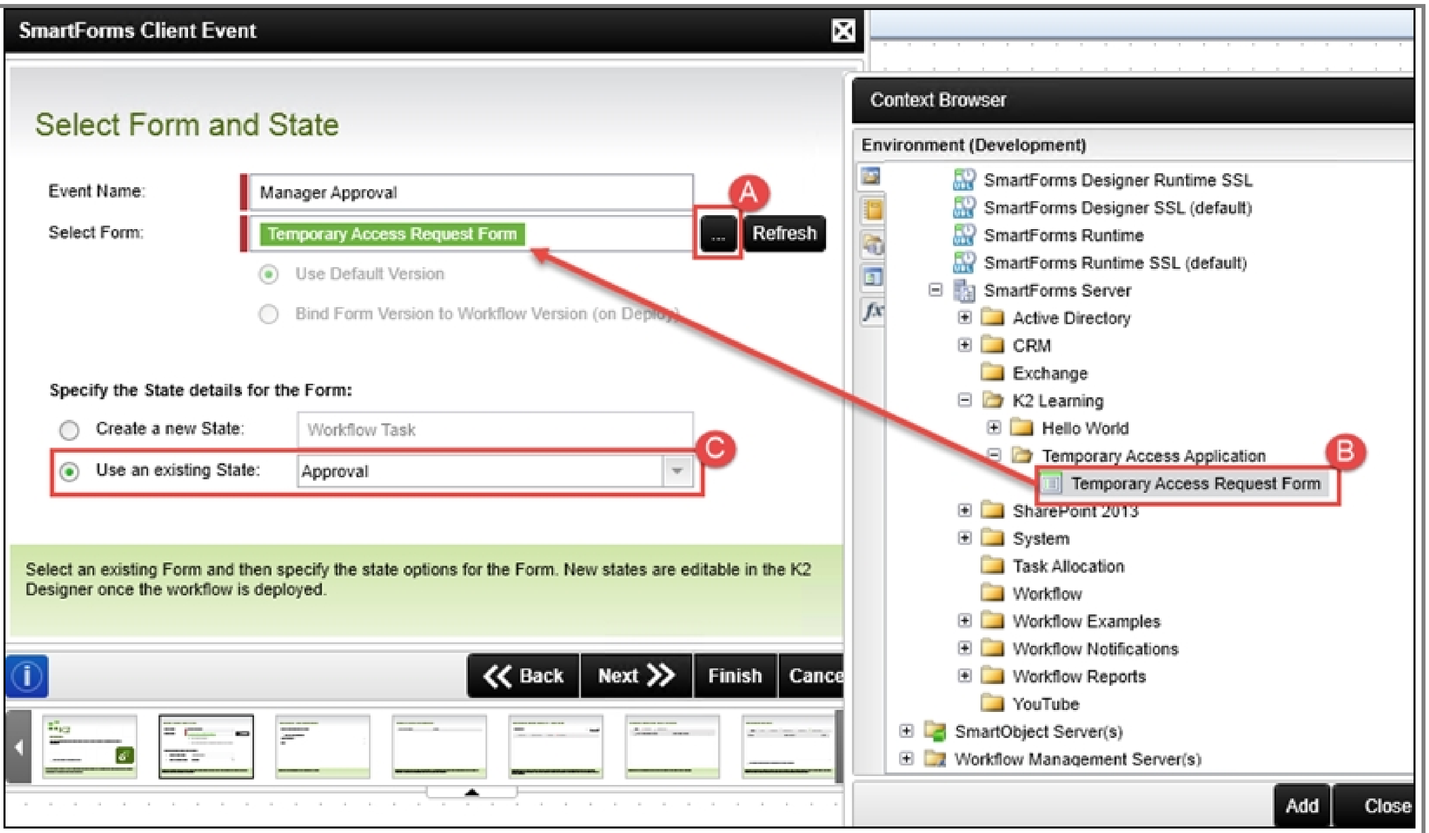
a. Select the **Event Wizards** pane and then drag and drop a **SmartForms Client Event** onto the workflow design canvas.



b. Click **Next** on the wizard welcome page.

c. On the **Select Form and State** screen, change the **Event Name** to
Manager Approval
Select the **Temporary Access Request Form**, then configure the **State** to use the existing **Approval State**.

You are now changing the state from the default state to a new state that you will later adjust so that it is specific to this user task.



SmartForms Client Event

Select Form and State

Event Name: Manager Approval

Select Form: Temporary Access Request Form

○ Use Default Version

○ Bind Form Version to Workflow Version (on Deploy)

Specify the State details for the Form:

○ Create a new State: Workflow Task

● Use an existing State: Approval

Select an existing Form and then specify the state options for the Form. New states are editable in the K2 Designer once the workflow is deployed.

Context Browser

Environment (Development)

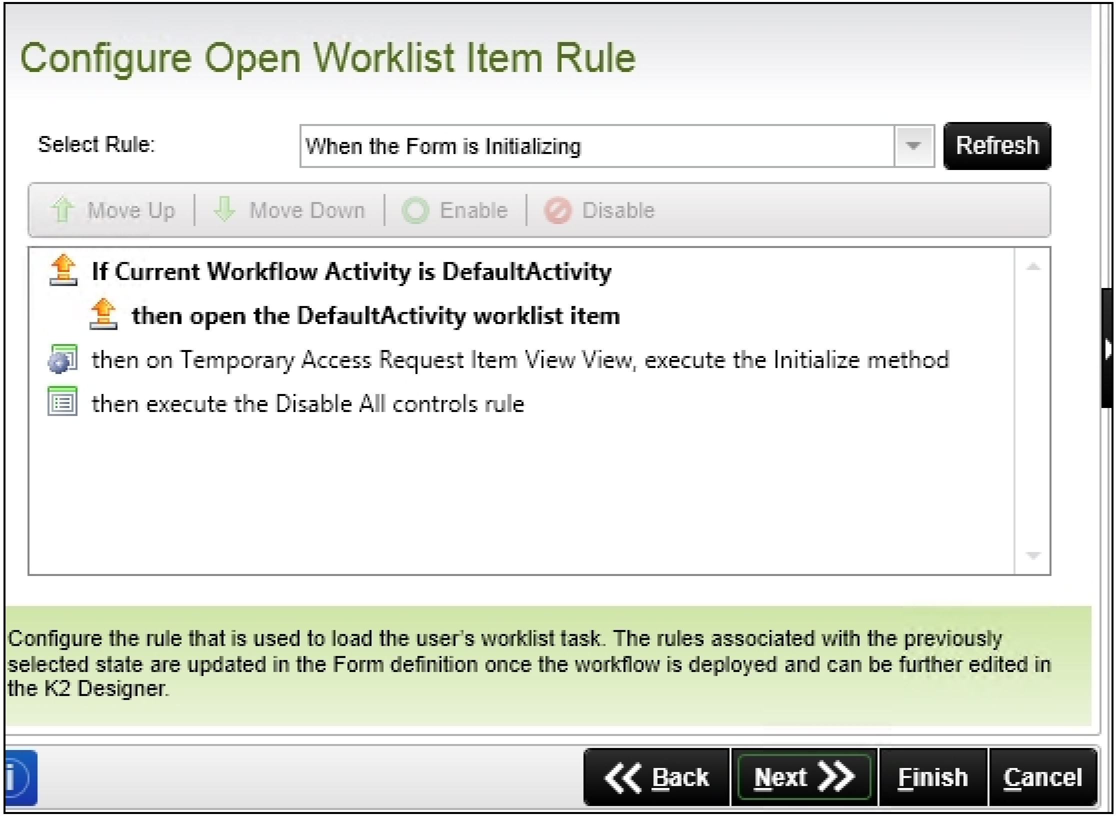
- SmartForms Designer Runtime SSL
- SmartForms Designer SSL (default)
- SmartForms Runtime
- SmartForms Runtime SSL (default)
- SmartForms Server
  - Active Directory
  - CRM
  - Exchange
  - K2 Learning
    - Hello World
    - Temporary Access Application
      - Temporary Access Request Form
  - SharePoint 2013
  - System
  - Task Allocation
  - Workflow
  - Workflow Examples
  - Workflow Notifications
  - Workflow Reports
  - YouTube
- SmartObject Server(s)
- Workflow Management Server(s)

d. **Close** the context browser and click **Next**.

e. Leave the **Configure Item Reference** screen as-is (it will automatically be configured to use the existing Temporary Access Request item reference) and click **Next**.

f. Click **Next** on the **Specify Form Parameters** screen.

g. The **Configure Open Worklist Item Rule** screen should be automatically configured to open the task **When the Form is initializing**, so click **Next**.



Configure Open Worklist Item Rule

Select Rule: When the Form is Initializing

If Current Workflow Activity is DefaultActivity

then open the DefaultActivity worklist item

then on Temporary Access Request Item View View, execute the Initialize method

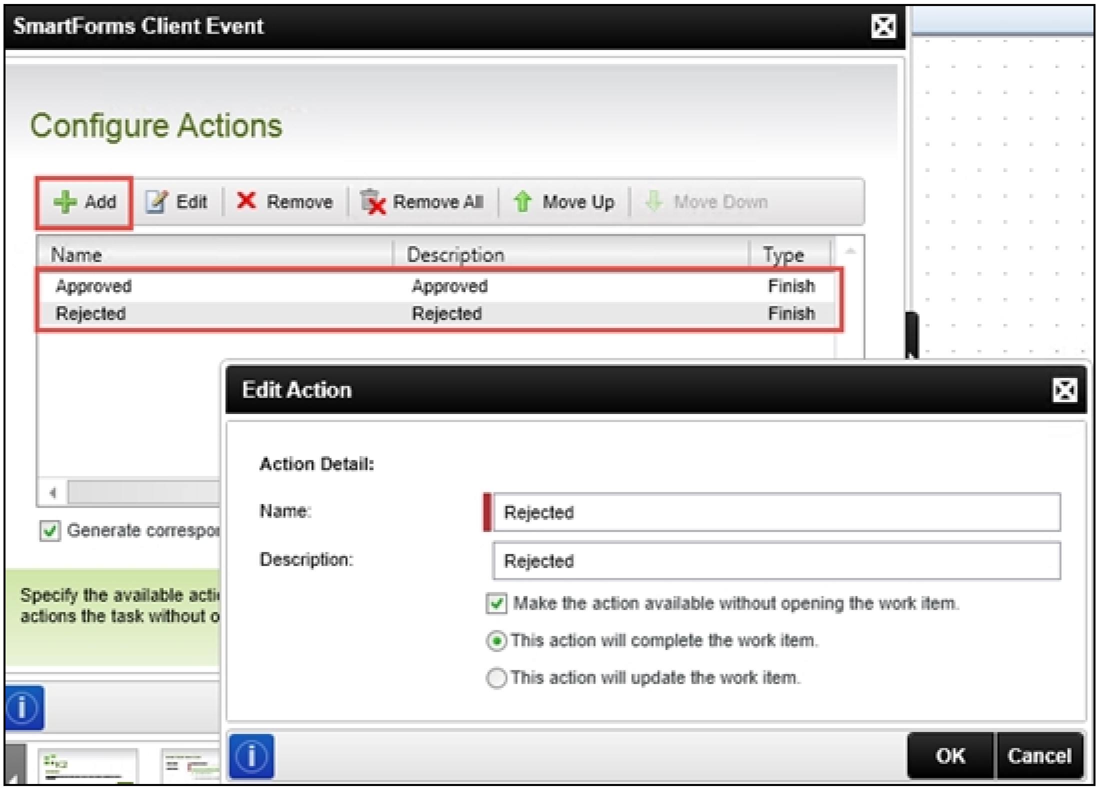
then execute the Disable All controls rule

Configure the rule that is used to load the user's worklist task. The rules associated with the previously selected state are updated in the Form definition once the workflow is deployed and can be further edited in the K2 Designer.

h. The **Transfer Workflow Data to Form** screen should be configured correctly by default, so click **Next**.

i. On the **Configure Actions** screen, use the **Add** button to add the following two actions:
Approved
Rejected

then click **Next.**



Configure Actions

Name	Description	Type
Approved	Approved	Finish
Rejected	Rejected	Finish

☑ Generate correspon

Specify the available acti
actions the task without o

Edit Action

Action Detail:

Name: Rejected

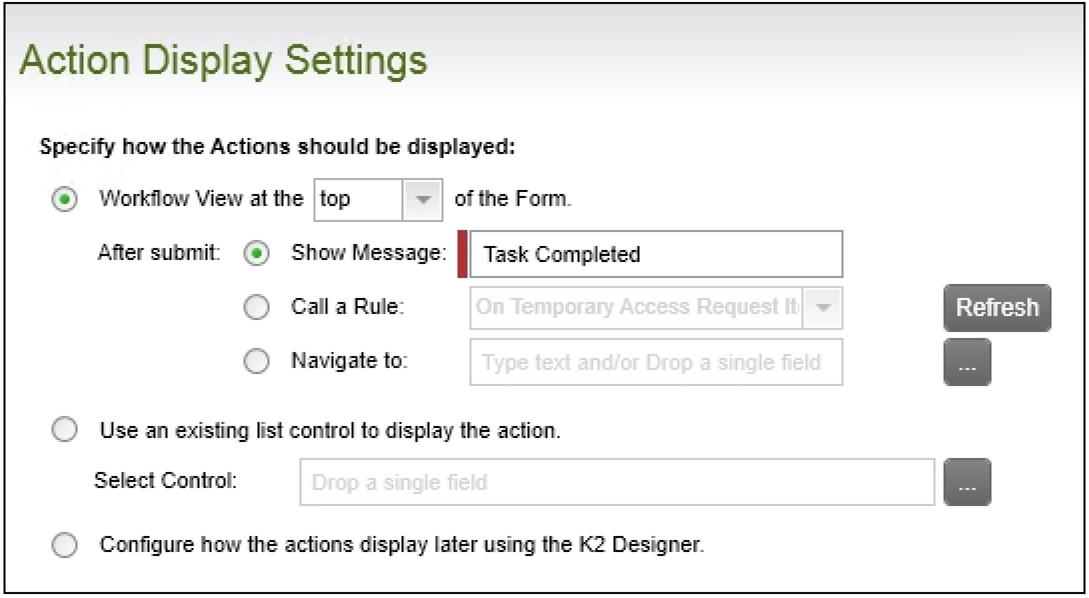
Description: Rejected

☑ Make the action available without opening the work item.

● This action will complete the work item.

○ This action will update the work item.

j. You can leave the **Action Display Settings** screen as-is and click **Next**.



Action Display Settings

Specify how the Actions should be displayed:

● Workflow View at the top of the Form.

After submit: ● Show Message: Task Completed

○ Call a Rule: On Temporary Access Request It

○ Navigate to: Type text and/or Drop a single field

○ Use an existing list control to display the action.

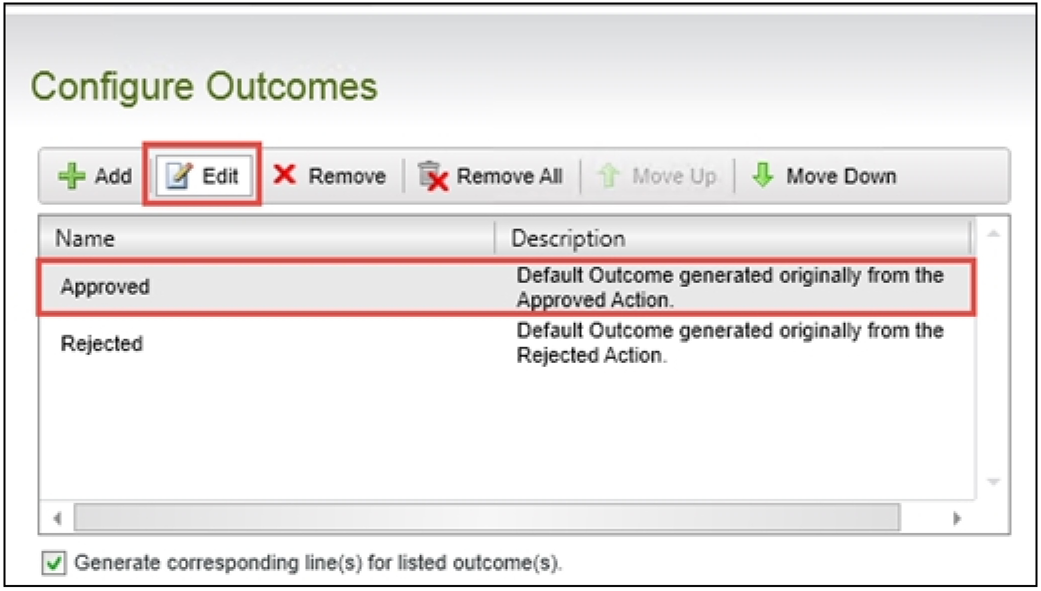
Select Control: Drop a single field

○ Configure how the actions display later using the K2 Designer.

k. You don't need to configure any data transfer, so click **Next** on the **Transfer Form Data to Workflow** screen.

l. On the **Configure Outcomes** screen, you will be adding an outcome and configuring the approved outcomes (remember one outcome is approved and AD account required, and the other is approved and AD account not required) In the next step, you will configure the existing approved outcome so that the AD account is required.

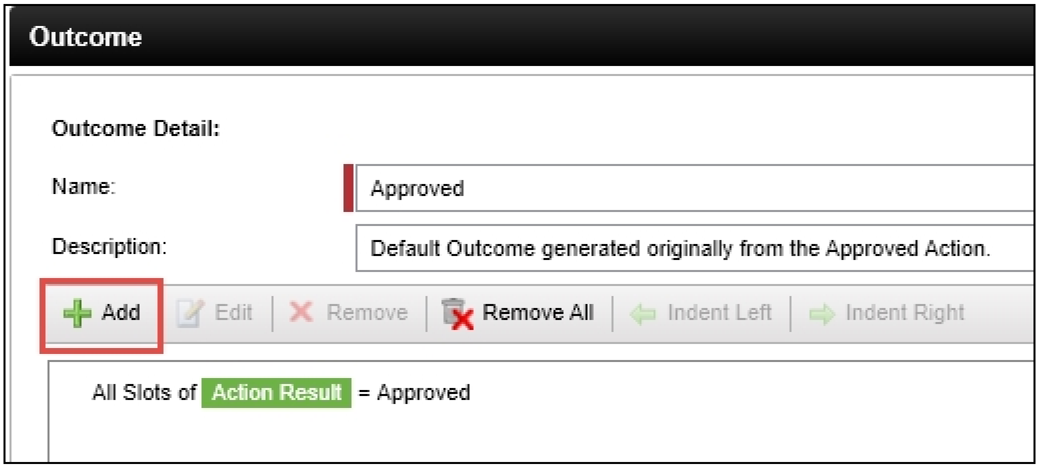
First, select the existing **Approved** outcome and click **Edit**.



Name	Description
Approved	Default Outcome generated originally from the Approved Action.
Rejected	Default Outcome generated originally from the Rejected Action.

☑ Generate corresponding line(s) for listed outcome(s).

m. Click the **Add** button to add another condition to this outcome.



Outcome Detail:

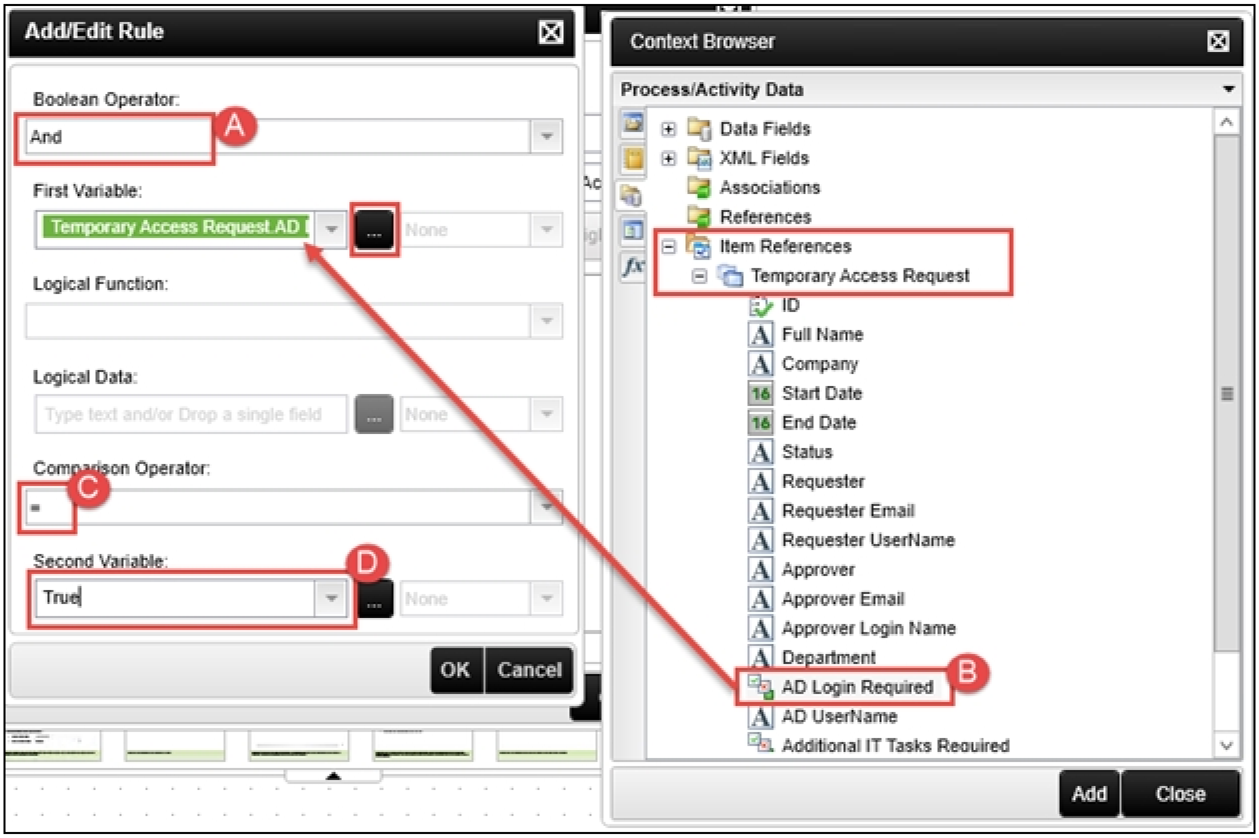
Name: Approved

Description: Default Outcome generated originally from the Approved Action.

All Slots of Action Result = Approved

n. Configure the outcome so that it evaluates as follows:
Boolean Operator: **And**
First Variable: **Temporary Access Request.AD Login Required**
Comparison Operator: **=**
Second Variable: *True*

See the screenshot below for reference.



Add/Edit Rule

Boolean Operator: And

First Variable: Temporary Access Request.AD L… None

Logical Function:

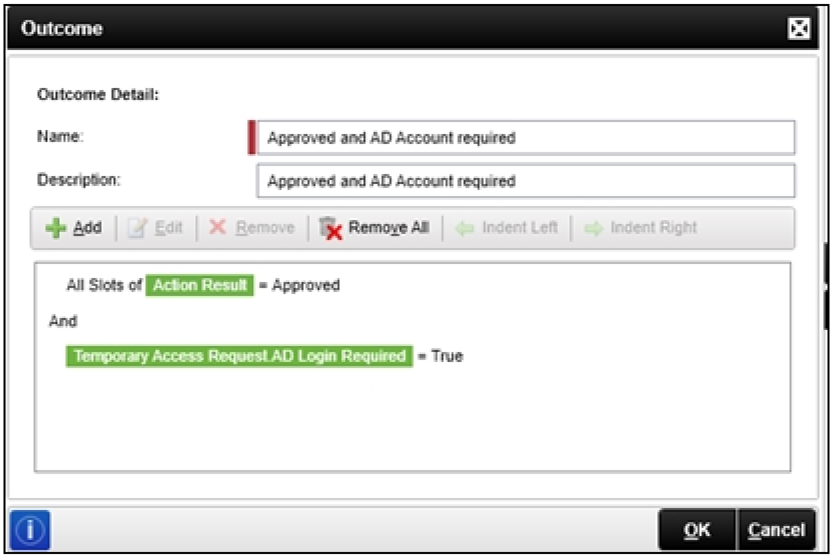
Logical Data: Type text and/or Drop a single field None

Comparison Operator: =

Second Variable: True None

o. Click **OK** to close the outcome configuration screen.

p. Change the edited outcome **Name** to
Approved and AD Account required
and change the **Description** to
Approved and AD Account required
Click **OK** to continue.



Outcome

Outcome Detail:

Name: Approved and AD Account required

Description: Approved and AD Account required

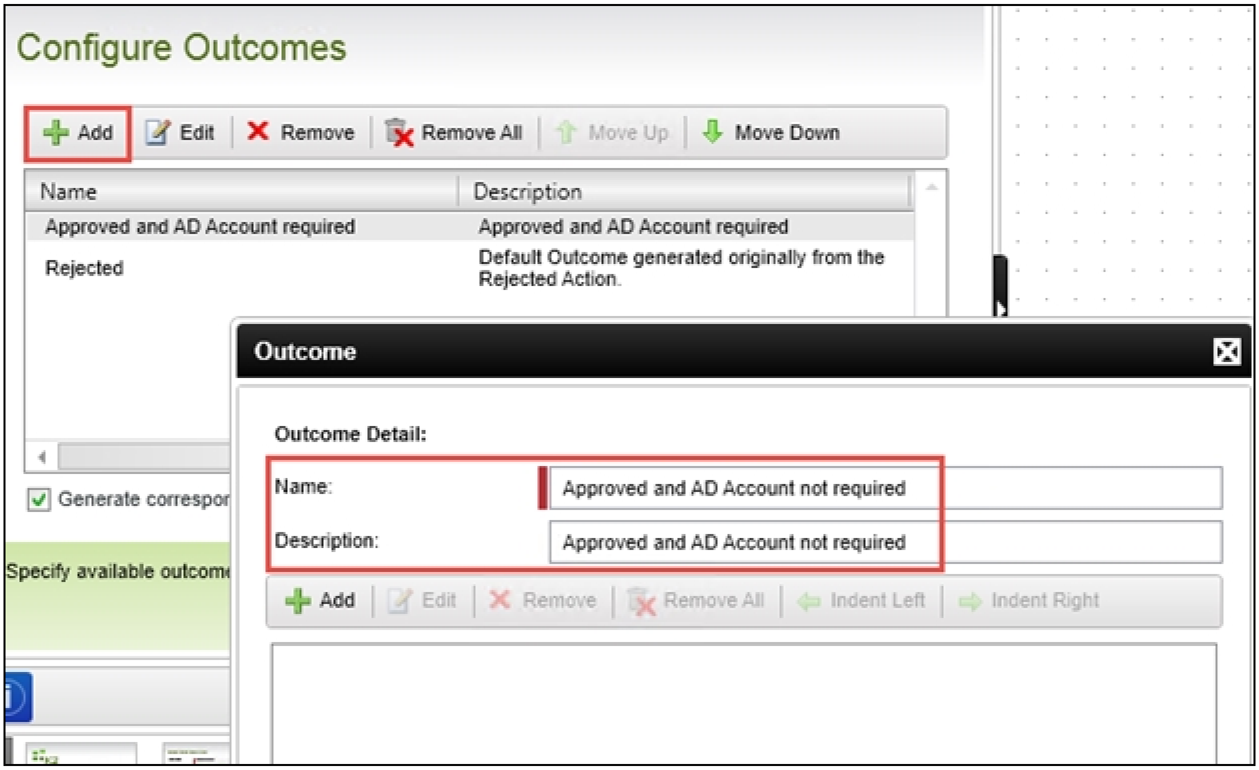
All Slots of Action Result = Approved

And

Temporary Access Request.AD Login Required = True

In this step, you edited the approved outcome to evaluate the form's AD Login Required field. If the login option was selected on the form, then it's value becomes true. In this case, the workflow will follow this outcome path to the next activity or event. If the login option was not selected on the form, the workflow will not follow this outcome path. In the next step, you will add and configure an additional outcome that will provide an outcome path in the event the request is approved, but the AD account option is not selected.

q. Click on the **Add** button. For the **Name** and **Description**, enter
Approved and AD Account not required
then click **OK**.



Configure Outcomes

Name	Description
Approved and AD Account required	Approved and AD Account required
Rejected	Default Outcome generated originally from the Rejected Action.

☑ Generate correspor

Specify available outcome

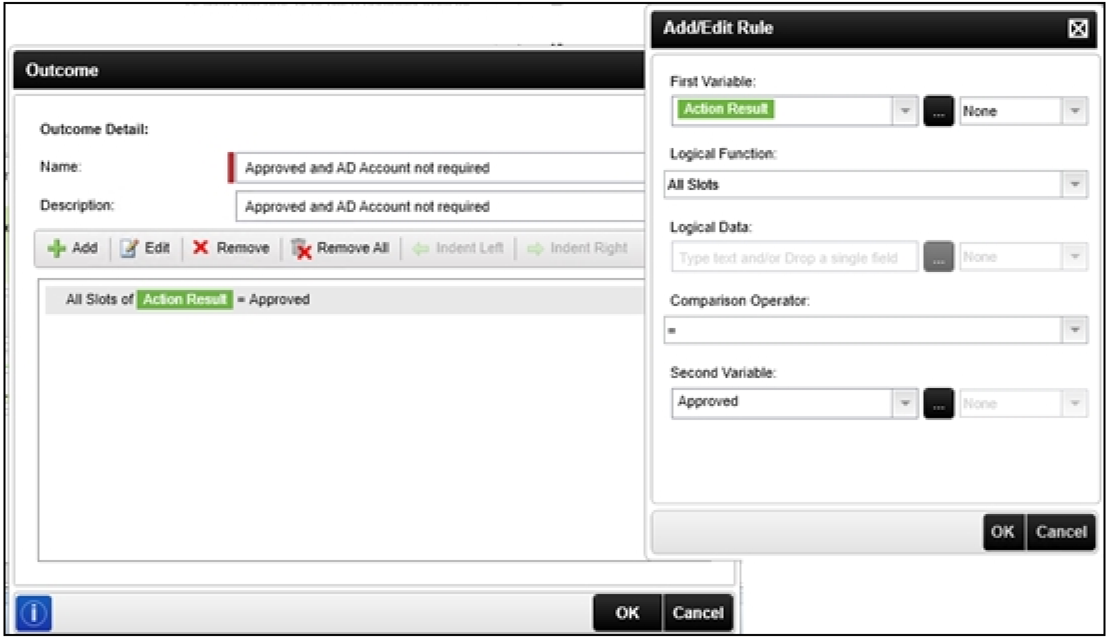
Outcome

Outcome Detail:

Name: Approved and AD Account not required

Description: Approved and AD Account not required

r. Click the **Add** button and add an condition as follows:
First Variable: **Action Result**
Logical Function: **All Slots**
Comparison Operator: **=**
Second Variable: **Approved**
In this step, you are telling K2 to evaluate if the request action is approved.



Outcome

Outcome Detail:

Name: Approved and AD Account not required

Description: Approved and AD Account not required

All Slots of Action Result = Approved

Add/Edit Rule

First Variable: Action Result ... None

Logical Function: All Slots

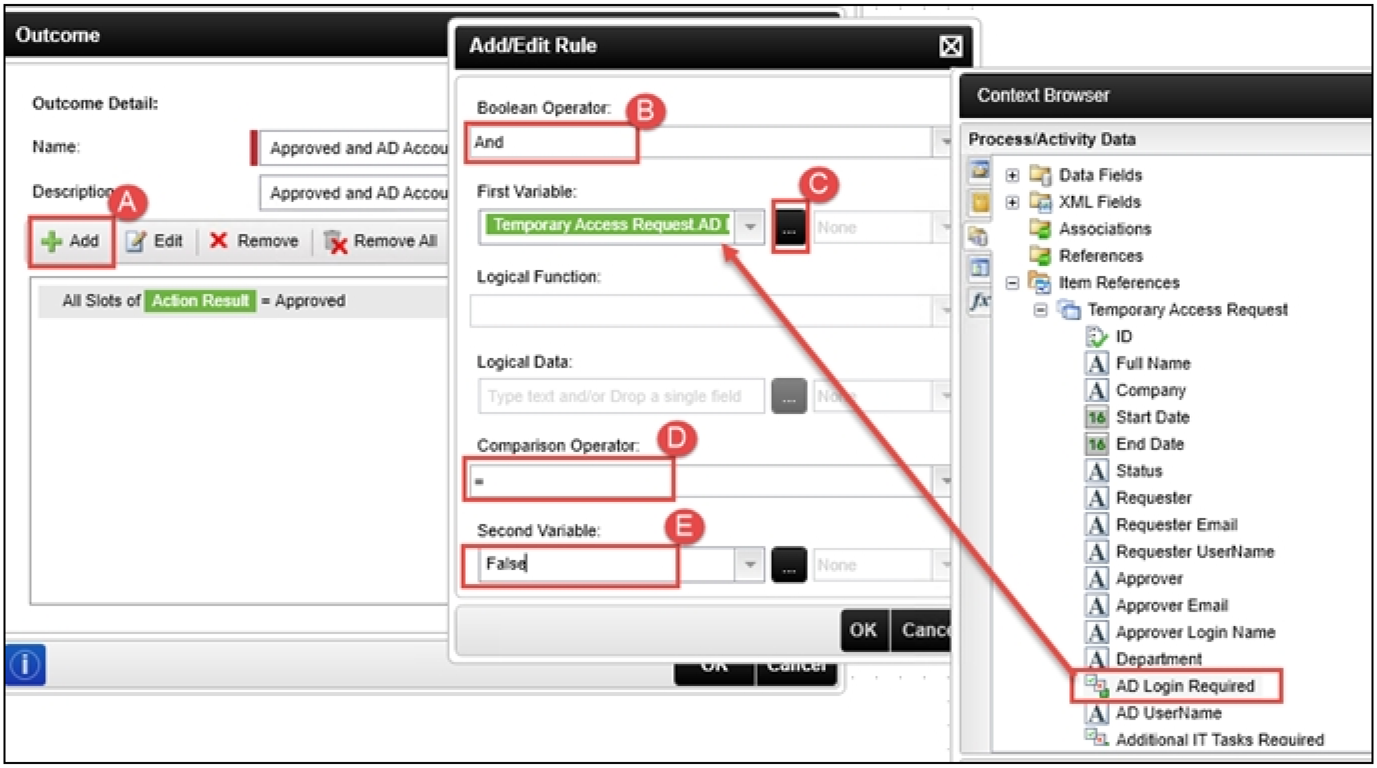
Logical Data: Type text and/or Drop a single field ... None

Comparison Operator: =

Second Variable: Approved ... None

s. **Add** a second condition to the outcome as follows:
Boolean Operator: **And**
First Variable: **Temporary Access Request.AD Login Required**
Comparison Operator: **=**

Second Variable: *False*
In this step, you are adding a condition to evaluate the form's AD Login Required field to see if it returns as false, or not selected. **Close** the context browser, then click **OK**.



Outcome

Outcome Detail:

Name: Approved and AD Accou

Description: Approved and AD Accou

All Slots of Action Result = Approved

Add/Edit Rule

Boolean Operator: And

First Variable: Temporary Access Request.AD L

Logical Function:

Logical Data: Type text and/or Drop a single field

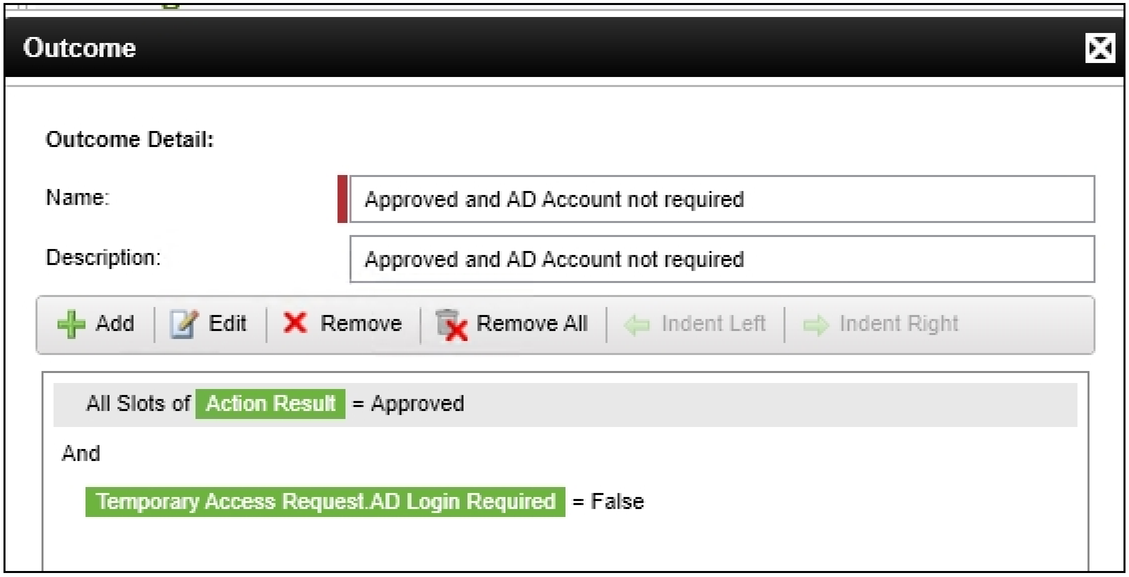
Comparison Operator: =

Second Variable: False

Context Browser

Process/Activity Data

t. The outcome should look like the image below. Click **OK** to continue.



Outcome

Outcome Detail:

Name: Approved and AD Account not required

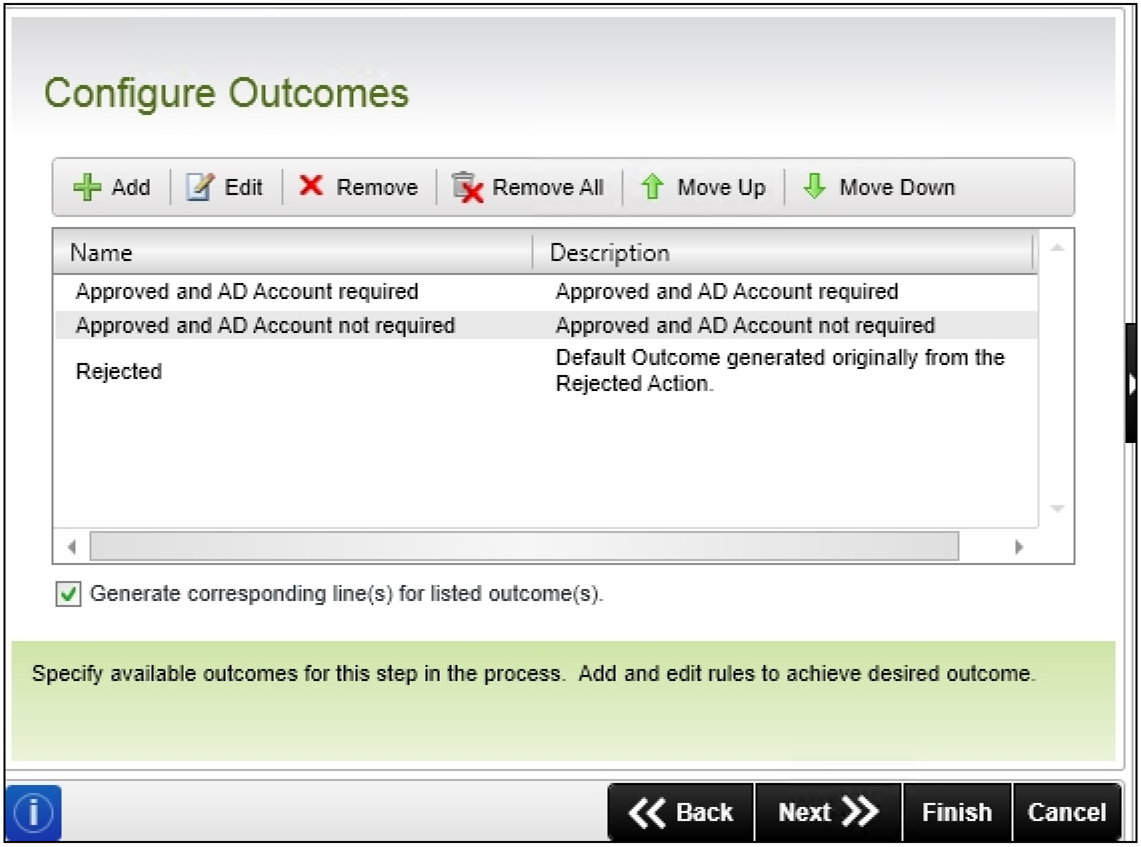
Description: Approved and AD Account not required

All Slots of Action Result = Approved

And

Temporary Access Request.AD Login Required = False

u. Use the **Move Up** button to move the two approved outcomes next to each other. The **Configure Outcomes** screen should look like the image below. Click **Next** to continue.



Configure Outcomes

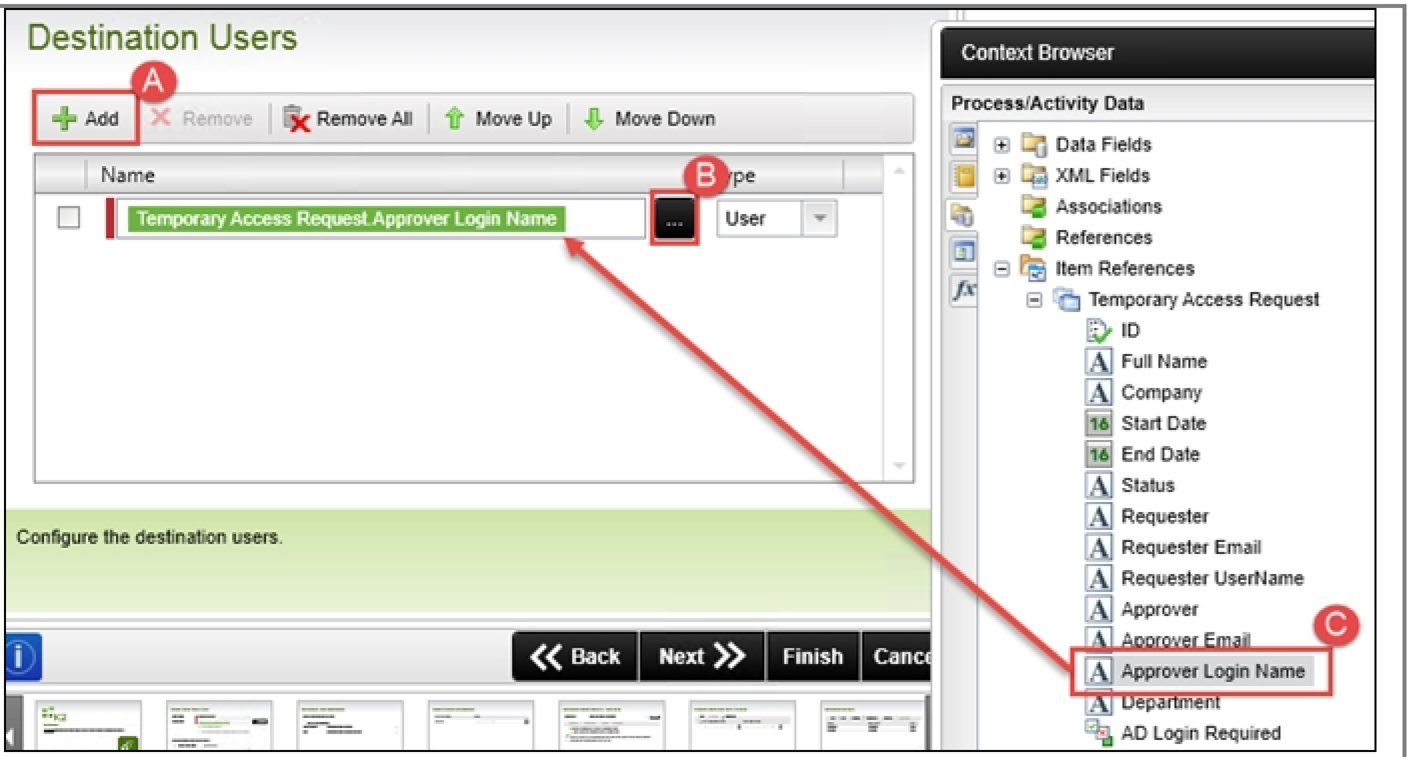
Name	Description
Approved and AD Account required	Approved and AD Account required
Approved and AD Account not required	Approved and AD Account not required
Rejected	Default Outcome generated originally from the Rejected Action.

☑ Generate corresponding line(s) for listed outcome(s).

Specify available outcomes for this step in the process. Add and edit rules to achieve desired outcome.

Next you will add the destination user, or the user that will be approving the request. (This user is also referred to as the task participant.) In this application, the approver name is being saved along with the request data, so you can read the approver login name from the Temporary Access Request SmartObject Item Reference.

v. On the **Destination Users** screen, click the **Add** button and then click the ellipsis to select the **Approver Login Name** field from the **Temporary Access Request** Item Reference, as shown below. **Close** the context browser and click **Next** to continue.

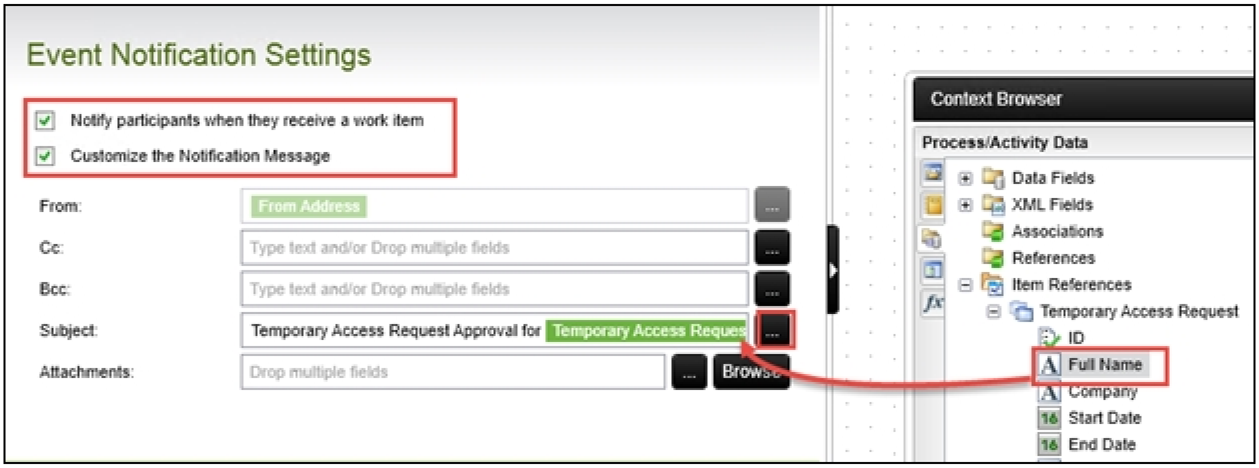


Destination Users

	Name	Type
□	Temporary Access Request.Approver Login Name	User

Configure the destination users.

w. On the **Event Notification Settings** screen, select the option to **Notify participants when they receive a work item**, then customize the notification email. Modify the subject line to better describe the task and add the **Full Name** from the Item Reference. **Close** the context browser and click **Next**.



Event Notification Settings

☑ Notify participants when they receive a work item

☑ Customize the Notification Message

From: From Address

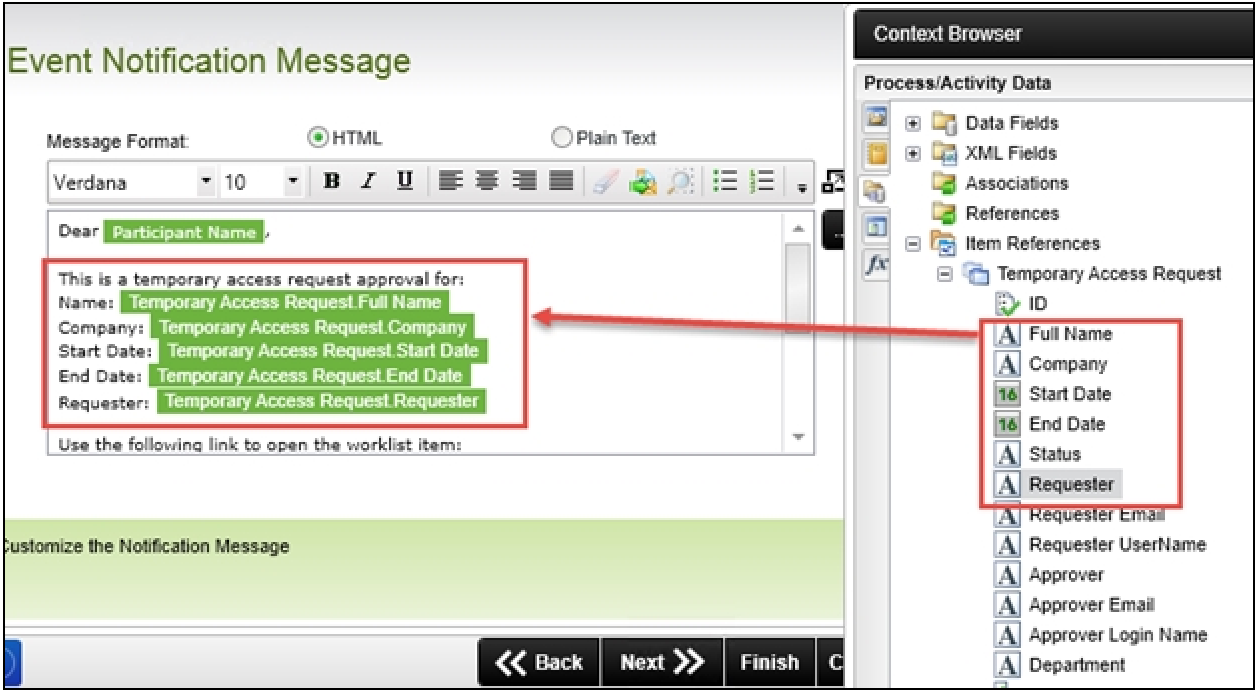
Cc: Type text and/or Drop multiple fields

Bcc: Type text and/or Drop multiple fields

Subject: Temporary Access Request Approval for Temporary Access Reques

Attachments: Drop multiple fields

x. For the **Event Notification Message** body, add some information from the Item Reference so that the approver can approve or reject the request using email if they want to. The sample below includes the **Full Name**, **Company**, **Start Date**, **End Date** and **Requester**. You can add whatever information you like from the Item Reference.



Event Notification Message

Message Format: ● HTML ○ Plain Text

Dear Participant Name,

This is a temporary access request approval for:
Name: Temporary Access Request.Full Name
Company: Temporary Access Request.Company
Start Date: Temporary Access Request.Start Date
End Date: Temporary Access Request.End Date
Requester: Temporary Access Request.Requester

Use the following link to open the worklist item:

y. **Close** the context browser, then click **Finish** to complete the client event wizard.

z. **Save** your workflow.

Step 4 Review

In this step you added a client event to the workflow for the manager approval task. You configured the event to use the Temporary Access Request Form and configured the outcomes to follow a different path depending on whether the workflow should create an AD account or not. This step demonstrated how you can configure more advanced outcomes that evaluate data as well as user input. You used a variable (the Approver Login Name) rather than a static user name as the user for the approval task. Finally, you customized the task notification email that is sent to the approver.

Once again, item references are shortcuts to the current SmartObject record that is created as part of the process instance. Using item references in your workflow creates dynamic content for your workflow properties, as well as user content.

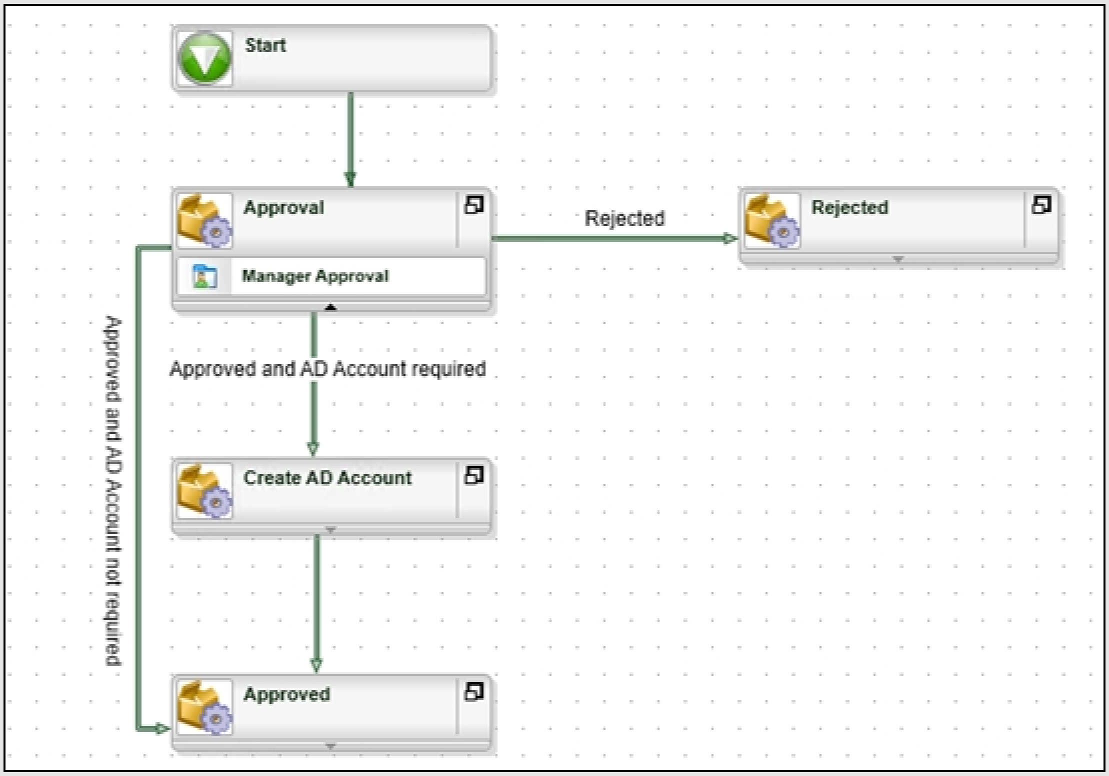
Step 5: Format the workflow and add blank activities

In this step you will format your workflow incorporating the outcomes you created in Step 4. You will add blank activities, that will eventually contain events for each of the outcome paths. This step is intended to further familiarize you with the K2 Studio design canvas.

Step 5 Tasks

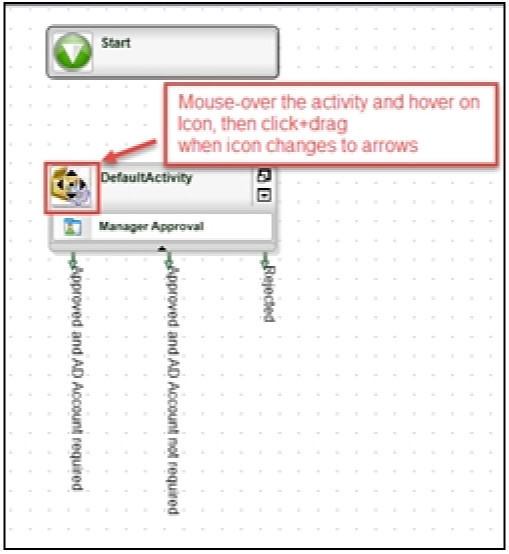
1. Rename the DefaultActivity to
   Approval
   (you can accept the warning message).
2. Add three new blank activities to the design canvas:
   Rejected
   Create AD Account
   Approved
   then either draw or move lines and labels around so that the workflow layout looks like the image

below.



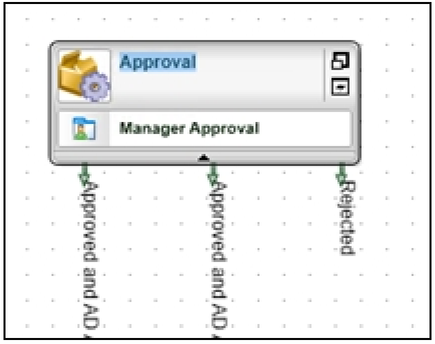
Step 5 Walkthrough

a. Move the new activity in the workflow so it lies directly underneath the **Start** step of the workflow. To do this, move your mouse over the box icon for the activity, and then click+drag when your mouse pointer changes to four arrows.

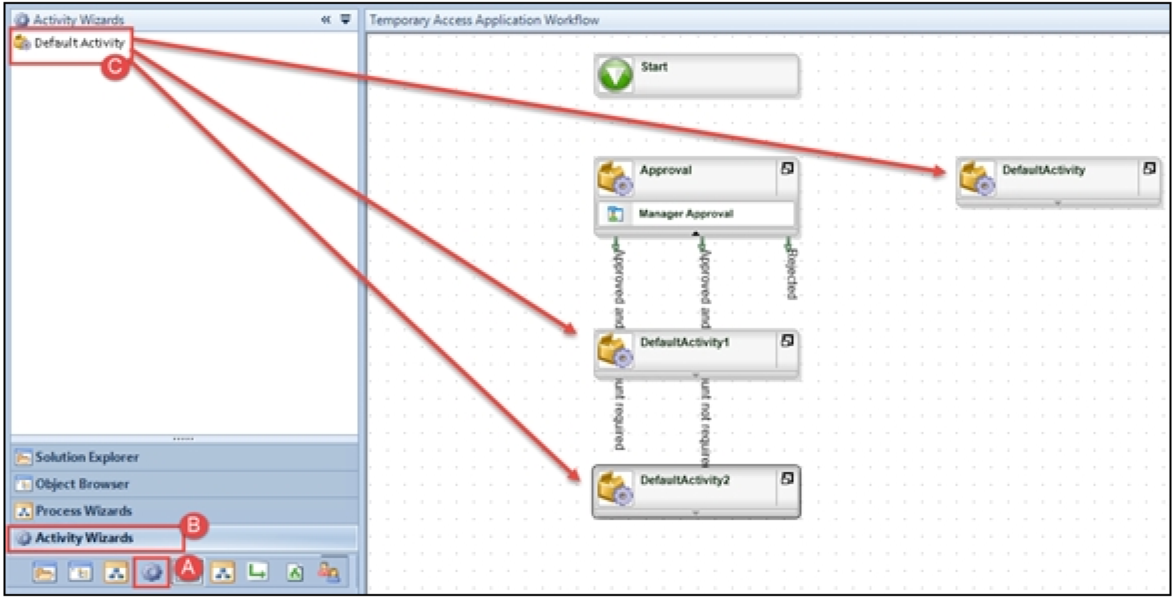

Start
Mouse-over the activity and hover on Icon, then click+drag when icon changes to arrows
DefaultActivity
Manager Approval
Approved and AD Account required
Approved and AD Account not required
Rejected

b. Rename the new activity by double-clicking the activity name and typing the new name
Approval
then click anywhere on the design canvas to set the new name. Click **OK** if you see a warning mes-

sage. You can safely ignore this warning.

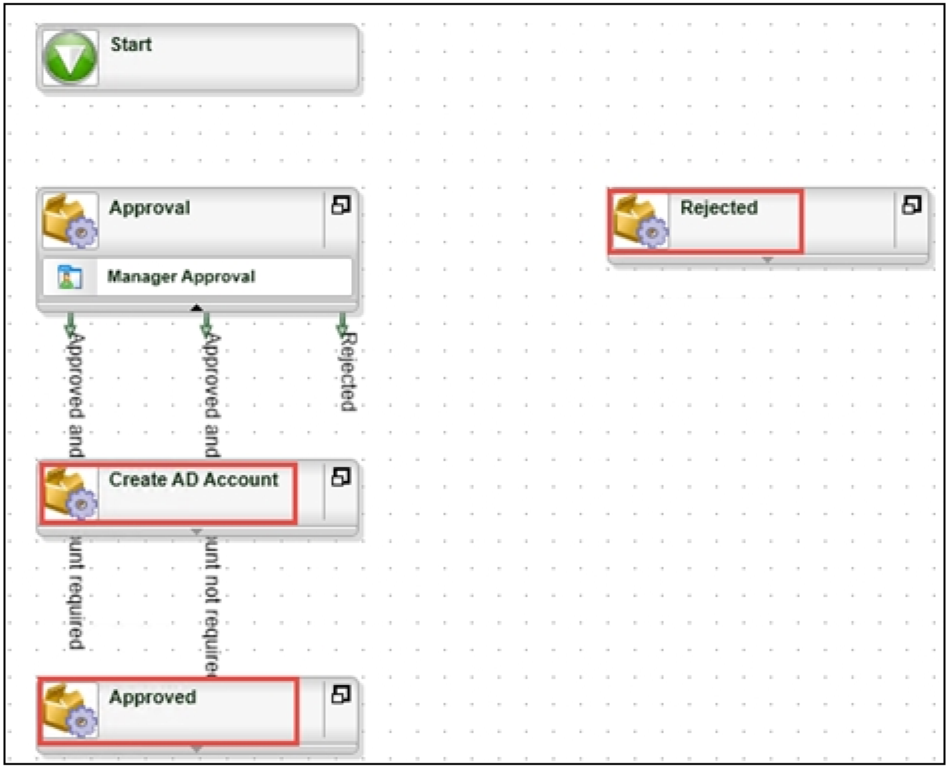


c. Open the **Activity Wizards** pane, then drag and drop three blank **Default Activities** onto the design canvas. You will configure the activities for each of the outcome lines.



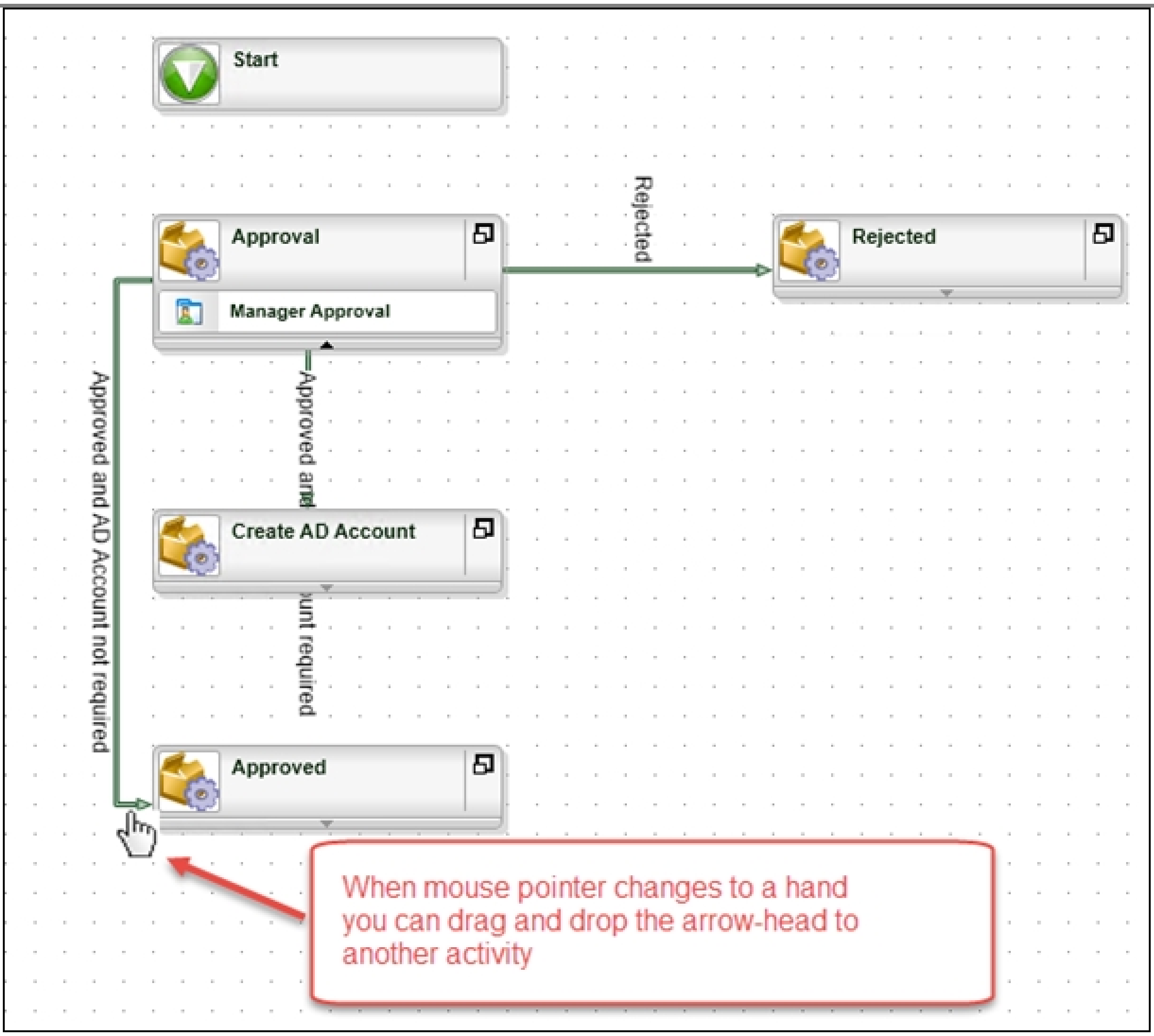
d. Double-click each of the activity names and rename them as follows:
Rejected
Create AD Account

Approved



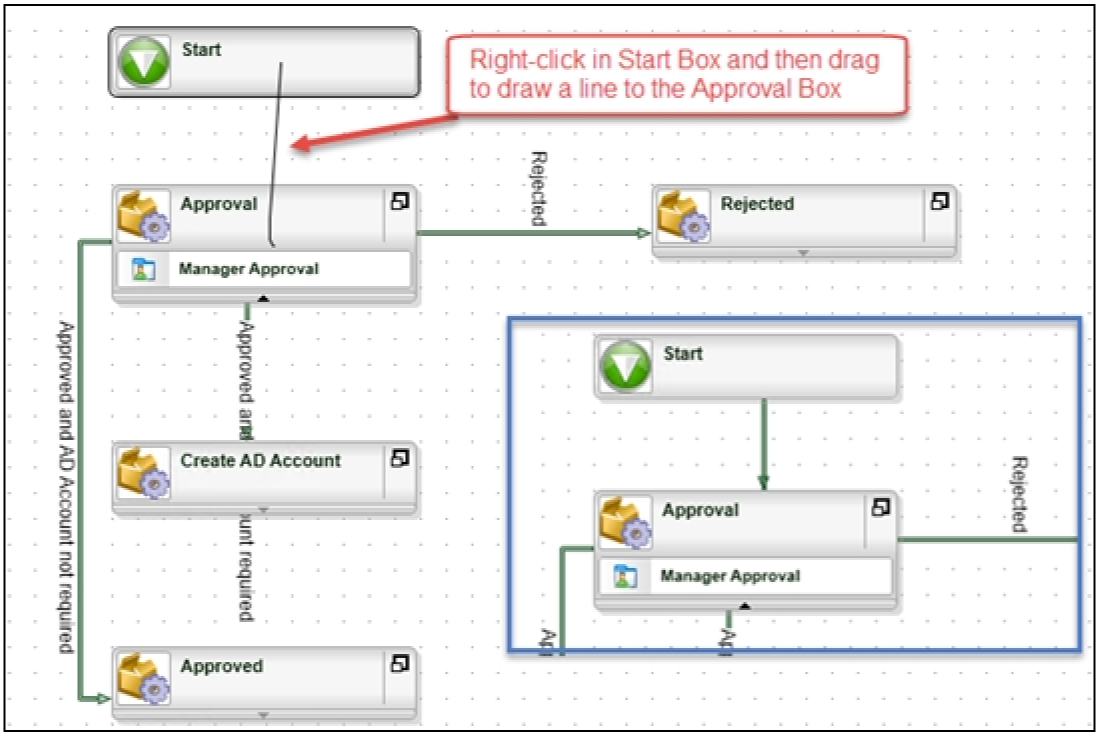
e. Using your mouse, grab the arrow-head for each line that comes out the **Approval** activity, then attach the lines to the new activities. Use the image below for reference.

Tip
When the mouse pointer changes to a hand, you can grab that end of the line and move it around to another activity. You can also move the start of a line around, but you cannot detach the start of a line from an activity. Lines flow out of activities and are bound to the source activity.

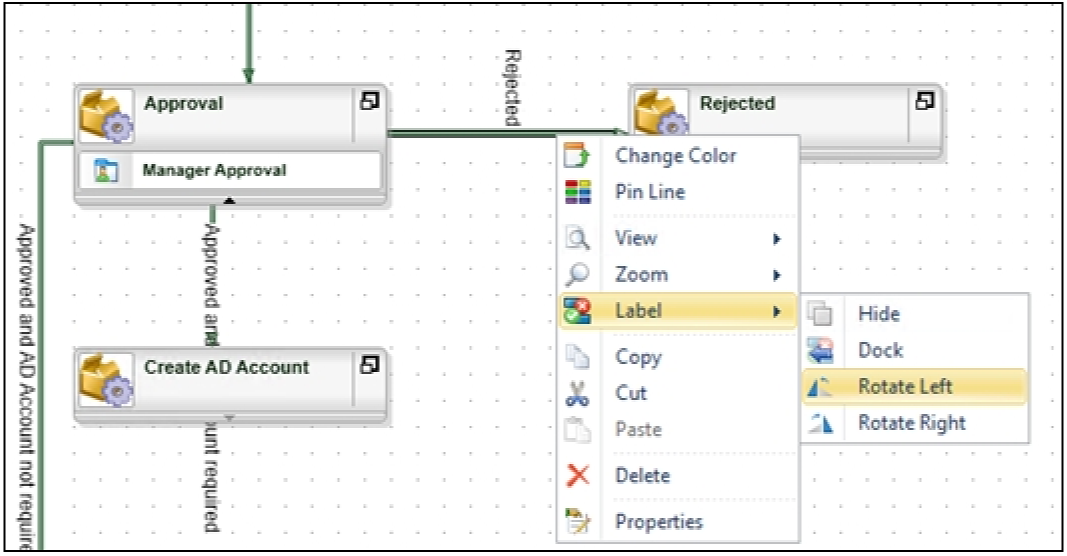


When mouse pointer changes to a hand you can drag and drop the arrow-head to another activity

f. Draw a line between the **Start** step and the **Approval** step by right-clicking inside of the **Start** activity, then dragging a new line into the **Approval** activity. K2 will draw an arrow-line between the two activities.

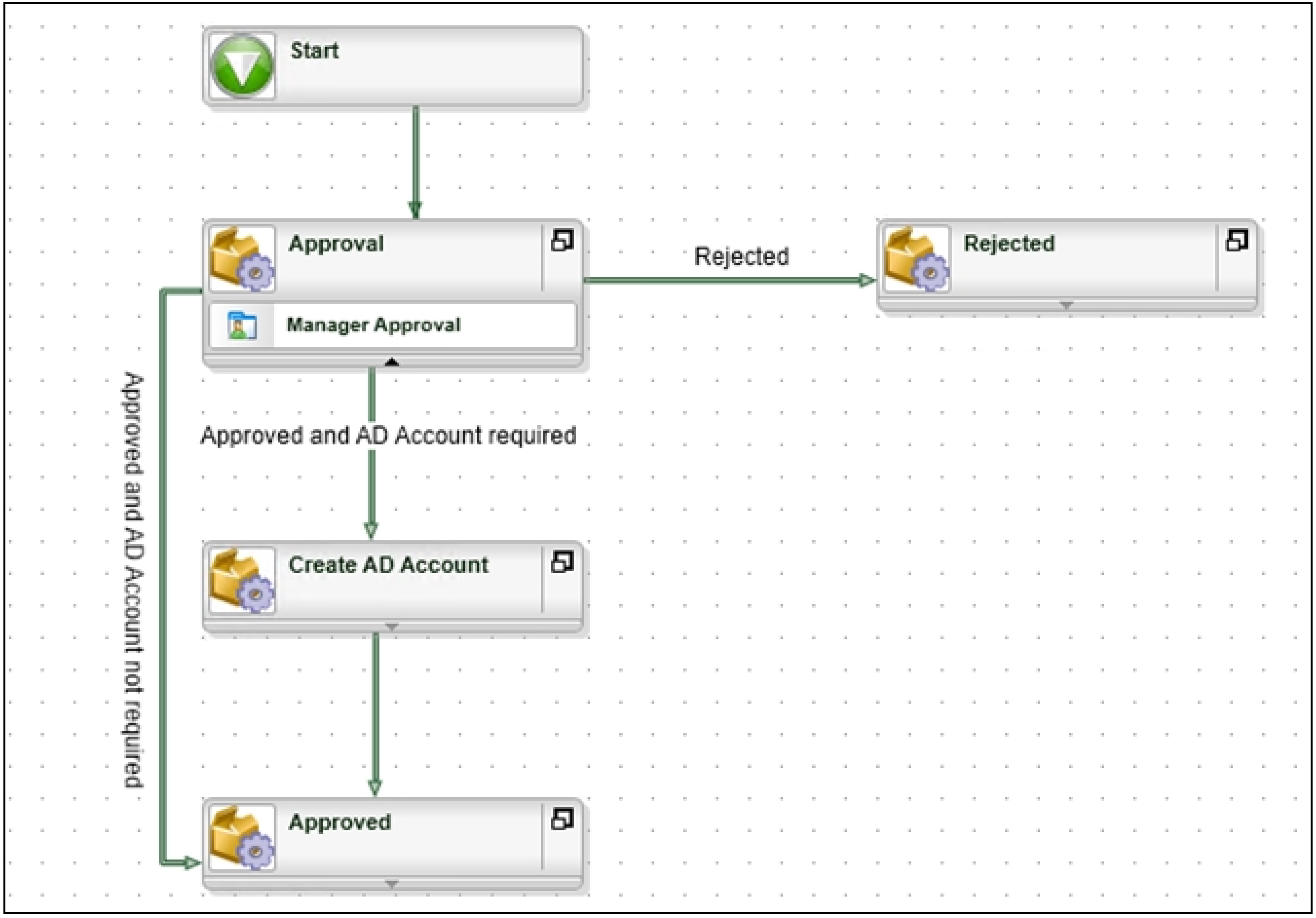


g. Right-click on the **Rejected** line and select **Label** > **Rotate Left** to rotate the label name. You can then mouse-over the line label and click+drag it around to align it with the outcome line.



h. Repeat the **Label** > **Rotate Left** for the **Approved and AD Account Required** line and move the label if required.

i. Draw a line between the **Create AD Account** activity and the **Approved** activity. Your workflow should look like the image below.



j. **Save** your workflow.

Step 5 Review

In this step you formatted your workflow layout, added new activities and drew lines to connect the activities. Depending on the form and approval options selected, the workflow will following the appropriate outcome path.

Step 6: Add system tasks to each new activity

In this step you will be adding system tasks to each blank activity you added in the previous step. System tasks are tasks that are performed by the K2 server and do not required any human interaction. The system tasks include updating the request status property, sending an email and creating (if required) the AD user account.

Step 6 Tasks

1. Add a SmartObject event to the **Rejected** activity to set the **Status** property of the **Temporary Access Request SmartObject** to
Rejected
2. Add a **Send E-mail** event to the **Rejected** activity to notify the originator that their request was rejected. Use some values from the Item Reference to add contextual information to the email such as the **Full Name, Company, Start Date** and **End Date**, so that the originator knows which request was rejected.
3. Add an **Active Directory** event to create a new account for the user in the **Create AD Account** activity, with the following properties:

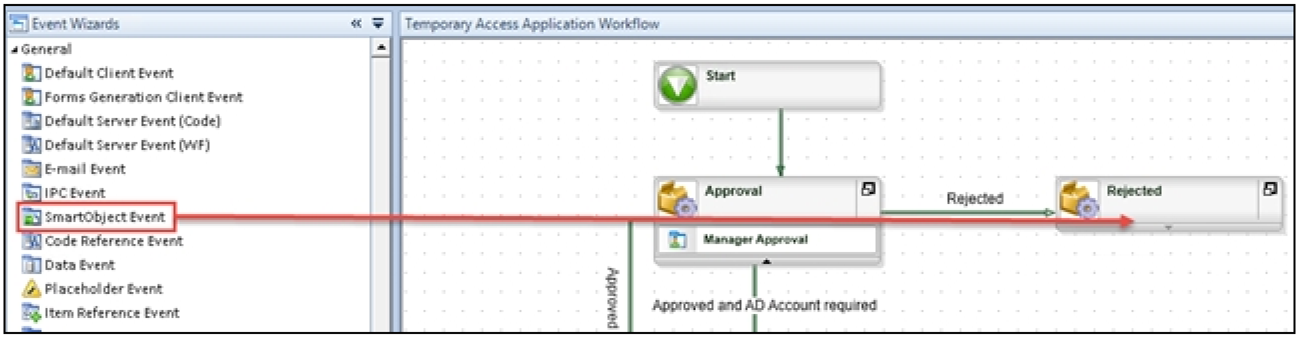
Field	Value

Full Name	Use the **Trim** inline function and select the **Full Name** property from the **Temporary Access Request** Item Reference, then add *(Temp)* to the end of the Full Name property.
Logon Name	Use the **Trim** inline function, and select the **AD UserName** property from the **Temporary Access Request** Item Reference
Password	*K2pass!*
Do not create an item reference, and do not create a mailbox or add the user to groups.	

4. Copy and paste the **Set Status Rejected** event into the **Approved** activity, and re-configure it to set the **Status** property of the Temporary Access Request SmartObject to
Approved
then rename the copied event to
Set Status Approved

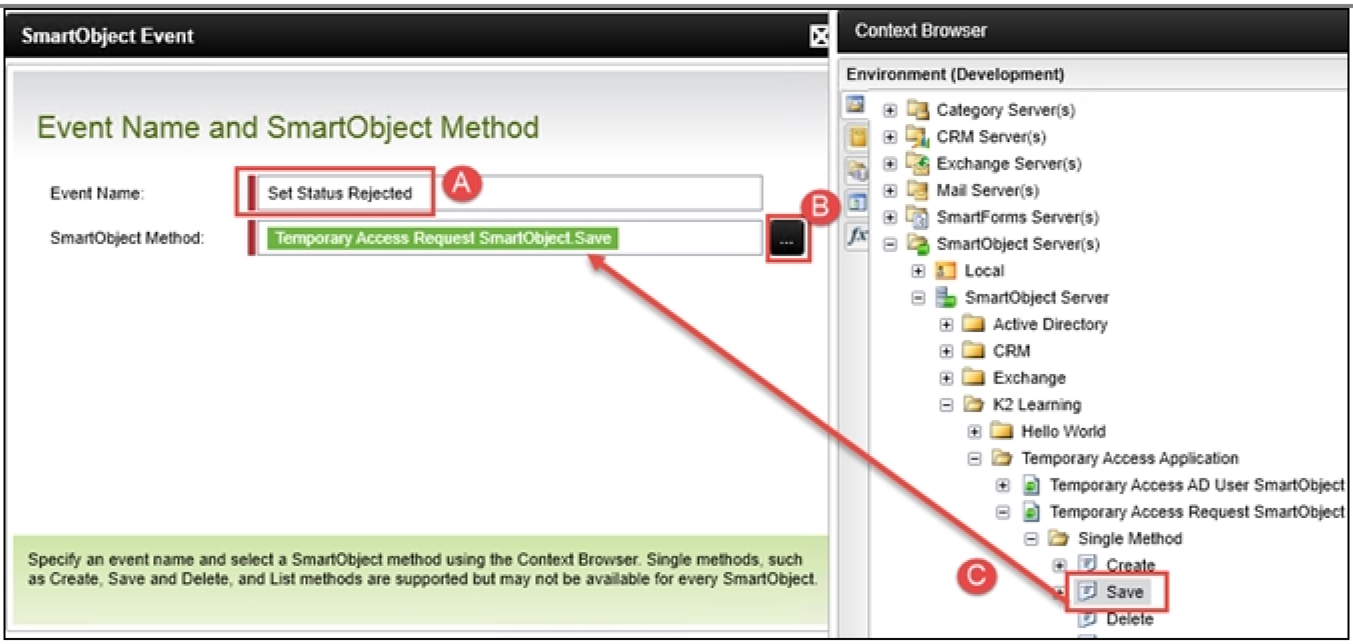
Step 6 Walkthrough

a. Switch to the **Event Wizards** pane, then drag and drop a **SmartObject Event** into the **Rejected** activity. When you drop the event into the activity, the wizard will automatically start. Click **Next** on the welcome screen.



b. On the **Event Name and SmartObject Method** screen, name the event
Set Status Rejected
and then use the ellipsis to locate the following method:
Environment > SmartObject Server(s) > SmartObject Server > K2 Learning > Temporary Access Application > Temporary Access Request SmartObject > Single Method > Save.
Drag the method into the **SmartObject Method** text box. **Close** the context browser. Click **Next** to continue.

Note
In K2 terms, save is the equivalent of update. In this step, you are updating the Temporary Access Request SmartObject > Status property so that it reflects the rejected decision.



SmartObject Event

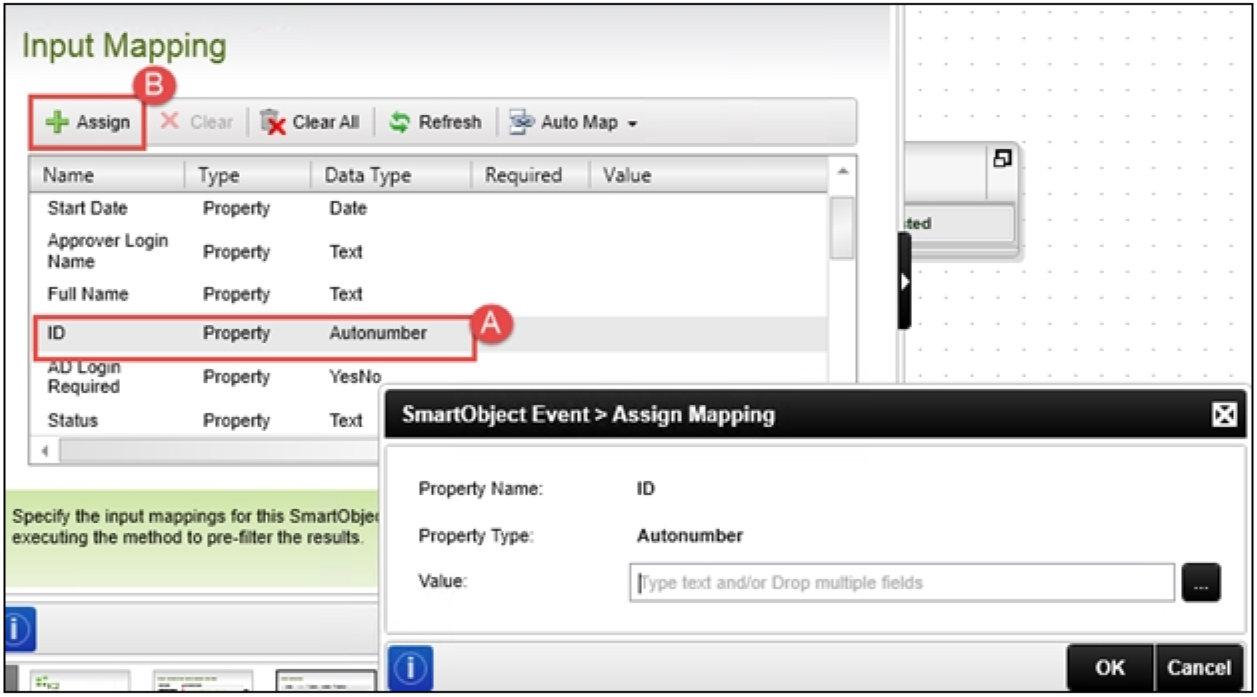
Event Name and SmartObject Method

Event Name: Set Status Rejected

SmartObject Method: Temporary Access Request SmartObject.Save

Specify an event name and select a SmartObject method using the Context Browser. Single methods, such as Create, Save and Delete, and List methods are supported but may not be available for every SmartObject.

c. On the **Input Mapping** screen, locate the **ID** property, then click the **Assign** button. You have to tell K2 *which* record you want to update. Remember that the ID of the access request record is saved in the **RecordID** data field - you will be mapping this data field to the ID input property.



Input Mapping

Name	Type	Data Type	Required	Value
Start Date	Property	Date		
Approver Login Name	Property	Text		
Full Name	Property	Text		
ID	Property	Autonumber		
AD Login Required	Property	YesNo		
Status	Property	Text		

Specify the input mappings for this SmartObjec executing the method to pre-filter the results.

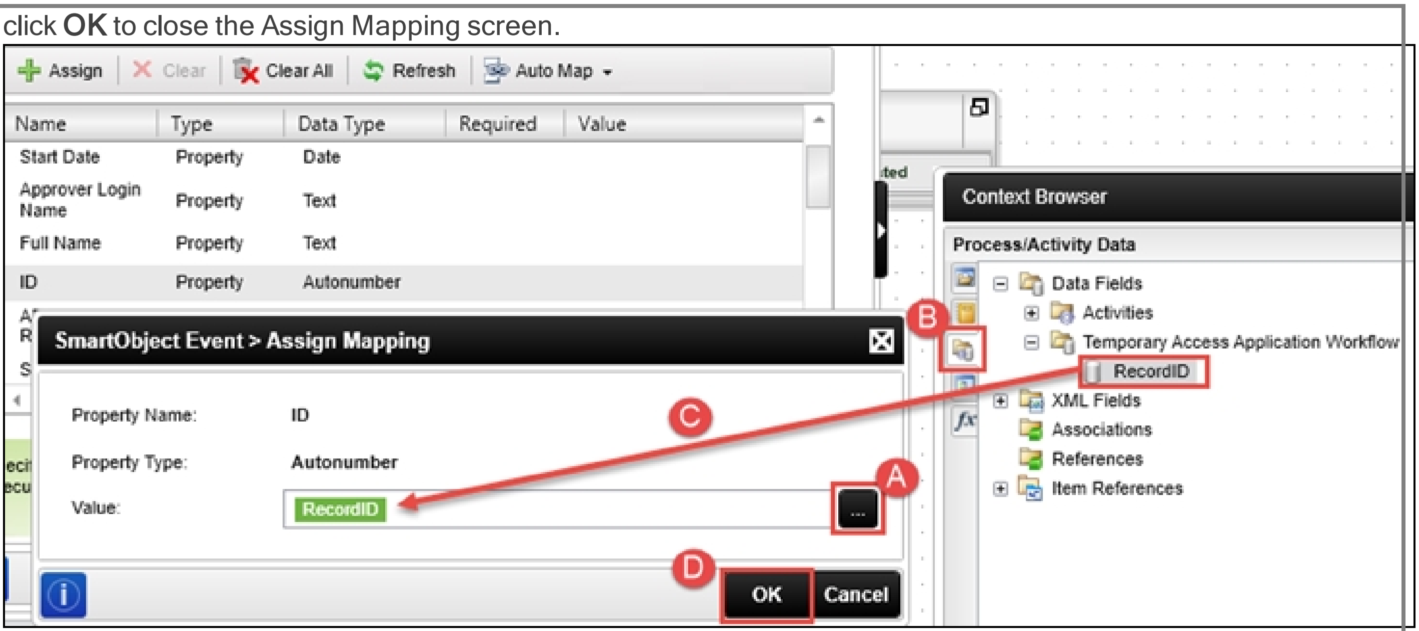
SmartObject Event > Assign Mapping

Property Name: ID

Property Type: Autonumber

Value: Type text and/or Drop multiple fields

d. On the **Assign Mapping** screen, click the ellipsis icon and then locate and drag and drop the **RecordID** data field that exists in the **Process/Activity Data** pane. **Close** the context browser, then click **OK** to close the Assign Mapping screen.



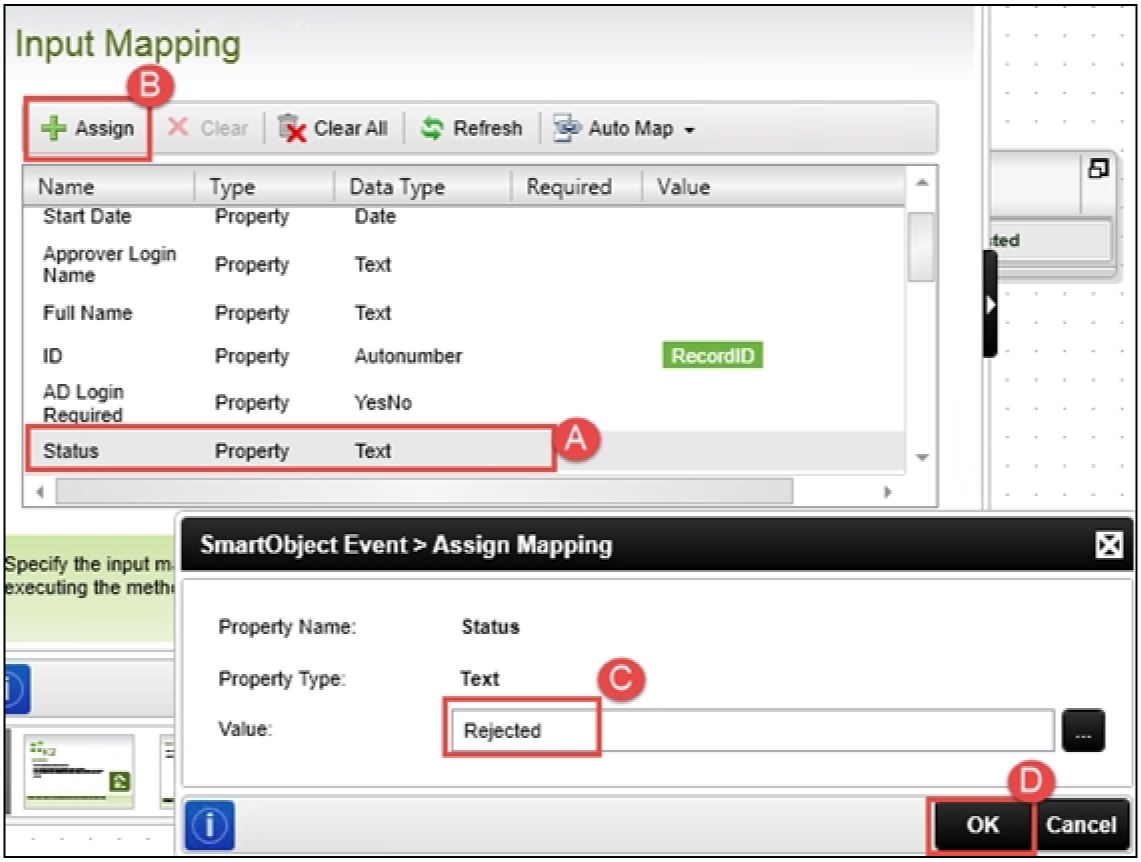
Name	Type	Data Type	Required	Value
Start Date	Property	Date		
Approver Login Name	Property	Text		
Full Name	Property	Text		
ID	Property	Autonumber		

Property Name: ID

Property Type: Autonumber

Value: RecordID

e. On the **Input Mapping** screen, locate the **Status** property and click **Assign**. Enter *Rejected* as the **Status** for this request, then click **OK**. Click **Next**, **Next** and then **Finish** to complete this wizard.



Name	Type	Data Type	Required	Value
Start Date	Property	Date		
Approver Login Name	Property	Text		
Full Name	Property	Text		
ID	Property	Autonumber		RecordID
AD Login Required	Property	YesNo		
Status	Property	Text		

Property Name: Status

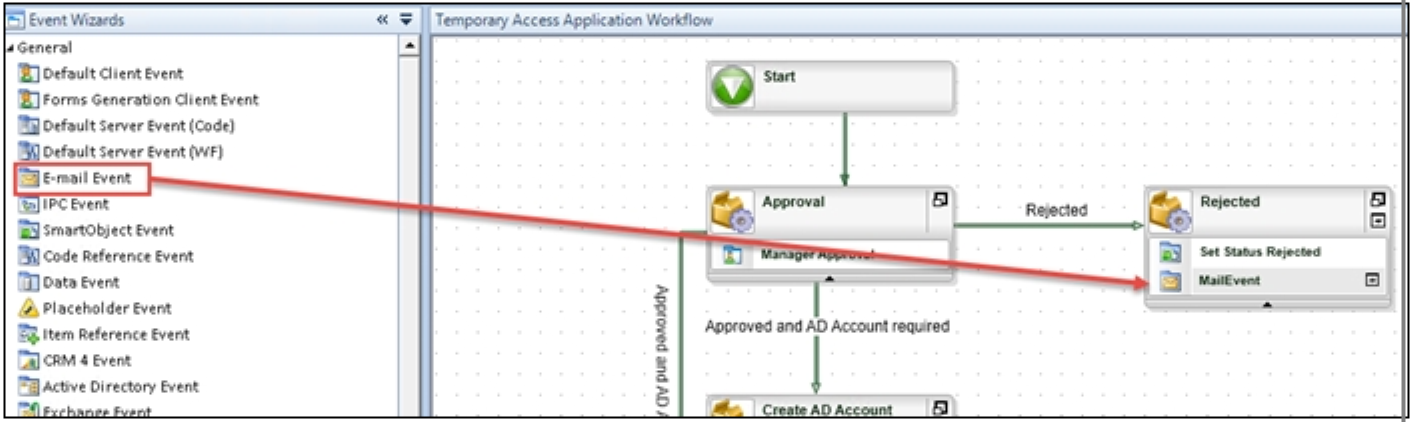
Property Type: Text

Value: Rejected

In this step, you assigned the save method of the Temporary Access Request SmartObject to update the status property. You mapped the ID of the SmartObject record to the RecordID data field so that K2 knows to update the current record. Recall that the RecordID data field value is set by the form when the workflow is started. You manually entered a value for the status property to show that the workflow request was rejected. This step was an example of how you can use the SmartObject Event in a workflow to call a SmartObject method as a system task.

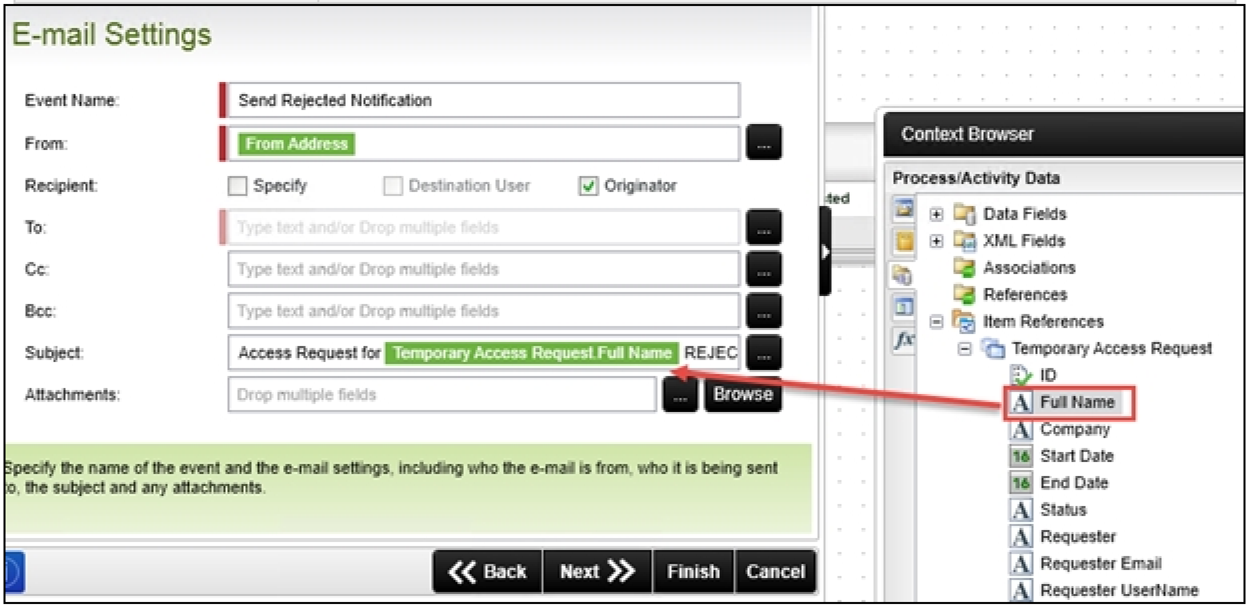
Now you will add an email event and configure it to let the originator know their request was rejected.

f. From the **Event Wizard** pane, drag and drop an **E-mail event** into the **Rejected** activity so that it is below the **Set Status Rejected** event. The E-mail Event wizard will start as soon as you drop the event. Click **Next** on the wizard home screen to continue.



g. Configure the E-mail settings using the table below as a guide. Use the **Item References** > **Full Name** property found in the context browser in place of the [Full Name] placeholder. **Close** the context browser and click **Next**.

Field	Value
Event Name	*Send Rejected Notification*
From	Leave default value unchanged, it should say **From Address**
Recipient	UNCHECK **Specify** and CHECK **Originator**
Subject	*Access Request for [Full Name] REJECTED*



E-mail Settings

Event Name: Send Rejected Notification

From: From Address

Recipient: □ Specify □ Destination User ☑ Originator

To: Type text and/or Drop multiple fields

Cc: Type text and/or Drop multiple fields

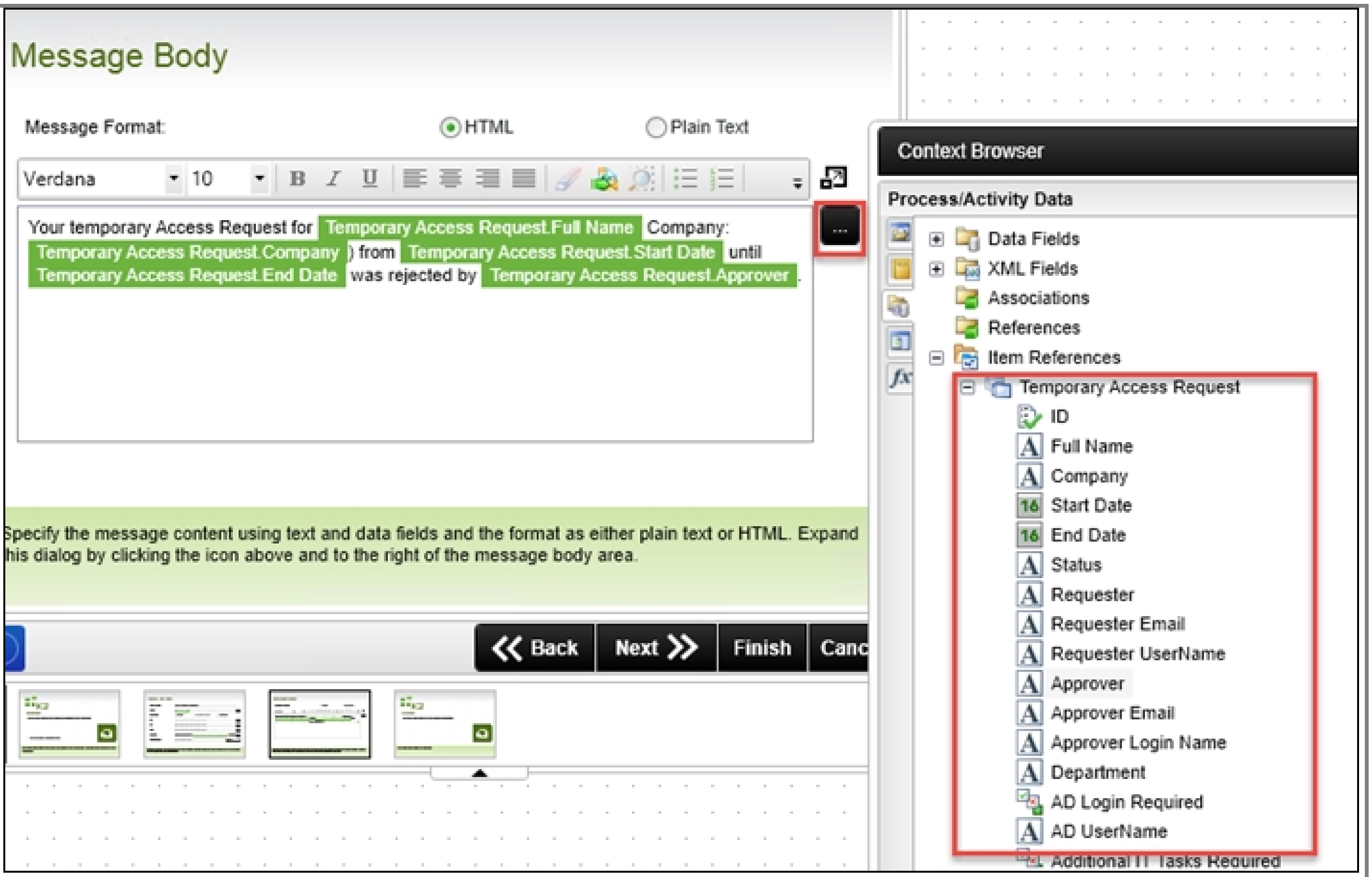
Bcc: Type text and/or Drop multiple fields

Subject: Access Request for Temporary Access Request.Full Name REJEC

Attachments: Drop multiple fields

Specify the name of the event and the e-mail settings, including who the e-mail is from, who it is being sent to, the subject and any attachments.

h. For the message body, use the context browser and Item Reference properties to add some contextual information about the request. Use the sample below as a guide, but feel free to use whatever text you like. **Close** the context browser after you have the message body set, then click **Next** and **Finish** to complete the wizard.



Message Body

Message Format: ● HTML ○ Plain Text

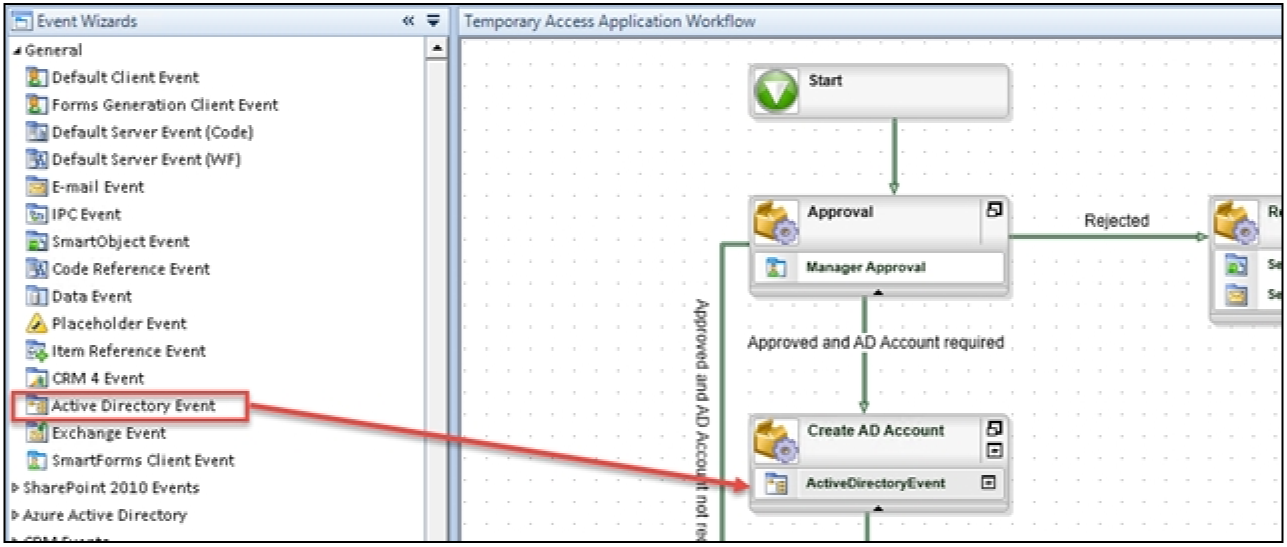
Your temporary Access Request for Temporary Access Request.Full Name Company: Temporary Access Request.Company) from Temporary Access Request.Start Date until Temporary Access Request.End Date was rejected by Temporary Access Request.Approver .

Specify the message content using text and data fields and the format as either plain text or HTML. Expand his dialog by clicking the icon above and to the right of the message body area.

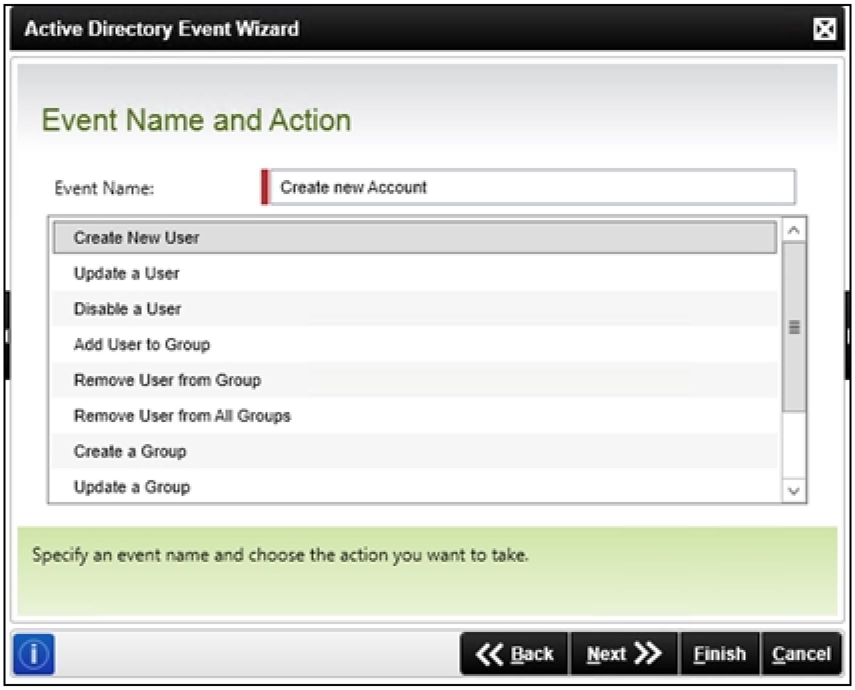
This concludes the events you want to execute if the request was rejected. Notice that when building workflows with K2 Studio, you can insert multiple events into the same activity. When the workflow runs, K2 will execute all of the events in an activity in sequence, assuming that activity was reached in the path of the workflow.

Next, you will add an event to create an Active Directory account if one is required.

i. From the **Event Wizards** pane, drag and drop an **Active Directory Event** into the **Create AD Account** activity. The event wizard will start as soon as you drop the event into the activity.



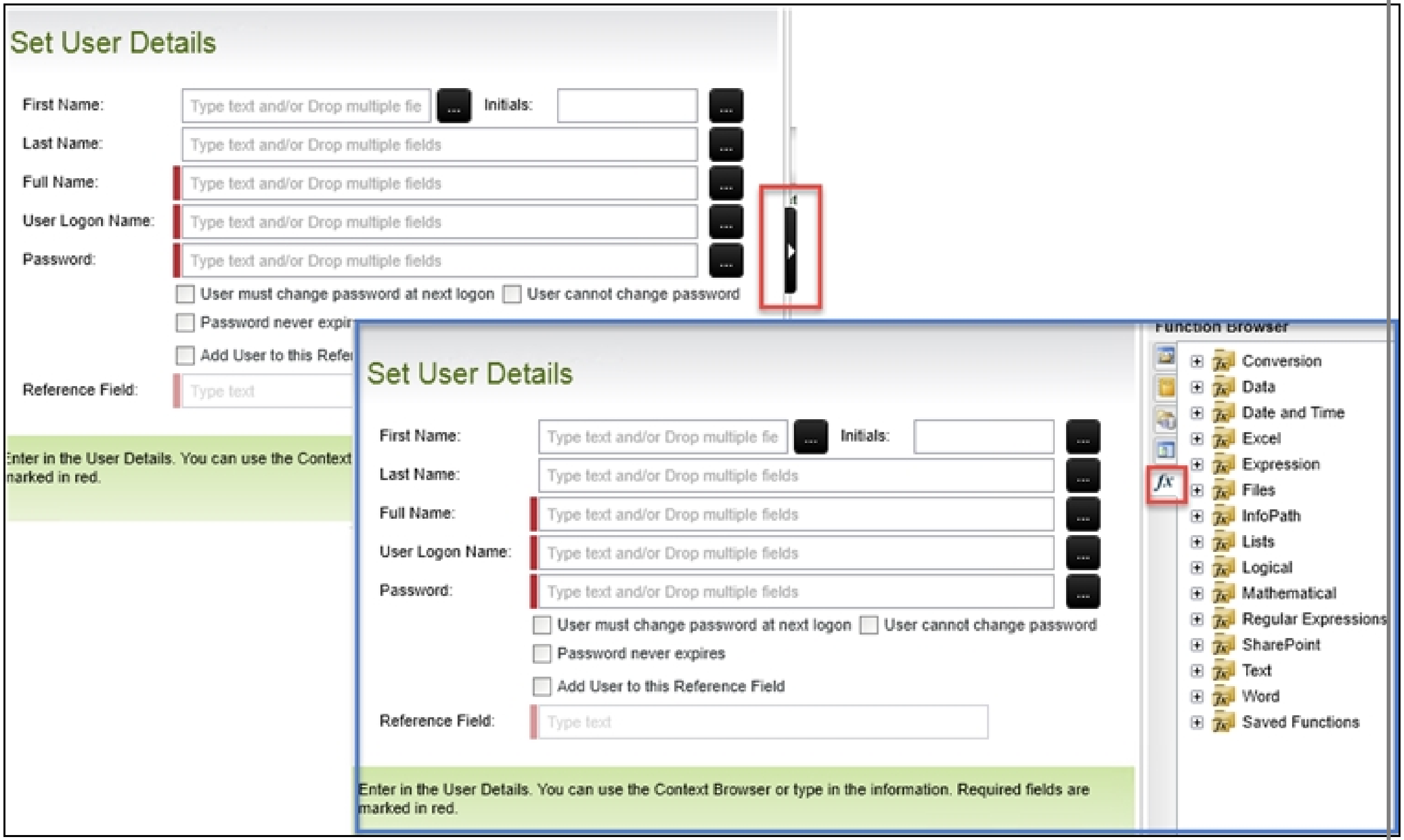
j. Click **Next** on the welcome page. On the **Event Name and Actions** screen, enter *Create New Account* as the **Event Name**, then select the **Create New User** action and click **Next**.



Event Name: Create new Account

Specify an event name and choose the action you want to take.

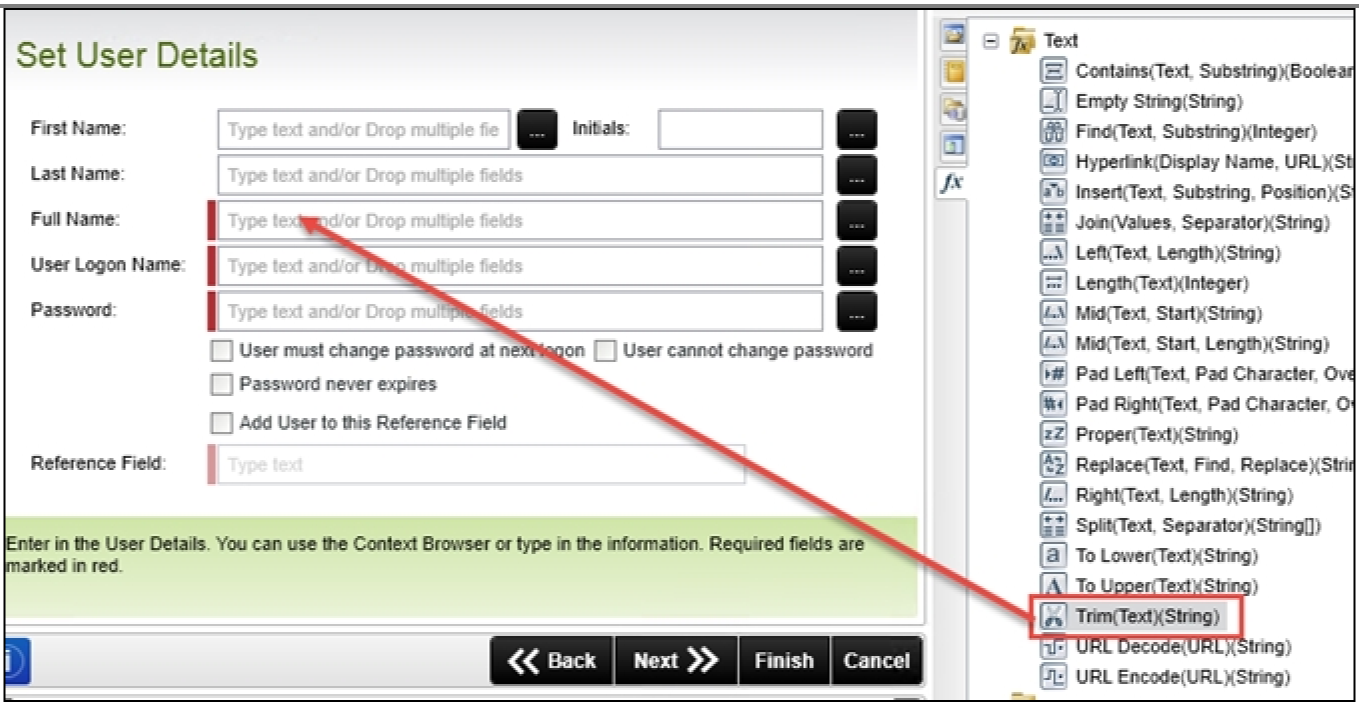
k. You will be using inline text functions to remove any extra spaces from the values entered by the requester as Active Directory is sensitive to leading and trailing spaces. On the **Set User Details** screen, click the arrow on the right side of the screen to expose the context browser, then select the **Inline Functions** tab.



Set User Details
First Name: Type text and/or Drop multiple fie
Initials:
Last Name: Type text and/or Drop multiple fields
Full Name: Type text and/or Drop multiple fields
User Logon Name: Type text and/or Drop multiple fields
Password: Type text and/or Drop multiple fields
□ User must change password at next logon
□ User cannot change password
□ Password never expires
□ Add User to this Reference Field
Reference Field: Type text
Function Browser
Conversion
Data
Date and Time
Excel
Expression
Files
InfoPath
Lists
Logical
Mathematical
Regular Expressions
SharePoint
Text
Word
Saved Functions

Enter in the User Details. You can use the Context Browser or type in the information. Required fields are marked in red.

l. Drag and drop the **Text** > **Trim** function into the **Full Name** text box. The trim function removes extra spaces from the text that was entered by the user.



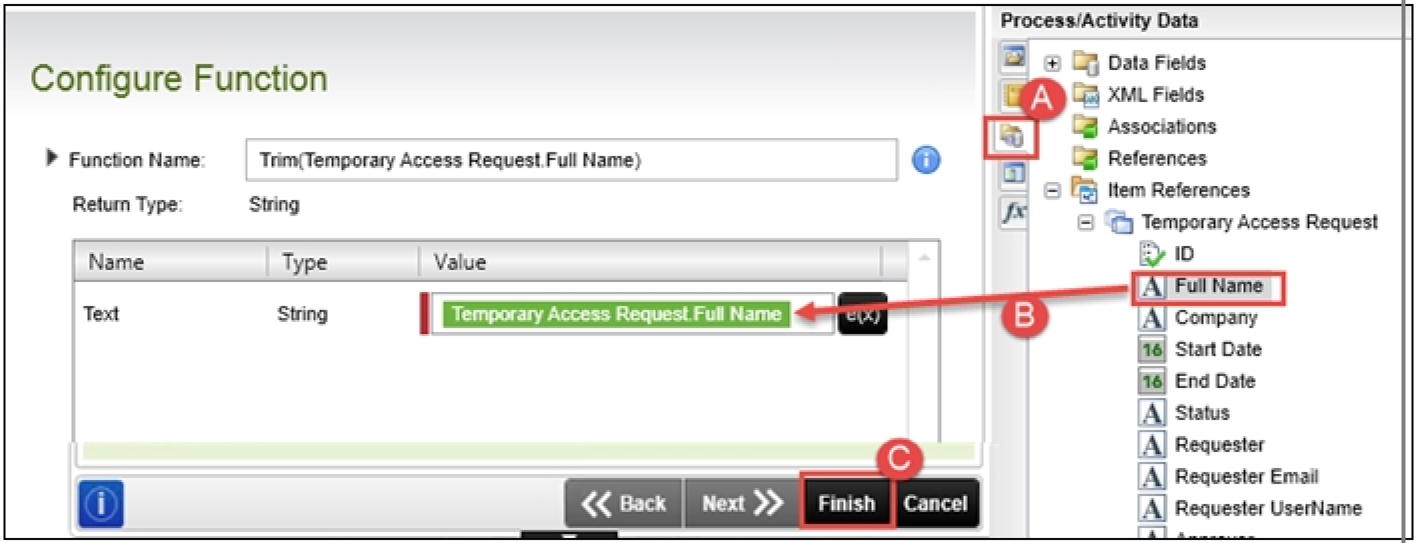
Set User Details

First Name: Type text and/or Drop multiple fie ... Initials: ...
Last Name: Type text and/or Drop multiple fields ...
Full Name: Type text and/or Drop multiple fields ...
User Logon Name: Type text and/or Drop multiple fields ...
Password: Type text and/or Drop multiple fields ...
□ User must change password at next logon □ User cannot change password
□ Password never expires
□ Add User to this Reference Field
Reference Field: Type text

Enter in the User Details. You can use the Context Browser or type in the information. Required fields are marked in red.

« Back
Next »
Finish
Cancel
Text
Contains(Text, Substring)(Boolear
Empty String(String)
Find(Text, Substring)(Integer)
Hyperlink(Display Name, URL)(St
Insert(Text, Substring, Position)(S
Join(Values, Separator)(String)
Left(Text, Length)(String)
Length(Text)(Integer)
Mid(Text, Start)(String)
Mid(Text, Start, Length)(String)
Pad Left(Text, Pad Character, Ove
Pad Right(Text, Pad Character, O
Proper(Text)(String)
Replace(Text, Find, Replace)(Strin
Right(Text, Length)(String)
Split(Text, Separator)(String[])
To Lower(Text)(String)
To Upper(Text)(String)
Trim(Text)(String)
URL Decode(URL)(String)
URL Encode(URL)(String)

m. The **Configure Function** window will open. From the Context Browser, drag the **Full Name** property from the **Temporary Access Request** Item Reference into the **Text** textbox, as shown below, then click **Finish**.

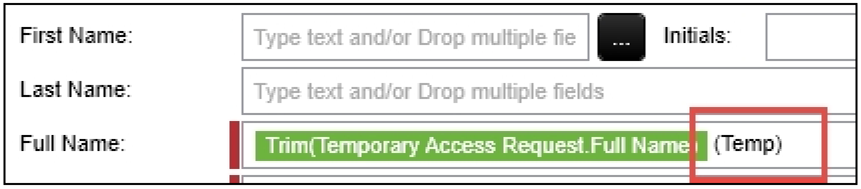


Configure Function

Function Name: Trim(Temporary Access Request.Full Name)
Return Type: String

Name	Type	Value
Text	String	Temporary Access Request.Full Name

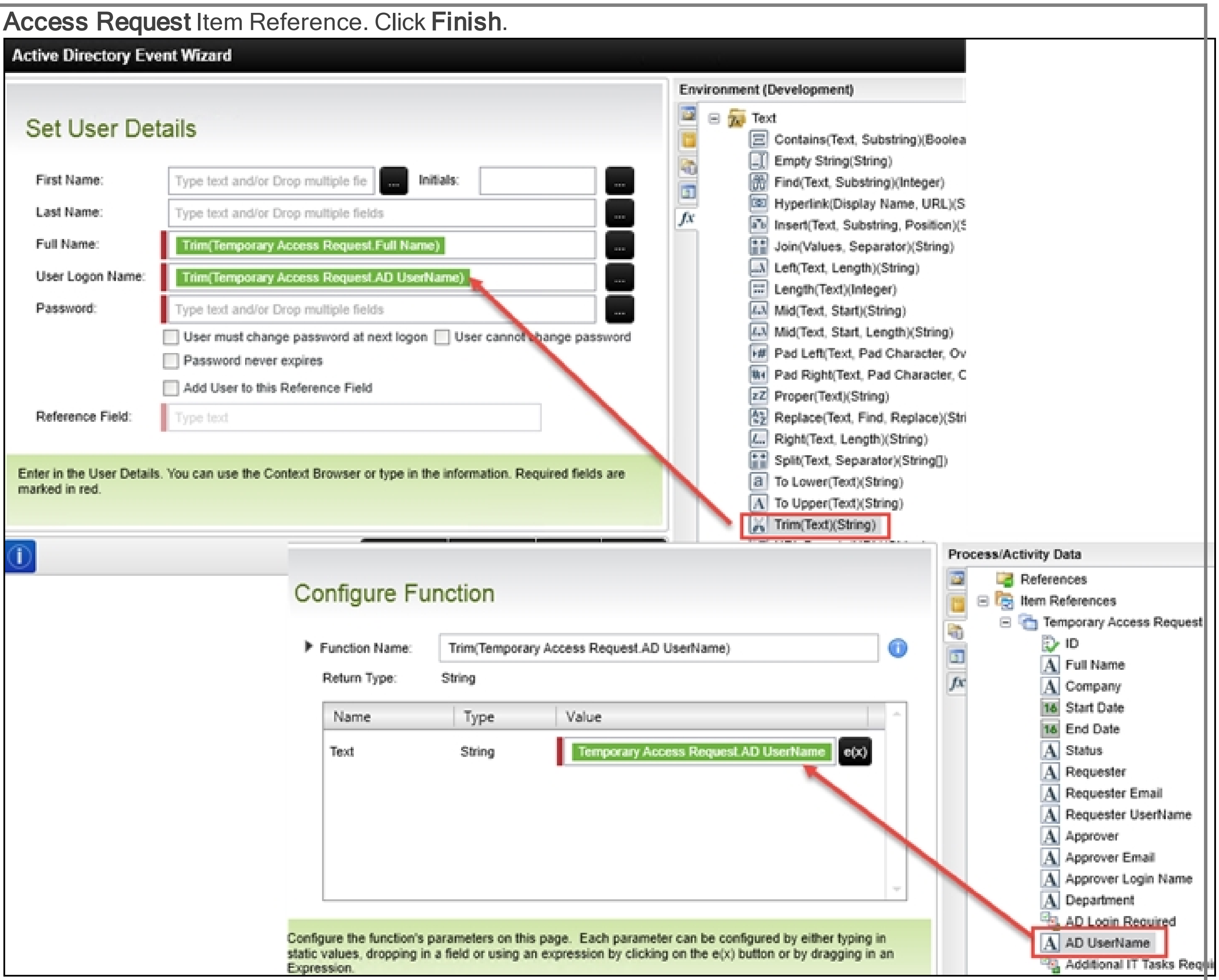
n. Manually add
(Temp)
after the **Full Name** to indicate that this is a temporary account.



First Name: Type text and/or Drop multiple fie ... Initials:
Last Name: Type text and/or Drop multiple fields
Full Name: Trim(Temporary Access Request.Full Name) (Temp)

o. Next you will set the User Logon Name property. Drag the **Trim** Inline Function into the **User Logon Name** text box, and configure the function to use the **AD UserName** property from the **Temporary**

Access Request Item Reference. Click **Finish**.



Active Directory Event Wizard

Set User Details

First Name: Type text and/or Drop multiple fie … Initials: …

Last Name: Type text and/or Drop multiple fields …

Full Name: Trim(Temporary Access Request.Full Name) …

User Logon Name: Trim(Temporary Access Request.AD UserName) …

Password: Type text and/or Drop multiple fields …

□ User must change password at next logon □ User cannot change password

□ Password never expires

□ Add User to this Reference Field

Reference Field: Type text

Enter in the User Details. You can use the Context Browser or type in the information. Required fields are marked in red.

Environment (Development)

- Text
  - Contains(Text, Substring)(Boolea
  - Empty String(String)
  - Find(Text, Substring)(Integer)
  - Hyperlink(Display Name, URL)(S
  - Insert(Text, Substring, Position)(S
  - Join(Values, Separator)(String)
  - Left(Text, Length)(String)
  - Length(Text)(Integer)
  - Mid(Text, Start)(String)
  - Mid(Text, Start, Length)(String)
  - Pad Left(Text, Pad Character, Ov
  - Pad Right(Text, Pad Character, C
  - Proper(Text)(String)
  - Replace(Text, Find, Replace)(Stri
  - Right(Text, Length)(String)
  - Split(Text, Separator)(String[])
  - To Lower(Text)(String)
  - To Upper(Text)(String)
  - Trim(Text)(String)

Configure Function

Function Name: Trim(Temporary Access Request.AD UserName)

Return Type: String

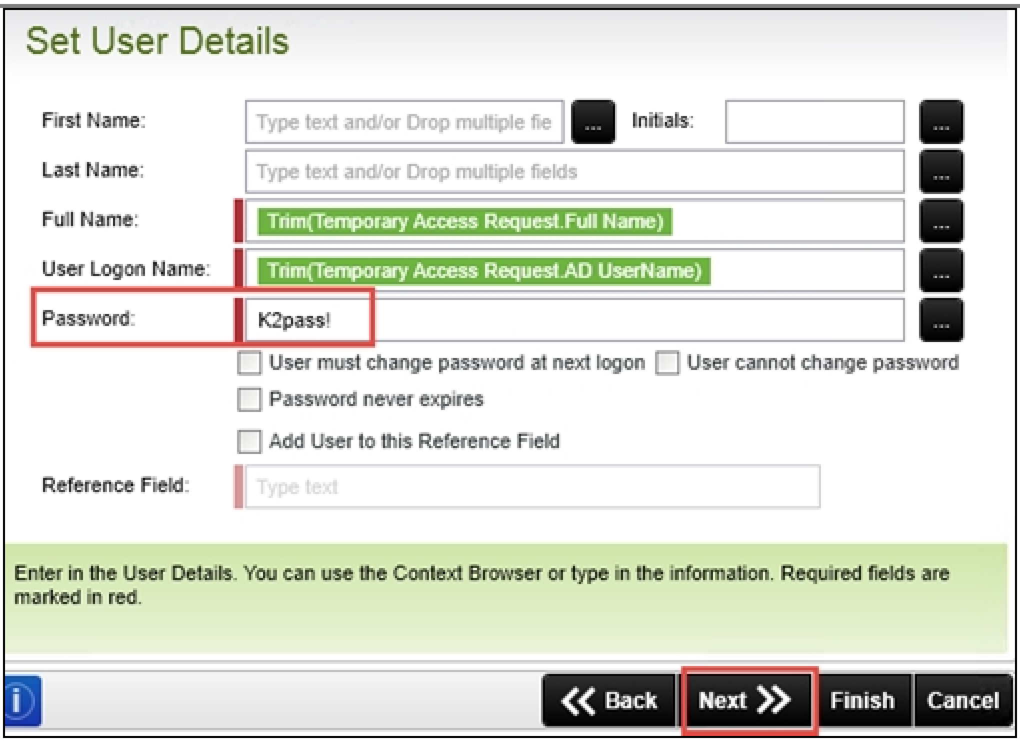
Name	Type	Value
Text	String	Temporary Access Request.AD UserName e(x)

Configure the function's parameters on this page. Each parameter can be configured by either typing in static values, dropping in a field or using an expression by clicking on the e(x) button or by dragging in an Expression.

Process/Activity Data

- References
- Item References
  - Temporary Access Request
    - ID
    - Full Name
    - Company
    - Start Date
    - End Date
    - Status
    - Requester
    - Requester Email
    - Requester UserName
    - Approver
    - Approver Email
    - Approver Login Name
    - Department
    - AD Login Required
    - AD UserName
    - Additional IT Tasks Requi

p. For the **Password**, type
K2pass!
then click **Next**.



Set User Details

First Name: Type text and/or Drop multiple fie Initials:

Last Name: Type text and/or Drop multiple fields

Full Name: Trim(Temporary Access Request.Full Name)

User Logon Name: Trim(Temporary Access Request.AD UserName)

Password: K2pass!

□ User must change password at next logon □ User cannot change password

□ Password never expires

□ Add User to this Reference Field

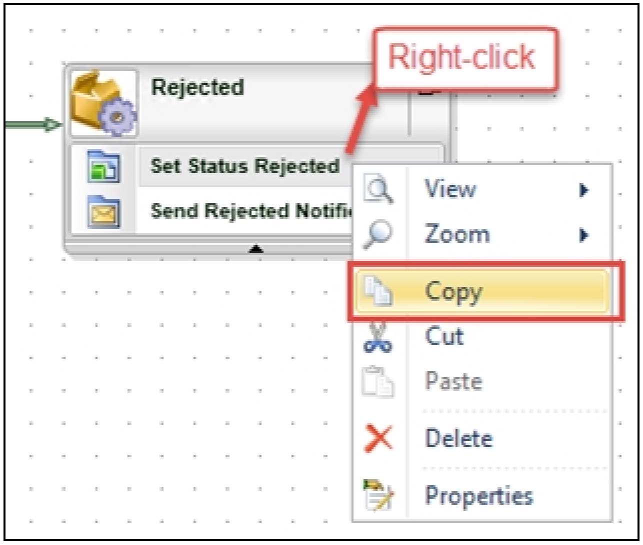
Reference Field: Type text

Enter in the User Details. You can use the Context Browser or type in the information. Required fields are marked in red.

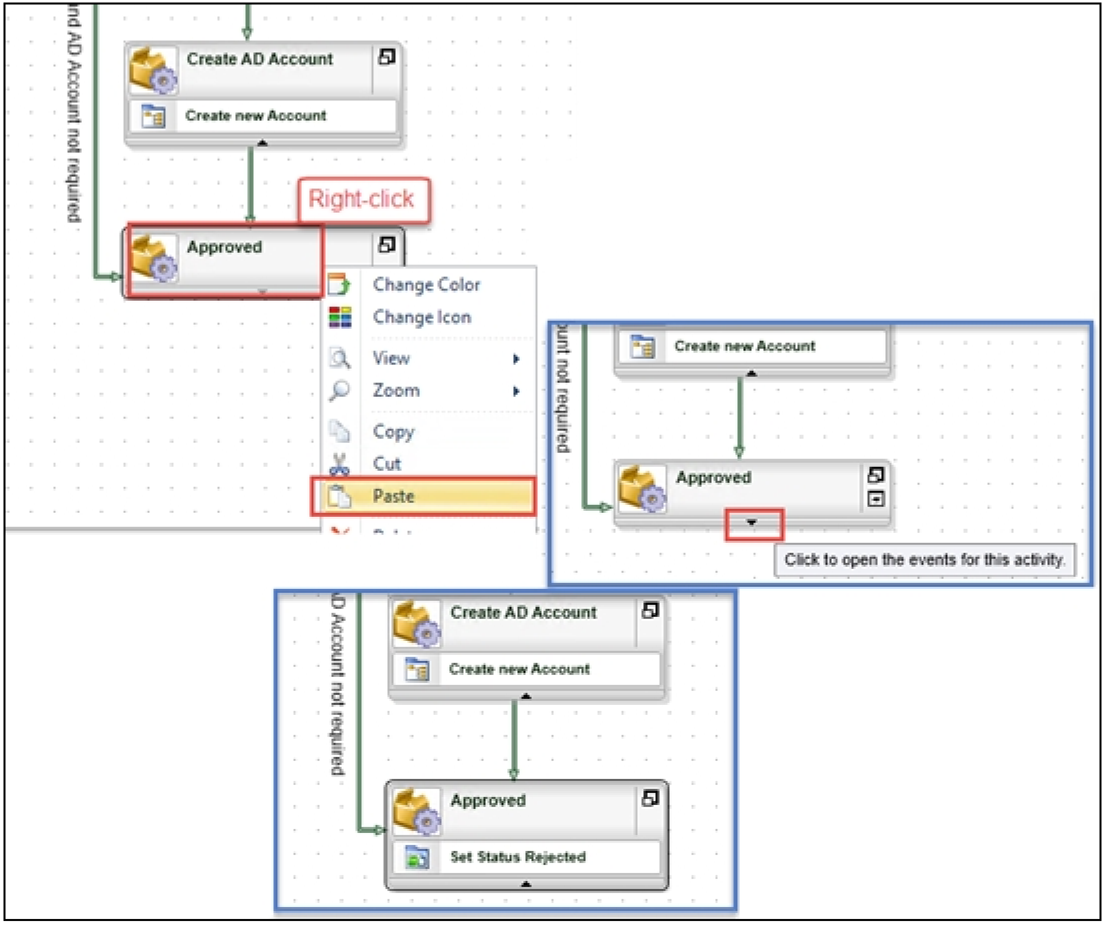
q. To save time you won't add more properties to the new AD account or create a mailbox for this user, so just click **Next**, **Next** and **Finish** to complete the Active Directory Event Wizard.
Do not select the options to add the user to AD Groups or to create a mailbox for the user. In a real-world environment you may want to populate additional properties for the user or create a mailbox, and you can use the context browser to drag and drop variables into fields as necessary just like you did for the first screens of the wizard. You can skip any additional properties to save time in this tutorial.

Now you will add a SmartObject event to set the **status** of the **request** to **approved**. Instead of dragging and dropping (and configuring) a new SmartObject event, you can simply copy and paste an existing SmartObject event and re-configure it, as it already does something similar to what you want to achieve.

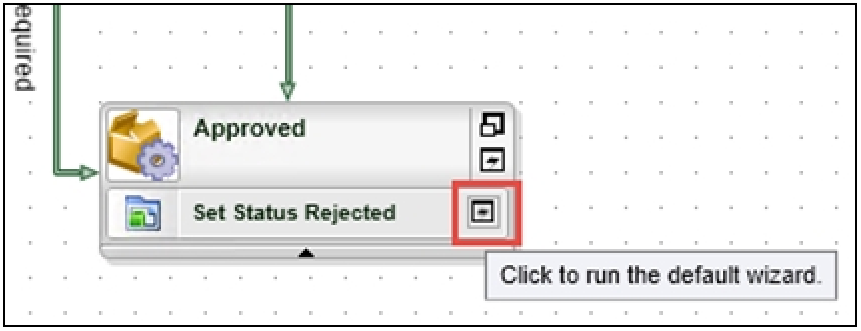
r. In the workflow canvas, right-click the **Set Status Rejected** event and select **Copy**.



s. Right-click on the **Approved** step and select **Paste**, and then click the expand-down arrow to show the pasted event.

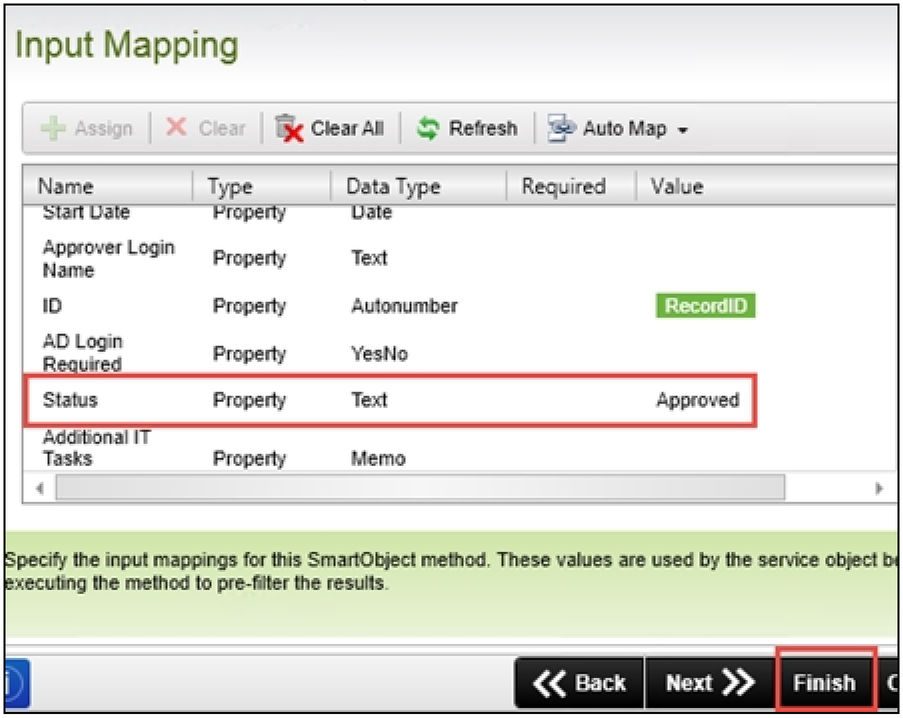


t. Move your mouse pointer over the copied event and click the event icon to re-run the wizard for the copied event.



u. Click **Next** to get to the **Input Mapping** screen for the wizard. Click to highlight the **Status** line, then click the **Assign** button and change the text box entry to
Approved

Click **OK**, then **Finish** to complete the wizard.

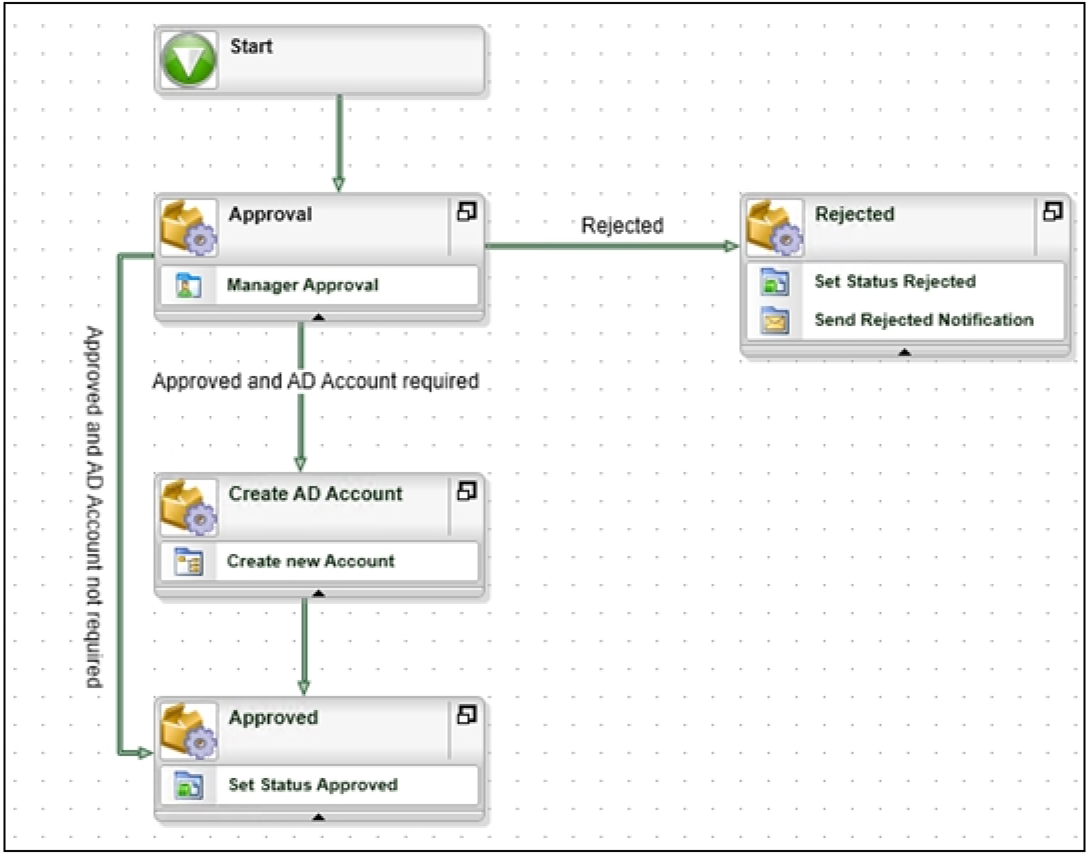


Input Mapping

Name	Type	Data Type	Required	Value
Start Date	Property	Date		
Approver Login Name	Property	Text		
ID	Property	Autonumber		RecordID
AD Login Required	Property	YesNo		
Status	Property	Text		Approved
Additional IT Tasks	Property	Memo		

Specify the input mappings for this SmartObject method. These values are used by the service object be executing the method to pre-filter the results.

v. Double-click the copied event name and rename it to *Set Status Approved*
Your workflow should look like the image below.



w. **Save** your work. (File > Save)

Step 6 Review

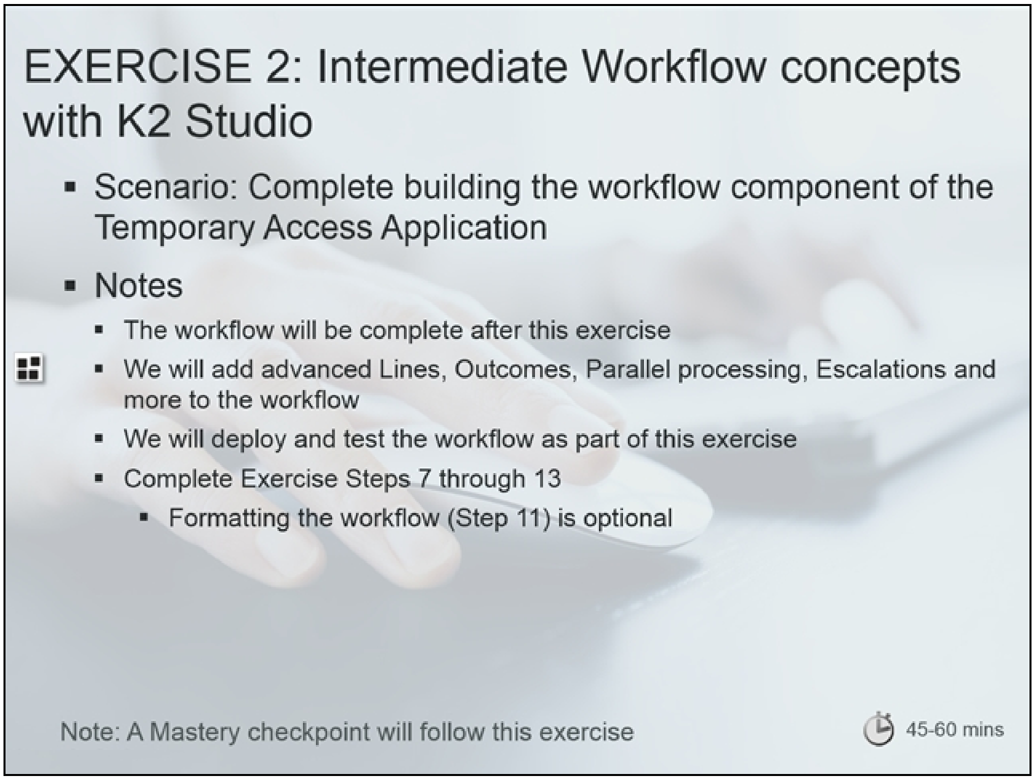
In this step you added system events so that the workflow will perform tasks depending on whether the request was rejected or approved. You also created a new user account in Active Directory for the temporary user (if an account was required). You learned how to use inline functions in a wizard, and how to copy and paste events into activities and re-run a wizard to reconfigure an event. You also learned how to call a SmartObject method (save - to update the request status) in a workflow and how to send an email during a workflow.

In the next part of this tutorial, you will add some more advanced workflow design concepts.

Part 1 Review

In this part of the tutorial you learned how to build a workflow with K2 Studio, how to use wizards and connect activities with lines. These are the basic principles used to assemble workflows with K2 Studio. In Part 2: Building a more complex workflow with K2 Studio, you will explore more advanced workflow concepts like conditional lines, evaluating user data and business data in a line, escalations, loop-back lines and more.

EXERCISE 2: Intermediate Workflow concepts with K2 Studio



EXERCISE 2: Intermediate Workflow concepts with K2 Studio

- Scenario: Complete building the workflow component of the Temporary Access Application
- Notes
  - The workflow will be complete after this exercise
  - We will add advanced Lines, Outcomes, Parallel processing, Escalations and more to the workflow
  - We will deploy and test the workflow as part of this exercise
  - Complete Exercise Steps 7 through 13
    - Formatting the workflow (Step 11) is optional

Note: A Mastery checkpoint will follow this exercise

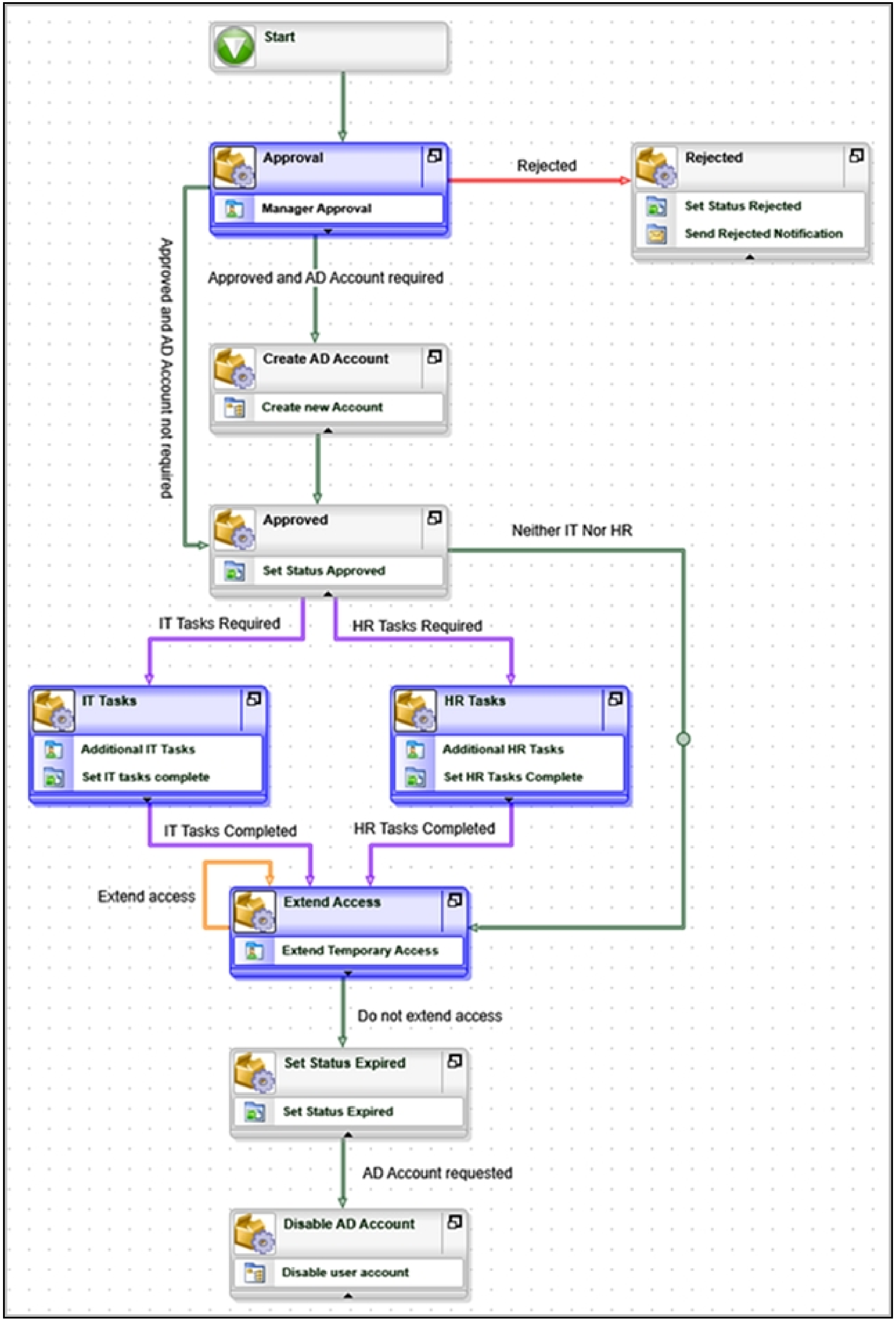
45-60 mins

Video

Now it is time to learn how to implement some of the more advanced workflow concepts we just covered. In this exercise, you will complete the workflow components of the Temporary Access Application. You will learn how to implement parallel execution, merging activities, conditional line rules, escalations, looping activities and how to format workflows for clarity.

You will be completing the workflow in this exercise, so you will also deploy and then test the workflow with various combinations of data that will control the flow of the workflow.

The Temporary Access workflow as it will look at the end of the exercise



Note
The step to format the workflow (Step 11) with colors is optional. If time is limited, this step can be skipped.

When you are ready, continue on with Temporary Access Application: Part 2 to complete the workflow, and then to Temporary Access Application: Part 3 (Testing the application) to test the workflow.

Temporary Access Application: Part 2 (Building a more complex workflow with K2 Studio)

In Part 2 of the Temporary Access Application tutorial, you will continue building the workflow that you begin in Part 1. You will explore more advanced concepts such as adding conditional lines, evaluating user data and business data in a line, enabling escalations, loop-back lines and more. You will also learn how to format workflows for clarity. Finally, you will deploy the finished workflow to a K2 environment.

> **Note**
> You must complete all of the steps from Part 1 of this tutorial before continuing with Part 2. Part 2 builds upon the workflow concepts that were started in Part 1.

Step 7: Configure conditional line rules to run in parallel mode

In this step you will add the HR user task and the IT user task to the workflow. You will configure a conditional rule to evaluate which task line the workflow should follow or if the workflow should follow both task lines. In this step, you are configuring the workflow to run in parallel mode.

Step 7 Tasks

1. Add two Default Activities called
   IT Tasks
   HR Tasks
2. Add lines to join the **Approved** activity to the **IT Tasks** and **HR Tasks** in parallel and then define line rules to conditionally follow the lines depending on whether the **Additional IT Tasks** and **Additional HR Tasks** properties of the **Temporary Access Request** Item Reference are *True*.
3. Add a **SmartForms Client Event** to the IT Tasks activity, and configure it as follows:

Field	Value
Name	*Additional IT Tasks*
Form	**Temporary Access Request Form**
State	Use existing **IT Tasks** State
Configure Item Reference	(no changes, use default values)
Specify Form Parameters	(no changes, use default values)
Configure Open Worklist Item Rule	(no changes, use default values)
Transfer Workflow Data to Form	(no changes, use default values)
Actions	*IT Tasks Completed*
Action Display Settings	(no changes, use default values)
Transfer Form Data to Workflow	(no changes, use default values)
Outcomes	(no changes, use default values)
Destination User	Select a group of users, in this sample you will use **Denallix\Operations**. Or use your own username for testing purposes.
Task Notification	(Do not send a task notification)

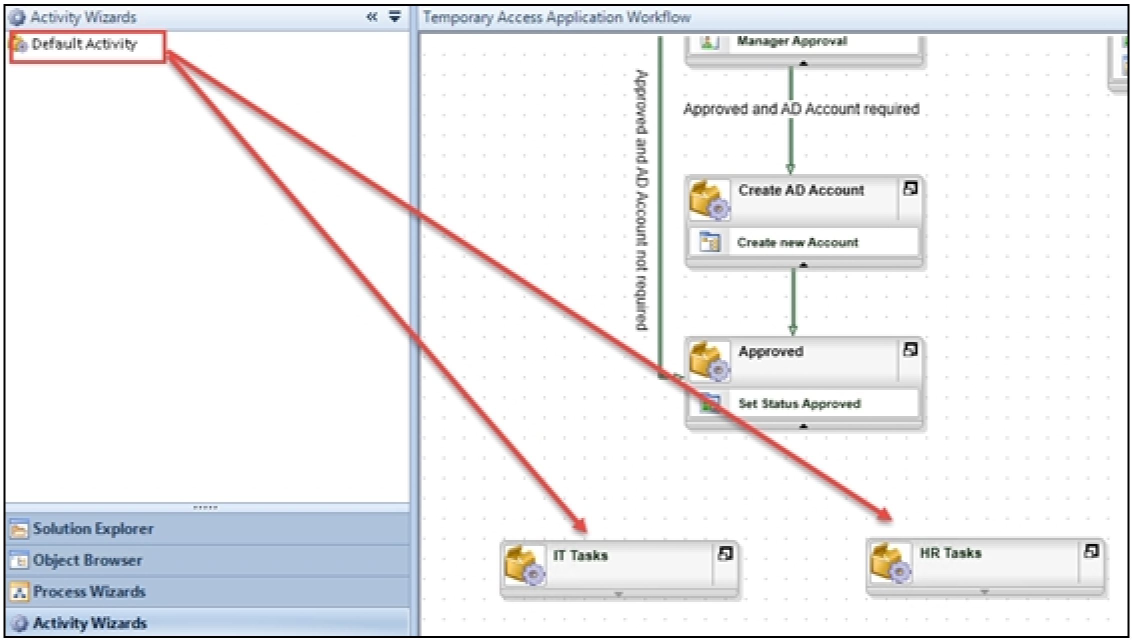
4. Add a **SmartForms Client Event** to the HR Tasks activity, and configure it as follows:

Field	Value
Name	*Additional HR Tasks*
Form	**Temporary Access Request Form**

State	Use existing **HR Tasks** State
Configure Item Reference	(no changes, use default values)
Specify Form Parameters	(no changes, use default values)
Configure Open Worklist Item Rule	(no changes, use default values)
Transfer Workflow Data to Form	(no changes, use default values)
Actions	*HR Tasks Completed*
Action Display Settings	(no changes, use default values)
Transfer Form Data to Workflow	(no changes, use default values)
Outcomes	(no changes, use default values)
Destination User	Select a group of users, in this sample you will use **Denallix\Human Resources**. Or use your own username for testing purposes.
Task Notification	(Do not send a task notification)

Step 7 Walkthrough

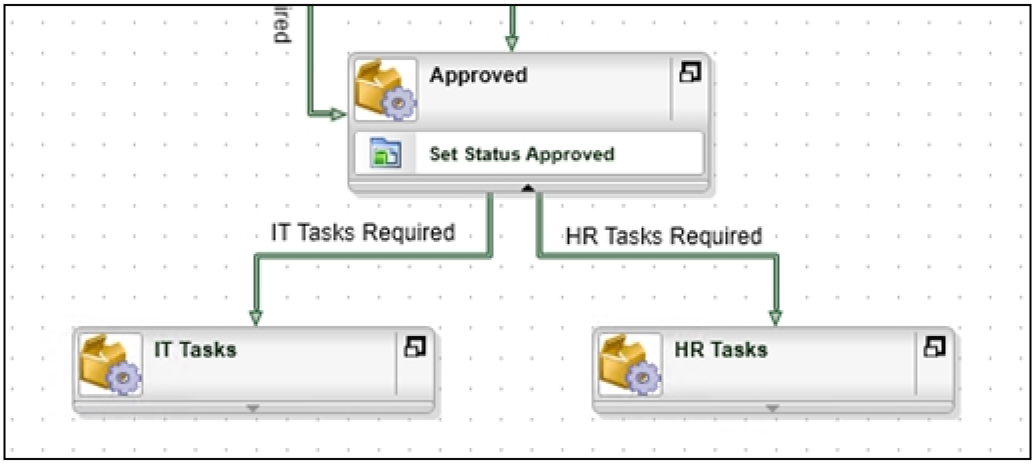
a. Select the **Activity Wizards** pane, then drag and drop two activities onto the design canvas. Name the activities as follows:
IT Tasks
HR Tasks



b. Draw two lines to connect the **Approved** activity with the two new activities. (Right-click in the first activity and drag down into the second activity to draw a line.) Double-click the lines to add line labels as follows:
IT Tasks Required
HR Tasks Required

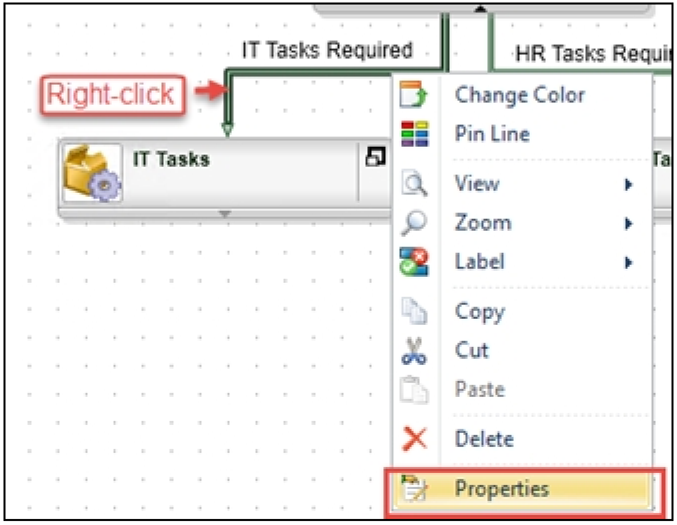
> **Note**
> In this step, you are configuring the IT Tasks and HR Tasks to run in *parallel* mode. If both IT Tasks and HR

Tasks are selected on the request form, you want them to run at the same time. Since each of these tasks are independent of the other (you don't need HR Tasks to wait on the IT Tasks and vice versa, you can configure them to run at the same time. You will configure a condition in a later step that will wait for both sets of tasks to complete before proceeding (if both sets of tasks were selected on the form).

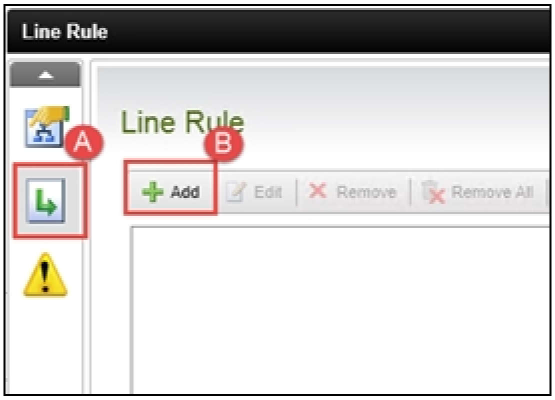


Now you will configure the line rules so that the lines will only be followed if the requester selected the associated option when filling out the form.

c. Right-click the **IT Tasks Required** line and select **Properties**.

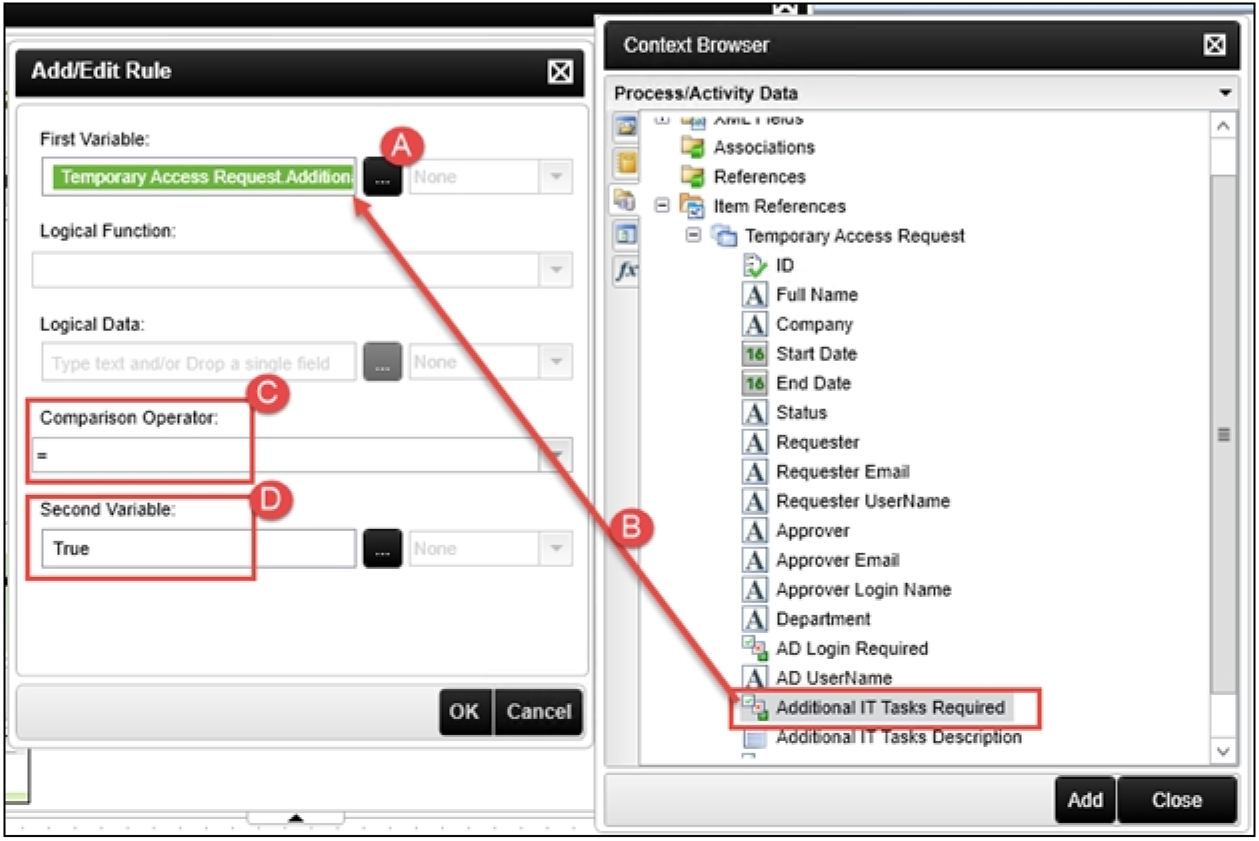


d. Select the Line Rule icon, then click the **Add** button.

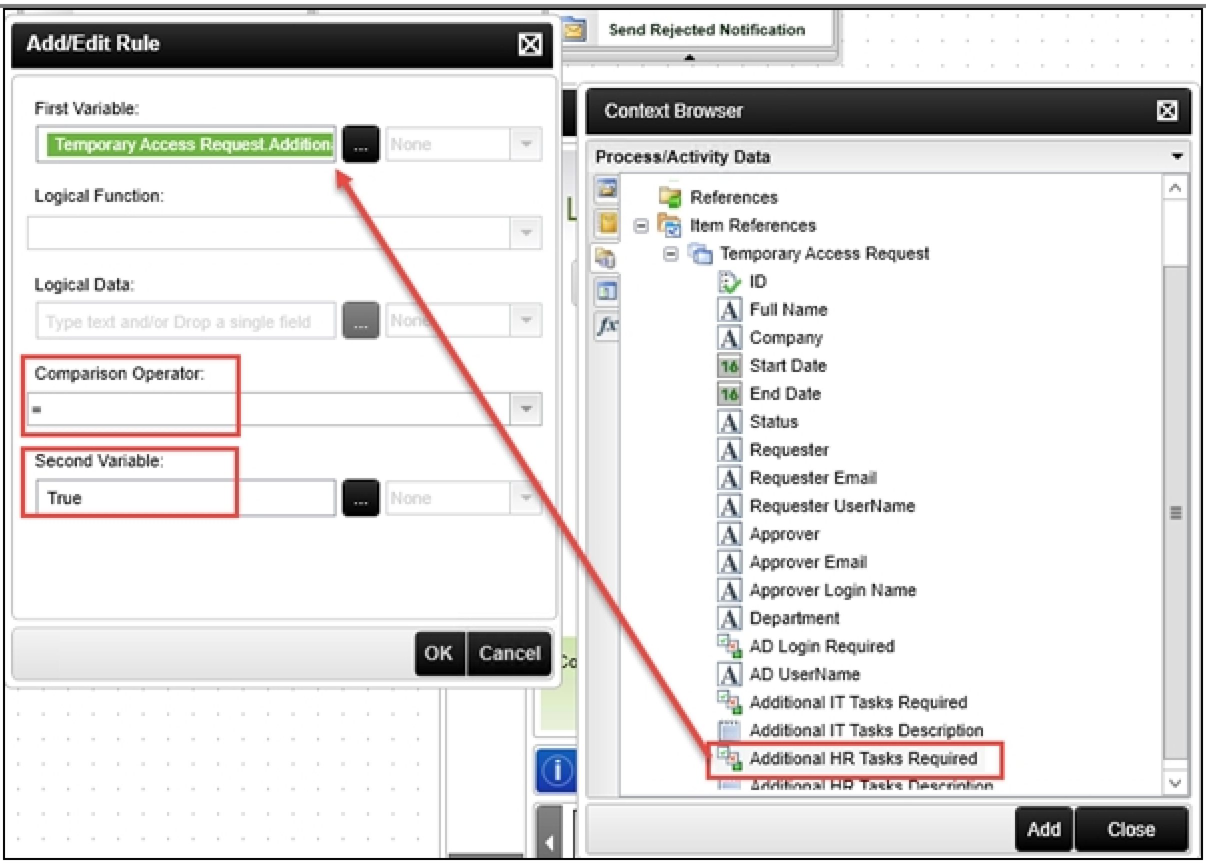


e. Configure the Line rule as follows:
First Variable: **Item References > Temporary Access Request > Additional IT Tasks Required**
Comparison Operator: **=**
Second Variable: *True*

then click **OK** and **Close** the context browser and click **Finish**. The line will now be followed if the Additional IT Tasks option was selected on the request form.

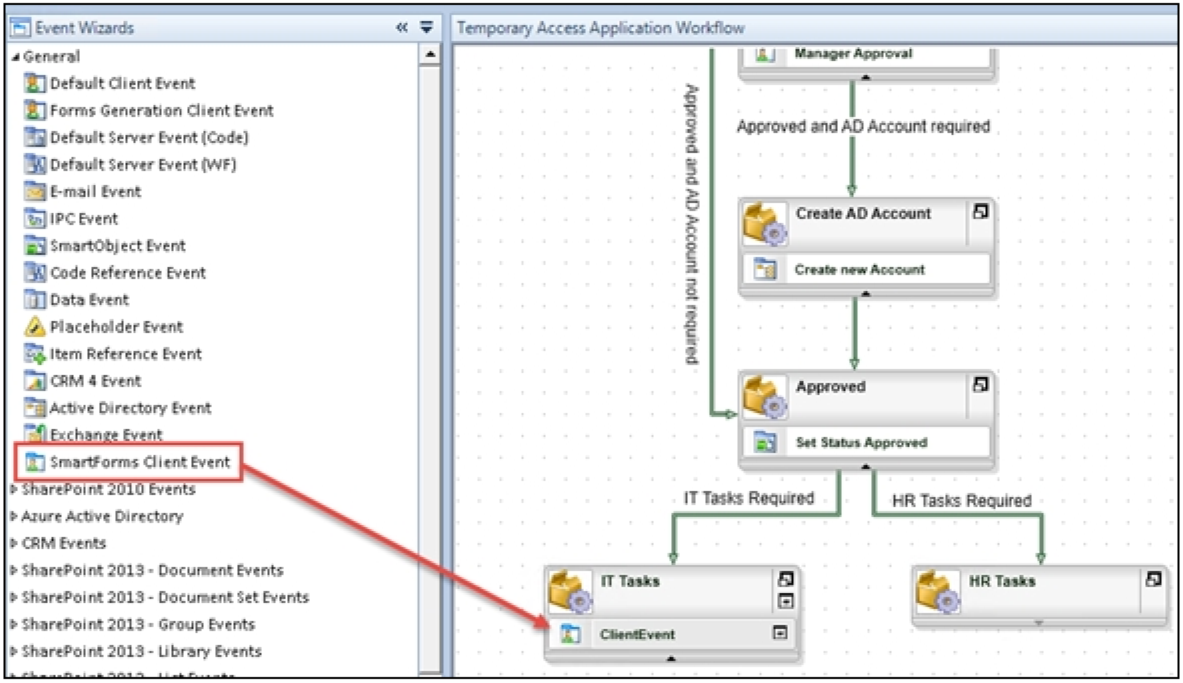


f. Next you will set up the HR Tasks line. Using the same steps as before, right-click the **HR Tasks Required** line, and configure the line rule as follows:
First Variable: **Item References** > **Temporary Access Request** > **Additional HR Tasks Required**
Comparison Operator: **=**
Second Variable: *True*
then click **OK** and then **Close** the context browser and click **Finish**. The line will now be followed if the Additional HR Tasks option was selected on the request form.

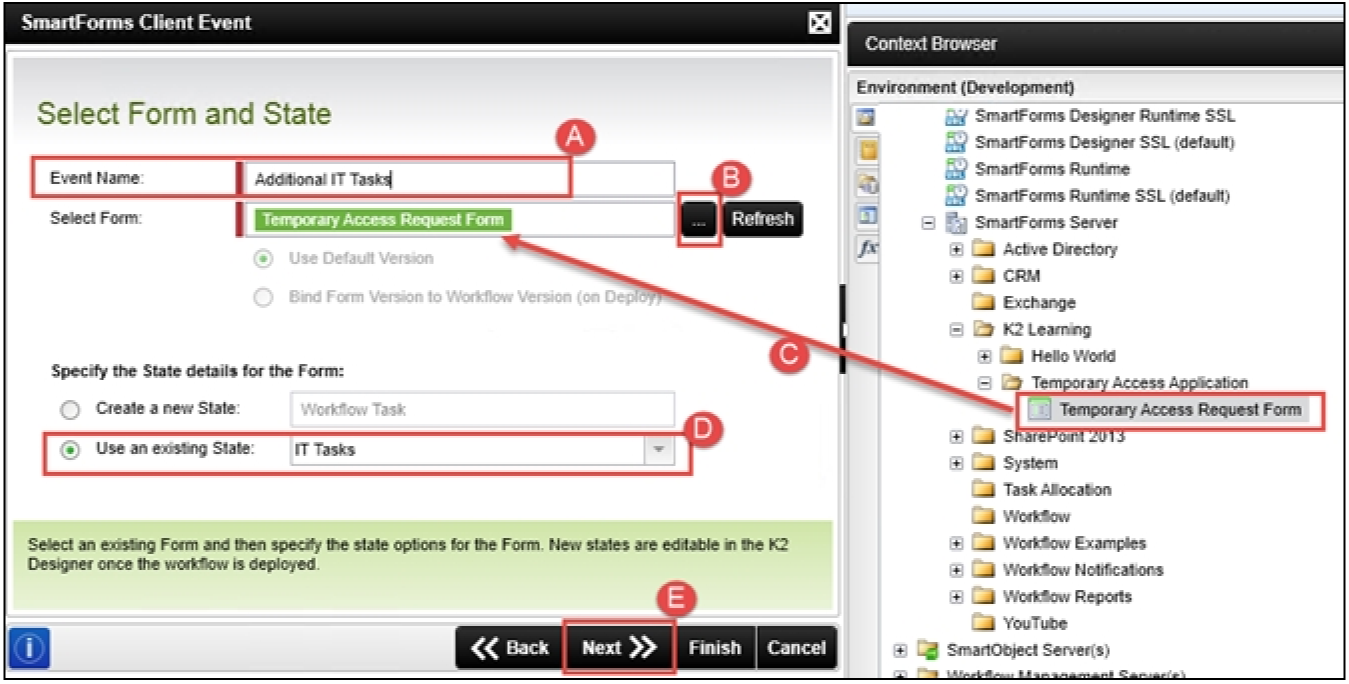


Now that you have set up the rules for your parallel lines, you will add user tasks to both the IT Tasks and HR Tasks activities. The user tasks will require a user from each group to action the task before the workflow can continue. (This is, of course, dependent on whether or not the additional task options where selected on the request form.)

g. Switch to the **Event Wizards** pane, then drag and drop a **SmartForms Client Event** into the **IT Tasks** activity. Click **Next** on the wizard welcome page.



h. On the **Select Form and State** screen, rename the event to
Additional IT Tasks
and use the ellipsis button to locate and select the **Temporary Access Request Form**.
Configure the state options to **Use an existing State** and select the **IT Tasks** state, then click **Next**.


SmartForms Client Event
Select Form and State
A
B
C
D
E
Event Name: Additional IT Tasks
Select Form: Temporary Access Request Form
...
Refresh
Use Default Version
Bind Form Version to Workflow Version (on Deploy)
Specify the State details for the Form:
Create a new State: Workflow Task
Use an existing State: IT Tasks
Select an existing Form and then specify the state options for the Form. New states are editable in the K2 Designer once the workflow is deployed.
Back
Next
Finish
Cancel
Context Browser
Environment (Development)
SmartForms Designer Runtime SSL
SmartForms Designer SSL (default)
SmartForms Runtime
SmartForms Runtime SSL (default)
SmartForms Server
Active Directory
CRM
Exchange
K2 Learning
Hello World
Temporary Access Application
Temporary Access Request Form
SharePoint 2013
System
Task Allocation
Workflow
Workflow Examples
Workflow Notifications
Workflow Reports
YouTube
SmartObject Server(s)
Workflow Management Server(s)

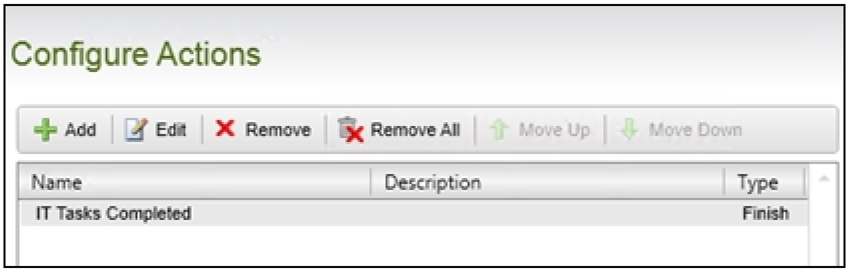
i. Click **Next** to skip over the **Configure Item Reference** screen.

j. Click **Next** to skip over the **Specify Form Parameters** screen.

k. The **Configure Open Worklist Item Rule** should be configured correctly to open the worklist item **When the Form is Initializing**, so click **Next** to continue.

l. Click **Next** to skip over the **Transfer Workflow Data to Form** screen.

m. On the **Configure Actions** screen, add one **Action** called
IT Tasks Completed
then click **OK**, then **Next**.



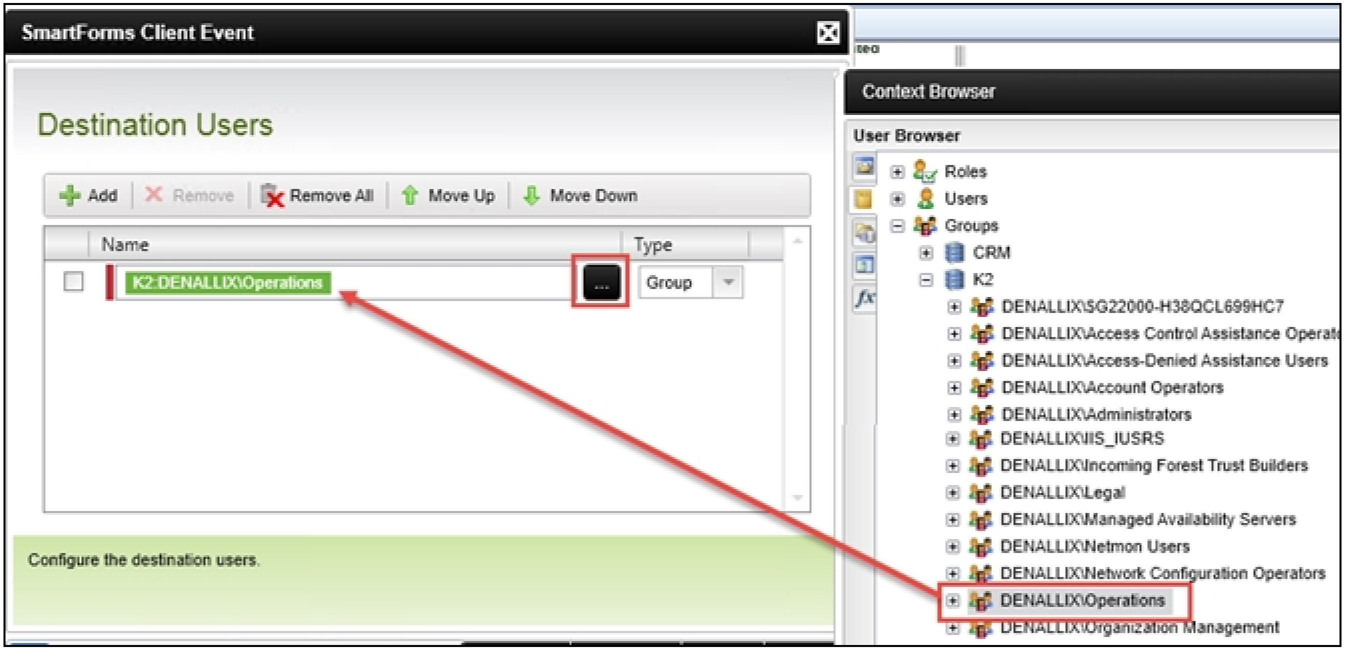
Name	Description	Type
IT Tasks Completed		Finish

n. Leave the **Action Display Settings** page as-is, and click **Next**.

o. Click **Next** to skip over the **Transfer Form Data to Workflow** screen.

p. Click **Next** to skip over the **Configure Outcomes** screen.

q. On the **Destination Users** screen, click the **Add** button to add a new destination user, then use the ellipsis button to select the user or group that will perform the IT tasks. In this sample, you will use the **Denallix\Operations** AD Group, but you may need to select a different group or user if you are using your own environment for this tutorial. You can also select the **Originator** value found in the **Workflow Context Browser** > **Process Instance** node to route the task back to you for convenience in testing, if you prefer. Click **Next** to continue.


SmartForms Client Event
Destination Users
Add
Remove
Remove All
Move Up
Move Down
Name
Type
...
Configure the destination users.
Context Browser
User Browser
Roles
Users
Groups
CRM
K2
DENALLIX\SG22000-H38QCL699HC7
DENALLIX\Access Control Assistance Operat
DENALLIX\Access-Denied Assistance Users
DENALLIX\Account Operators
DENALLIX\Administrators
DENALLIX\IIS_IUSRS
DENALLIX\Incoming Forest Trust Builders
DENALLIX\Legal
DENALLIX\Managed Availability Servers
DENALLIX\Netmon Users
DENALLIX\Network Configuration Operators
DENALLIX\Operations
DENALLIX\Organization Management

Name	Type
K2:DENALLIX\Operations	Group

r. To save time you won't send a notification for this task, so click **Finish** to complete the wizard. (If you wish, you can select the notify option and customize the notification message or come back to the activity later and add a custom notification. Remember that you can use item references to further customize your message body.)

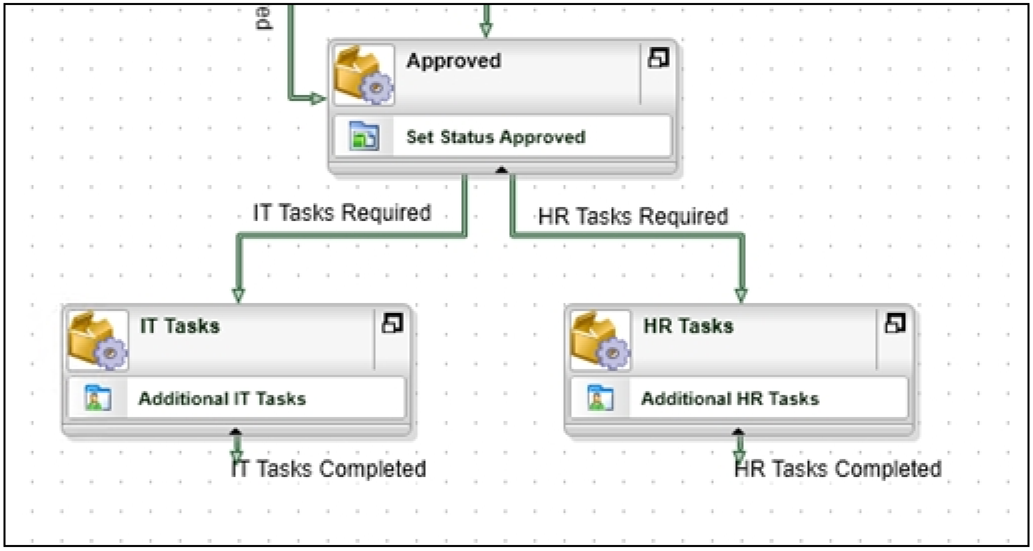
You have completed the IT Tasks user task step. Next, you will add the HR Tasks user task step following a similar procedure. You should be familiar enough now with the wizard screens, so we will just provide the configuration values for the wizard screens in a table. If you need navigation help to work through the screens, follow the same steps you just completed in configuring the IT Tasks event.

s. Drag and drop a **SmartForms Client Event** into the **HR Tasks** Activity, and configure the wizard screens using the table below as a guide.

Screen	Field	Value
Select Form and State	Name	*Additional HR Tasks*
	Form	**Temporary Access Request Form**
	State	Use existing **HR Tasks** State
Configure Item Reference		(no changes, use default values)
Specify Form Parameters		(no changes, use default values)
Configure Open Worklist Item Rule		(no changes, use default values)
Transfer Workflow Data to Form		(no changes, use default values)
Configure Actions	Actions	*HR Tasks Completed*
Action Display Settings		(no changes, use default values)
Transfer Form Data to Workflow		(no changes, use default values)

Outcomes		(no changes, use default values)
Destination User		Select a group of users, in this sample you will use **Denallix\Human Resources**. Or use the **Originator** property found in the **Workflow Context Browser** to route the task back to you for testing convenience.
Task Notification		(Do not send a task notification)

t. Your workflow should now look like the image below. **Save** your work.



Step 7 Review

In this step you defined conditional line rules to determine the flow of a workflow. In this case, the workflow will go to the IT Tasks, HR Tasks, or both, depending on the options selected by the requester in the original form. You learned how to define a line rule to conditionally follow a path in the workflow based on data instead of a user's task decision.

Next you will do some work to handle the parallel execution (in other words what happens if both HR and IT tasks were selected) and also add a line to define what happens if neither HR or IT tasks were selected.

Step 8: Merge parallel paths and add a line to handle cases for no additional tasks

In this step you will add a *merge* activity that will determine whether it needs to wait for IT Tasks, HR Tasks, or both to be completed. You will also add a line for the workflow to follow if neither IT nor HR tasks were selected. When building workflows with parallel paths, you need to add an activity that will act as the merging point for the parallel lines. This activity will have rules that tell K2 to wait until both parallel activities have completed, before continuing on. If only one additional task was selected (either IT or HR), then the merging activity will simply continue on after the task has completed.

Step 8 Tasks

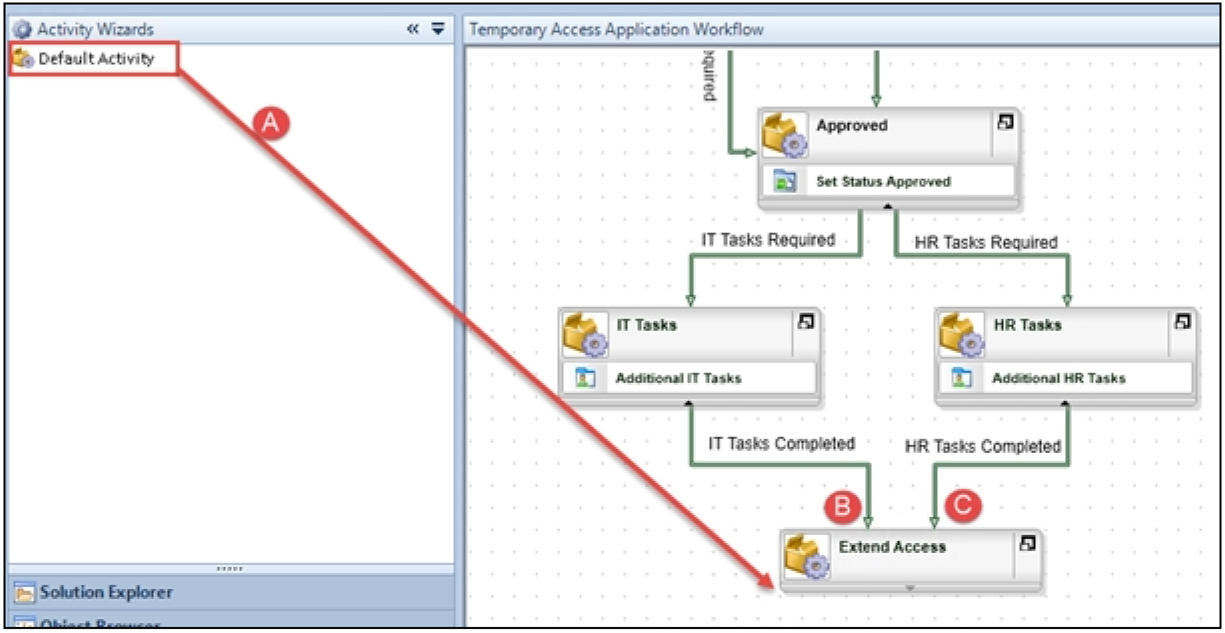
1. Add a new activity called
   Extend Access
   then connect the **IT Tasks Completed** and **HR Tasks Completed** lines to the new activity.
2. Draw a line that goes from **Approved** to **Extend Access**, and configure the line rule so that it is followed if BOTH the **Additional IT Tasks** and **Additional HR Tasks** properties of the **Temporary Access Request** Item Reference are *False*.
3. Define a preceding rule on the Extend Access activity as follows:
   Item Reference > Temporary Access Request > Additional IT Tasks Required
   =
   False
   AND
   Item Reference > Temporary Access Request > Additional HR Tasks Required

=
False

4. Add a SmartObject event to the **IT Tasks** activity to set the value of the **Additional IT Tasks Required** property to
False
5. Add a SmartObject event to the **HR Tasks** activity to set the value of the **Additional HR Tasks Required** property to
False
so that the preceding rule will execute correctly.

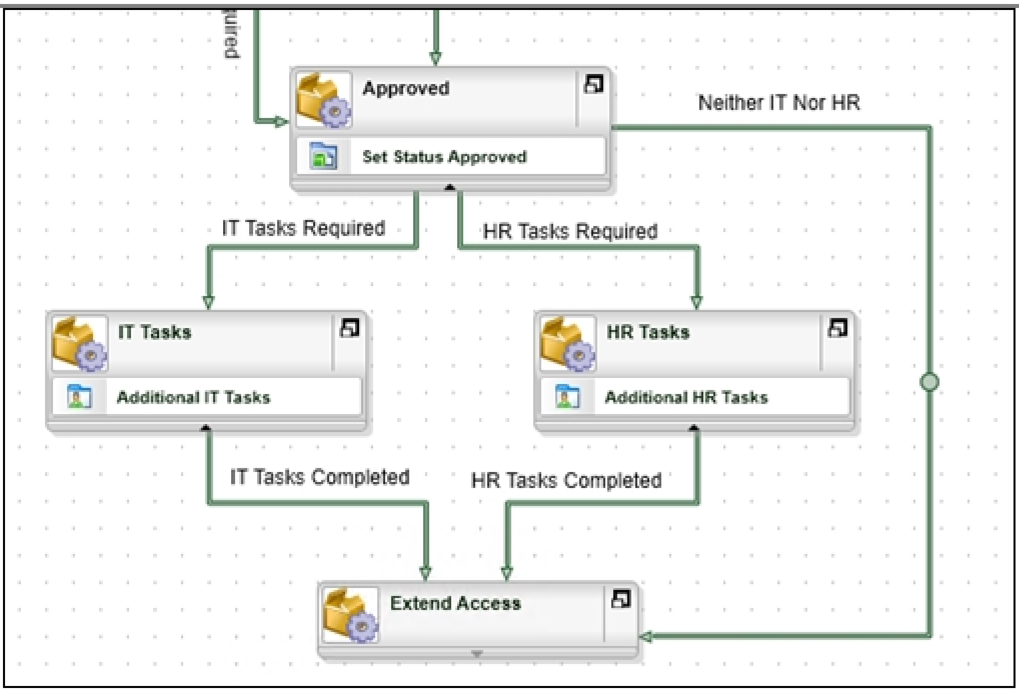
Step 8 Walkthrough

a. Add a new activity called
Extend Access
then grab and drag the line arrow-heads to connect the **IT Tasks Completed** and **HR Tasks Completed** to the new activity.

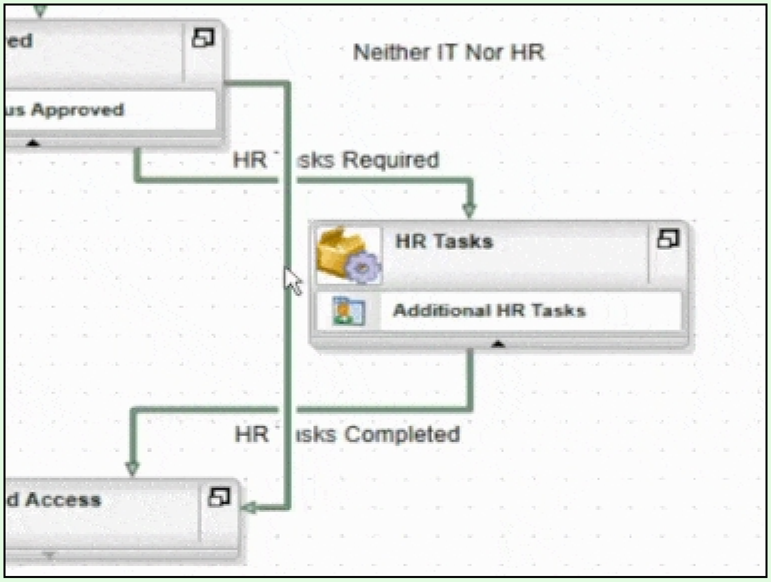


This activity will contain a user task so that the original requester can decide whether to extend the temporary access or not. Conveniently, it will also act as the merging activity for the IT Tasks and HR Tasks, if they were run in parallel. Once again, you need a merging activity so that you can configure a condition to wait for both the IT Tasks and HR Tasks to complete before moving forward with the workflow.

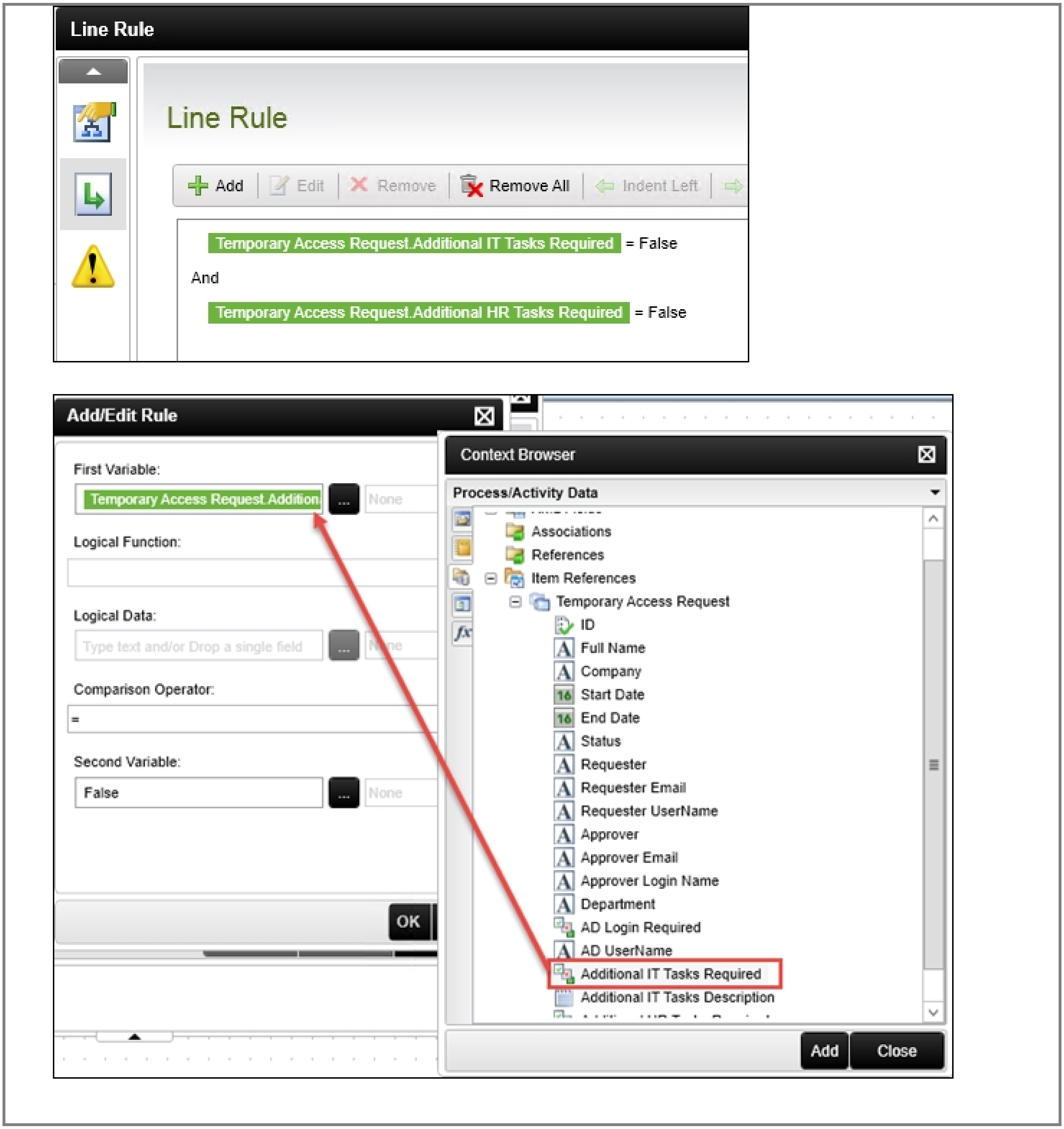
b. You also need to cater for cases where neither the Additional HR Tasks nor the Additional IT Tasks options were selected. Draw a line that goes from the **Approved** activity to the **Extend Access** activity. This is the line you will follow if neither IT nor HR tasks were selected. Double-click the line and set the label to
Neither IT Nor HR



Tip
You can use the Pin Line function to add a draggable pin to the line. This allows you to grab the pinned handle and drag the line to a different place on the canvas. In the example below, you are dragging the line so that it creates an outer loop of sorts around the HR Tasks activity. (Hint: once you have added the pins, click and drag the handles fairly quickly. It can take a few attempts to get the hang of it.)



c. Right-click the new line and add a line rule as follows
First Variable: **Item Reference > Temporary Access Request > Additional HR Tasks Required**
Comparison Operator: **=**
Second Variable: *False*
Boolean Operator: **AND**
First Variable: **Item Reference > Temporary Access Request > Additional IT Tasks Required**
Comparison Operator: **=**
Second Variable: *False*
See the screen shots for reference. Here you are simply telling K2 to follow this line if neither the Additional IT Tasks nor the Additional HR Tasks options were selected on the request form.



Line Rule

Temporary Access Request.Additional IT Tasks Required = False

And

Temporary Access Request.Additional HR Tasks Required = False

Add/Edit Rule

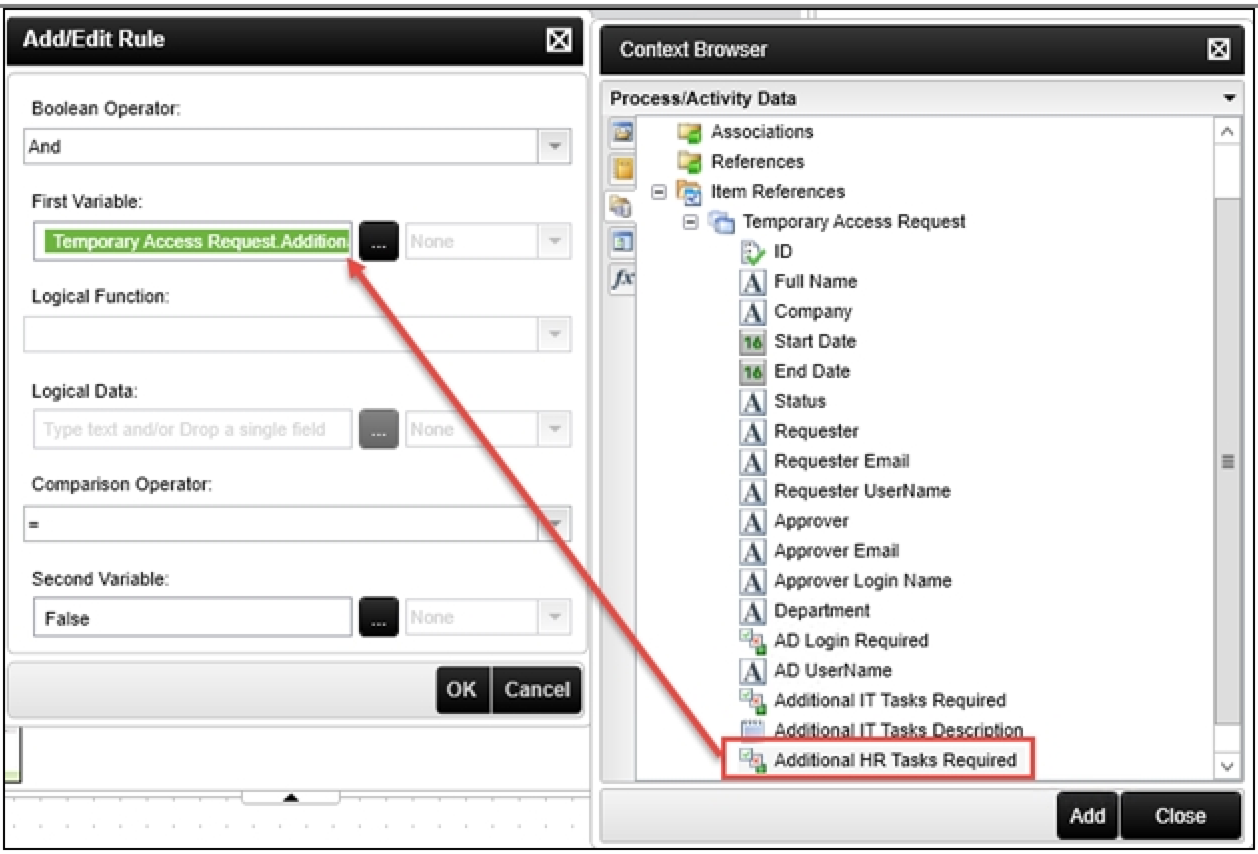
First Variable: Temporary Access Request.Addition

Logical Function:

Logical Data: Type text and/or Drop a single field

Comparison Operator: =

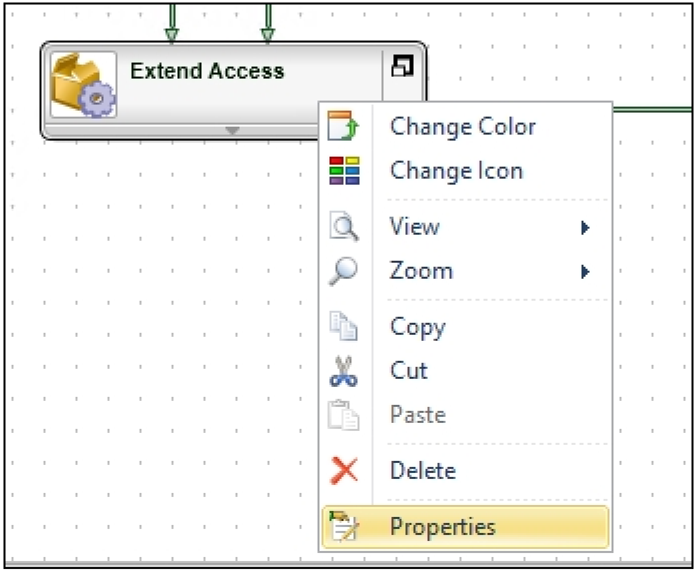
Second Variable: False



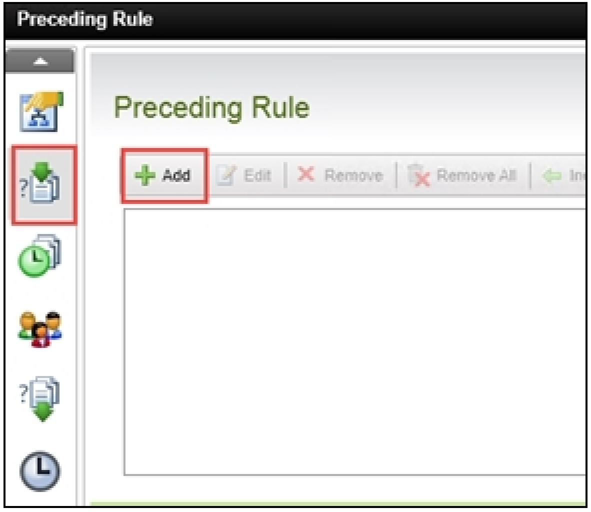
The new line you just added is necessary because the workflow would simply end after the Approved step if the user did not select either the Additional IT Tasks option nor the Additional HR Tasks option on the request form.

Next you will configure the Extend Access task to only run when *both* IT Tasks and HR Tasks are complete. This is a requirement when you want to implement a parallel path in a workflow, otherwise the Extend Access task would start once after the IT Tasks are complete and once again when the HR Tasks are complete. You will be adding a preceding rule to the Extend Access task so that it checks that both IT Tasks and HR Tasks are complete before the Extend Access task is started.

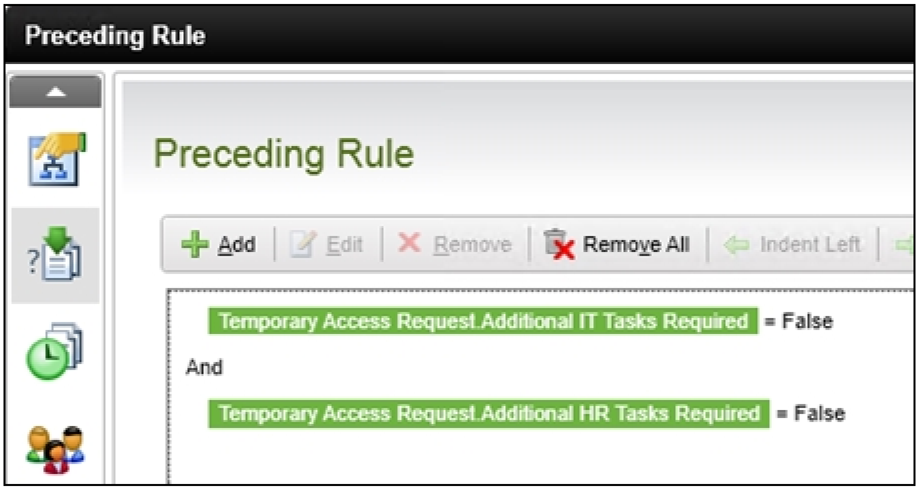
d. Right-click the **Extend Access** activity and select **Properties**.



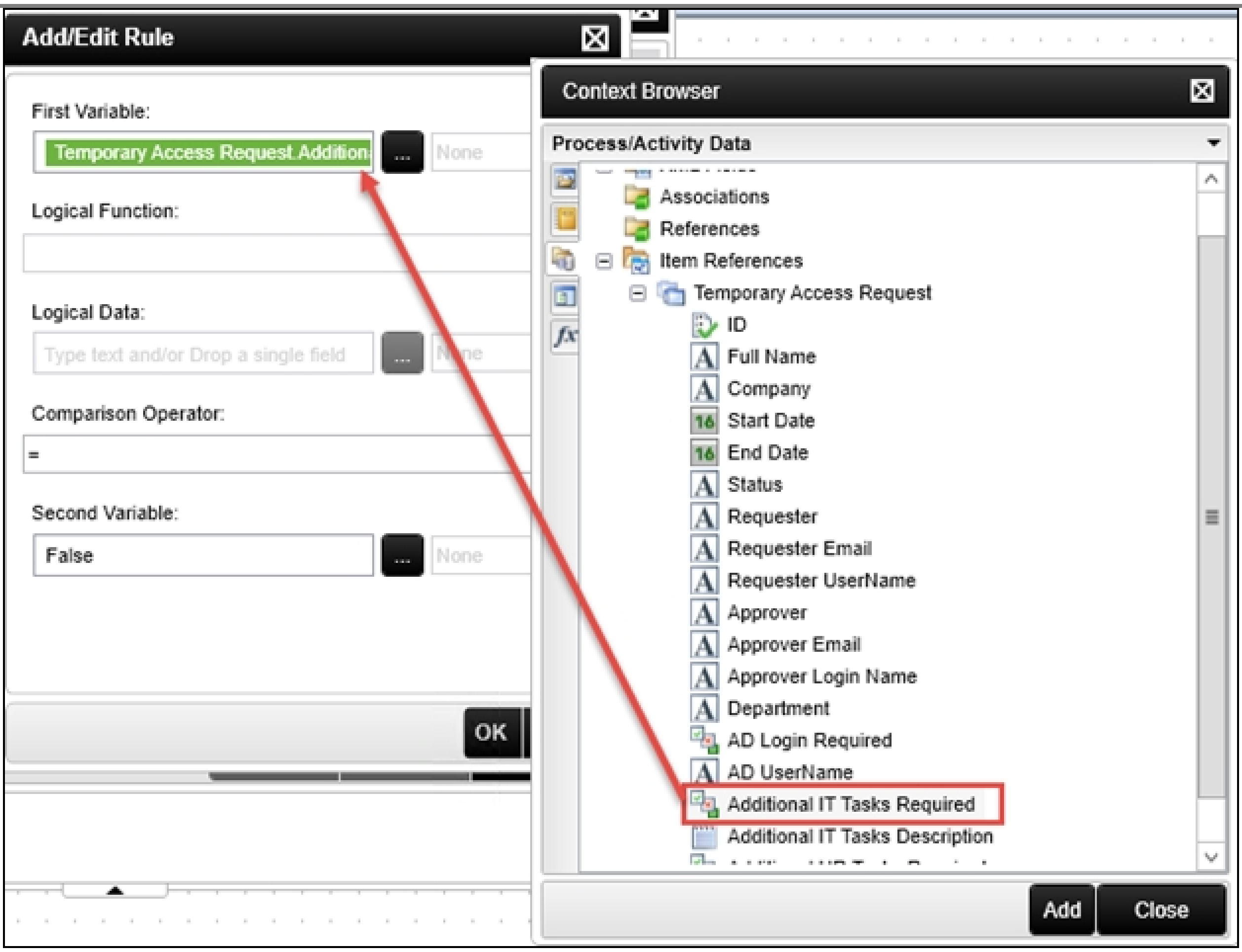
e. Select the **Preceding Rule** tab, then click **Add**.



f. Configure the rule as follows
First Variable: **Item Reference > Temporary Access Request > Additional HR Tasks Required**
Comparison Operator: **=**
Second Variable: *False*
Boolean Operator: **AND**
First Variable: **Item Reference > Temporary Access Request > Additional IT Tasks Required**
Comparison Operator: **=**
Second Variable: *False*
See the screen shots below for guidance. Click **Finish** to complete the rule definition.



Temporary Access Request.Additional IT Tasks Required = False
And
Temporary Access Request.Additional HR Tasks Required = False



Add/Edit Rule

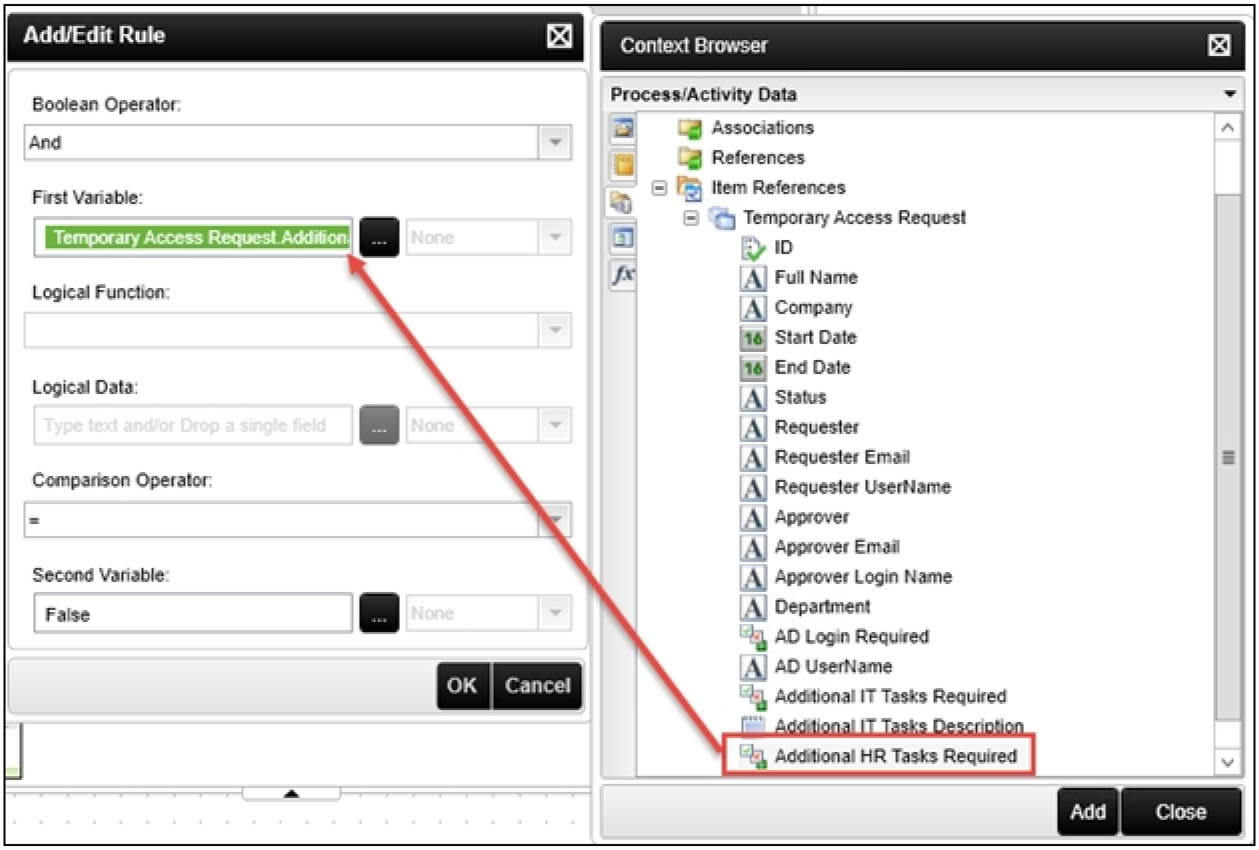
First Variable: Temporary Access Request.Addition

Logical Function:

Logical Data: Type text and/or Drop a single field

Comparison Operator: =

Second Variable: False



Add/Edit Rule

Boolean Operator: And

First Variable: Temporary Access Request.Addition

Logical Function:

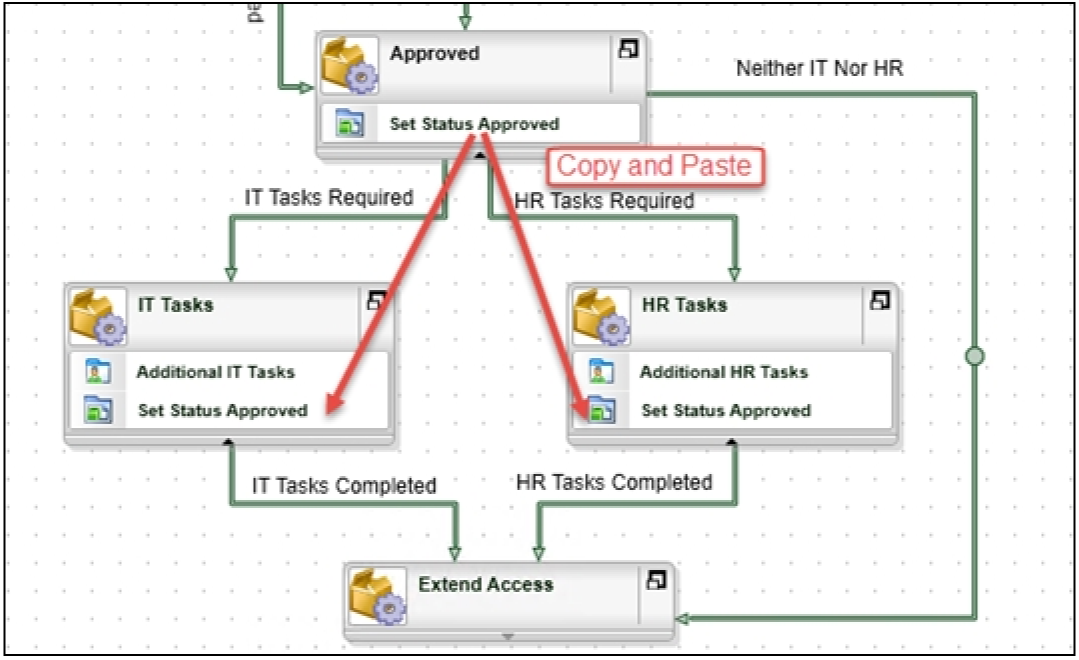
Logical Data: Type text and/or Drop a single field

Comparison Operator: =

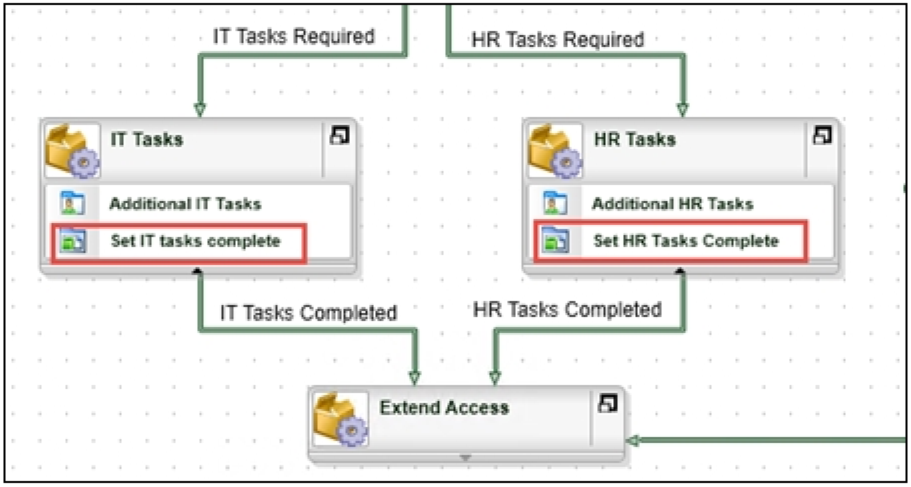
Second Variable: False

There is one more component to configuring successful parallel paths. When the IT Tasks are completed, you need to set the Additional IT Tasks back to false, and when the HR Tasks are completed, you need to set the Additional HR Tasks back to false as well. This essentially "tricks" K2 into thinking the two sets of tasks are not required, thus fulfilling the preceding rule you just set. But since you are making this setting *after* they have completed, the setting will not effect the tasks from running. You will do this by calling the Temporary Access Request SmartObject and updating those properties once the respective tasks are completed.

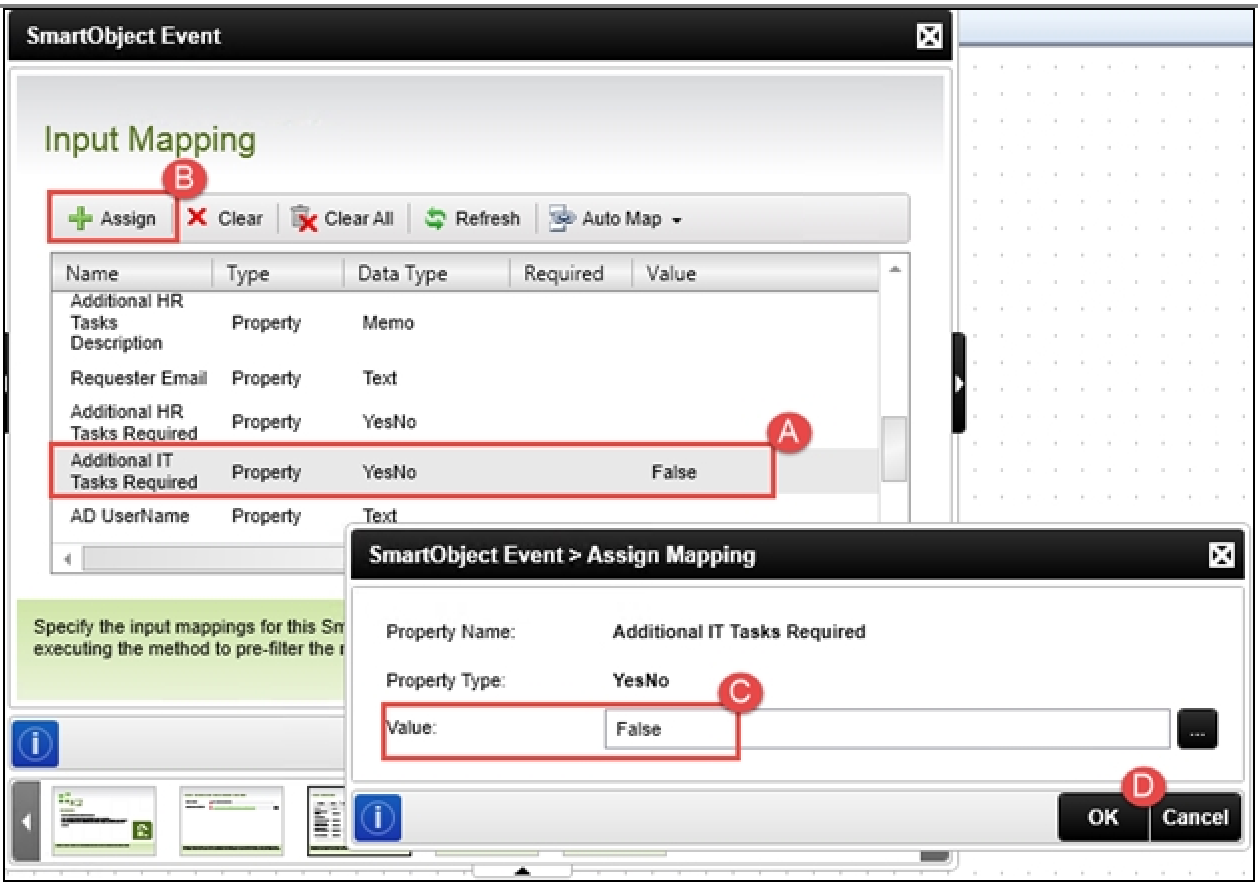
g. Copy and paste the **Set Status Approved** event into the IT Tasks and HR Tasks activities. Make sure the event is copied UNDERNEATH the existing user tasks, since you want these events to fire after the user tasks. (Once again, you are reusing workflow content that performs a similar task, to save time.)



h. Rename the events to
Set IT Tasks Complete
Set HR Tasks complete



i. Run the wizard for the copied **Set IT Tasks Complete** event, and set the value of the **Additional IT Tasks Required** property to
False
then click **Finish** to complete the wizard. (You can leave the other properties as is.)



Input Mapping

Name	Type	Data Type	Required	Value
Additional HR Tasks Description	Property	Memo		
Requester Email	Property	Text		
Additional HR Tasks Required	Property	YesNo		
Additional IT Tasks Required	Property	YesNo		False
AD UserName	Property	Text		

Specify the input mappings for this Sm… executing the method to pre-filter the r…

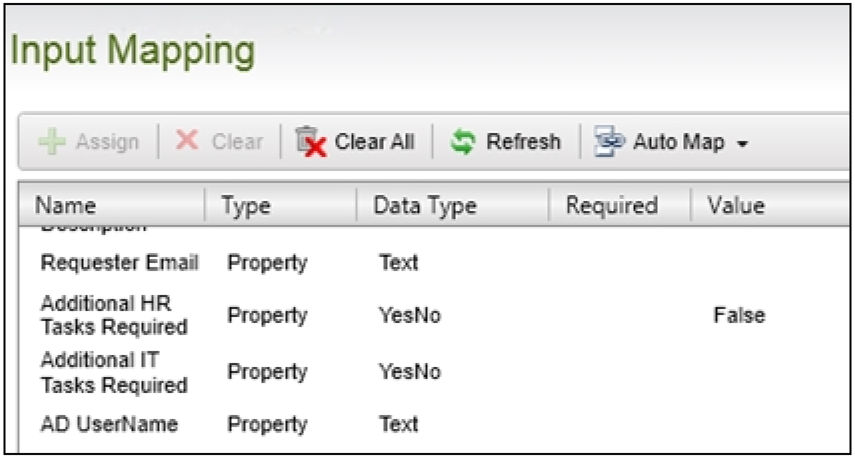
SmartObject Event > Assign Mapping

Property Name: Additional IT Tasks Required

Property Type: YesNo

Value: False

j. Run the wizard for the copied **Set HR Tasks Complete** event, and set the value of the **Additional HR Tasks Required** property to
False
then click **Finish** to complete the wizard. (You can leave the other properties as is.)



Input Mapping

Name	Type	Data Type	Required	Value
Requester Email	Property	Text		
Additional HR Tasks Required	Property	YesNo		False
Additional IT Tasks Required	Property	YesNo		
AD UserName	Property	Text		

k. **Save** your workflow.

Step 8 Review

In this step, you configured the **Extend Access** step to wait for both **IT Tasks** and **HR Tasks** to be completed before it starts. You accomplished this by setting the value of the **Additional IT Tasks Required** and **Additional HR Tasks Required** back to false after the respective tasks were completed. This will now allow for all combinations of IT Tasks and HR Tasks. What is important to note here is that for a parallel workflow that merges again later on, you need to define a *preceding rule* for the merging activity.

You also added a line that will be followed if neither IT nor HR Tasks are required. This way, the workflow will ultimately arrive at the **Extend Access** step regardless of the options selected by the requester.

Step 9: Add a client event with a start rule, an escalation and a loopback line

In this step, you will add a workflow step for the requester to decide whether or not the temporary access should be extended. (For example, a project may be lasting longer than anticipated and therefore the temporary user needs access for longer.) You want this task to start one day before the temporary access is due to expire to prevent adding a task to the requester's worklist until it is actually necessary to extend access. As a security measure, you also want this task to expire and the temporary access to expire if the requester forgets to action the task by one day after the end date of the access request.

Step 9 Tasks

1. Define a start rule for the **Extend Access** activity so that it starts one day before the **End Date**.
2. Add a SmartForms Client Event to the **Extend Access** and configure it as follows:

Field	Value
Name	*Extend Temporary Access*
Form	**Temporary Access Request Form**
State	Use existing **Extend** state
Configure Item Reference	(no changes, use default values)
Specify Form Parameters	(no changes, use default values)
Configure Open Worklist Item Rule	(no changes, use default values)
Transfer Workflow Data to Form	(no changes, use default values)
Actions	*Extend access* *Do not extend access*
Action Display Settings	(no changes, use default values)
Transfer Form Data to Workflow	(no changes, use default values)
Outcomes	(no changes, use default values)
Destination User	**Process Originator**
Task Notification	(Do not send a task notification message)

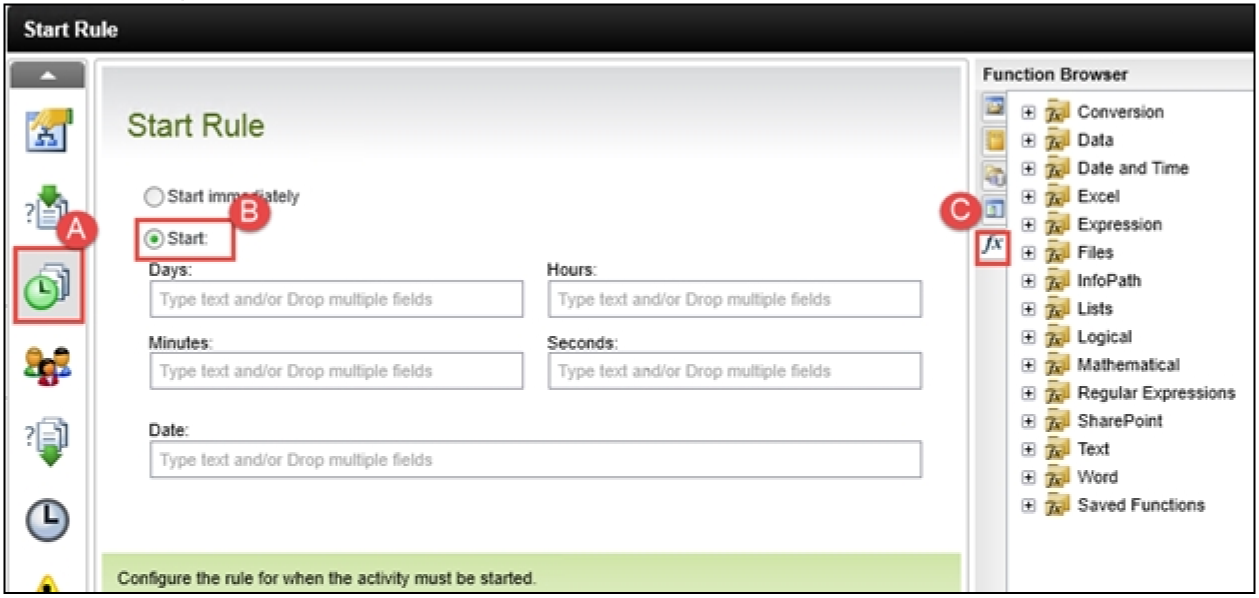
3. Configure the **Extend Access** line so that it loops back to the same **Extend Access** activity.
4. Add an activity called
   Set Status Expired
   and connect the **Do not extend access** line into this activity.
5. Add an escalation for the **Extend Access** activity that fires one day after the **End Date**, and configure it to **Go To** the **Set Status Expired** activity.

Step 9 Walkthrough

First you will add the rule that delays the start of the Extend Access step until one day before the Access End Date.

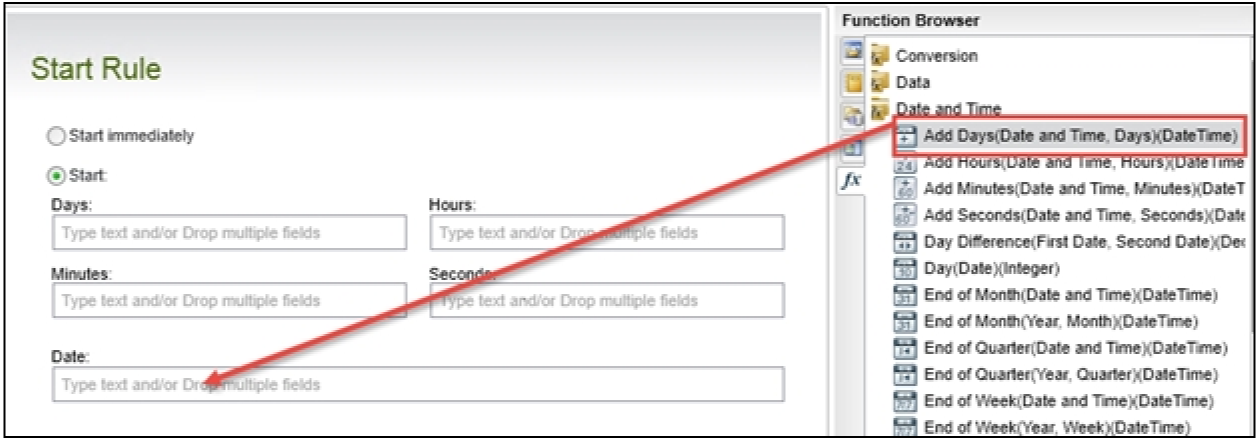
a. Right-click the **Extend Access** activity and select **Properties**.

b. Select the **Start Rule** tab, then select the **Start** option and expand the context browser on the right of the window. Open the **Inline Functions** tab.



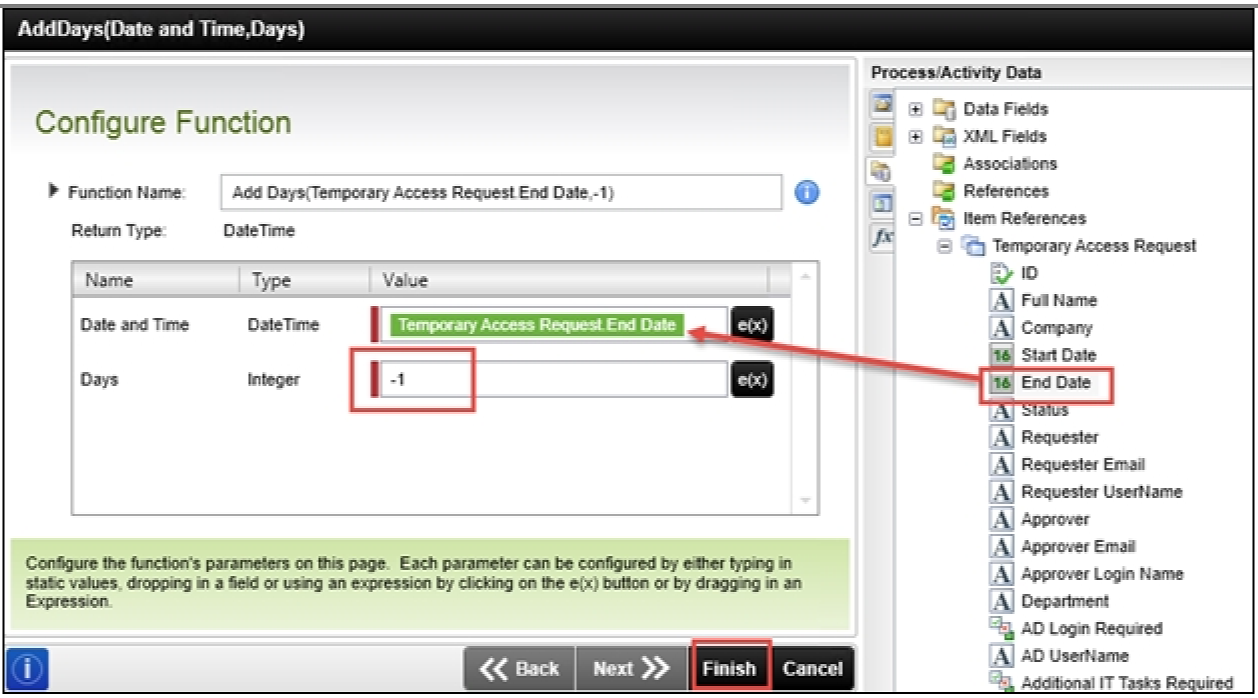
Start Rule
Start Rule
○ Start immediately
● Start:
A
B
C
Days:
Type text and/or Drop multiple fields
Hours:
Type text and/or Drop multiple fields
Minutes:
Type text and/or Drop multiple fields
Seconds:
Type text and/or Drop multiple fields
Date:
Type text and/or Drop multiple fields
Configure the rule for when the activity must be started.
Function Browser
Conversion
Data
Date and Time
Excel
Expression
Files
InfoPath
Lists
Logical
Mathematical
Regular Expressions
SharePoint
Text
Word
Saved Functions

c. Drag and drop the **Date and Time > Add Days** function into the **Date** textbox.



Start Rule
○ Start immediately
● Start:
Days:
Type text and/or Drop multiple fields
Hours:
Type text and/or Drop multiple fields
Minutes:
Type text and/or Drop multiple fields
Seconds:
Type text and/or Drop multiple fields
Date:
Type text and/or Drop multiple fields
Function Browser
Conversion
Data
Date and Time
Add Days(Date and Time, Days)(DateTime)
Add Hours(Date and Time, Hours)(DateTime
Add Minutes(Date and Time, Minutes)(DateT
Add Seconds(Date and Time, Seconds)(Date
Day Difference(First Date, Second Date)(De
Day(Date)(Integer)
End of Month(Date and Time)(DateTime)
End of Month(Year, Month)(DateTime)
End of Quarter(Date and Time)(DateTime)
End of Quarter(Year, Quarter)(DateTime)
End of Week(Date and Time)(DateTime)
End of Week(Year, Week)(DateTime)

d. For the Days value, type (minus 1)
-1
then drag the **Item References > Temporary Access Request > End Date** property into the **Date and Time Value** text box, then click **Finish**. Click **Finish** once again to close the Extend Access Properties wizard.



AddDays(Date and Time,Days)

Configure Function

Function Name: Add Days(Temporary Access Request.End Date,-1)

Return Type: DateTime

Name	Type	Value
Date and Time	DateTime	Temporary Access Request.End Date
Days	Integer	-1

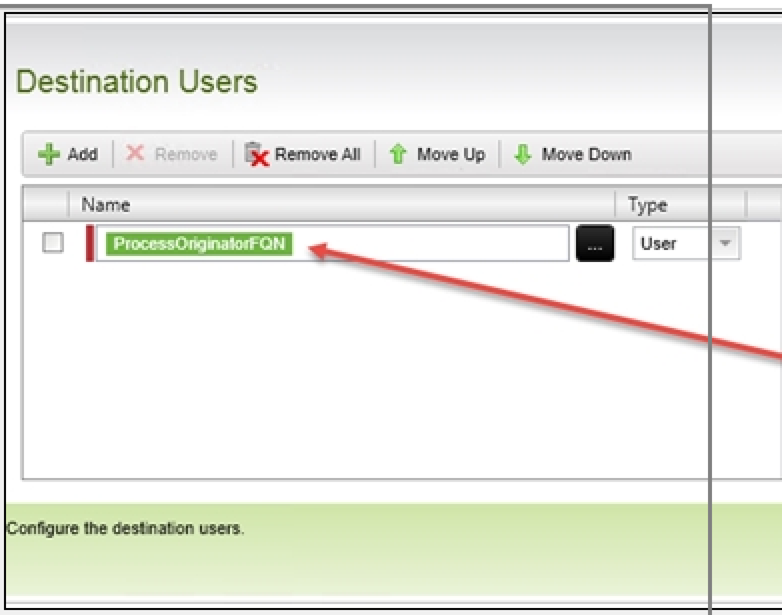
Configure the function's parameters on this page. Each parameter can be configured by either typing in static values, dropping in a field or using an expression by clicking on the e(x) button or by dragging in an Expression.

You want the Extend Access user task (that you will configure next) to fire off one day before the temporary access end date. The user task will be routed to the form originator, who must decide whether or not to extend the temporary access. Now you will add the user task and configure the lines that result from the user task.

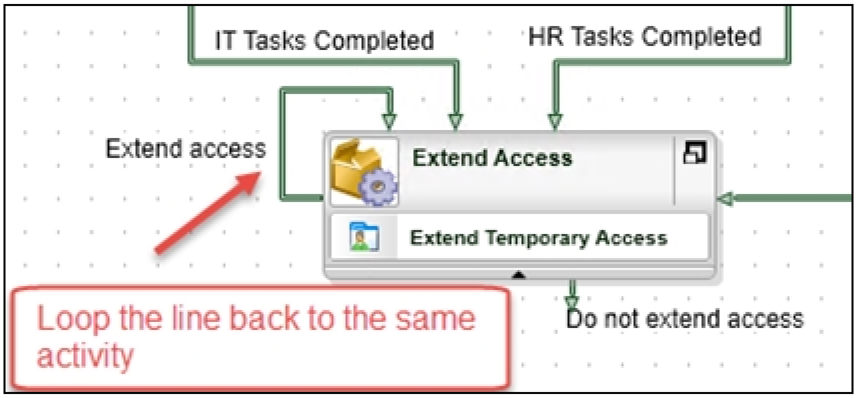
e. Drag and drop a **SmartForms Client event** into the **Extend Access** activity from the Events Wizards pane. Click **Next** on the welcome page.

f. As you did for the HR Tasks activity, you should be familiar enough with the SmartForms Client Event wizard screens to use the table below to configure the wizard:

Screen	Field	Value
Select Form and State	Name	*Extend Temporary Access*
	Form	**Temporary Access Request Form**
	State	Use existing **Extend** state
Configure Item Reference		(no changes, use default values)
Specify Form Parameters		(no changes, use default values)
Configure Open Worklist Item Rule		(no changes, use default values)
Transfer Workflow Data to Form		(no changes, use default values)
Configure Actions	Actions	*Extend access* *Do not extend access*
Action Display Settings		(no changes, use default values)
Transfer Form Data to Workflow		(no changes, use default values)
Outcomes		(no changes, use default values)
Destination User		Select the **Process Instance > Originator** from the Workflow Context tab

		 Destination Users Add · Remove · Remove All · Move Up · Move Down Name: ProcessOriginatorFQN · Type: User Configure the destination users.
Task Notification		(Do not send a task notification)

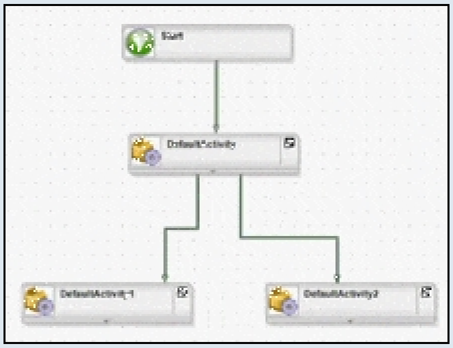
g. You should see two lines that flow out of the activity. Grab the arrow-head of the **Extend Access** line, and loop it back into the same activity. (It may take a few tries before the arrow will snap to the activity. Aim for the activity icon.) This step is creating a loop in the user task that will continue to loop until the user or system expires the event. If the originator chooses to extend the access, they will need to input a new end date. The workflow will then loop back and pause until one day before the new end date.



Loop the line back to the same activity

Note

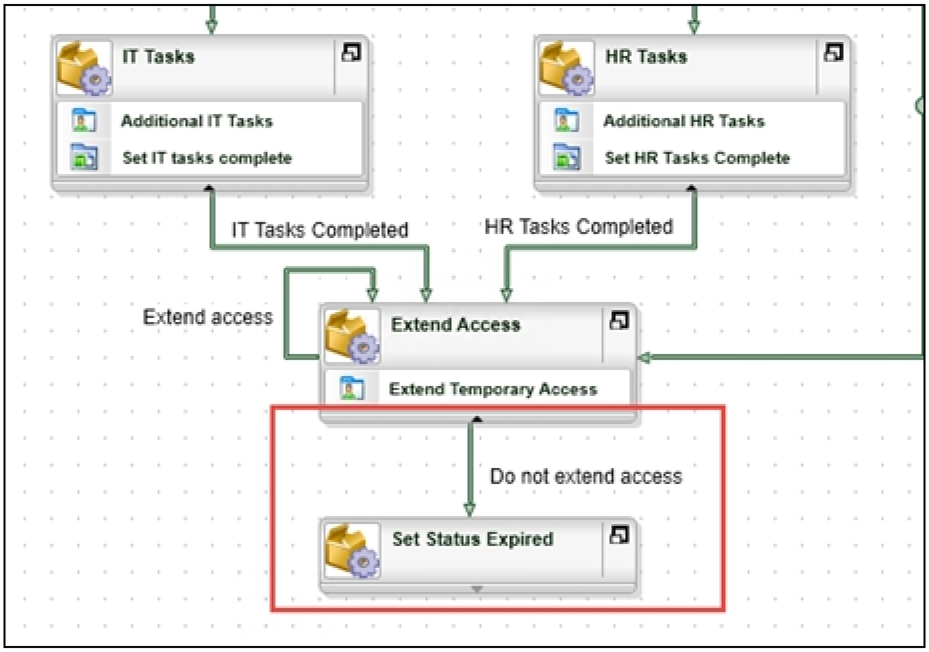
It can be tricky at first to get the loop-back line snapped into place. Grab the arrow-head of the line and drag it around the side of the activity and snap it to the top of the activity, near the icon. The animated .gif below may help:



Caution

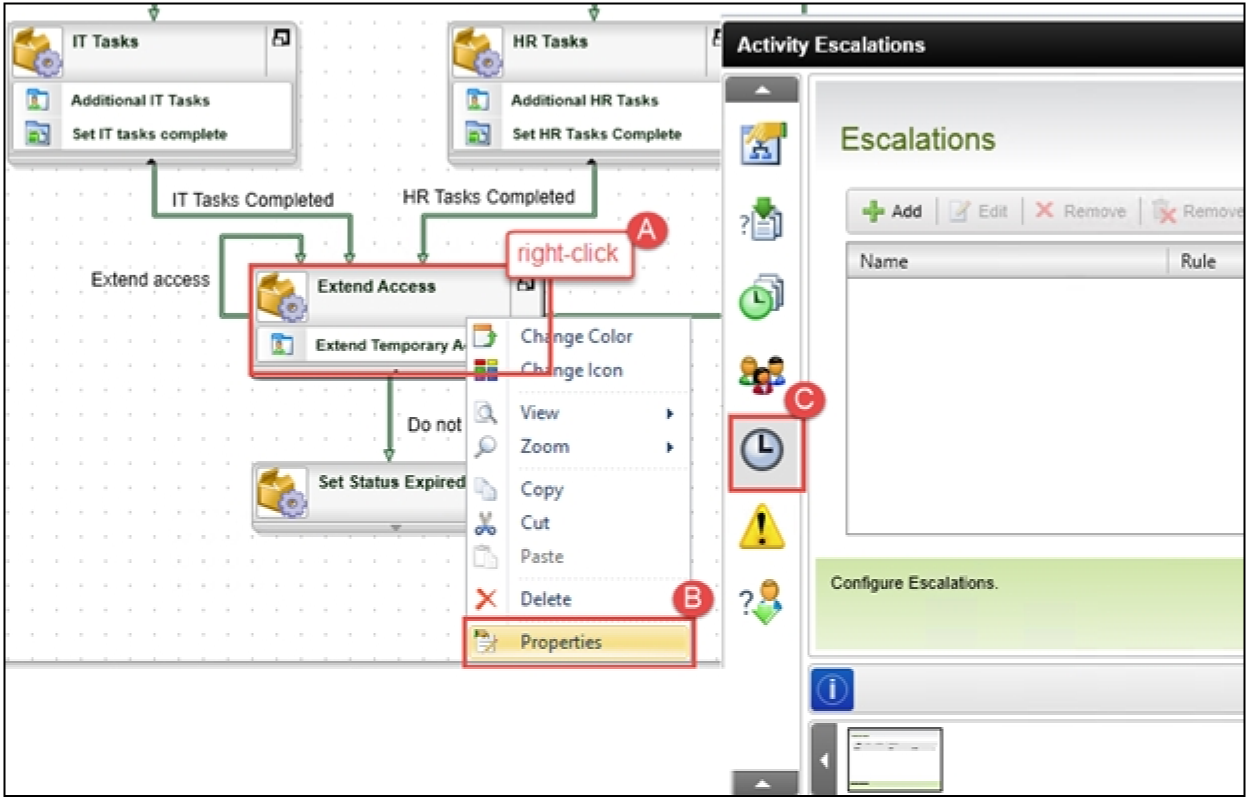
Do NOT draw lines that loop back to activities that contain only server events if those activities do not have a start date that increments somehow. Doing this will result in an unresponsive process at runtime and can affect the performance of your K2 environment, because your workflow could end up going into an infinite loop. You should only use this loop-back line approach with activities that contain user tasks (because the workflow will wait for the user to action their task) or with activities that have a start rule (because the start rule will delay the start of the activity until a certain date, which will prevent the infinite loop from putting your workflow into an infinite running state).

h. Add a new activity called
Set Status Expired
to the workflow and then connect the **Do not extend Access** line to this new activity.



i. Next you will define an escalation for the Extend Access step, so that the workflow will go to the Set Status Expired activity automatically one day after the end date of the access. This is a security measure, in case the requester forgets to action the task to extend the temporary access.

Right-click the **Extend Access** activity and select **Properties**, then open the **Escalations** tab.

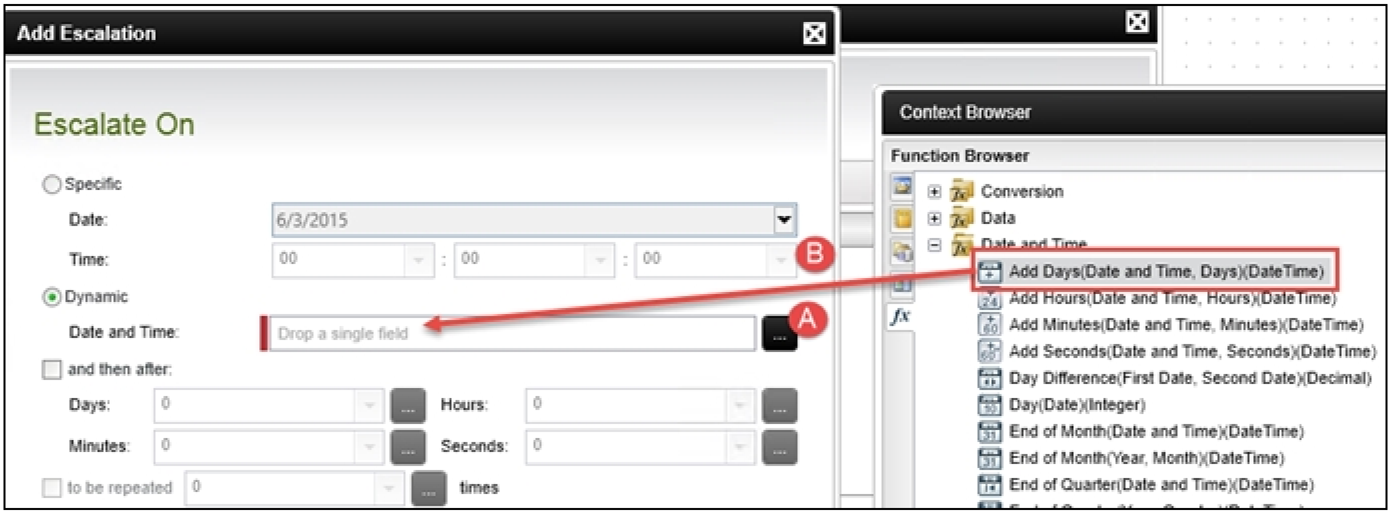


j. **Add** a new escalation and click **Next** on the welcome screen.

k. Name the escalation
Expire Access
and click **Next**.

l. On the **Rule Template** screen, select **Escalate On** and click **Next**.

m. You want the escalation to fire one day after the end date. On the **Escalate On** screen, select the **Dynamic** radio button, then use the ellipsis icon to drag and drop the **Add Days** inline function into the **Date and Time** text box.

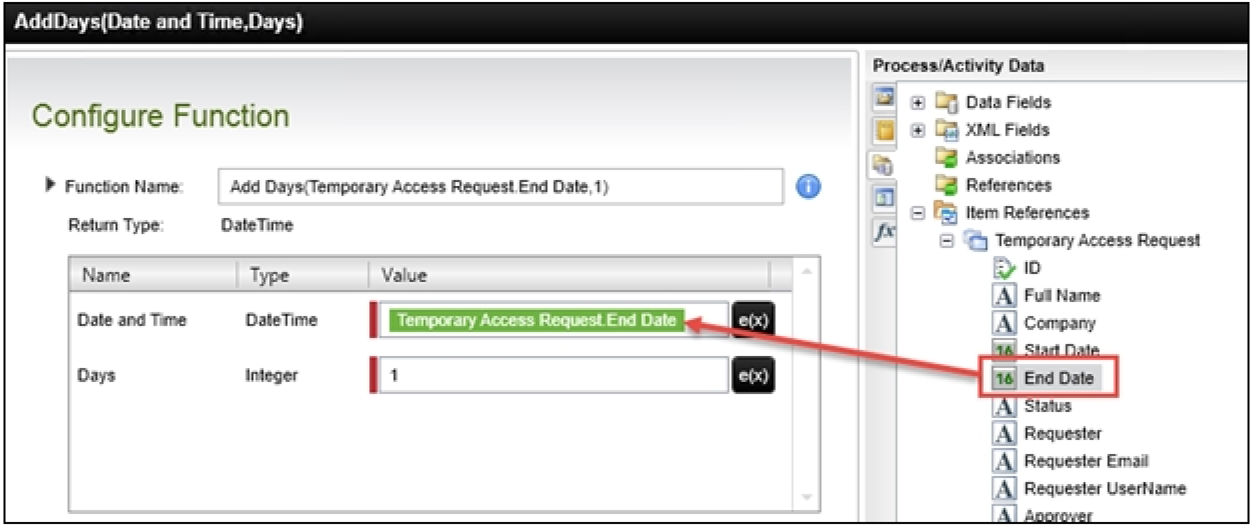

Add Escalation

Escalate On

○ Specific
Date: 6/3/2015
Time: 00 : 00 : 00
● Dynamic
Date and Time: Drop a single field
□ and then after:
Days: 0 Hours: 0
Minutes: 0 Seconds: 0
□ to be repeated 0 times

B
A
Context Browser
Function Browser
Conversion
Data
Date and Time
Add Days(Date and Time, Days)(DateTime)
Add Hours(Date and Time, Hours)(DateTime)
Add Minutes(Date and Time, Minutes)(DateTime)
Add Seconds(Date and Time, Seconds)(DateTime)
Day Difference(First Date, Second Date)(Decimal)
Day(Date)(Integer)
End of Month(Date and Time)(DateTime)
End of Month(Year, Month)(DateTime)
End of Quarter(Date and Time)(DateTime)

n. Configure the function to add one day to the **Item References > Temporary Access Request > End Date** property, then click **Finish**. In this step, you are telling K2 that if you reach the end date (day), the wait one day and expire the task if the requester did not extend the end date.



AddDays(Date and Time,Days)

Configure Function

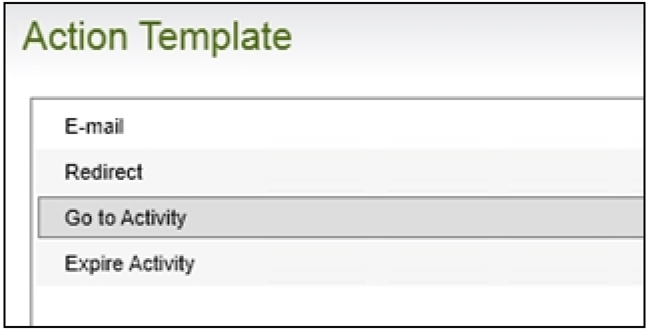
Function Name: Add Days(Temporary Access Request.End Date,1)

Return Type: DateTime

Name	Type	Value
Date and Time	DateTime	Temporary Access Request.End Date
Days	Integer	1

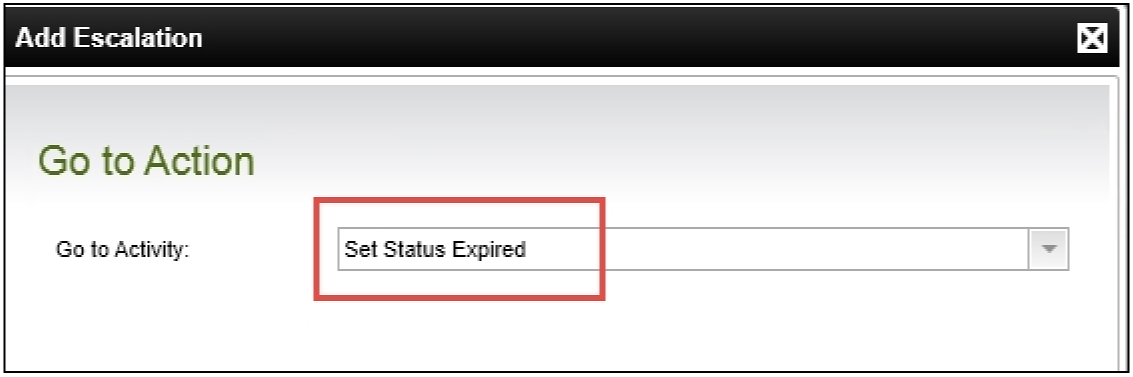
o. Click **Next** to go to the **Working Hours** screen and then select the **None** radio button. Click **Next**. (This escalation will not be associated with any working hours.)

p. For the **Action Template**, select **Go To Activity** and click **Next**.



Action Template

q. On the **Go to Action** screen, select **Set Status Expired**, then click **Next**. Click **Finish** to complete the Escalation wizard, then click **Finish** to close the activity properties page.



Add Escalation

Go to Action

Go to Activity: Set Status Expired

Step 9 Review

In this step you configured a user task with a start rule (so that the task is delayed until a specific date) and an outcome that loops back to the same activity (so that the task is restarted on a later date if the user decided to extend access to a future date). You also added an escalation that will expire the active task and go to another activity in the workflow if the user does not action the task in time. You used inline functions to calculate the start date and escalation date.

In your scenario, this step is used so that the requester can extend the temporary access if needed, but if they forget to do so, the system will automatically expire the temporary access one day after the end date.

Step 10: Add server tasks to complete the workflow

In this step you will add some additional server tasks that need to be performed at the end of the workflow. One task will update the status of the temporary access request and another will disable the AD account if one was created for the temporary user. You will define a conditional line that will check to see if an AD account was created.

Step 10 Tasks

1. Add an event in the **Set Status Expired** activity to set the **Status** of the request to *Expired*.
2. Add a **Disable AD Account** activity to the workflow and add an Active Directory Event and configure it to disable the AD account using the **Item References > Temporary Access Request > AD UserName** property and the **trim** inline function.
3. Add a line between the **Set Status Expired** activity and the **Disable AD Account** activity and configure the line rule to only execute if **Item References > Temporary Access Request > AD Login Required** property is *True*

Step 10 Walkthrough

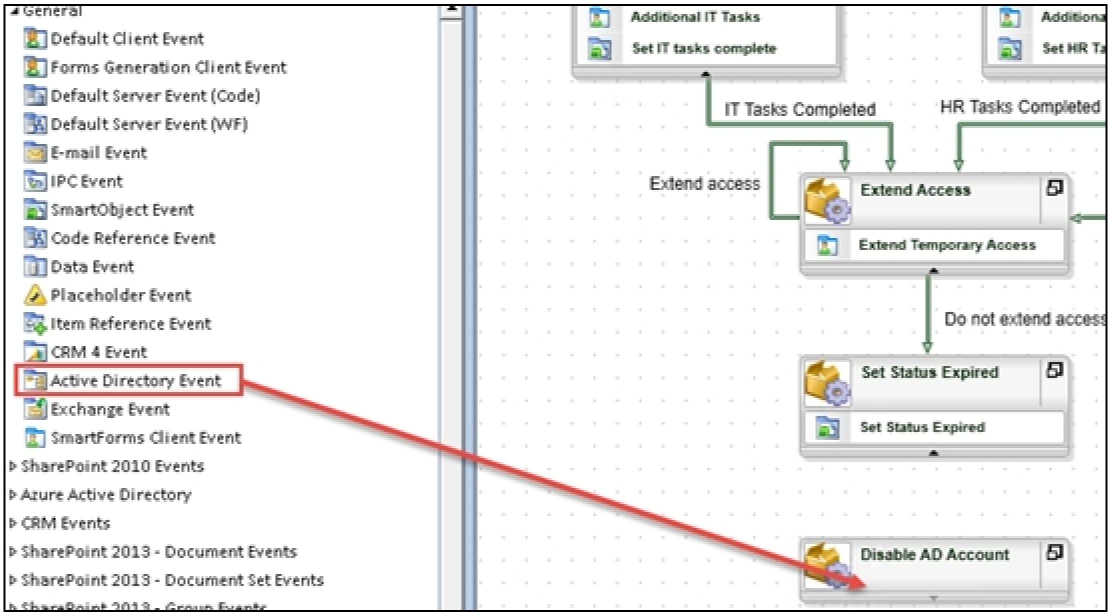
a. Copy and paste the **Set Status Approved** event into the **Set Status Expired** activity. Rename the event to *Set Status Expired*

b. Re-run the event wizard and configure the event to set the **Status** property to *Expired* **Finish** the wizard.

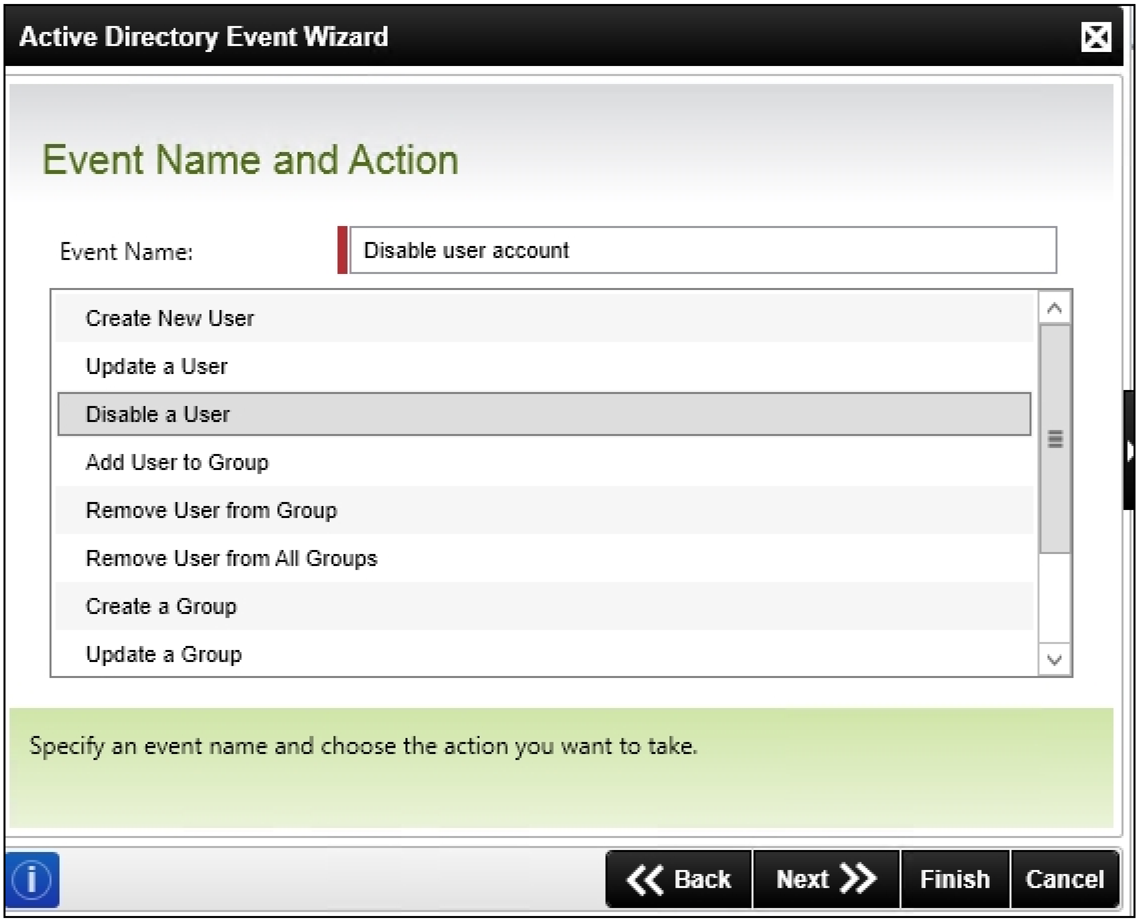
Now you will add an event that will disable the AD account if one was created.

c. Add a new activity to the workflow and name it *Disable AD Account*

d. Drag and drop an **Active Directory Event** into the **Disable AD Account** activity. Click **Next** on the welcome screen.



e. Name the event *Disable user account* then select the **Disable a User** action and click **Next**.



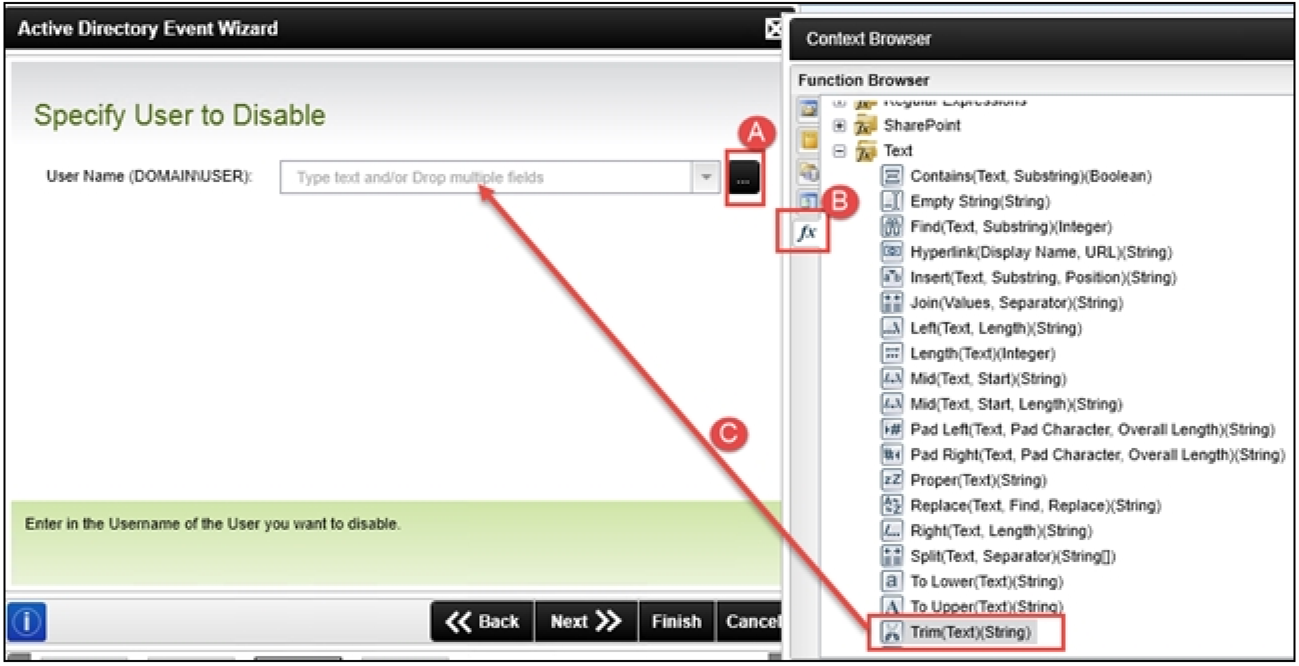
Active Directory Event Wizard

Event Name and Action

Event Name: Disable user account

Specify an event name and choose the action you want to take.

f. On the **Specify User to Disable** screen, drag and drop the **Trim** inline function into the **User Name** field. (Once again, you are using the trim in-line function so that any extra spaces are removed, as Active Directory can be particular when it comes to extra spaces.)



Active Directory Event Wizard

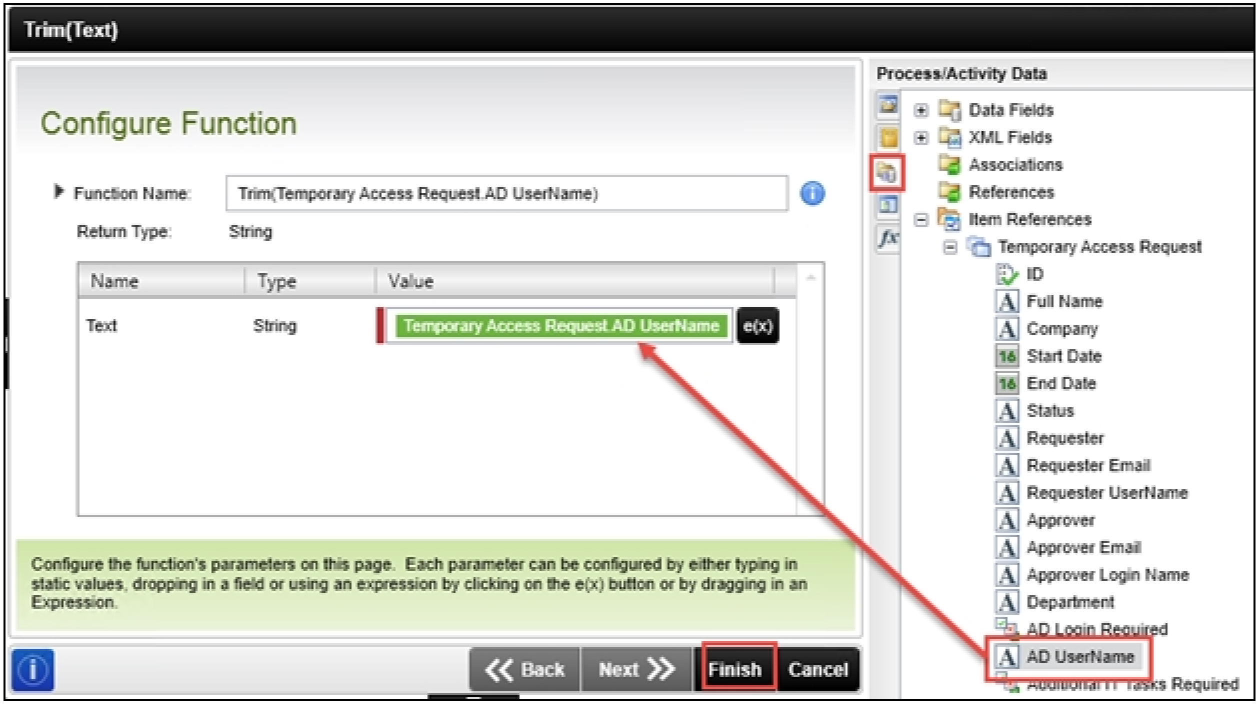
Specify User to Disable

User Name (DOMAIN\USER): Type text and/or Drop multiple fields

Enter in the Username of the User you want to disable.

A
B
C
Back
Next
Finish
Cancel
Context Browser
Function Browser
SharePoint
Text
Contains(Text, Substring)(Boolean)
Empty String(String)
Find(Text, Substring)(Integer)
Hyperlink(Display Name, URL)(String)
Insert(Text, Substring, Position)(String)
Join(Values, Separator)(String)
Left(Text, Length)(String)
Length(Text)(Integer)
Mid(Text, Start)(String)
Mid(Text, Start, Length)(String)
Pad Left(Text, Pad Character, Overall Length)(String)
Pad Right(Text, Pad Character, Overall Length)(String)
Proper(Text)(String)
Replace(Text, Find, Replace)(String)
Right(Text, Length)(String)
Split(Text, Separator)(String[])
To Lower(Text)(String)
To Upper(Text)(String)
Trim(Text)(String)

g. Configure the **Trim** function to use the **Item References > Temporary Access Request > AD User-Name** property, then click **Finish**.



Trim(Text)

Configure Function

Function Name: Trim(Temporary Access Request.AD UserName)

Return Type: String

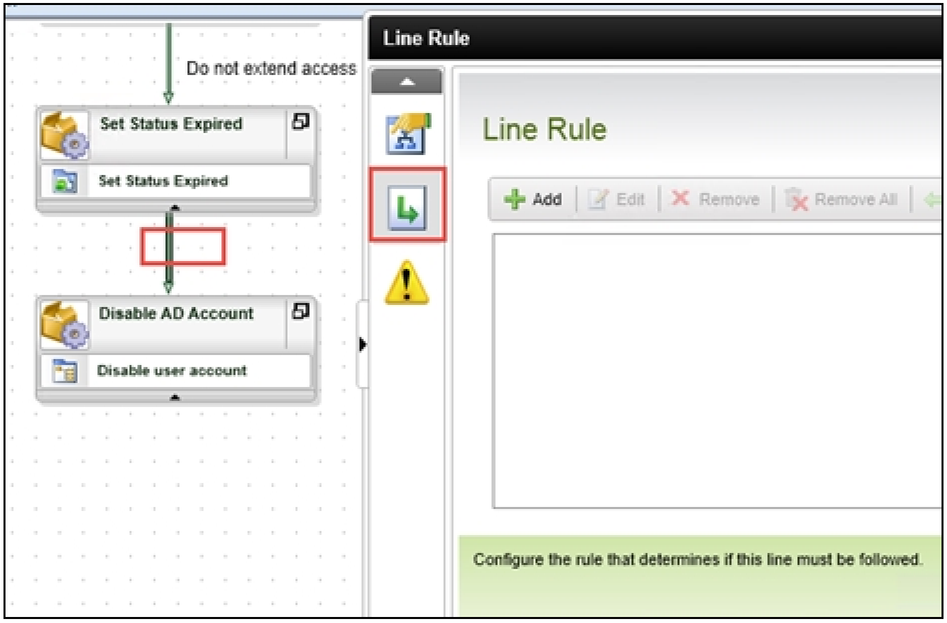
Name	Type	Value
Text	String	Temporary Access Request.AD UserName

Configure the function's parameters on this page. Each parameter can be configured by either typing in static values, dropping in a field or using an expression by clicking on the e(x) button or by dragging in an Expression.

h. Click **Finish** to complete the wizard.
In this step, you added an event that will disable the account of the temporary user, using the AD UserName property to find the correct account.
Next you will add a line to the workflow that will only go to this activity if an account was created for the temporary user.

i. Draw a line between the **Set Status Expired** activity and the **Disable AD Account** activity.

j. Right-click the new line and click the **Line Rule** tab from the line properties.

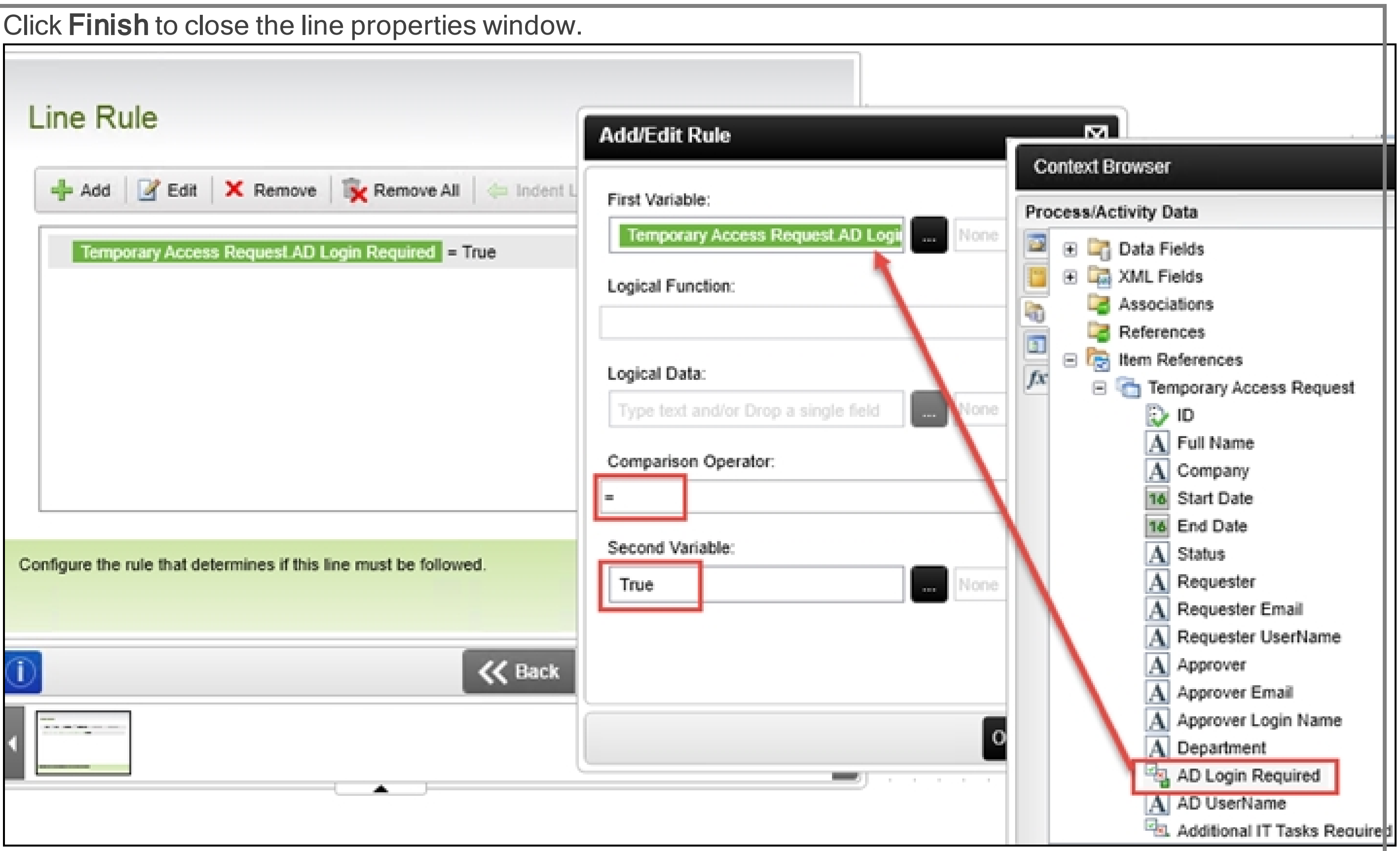


Line Rule

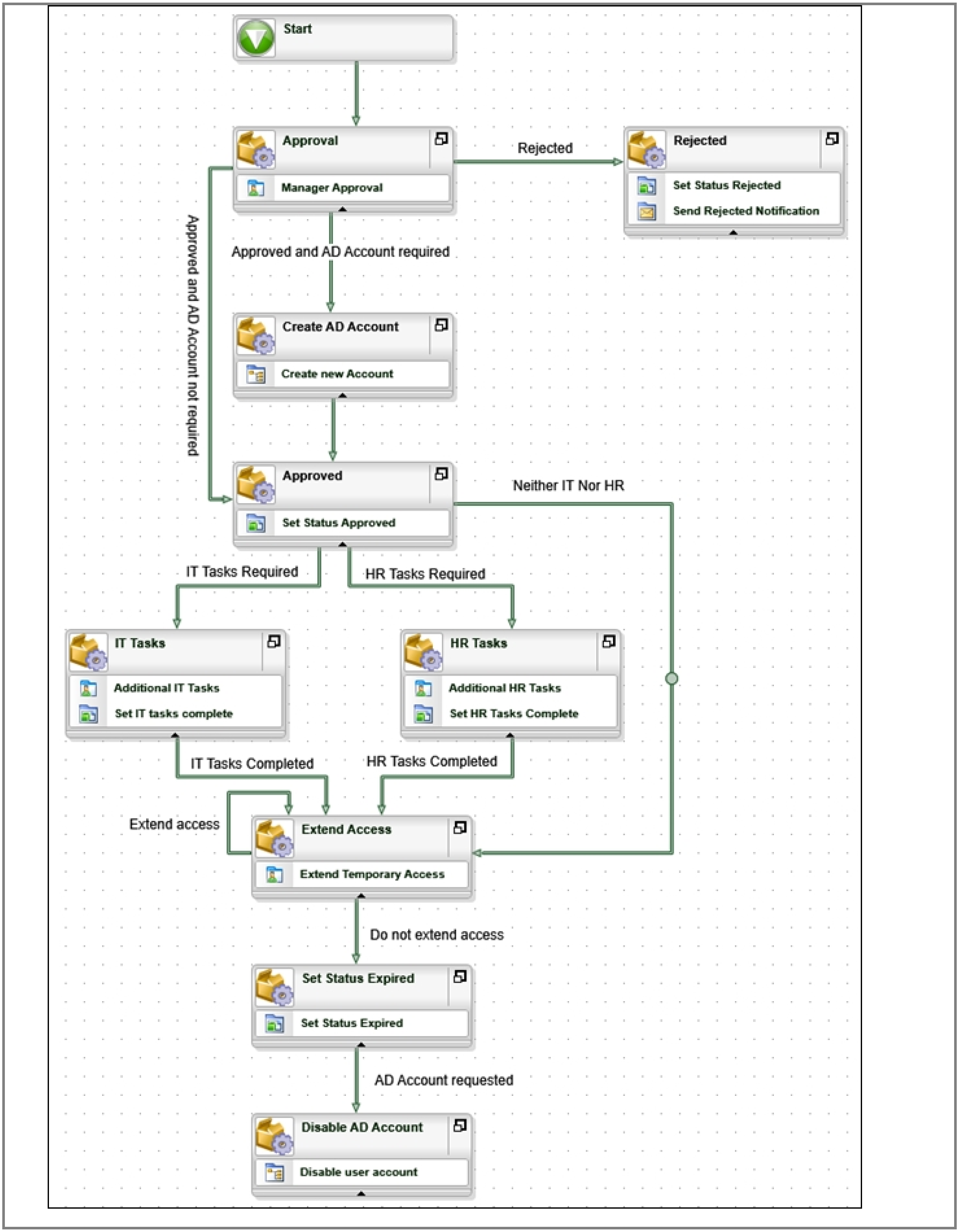
Configure the rule that determines if this line must be followed.

k. Add a new rule to the line and configure it so that the line is only followed if the **Item References > Temporary Access Request > AD Login Required** property is
True

Click **Finish** to close the line properties window.



l. Add a line label that indicates that the line will only be followed if the create account value was true. Your workflow should look like the image below.


Start
Approval
Manager Approval
Rejected
Rejected
Set Status Rejected
Send Rejected Notification
Approved and AD Account required
Approved and AD Account not required
Create AD Account
Create new Account
Approved
Set Status Approved
Neither IT Nor HR
IT Tasks Required
HR Tasks Required
IT Tasks
Additional IT Tasks
Set IT tasks complete
HR Tasks
Additional HR Tasks
Set HR Tasks Complete
IT Tasks Completed
HR Tasks Completed
Extend access
Extend Access
Extend Temporary Access
Do not extend access
Set Status Expired
Set Status Expired
AD Account requested
Disable AD Account
Disable user account

m. **Save** your workflow

Step 10 Review

In this step you added a SmartObject method to set the status of the request to expired. You also added a new event to disable the AD account and configured a line rule so that the account is only disabled if an account was created in the first place.

Step 11: Format the workflow with color-coding and align (OPTIONAL STEP)

Note
This is an optional step. Formatting the workflow will not change the behavior of the workflow. If you are short on time, skip ahead to Step 12.

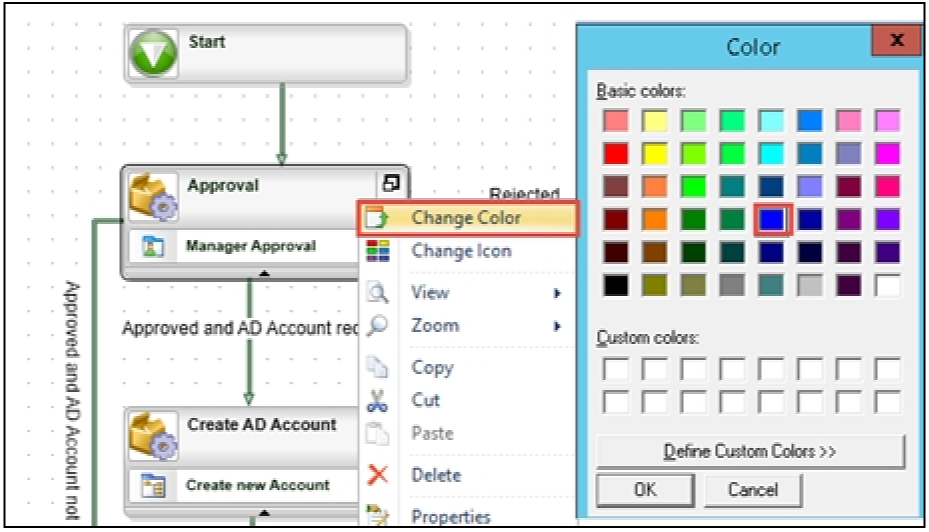
Using colors in your workflow can help you visually distinguish user tasks from system tasks and outcome line actions. This is especially useful for larger processes.

Step 11 Tasks

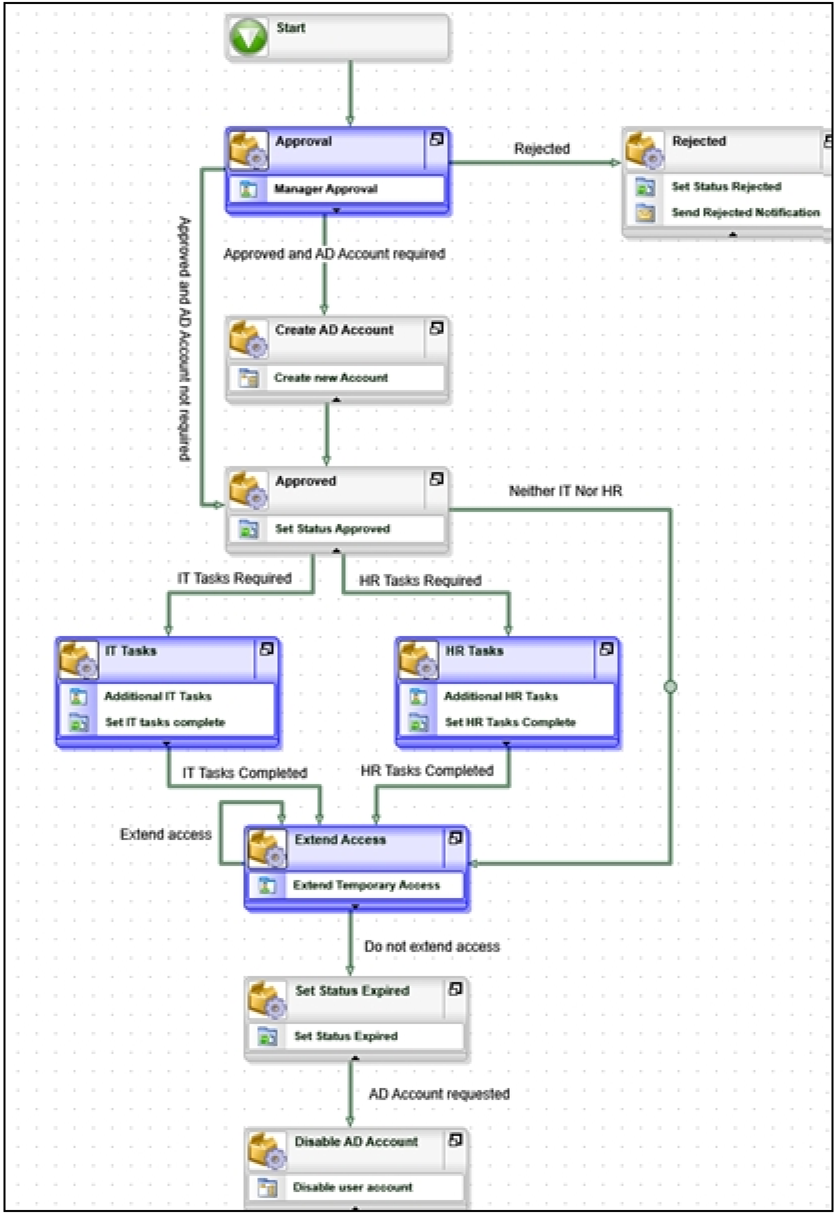
1. Change the activity colors for activities that contain user tasks to blue.
2. Change line colors as follows
   Rejected - red
   Looping Extend access - orange
   Parallel paths - purple
3. Align activities and move them around so that the workflow is easy to read. See the screenshot in the walkthrough for a sample layout.

Step 11 Walkthrough

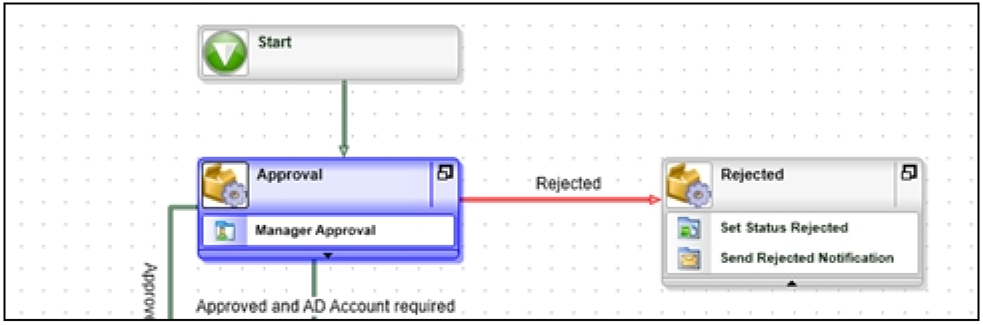
a. First you will change the color of all user activities to blue, so that you can easily distinguish activities that require user input from activities that are performed by the K2 server only. Right-click the **Approval** activity and select the **Change Color** option. Change the color of the activity to blue.



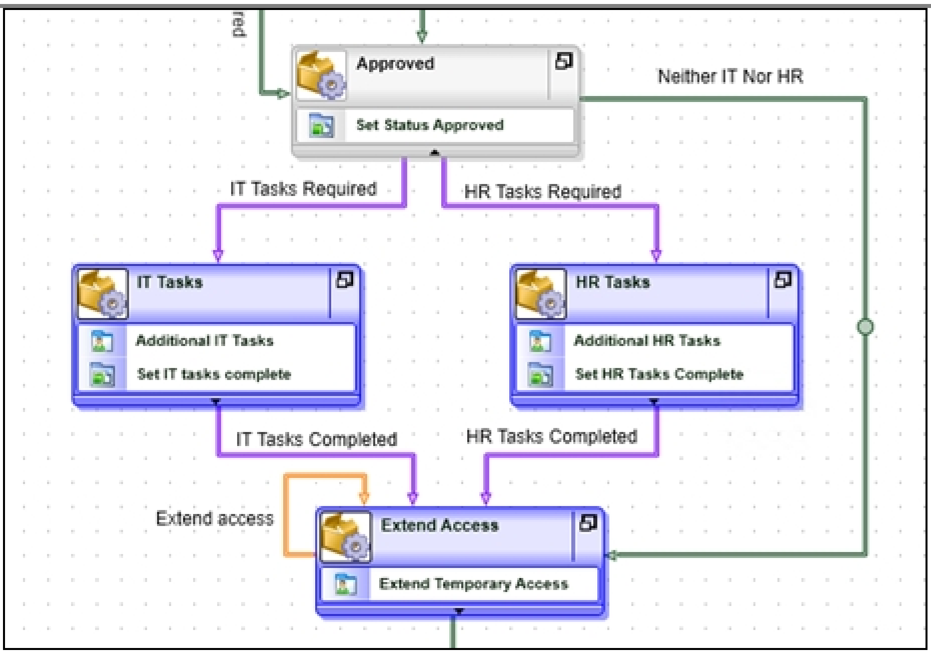
b. Use the same approach to change the color of the IT Tasks, HR Tasks and Extend Access activities to blue. Now it is easy to see which tasks in the workflow require user action.



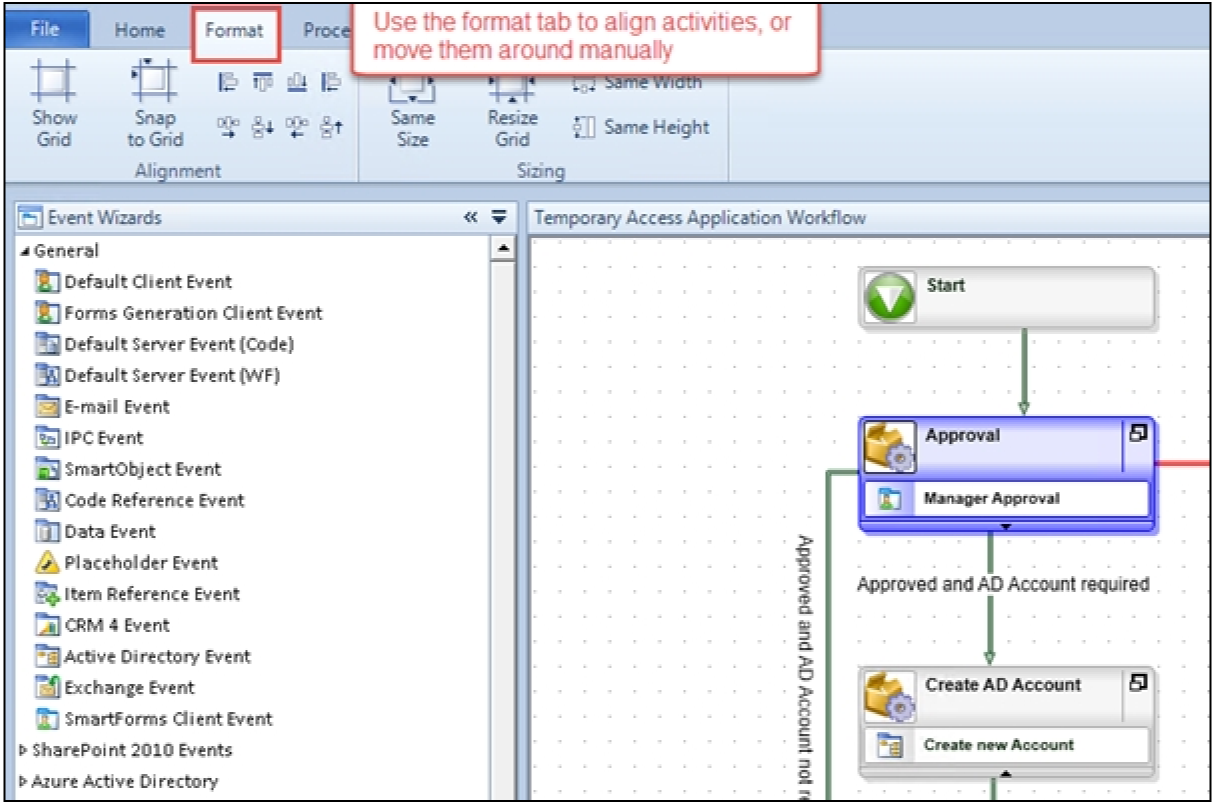
c. You can also use color-coding to illustrate the different paths that may be followed (for example, to illustrate the exception path of a workflow). Right-click the **Rejected** line and change its color to red to indicate that this is the exception path.



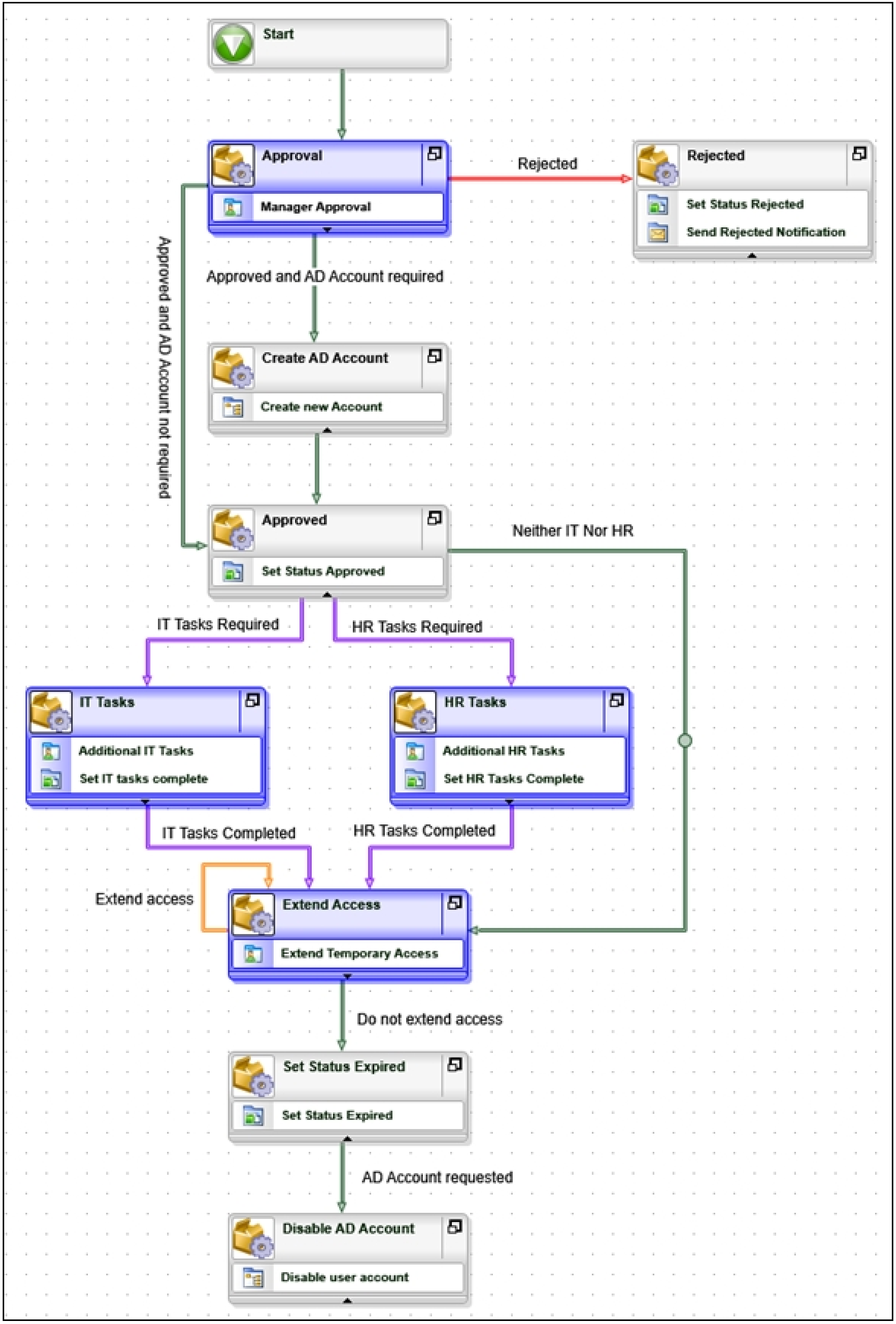
d. Change the color of the loop-back **Extend Access** line to orange to indicate that this is an optional path, then change the color of the **IT Tasks Required** and **HR Tasks required** lines to purple to indicate that they are parallel paths.



If necessary, move activities around and use the alignment functions so that activities are nicely aligned and separated from one another. Try not to have any activities overlap, and try to lay the workflow out so that lines do not cross over each other. It won't affect the operation of the workflow if they do, but following these guidelines just makes for an easy-to-understand workflow.


Use the format tab to align activities, or move them around manually

An example of how you might organize your workflow.



Step 11 Review

In this step, you just explored some formatting options for workflows. Applying some consistent and logical formatting language to a workflow will make it easier to understand visually.

Step 12: Configure process properties and deploy the workflow

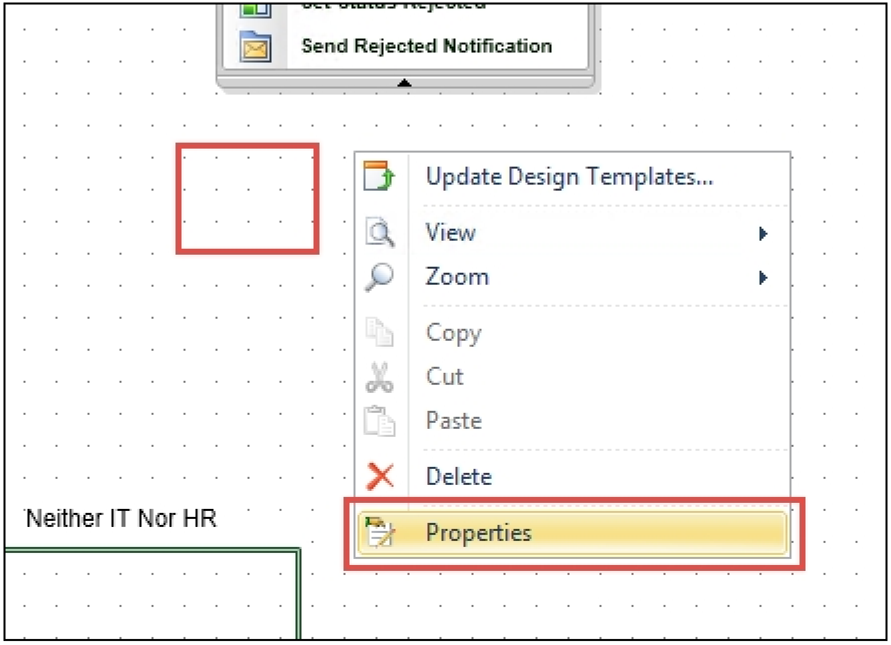
In this step, you will briefly explore the process properties, deploy the workflow to your K2 environment and then give the users in your organization permission to start and report on the workflow.

Step 12 Tasks

1. Confirm that the **Create Workflow Reporting SmartObjects** option is selected.
2. Save the workflow.
3. Deploy the workflow. (If the deployment does not complete within 5 minutes, close K2 Studio, re-open the project, save your workflow and try to deploy again.)
4. Give domain users **start** and **view** rights to the workflow (from the K2 Management site).

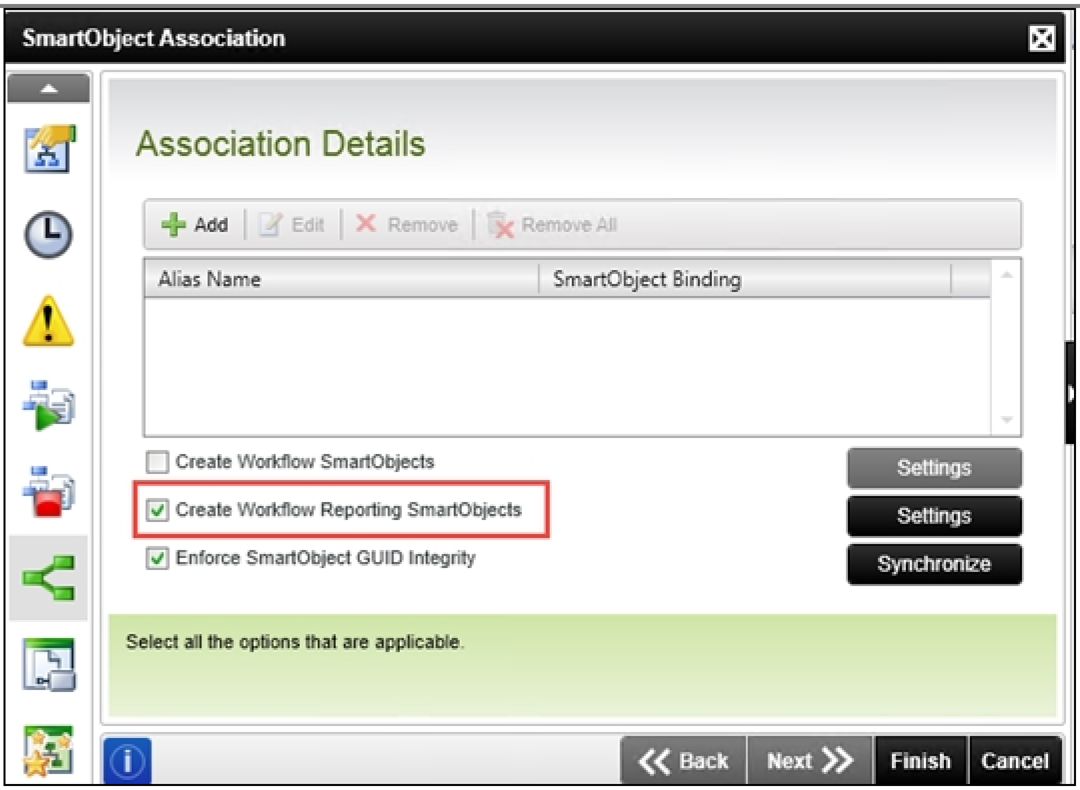
Step 12 Walkthrough

a. First you will make sure that the workflow will generate *reporting* SmartObjects when it is deployed.
You will be using these SmartObjects in Part 4 of this tutorial.
Right-click anywhere in the canvas of the workflow and select **Properties**.



b. Open the **Association** tab and confirm that the **Create Workflow Reporting SmartObjects** option is selected. If it is not selected, CHECK the box to select it, then click the **Settings** button to the right of that line. CHECK all three options (**Process**, **Activities**, **Events**) then click **OK**. Click **Finish**.

This setting tells K2 whether or not to gather reporting data for each workflow process and process instance. If you select this option, K2 will create reporting SmartObjects specific to this workflow. If you know for certain you don't need reporting data on a process, you can turn this option off, to keep server performance optimized.



Association Details

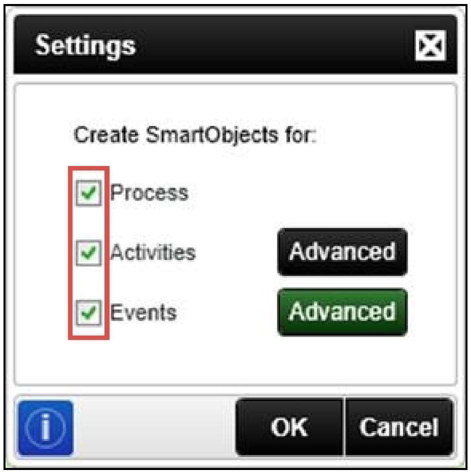
Alias Name	SmartObject Binding

□ Create Workflow SmartObjects

☑ Create Workflow Reporting SmartObjects

☑ Enforce SmartObject GUID Integrity

Select all the options that are applicable.



Create SmartObjects for:

☑ Process

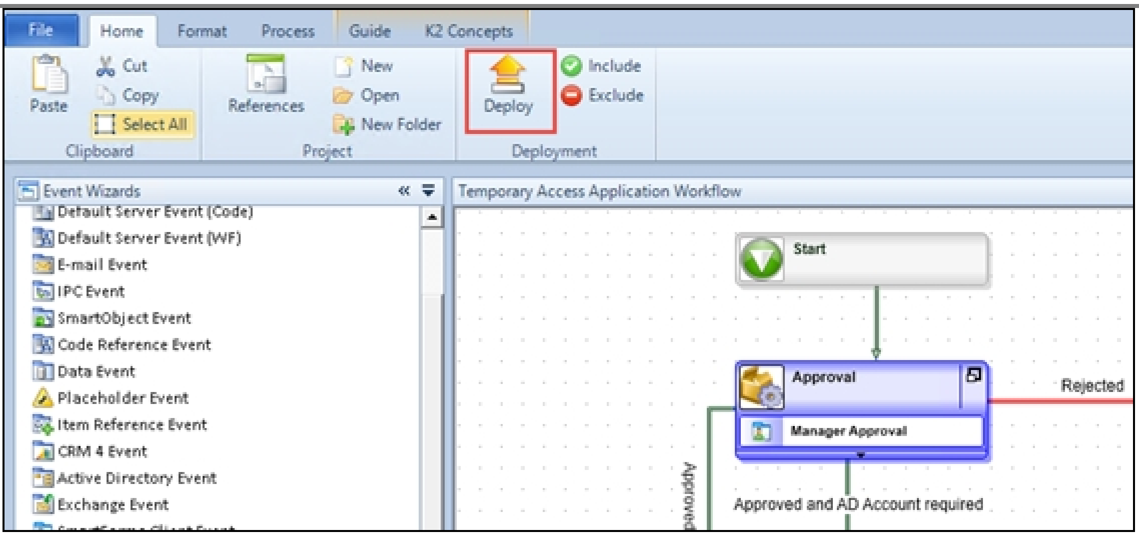
☑ Activities

☑ Events

c. Save your workflow using the **File** > **Save** command

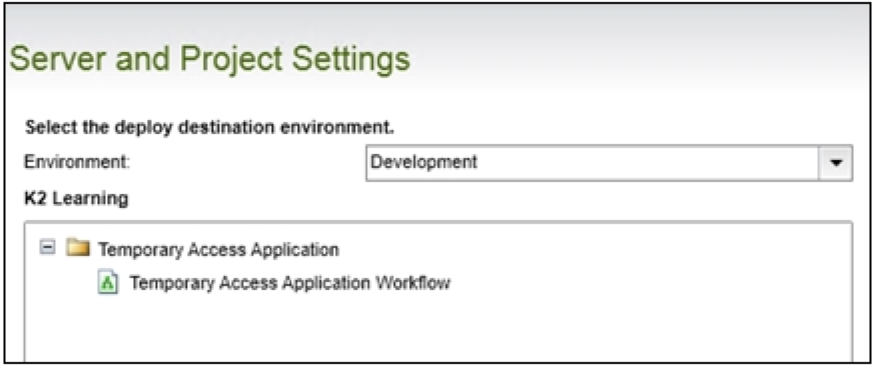
d. From the Deployment pane in the ribbon bar, select **Deploy**. (If this operation appears to hang for more than 5 minutes, close K2 Studio, re-open the project and try to deploy again).

Note
Deploying a workflow publishes it and makes it available to your users. If you make a change to a workflow, you must then redeploy so that your changes are exposed to your users. One word of caution however, deploying a workflow does not update previous instances of that workflow. In other words, any workflows that are currently active, will continue to use the workflow version they were on when they started. Depending on the changes you make to a workflow, you might consider allowing any active processes to complete prior to redeploying a new workflow version, especially if the updates included updates to the Forms or SmartObjects used in the workflow. .



e. Wait for the project compilation to finish and then on the deployment wizard welcome screen click **Next**.

f. In the **Server and Project Settings** screen, select **Development** and click **Next**.



Server and Project Settings

Select the deploy destination environment.

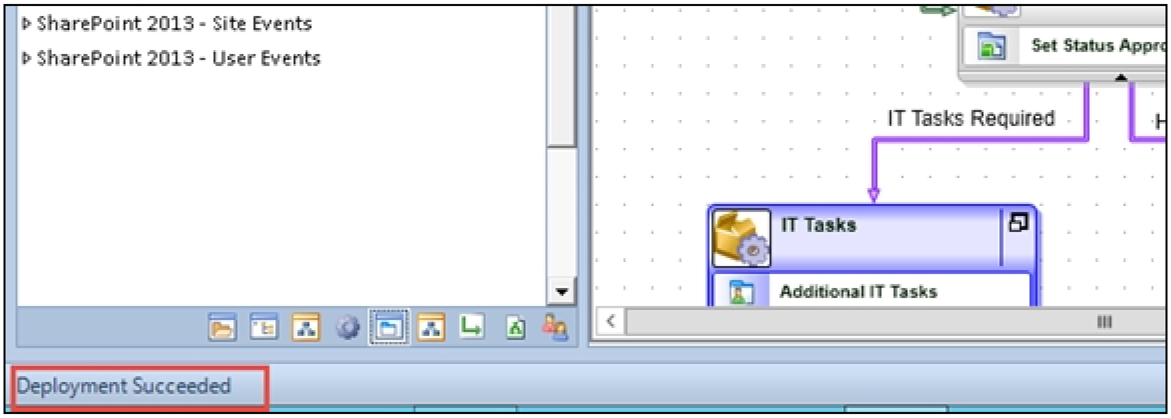
Environment: Development

K2 Learning

- Temporary Access Application
  - Temporary Access Application Workflow

g. Click **Finish** to start deploying the workflow. It may take a few seconds for the workflow to finish deploying. Once the deployment is done, you should see a **Deployment Succeeded** message in the status bar.

Note
If the deployment does not complete within 5 minutes, close K2 Studio (you may need to use windows task manager to stop the K2Studio.exe process), re-open the project, save your workflow and try to deploy again.

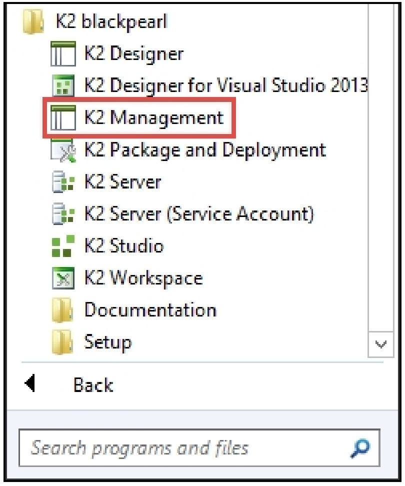


Deployment Succeeded

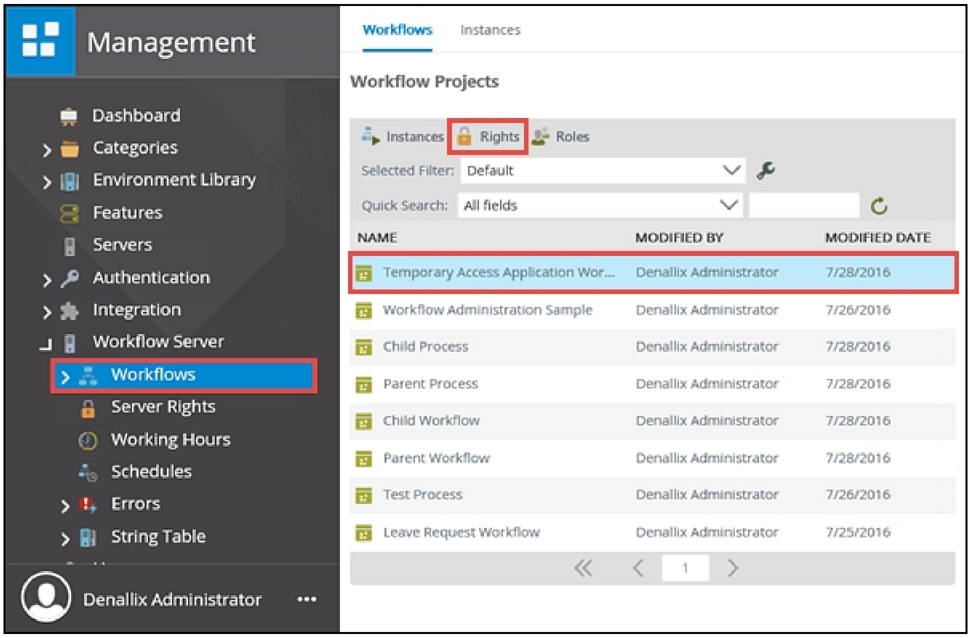
h. **Save** your workflow.

i. Next you will give all of the users in your domain rights to start the workflow and view the workflow reports.
Launch the K2 Management site (**Start > All Programs > K2 blackpearl > K2 Management**).

(If the K2 Management site link is not in your menu, ask your K2 administrator for the URL to the site. If you do not have administrator rights on your K2 environment, your K2 administrator will need to assign the process rights described below.)


K2 blackpearl
K2 Designer
K2 Designer for Visual Studio 2013
K2 Management
K2 Package and Deployment
K2 Server
K2 Server (Service Account)
K2 Studio
K2 Workspace
Documentation
Setup
Back
Search programs and files

j. From the Management menu, expand the **Workflow Server** node, then click on **Workflows**. All of the current workflow projects will be displayed in the central pane. Locate the **Temporary Access Application Workflow** project, then click it so that it is highlighted. Click the **Rights** button in the top toolbar.



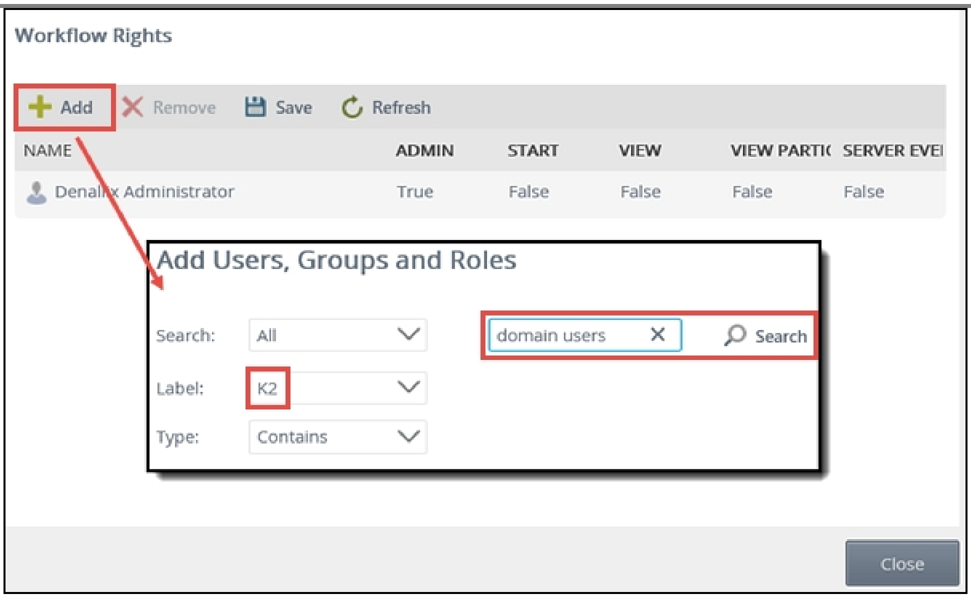
Selected Filter: Default

Quick Search: All fields

NAME	MODIFIED BY	MODIFIED DATE
Temporary Access Application Wor...	Denallix Administrator	7/28/2016
Workflow Administration Sample	Denallix Administrator	7/26/2016
Child Process	Denallix Administrator	7/28/2016
Parent Process	Denallix Administrator	7/28/2016
Child Workflow	Denallix Administrator	7/28/2016
Parent Workflow	Denallix Administrator	7/28/2016
Test Process	Denallix Administrator	7/26/2016
Leave Request Workflow	Denallix Administrator	7/25/2016

k. On the Workflow Rights screen, notice that the deploying user already has ADMIN rights. K2 assigns the admin process rights automatically to the user that deploys the workflow. Click **Add**.

l. On the **Add Users, Groups and Roles** screen, change the **Label** to **K2** (if necessary), then enter
domain users
into the search text box and click the **spyglass** or Search button.



NAME	ADMIN	START	VIEW	VIEW PARTI(	SERVER EVEI
Denallix Administrator	True	False	False	False	False

Add Users, Groups and Roles

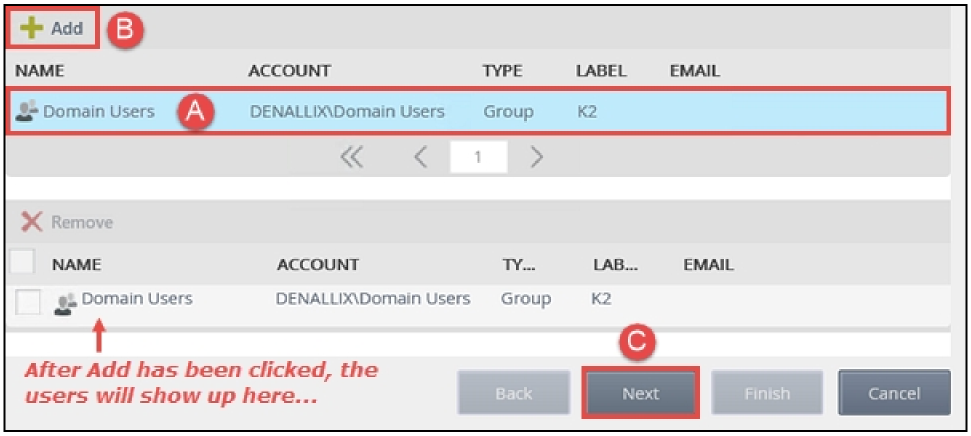
Search: All

Label: K2

Type: Contains

domain users

m. The results pane on the **Add Users, Groups and Roles** screen will display any search results that K2 finds. Click to highlight the **Domain Users** search result, then click **Add** once again to select the domain users. Click **Next**.

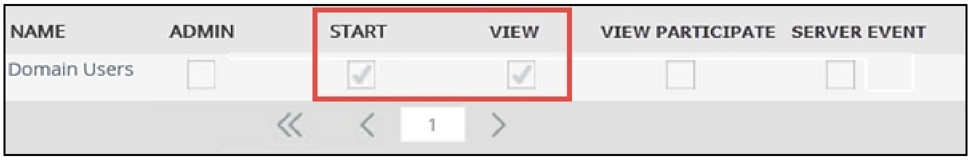


NAME	ACCOUNT	TYPE	LABEL	EMAIL
Domain Users	DENALLIX\Domain Users	Group	K2	

□	NAME	ACCOUNT	TY...	LAB...	EMAIL
□	Domain Users	DENALLIX\Domain Users	Group	K2	

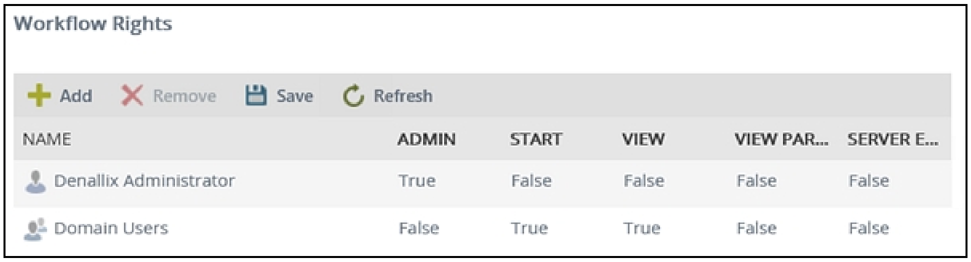
After Add has been clicked, the users will show up here...

n. CHECK the boxes for **START** and **VIEW**. Click **Finish**.



NAME	ADMIN	START	VIEW	VIEW PARTICIPATE	SERVER EVENT
Domain Users	□	☑	☑	□	□

You should see that the deploying user has ADMIN rights on the workflow, and domain users now have START and VIEW rights.



NAME	ADMIN	START	VIEW	VIEW PAR...	SERVER E...
Denallix Administrator	True	False	False	False	False
Domain Users	False	True	True	False	False

Leave the K2 Management site open as you will return to it in later steps.

Step 12 Review

In this step you reviewed the process properties, specifically ensuring that K2 will create reporting SmartObjects for this workflow. You will be using the reporting objects later to build a composite SmartObject. You also deployed the workflow and then set permissions so that users in your domain can start the workflow and run reports against the workflow.

The solution is now ready to test!

Part 2 Review

In this part of the tutorial you learned how to implement more advanced workflow concepts like conditional lines, evaluating user data and business data in a line, escalations, loop-back lines and more. You also learned how to format a workflow for clarity and then deployed the finished workflow to the K2 environment.

The application is now ready to test, which you can do by following Part 3: Testing the Application.

Temporary Access Application: Part 3 (Testing the application)

In this part of the tutorial, you will test your new Temporary Access Application by starting several new instances with different combinations of requirements and dates.

Note
You must complete Part 1 and Part 2 of this tutorial before continuing with Part 3. Part 3 is dependent on the first two parts.

Step 13: Test the workflow

Now that the workflow is deployed it is ready to test. In this step you will start a few new instances of the Temporary Access Application Workflow with different combinations of additional tasks and dates, to verify that all of the conditional executions and escalations in our workflow work as expected.

Step 13 Tasks

1. Open the **Temporary Access Request Form** and submit four new temporary access requests.

Field	Value
First Test Item	
Full name	First Test
Company	First Test Company
Start Date	(today's date)
End Date	(tomorrow's date)
AD Login Required	No
Additional IT Tasks	No
Additional HR Tasks	No
Second Test Item	
Full name	Second Test
Company	Second Test Company
Start Date	(today's date)
End Date	(2 days from today's date)
AD Login Required	No
Additional IT Tasks	Yes
Additional IT Tasks Description	(Additional IT Tasks)
Additional HR Tasks	No
Third Test Item	
Full name	Third Test
Company	Third Test Company
Start Date	(3 days before today's date)
End Date	(2 days before today's date)
AD Login Required	No
Additional IT Tasks	No
Additional HR Tasks	Yes
Additional HR Tasks Description	(Additional HR Tasks)
Fourth Test Item	
Full name	Fourth Test
Company	Fourth Test Company
Start Date	(today's date)
End Date	(today's date)
AD Login Required	Yes
AD-User Name	Set to Default
Additional IT Tasks	Yes
Additional IT Tasks Description	(Additional IT Tasks)
Additional HR Tasks	Yes
Additional HR Tasks Description	(Additional HR Tasks)

2. Using the appropriate approver account, approve all four requests. (The approver may vary, depending on where you are logged in. If on a K2-provided VM, logged in as Administrator, use Jonno's IE account found in the User Shortcuts.)

3. In the **K2 Management** site, open the process instances for the Temporary Access Application Workflow. Use the **View Flow Report** to verify that the workflows are at the expected activities, based on the selected options.

   First Test – skipped the **Create AD Account** step, skipped the **IT Tasks Step** and skipped the **HR Tasks** step, and is waiting at the **Extend Access** step.

   Second Test – skipped the **Create AD Account** step, waiting at the **IT Tasks** step.

   Third Test - skipped the **Create AD Account** step, waiting at the **HR Tasks** step.

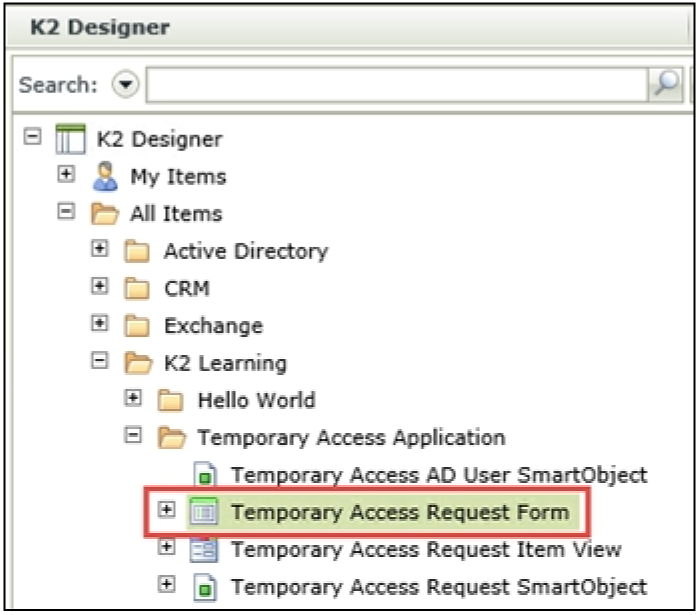
   Fourth Test – executed the **Create AD Account** step, waiting at the **IT Tasks** step and waiting at the **HR Tasks** step.

4. Submit another request and ask the approver to **Reject** the request, or reject the request as the approver. (Optional)
5. Complete the **IT Tasks** for the **Fourth Test** and the **Second Test**. Use the View Flow report to verify that the **Second Test** moved to the **Extend Access** step, but the **Fourth Test** is still waiting for **HR Tasks** to be completed.
6. Complete the **HR Tasks** for the **Third Test** and the **Fourth Test**. Use the View Flow report to verify that the **Third Test** moved to the **Extend Access** step, and the **Fourth Test** has also moved on to **Extend Access** because the HR task has completed.
7. As the originator, complete the **First Test** with **Do Not Extend Access**.
8. As the originator, open the **Third Request**, select an **End Date** in the future and complete the task with **Extend Access**.
9. Using View Flow reports, verify that the **Third Request** and the **Second Request** workflows are waiting to start the **Extend Access** step, because the end date is in the future.
10. Verify that the **Third Test** workflow expired at the **Extend Access** step and continued on to the **Set Status Expired** step. It did not go further than this because there was no AD account for this request.

Step 13 Walkthrough

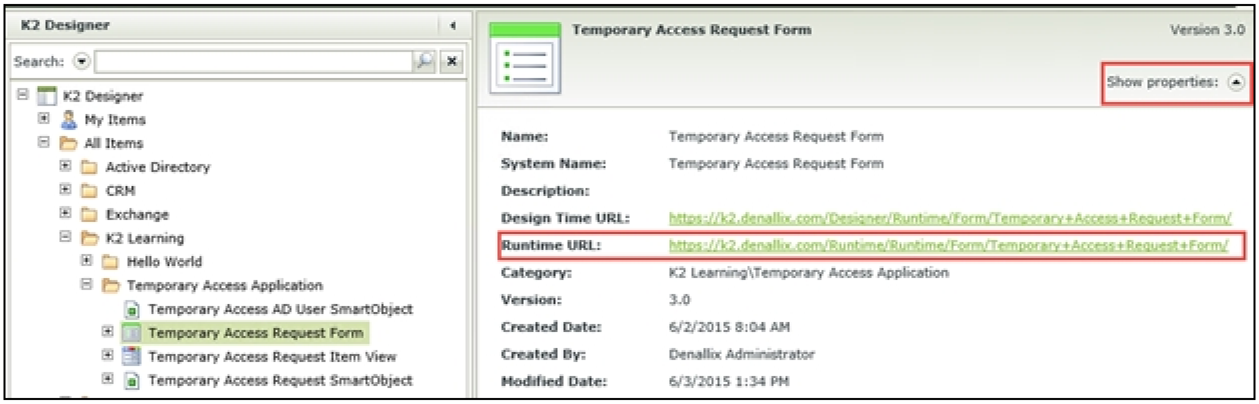
a. You will use the Temporary Access Request Form to start four new instances of the workflow. Open K2 Designer: **Start > All Programs > K2 blackpearl > K2 Designer**. (If the link does not appear in your menu, contact your K2 administrator for the URL to your K2 Designer environment.)

b. Locate the **K2 Learning > Temporary Access Application** folder, then select the **Temporary Access Request Form**.


K2 Designer
Search:
K2 Designer
My Items
All Items
Active Directory
CRM
Exchange
K2 Learning
Hello World
Temporary Access Application
Temporary Access AD User SmartObject
Temporary Access Request Form
Temporary Access Request Item View
Temporary Access Request SmartObject

c. In the properties pane, click the **Runtime URL** to open the form in a new tab.

> **Note**
> The Runtime URL is the web address (or link) that allows your users to access SmartForms. This is the URL that you can use to link to this form from a web page, for example.


K2 Designer
Search:
K2 Designer
My Items
All Items
Active Directory
CRM
Exchange
K2 Learning
Hello World
Temporary Access Application
Temporary Access AD User SmartObject
Temporary Access Request Form
Temporary Access Request Item View
Temporary Access Request SmartObject
Temporary Access Request Form
Version 3.0
Show properties:

Name: Temporary Access Request Form
System Name: Temporary Access Request Form
Description:
Design Time URL: https://k2.denallix.com/Designer/Runtime/Form/Temporary+Access+Request+Form/
Runtime URL: https://k2.denallix.com/Runtime/Runtime/Form/Temporary+Access+Request+Form/
Category: K2 Learning\Temporary Access Application
Version: 3.0
Created Date: 6/2/2015 8:04 AM
Created By: Denallix Administrator
Modified Date: 6/3/2015 1:34 PM

d. Once the form loads, enter some data for the test requests, then **Submit** the form. Use the table below to capture the details for each request. Make sure to select the options exactly as shown, since you will use reporting to validate that the workflow is executing as expected. Make a note of the **approver** value, because this user will need to approve the requests. When you submit the form you should see a confirmation message. In the background, the workflow will be starting.

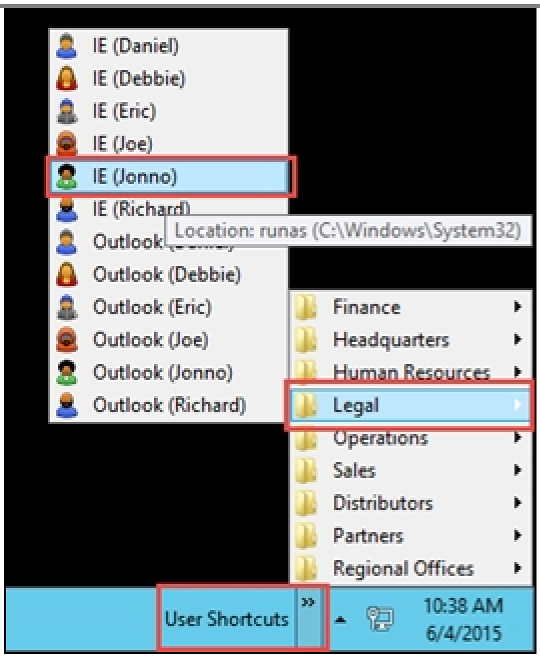
Field	Value
First Test Item	
Full name	First Test
Company	First Test Company
Start Date	(today's date)
End Date	(tomorrow's date)
AD Login Required	No
Additional IT Tasks	No

Additional HR Tasks	No
Second Test Item	
Full name	Second Test
Company	Second Test Company
Start Date	(today's date)
End Date	(2 days from today's date)
AD Login Required	No
Additional IT Tasks	Yes
Additional IT Tasks Description	(Additional IT Tasks)
Additional HR Tasks	No
Third Test Item	
Full name	Third Test
Company	Third Test Company
Start Date	(3 days before today's date)
End Date	(2 days before today's date)
AD Login Required	No
Additional IT Tasks	No
Additional HR Tasks	Yes
Additional HR Tasks Description	(Additional HR Tasks)
Fourth Test Item	
Full name	Fourth Test
Company	Fourth Test Company
Start Date	(today's date)
End Date	(today's date)
AD Login Required	Yes
AD-User Name	Set to Default
Additional IT Tasks	Yes
Additional IT Tasks Description	(Additional IT Tasks)
Additional HR Tasks	Yes
Additional HR Tasks Description	(Additional HR Tasks)

e. **Exit** the form.

Now, you will action the requests as the approver.

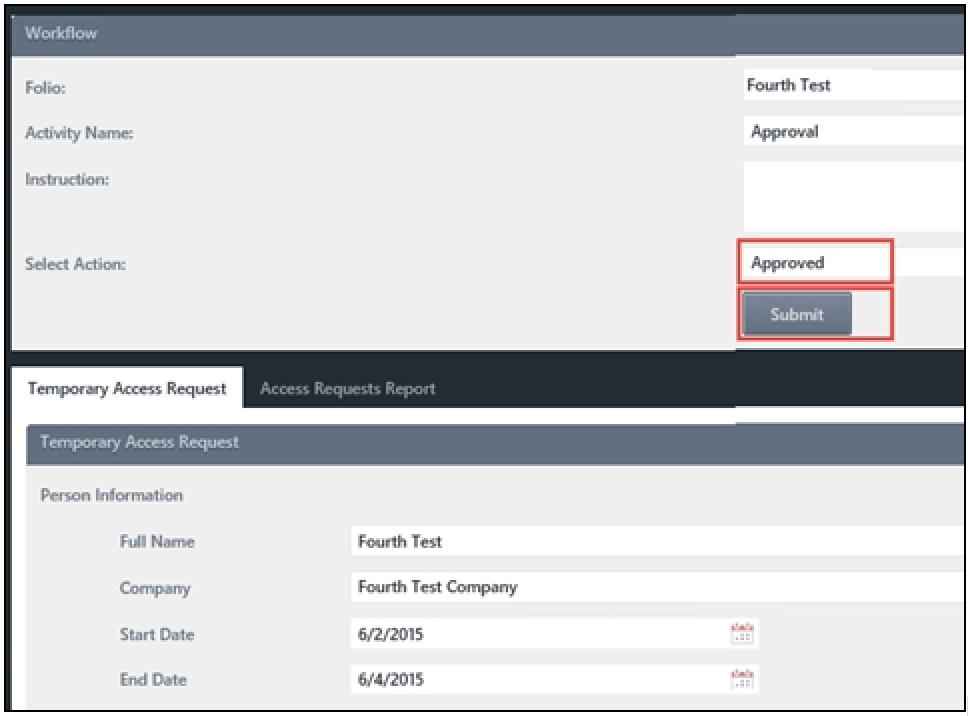
f. If you are building this tutorial using a K2-provided VM, open the approver's Internet Explorer (most likely Jonno) using the **User Shortcuts**. If prompted, login with the following credentials:
Username: *Denallix\Jonno*
Password: *K2pass!*
Alternatively, ask the approver that you noted while completing the requests to log in so that they can approve the requests.



g. Open the approver’s email or Outlook Web Access (OWA). You should see four approval tasks for the approver, one for each test you started.

h. Open the link for each of the tasks, and then approve each request. Alternatively, just reply to the emails with
Approved
for each item.

Note
SmartActions. SmartActions allow you users to simply reply to the task notification email with their action decision as the message body. For example, to approve a request, the user can simply reply to the task notification email with "Approved" as the message body content. K2 will read this action and move the workflow along accordingly. This feature is for convenience as it allows your users to action a task assigned to them without having to open and submit the form.



Workflow

Folio: Fourth Test

Activity Name: Approval

Instruction:

Select Action: Approved

Temporary Access Request

Person Information

Full Name: Fourth Test

Company: Fourth Test Company

Start Date: 6/2/2015

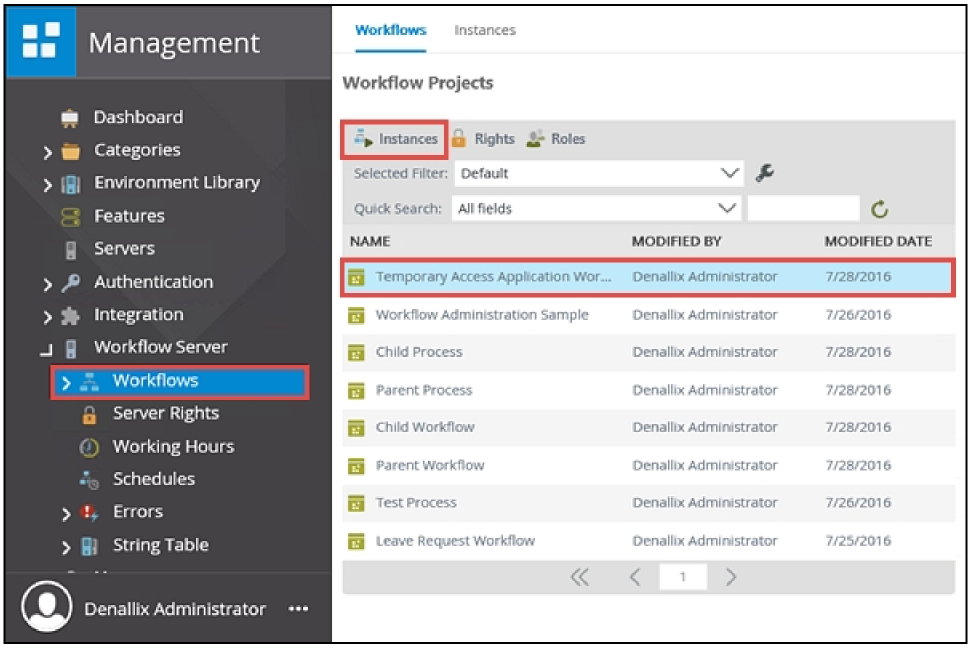
End Date: 6/4/2015

p

i. Close the approver's browser.

j. Switch back to the **K2 Management** site tab in your browser. (If you closed the K2 Management site, use the start menu link to open it once again: **Start > All Programs > K2 blackpearl > K2 Management**)

k. Expand the Workflow Server node, then click **Workflows** so that the workflow projects are displayed in the central pane. Locate, then click to highlight, the **Temporary Access Application Workflow**. Click the **Instances** button located in the toolbar.



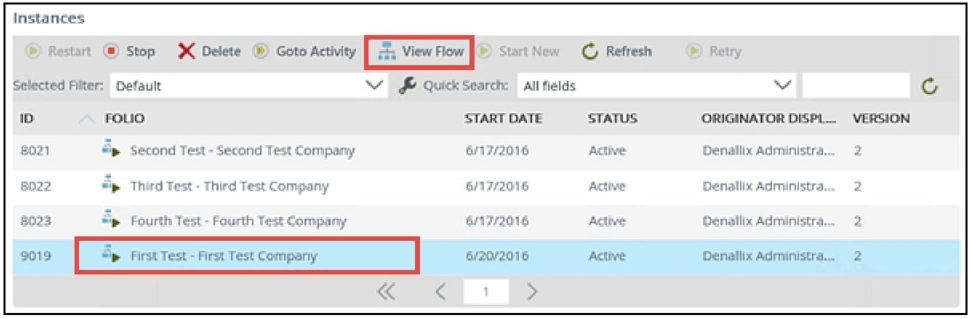
Selected Filter: Default

Quick Search: All fields

NAME	MODIFIED BY	MODIFIED DATE
Temporary Access Application Wor...	Denallix Administrator	7/28/2016
Workflow Administration Sample	Denallix Administrator	7/26/2016
Child Process	Denallix Administrator	7/28/2016
Parent Process	Denallix Administrator	7/28/2016
Child Workflow	Denallix Administrator	7/28/2016
Parent Workflow	Denallix Administrator	7/28/2016
Test Process	Denallix Administrator	7/26/2016
Leave Request Workflow	Denallix Administrator	7/25/2016

You should see the four instances that you started in the previous step. (The order of your process instances may not match the image below.) You will be opening the View Flow Report for each process instance to observe the current status of the workflow.

Process instances of the Temporary Access Application Workflow



Selected Filter: Default   Quick Search: All fields

ID	FOLIO	START DATE	STATUS	ORIGINATOR DISPL...	VERSION
8021	Second Test - Second Test Company	6/17/2016	Active	Denallix Administra...	2
8022	Third Test - Third Test Company	6/17/2016	Active	Denallix Administra...	2
8023	Fourth Test - Fourth Test Company	6/17/2016	Active	Denallix Administra...	2
9019	First Test - First Test Company	6/20/2016	Active	Denallix Administra...	2

l. Located the **First Test** process instance and click it to highlight it. Click the **View Flow** button located in the toolbar. Use the View Flow Report to verify that the workflows are at the expected activities, based on the selected options: (Leave the View Flow browser pages open, as you will return to them in a later step.)

First Test: skipped the **Create AD Account** step, skipped the **IT Tasks Step** and skipped the **HR Tasks** step, and is waiting at the **Extend Access** step.

Second test: skipped the **Create AD Account** step, and is waiting at the **IT Tasks** step.

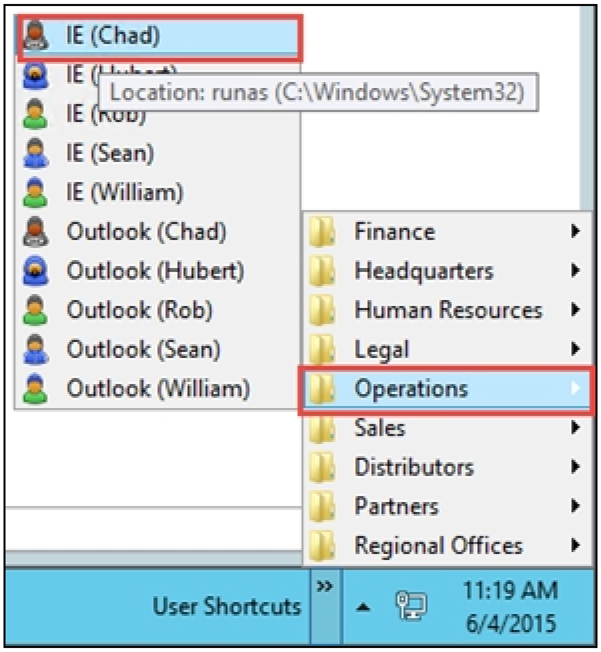
Third Test: skipped the **Create AD Account** step, and is waiting at the **HR Tasks** step.

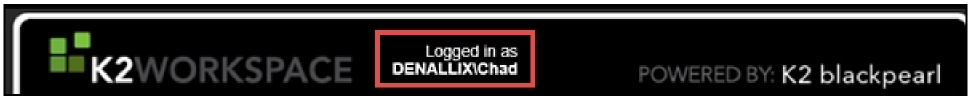
Fourth Test: executed the **Create AD Account** step, is waiting at the **IT Tasks** step and also waiting at the **HR Tasks** step.

m. If you have time, submit another request and ask the approver to **Reject** the request, or reject the request as the approver.

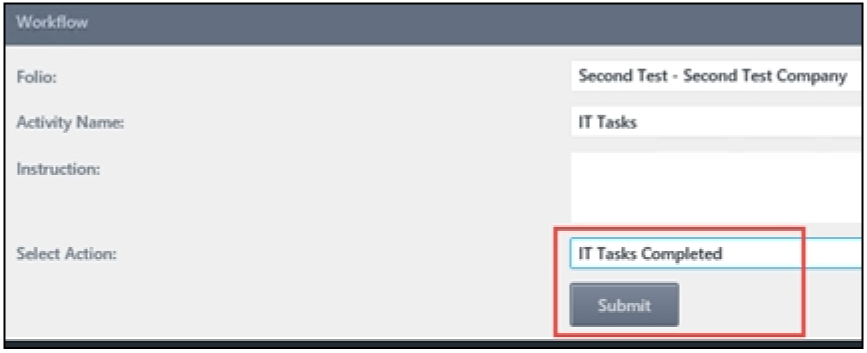
In next several steps, you will action the IT and HR tasks as the user assigned the task. If you followed the tutorial exactly, open the group member that is described in the steps below. If you assigned a different user as the participant, open that user's K2 Workspace to action the task. To open a user's K2 Workspace, begin by opening IE for that user. After IE is launched, confirm that it is the active window. Launch K2 Workspace (**Start** > **All Programs** > **K2 blackpearl** > **K2 Workspace**) so that it opens for the target user.

n. For the IT tasks, open the any user from the **Operations** group. Launch **K2 Workspace** from the start menu link.





o. You should see two tasks for the IT Tasks step (**Fourth Test** and **Second Test**). For both tasks, open the task form and then submit the form with the **IT Tasks Completed** task action.



Workflow

Folio: Second Test - Second Test Company

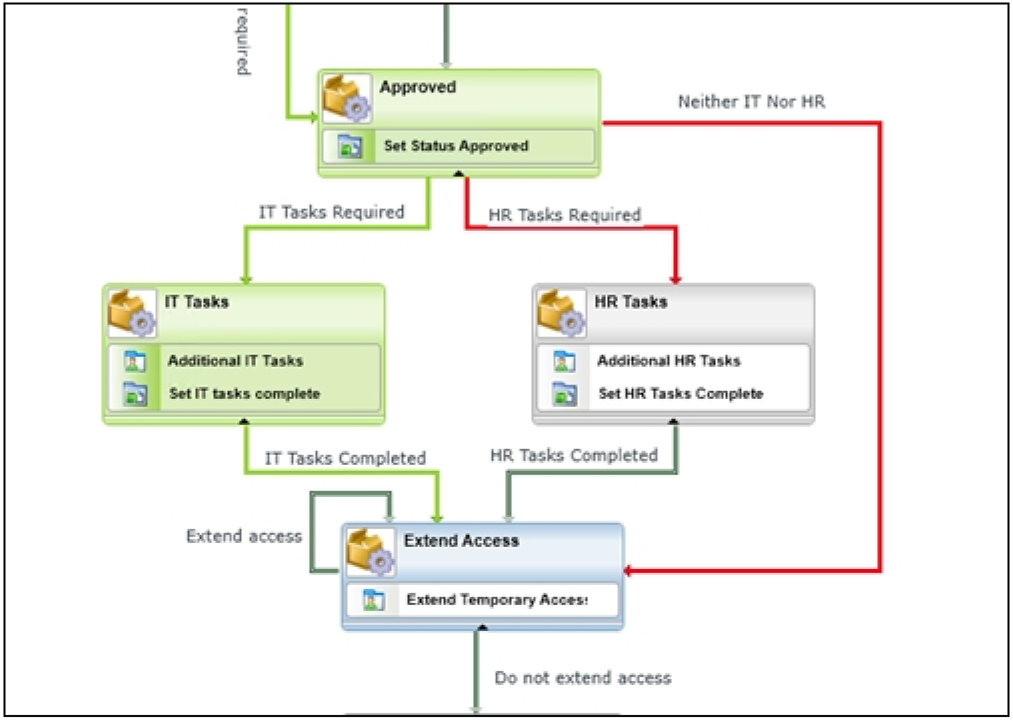
Activity Name: IT Tasks

Instruction:

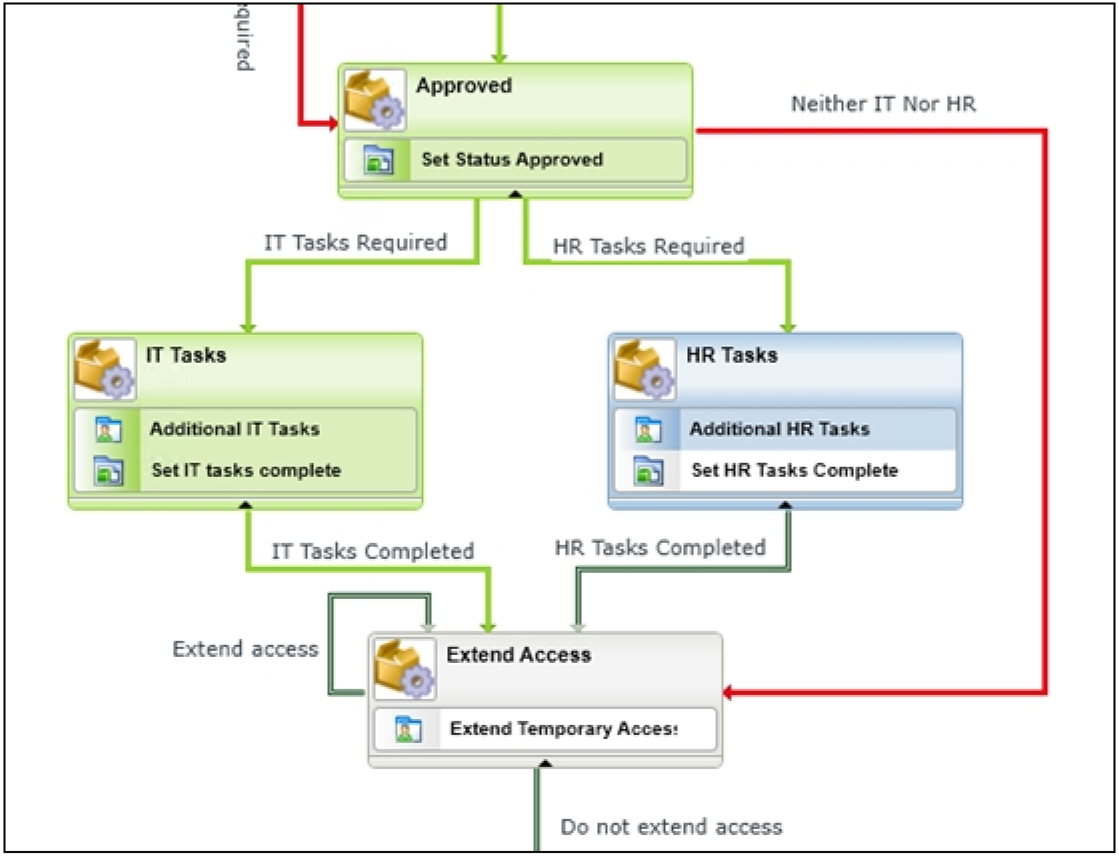
Select Action: IT Tasks Completed

Submit

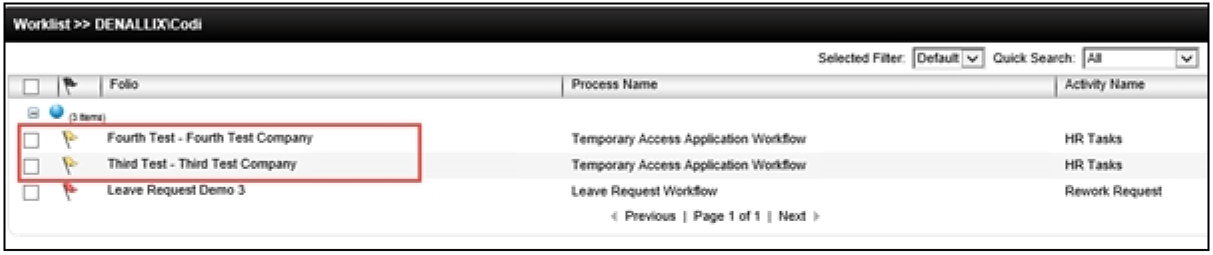
p. Return to the View Flow Reports that you left open (or re-open the View Flow Reports for the workflows that you started). For the **Second Test** workflow, you should see that the **IT Tasks** step has completed and the workflow moved on to the **Extend Access** step, because there were no HR Tasks for this request.



q. For the Fourth Test workflow, you should see that the **IT Tasks** step has completed but the workflow has not moved on to the **Extend Access** step because there is still an active **HR Tasks** for this request.



r. Next you will complete the outstanding HR Tasks. If you are using the K2-provided virtual server, open Internet Explorer as a user in the **Human Resources** group. If you assigned a different user, open that user's browser instead. As you did for the IT Tasks, open both HR Tasks and submit the forms with **HR Tasks Completed** as the action.



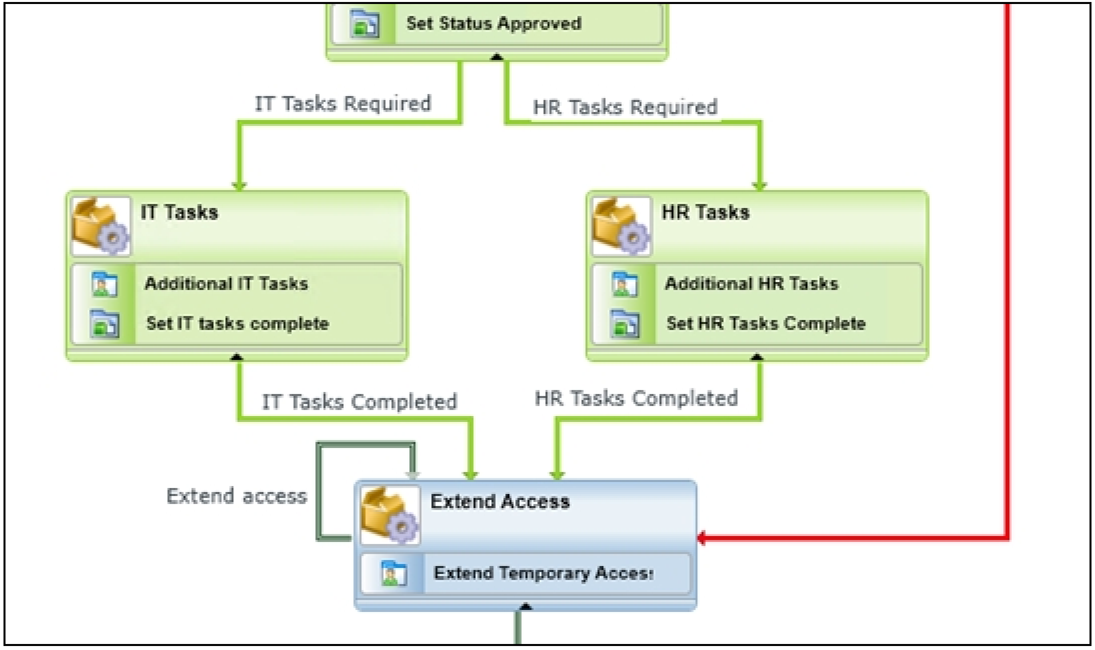
Worklist >> DENALLIX\Codi

Selected Filter: Default Quick Search: All

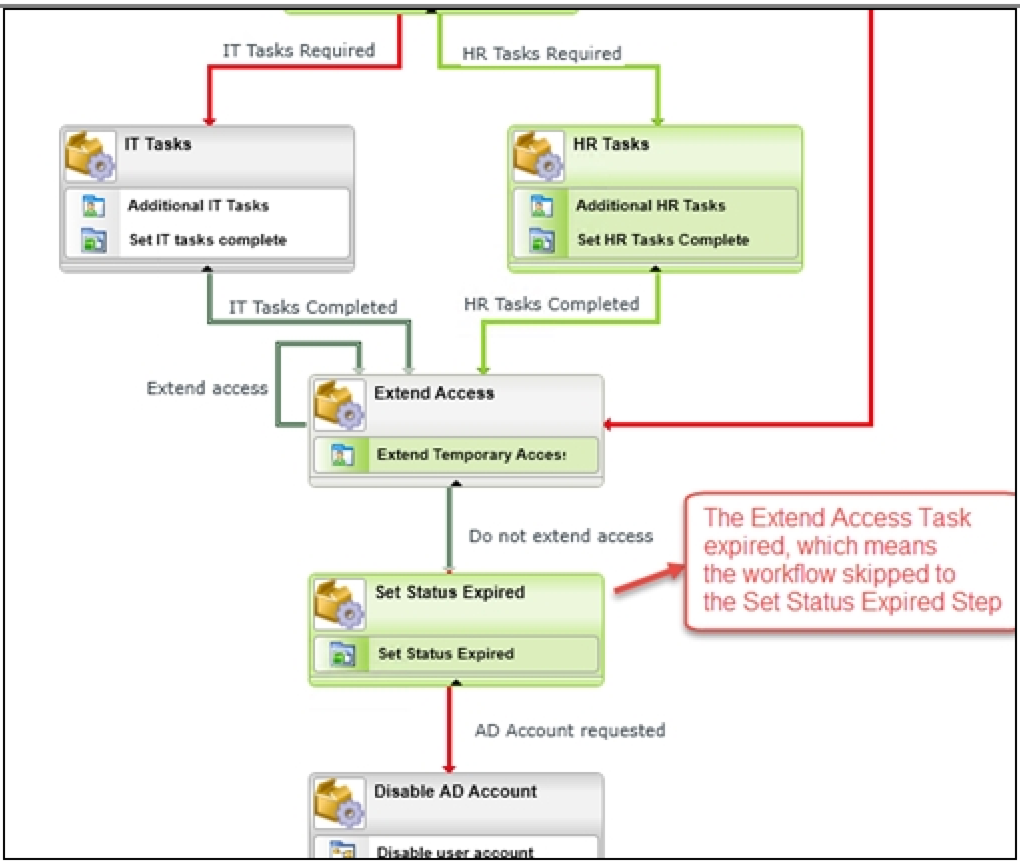
	Folio	Process Name	Activity Name
	(3 items)		
□	Fourth Test - Fourth Test Company	Temporary Access Application Workflow	HR Tasks
□	Third Test - Third Test Company	Temporary Access Application Workflow	HR Tasks
□	Leave Request Demo 3	Leave Request Workflow	Rework Request

Previous | Page 1 of 1 | Next

s. Return to the View Flow Reports that you had left open. You should see that the **Fourth Test** workflow has now progressed to the **Extend Access** step, because both the **IT Tasks** and the **HR Tasks** were completed. This verifies that our parallel workflow configuration is working as expected.

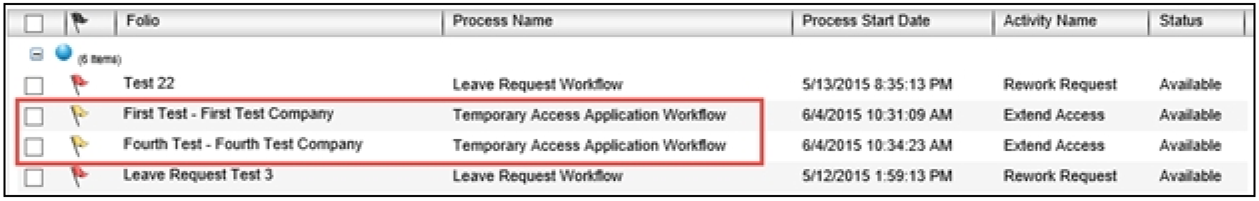


t. The workflow for the **Third Test** instance should also have progressed to the **Extend Access** step. If you entered the dates correctly, the **Third Test** instance should actually already have moved on to the **Set Status Expired** step, because the end date for this request was two days ago. Remember that you defined an escalation for the Extend Access task to expire it one day after the end date of the request.
Because there was no AD account for the Third Test request, the workflow did not try to disable the account and the workflow competed.
If this is what you see in the View Flow report, it means that the escalation and conditional line rules are working as expected.



The Extend Access Task expired, which means the workflow skipped to the Set Status Expired Step

u. Open K2 Workspace as the user who initially requested the temporary access. You should see two instances for the temporary access request workflow: **First Test** and **Fourth Test**

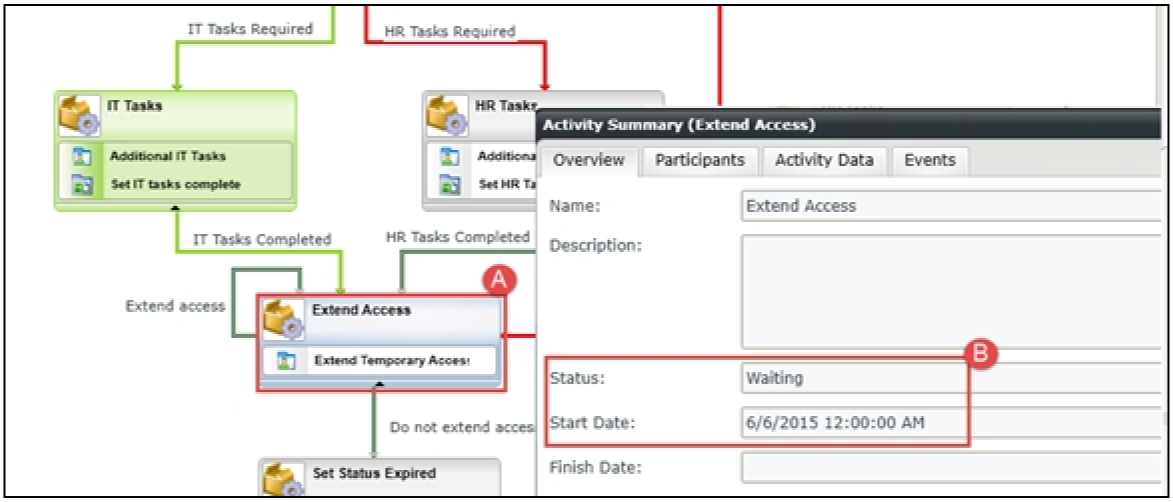


□		Folio	Process Name	Process Start Date	Activity Name	Status
⊟	●	(6 items)				
□	🚩	Test 22	Leave Request Workflow	5/13/2015 8:35:13 PM	Rework Request	Available
□	🚩	First Test - First Test Company	Temporary Access Application Workflow	6/4/2015 10:31:09 AM	Extend Access	Available
□	🚩	Fourth Test - Fourth Test Company	Temporary Access Application Workflow	6/4/2015 10:34:23 AM	Extend Access	Available
□	🚩	Leave Request Test 3	Leave Request Workflow	5/12/2015 1:59:13 PM	Rework Request	Available

v. Open the **First Test** request form, select the **Do Not Extend Access** option and **Submit** the form.

w. Open the **Fourth Test** form, select a date in the future for the **End Date**, select the **Extend Access** action and **Submit** the form.

x. If you refresh you task list, you should not see any **Extend Access** tasks because the remaining tasks are delayed until one day before the end date. For example, **Second Test** does not show up in the requester's task list yet, because you delayed this task to start one day before the temporary access is due to expire. In fact, if you open the View Flow Report for the **Second Test** and then double-click the **Extend Access** step, you should see that the status is waiting and the start date is a date in the future.



Name: Extend Access

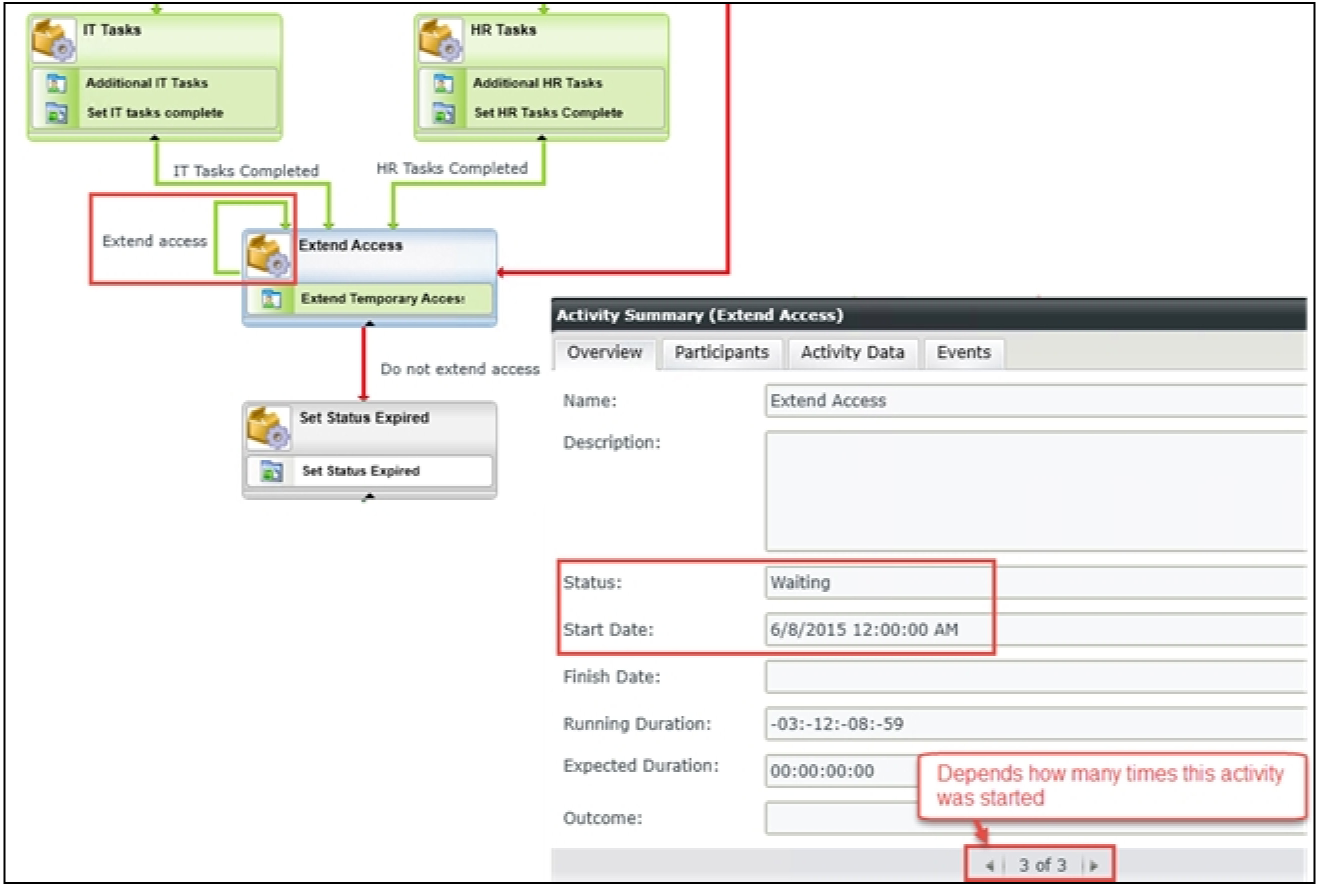
Description:

Status: Waiting

Start Date: 6/6/2015 12:00:00 AM

Finish Date:

y. If you open the View Flow Report for the **Fourth Test** workflow, you should see that the **Extend Access** line was followed and the activity looped back to itself. If you double-click the **Extend Access** step, notice that the status is waiting with a start date sometime in the future, based on the end date you selected. Notice at the bottom of this screen that there are multiple instances of this activity (probably two in our case), because the same activity was started twice. You can click the arrows to move between the reporting data for each instance of the activity.



Name: Extend Access

Description:

Status: Waiting

Start Date: 6/8/2015 12:00:00 AM

Finish Date:

Running Duration: -03:-12:-08:-59

Expected Duration: 00:00:00:00

Outcome:

z. Your are done with the tests, so you can close all browser windows. Close the Instances screen in the K2 Management site.

Step 13 Review

In this step you tested your workflow to verify that the workflow is executing as expected. Between the conditional paths, the start rules and escalation rules you have learned quite a lot about building more complex workflows with K2 Studio!

In Part 4: Building a composite SmartObject of this tutorial, you will look at creating a composite SmartObject and a view so that it is easier to report on temporary access requests.

Part 2: SmartObjects in K2 Studio

PART 2

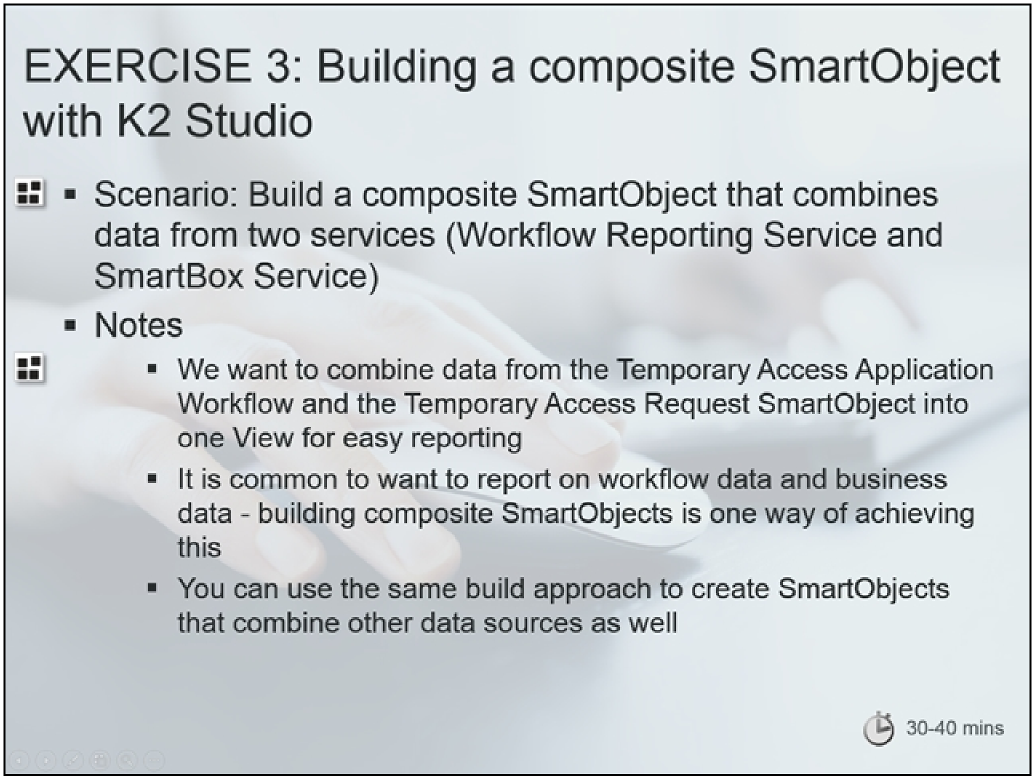
SmartObjects in K2 Studio

▶ Video

In Part 2 of this module, we will look at SmartObjects in K2 Studio, including more advanced SmartObject concepts like composite SmartObjects.

There is one exercise in this part where we will build a composite SmartObject for our Temporary Access Application. Note that the SmartObject exercise requires the workflow that you build in Part 1 of this module.

EXERCISE 3: Building a composite SmartObject with K2 Studio



EXERCISE 3: Building a composite SmartObject with K2 Studio

- Scenario: Build a composite SmartObject that combines data from two services (Workflow Reporting Service and SmartBox Service)
- Notes
  - We want to combine data from the Temporary Access Application Workflow and the Temporary Access Request SmartObject into one View for easy reporting
  - It is common to want to report on workflow data and business data - building composite SmartObjects is one way of achieving this
  - You can use the same build approach to create SmartObjects that combine other data sources as well

30-40 mins

Video

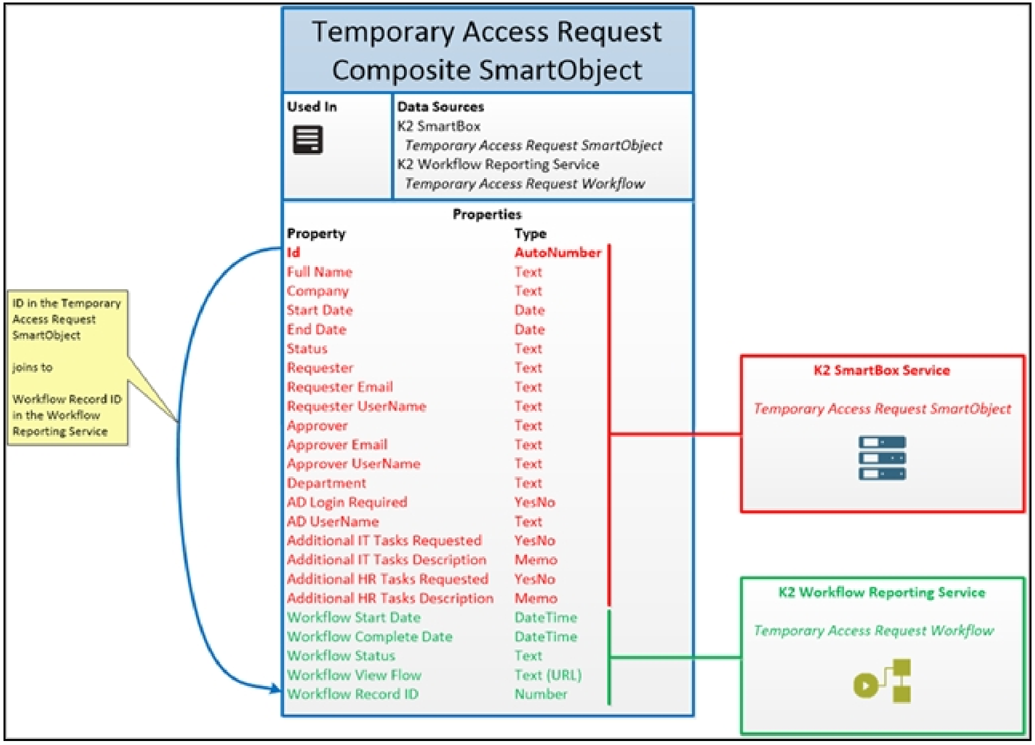
In this exercise, you will learn how to build a composite SmartObject with K2 Studio, how to test a SmartObject with the SmartObject Service Tester utility, and how to use a SmartObject in a SmartForm View. We will be using the SmartObjects from our Temporary Access Application to build a report that combines data from the workflow reporting service. In essence, this combines workflow data with business data so that it is easy for people in the organization to see who is allowed to be there, and easy to follow up on the status of the workflow. (It is a common requirement to report on workflow data and business data, and building composite SmartObjects is one way of achieving this.)

You can use the same build approach to create SmartObjects that combine other data sources as well. We are just using standard K2 Data Sources and Service Types (the Workflow Reporting Service and the SmartBox Service) because those are predictable services in all K2 environments, but the concepts apply equally to other providers and service types as well.

The diagram below illustrates the basic concept: we will be creating a new SmartObject (Temporary Access Request Composite SmartObject) that combines data from the SmartBox Service (Red fields) with data from the Workflow Reporting Service (green fields).

Because there is a common property between the two data sources (Id and Workflow Record ID), we are able to create this composite SmartObject.

The conceptual design of the composite SmartObject



Temporary Access Request Composite SmartObject

Used In	Data Sources
▤	K2 SmartBox *Temporary Access Request SmartObject* K2 Workflow Reporting Service *Temporary Access Request Workflow*

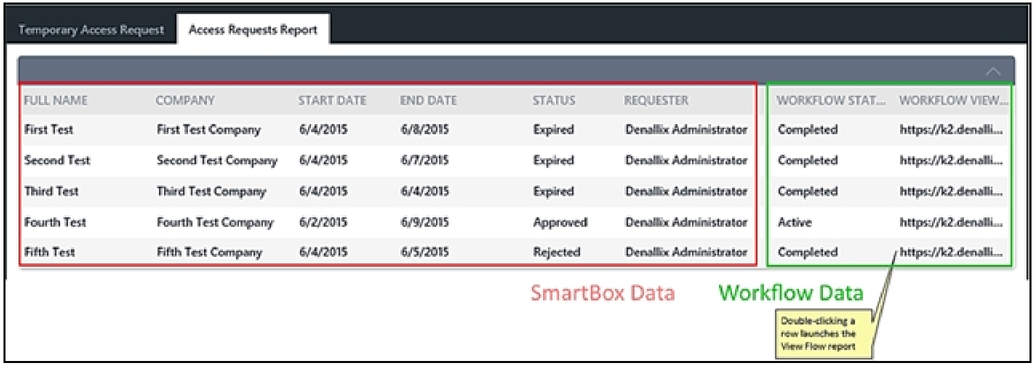
Properties

Property	Type
Id	AutoNumber
Full Name	Text
Company	Text
Start Date	Date
End Date	Date
Status	Text
Requester	Text
Requester Email	Text
Requester UserName	Text
Approver	Text
Approver Email	Text
Approver UserName	Text
Department	Text
AD Login Required	YesNo
AD UserName	Text
Additional IT Tasks Requested	YesNo
Additional IT Tasks Description	Memo
Additional HR Tasks Requested	YesNo
Additional HR Tasks Description	Memo
Workflow Start Date	DateTime
Workflow Complete Date	DateTime
Workflow Status	Text
Workflow View Flow	Text (URL)
Workflow Record ID	Number

ID in the Temporary Access Request SmartObject
joins to
Workflow Record ID in the Workflow Reporting Service
K2 SmartBox Service
Temporary Access Request SmartObject
K2 Workflow Reporting Service
Temporary Access Request Workflow

Eventually, we will be displaying this data on a SmartForm which we will then add to the Temporary Access Request Form in a separate tab. This will allow our users to easily see all Temporary Access Requests if they need to determine whether a specific person is cleared for access in the organization. (Again, you could use the SmartObject anywhere, we are just using SmartForms for this exercise since it is an easy way to create a Form that shows data.)

Showing data from the composite SmartObject on a SmartForm



FULL NAME	COMPANY	START DATE	END DATE	STATUS	REQUESTER	WORKFLOW STAT...	WORKFLOW VIEW...
First Test	First Test Company	6/4/2015	6/8/2015	Expired	Denallix Administrator	Completed	https://k2.denalli...
Second Test	Second Test Company	6/4/2015	6/7/2015	Expired	Denallix Administrator	Completed	https://k2.denalli...
Third Test	Third Test Company	6/4/2015	6/4/2015	Expired	Denallix Administrator	Completed	https://k2.denalli...
Fourth Test	Fourth Test Company	6/2/2015	6/9/2015	Approved	Denallix Administrator	Active	https://k2.denalli...
Fifth Test	Fifth Test Company	6/4/2015	6/5/2015	Rejected	Denallix Administrator	Completed	https://k2.denalli...

When you are ready, continue on to Temporary Access Application: Part 4 (Building a composite SmartObject with K2 Studio) to start this exercise.

Temporary Access Application: Part 4 (Building a composite SmartObject with K2 Studio)

In Part 1 and Part 2 you built a workflow in K2 Studio to manage temporary access requests. In this part, you will create a report against this data by combining the workflow reporting data with the request's record data. In effect, you are going to combine workflow reporting data with business data to build a report that is relevant to your users. In this case, you want to create a view of temporary access requests that anyone in the organization can review at any time, along with the workflow status and a link to the workflow. This is a security measure so that it is easy to see who is allowed to access the premises and so that it is easy to track down a workflow if it is stuck at a user task or to identify if the workflow failed.

To achieve this, you will build a composite SmartObject in K2 Studio, build a view against the SmartObject, then add the view to your Temporary Access Request Form.

From a technical perspective, what this tutorial illustrates is how to build an advanced composite SmartObject that combines data from different systems, so that you can create logical business entities that present relevant information to users, without them having to look in several different places for the information they need.

Note
You must complete Part 1, Part 2 and Part 3 of this tutorial before continuing on. Part 4 is dependent on the first three parts.

Step 14: Create a new composite SmartObject in K2 Studio

In this step you will be adding a new SmartObject to your existing K2 Learning project using K2 Studio. You will then configure the new SmartObject as a composite SmartObject that retrieves and combines data from two different systems.

Step 14 Tasks

1. In K2 Studio, open the project from Part 1 and add a new SmartObject called *Temporary Access Request Composite SmartObject* to the **Temporary Access Application** project folder.
2. Create an advanced method for the SmartObject that combines data from the **Service Object Server(s) > Service Object Server > SmartBox Service > Temporary Access Request SmartObject > Get List** method to the **Service Object Server(s) > Service Object Server > Workflow Reporting Service > K2 Learning\Temporary Access Application > Temporary Access Application Workflow > K2 Learning\Temporary Access Application Workflow > List Process Instances** method.

   Auto-create properties for all the properties returned by the **Temporary Access Request SmartObject > Get List** method.

   Manually create properties for the following properties returned by the **List Process Instances** method:

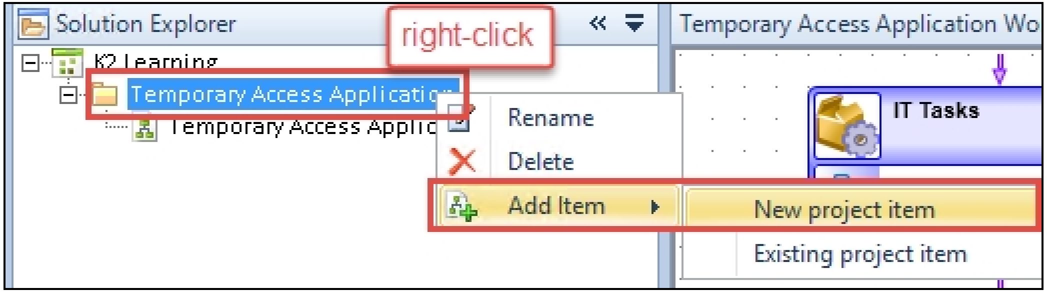
Service Property	SmartObject Property
Start Date	*Workflow Start Date*
Finish Date	*Workflow Complete Date*
Status	*Workflow Status*
View Flow	*Workflow View Flow*
RecordID	*Workflow Record ID*

3. Map the service object methods with the **Get List Method.ID** = **List Process Instances.RecordID**

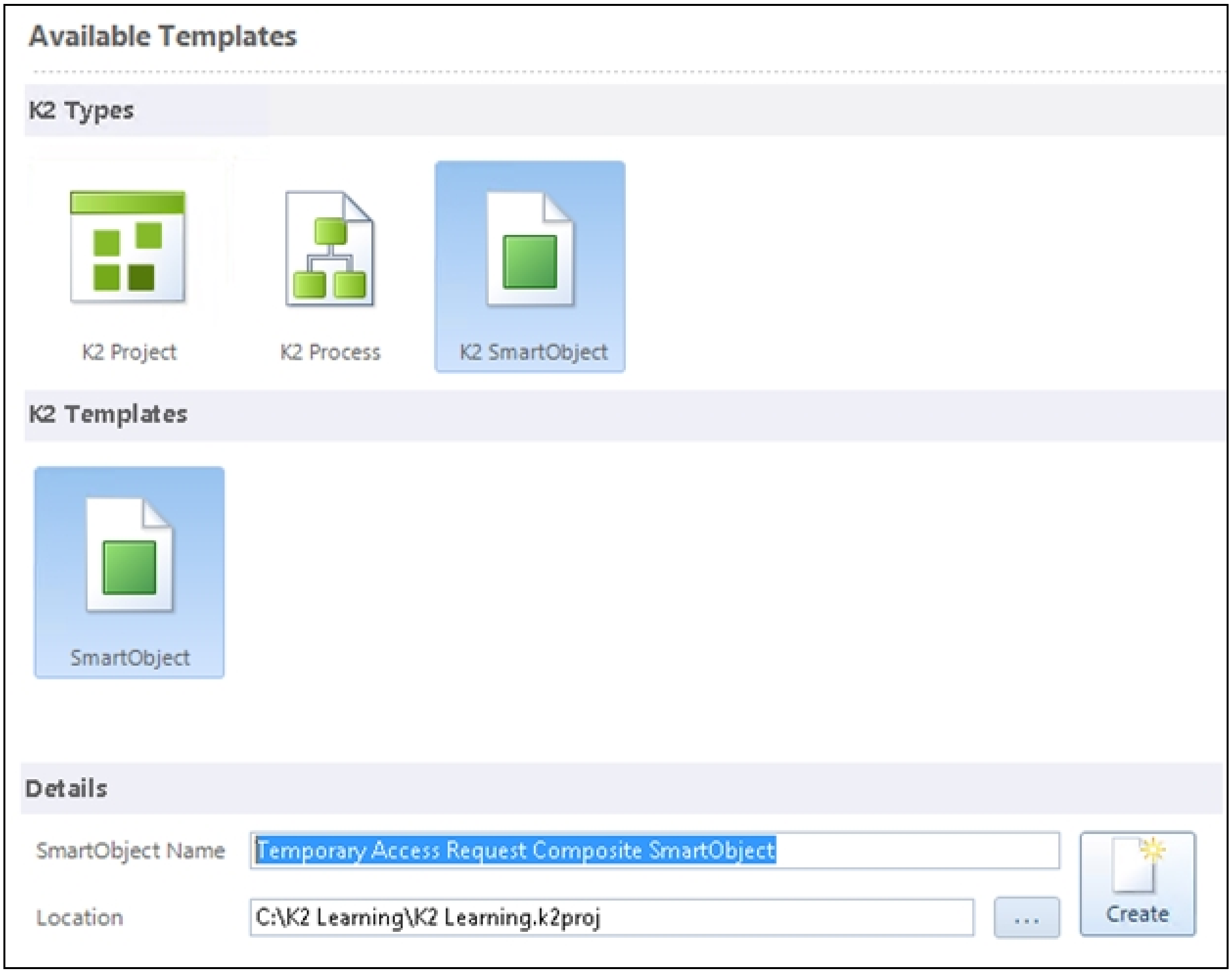
Step 14 Walkthrough

a. If necessary, open K2 Studio and open the project that you created in Part 1 of this tutorial. (Called K2 Learning.)

b. Expand the **Solution Explorer**. Right-click on the **Temporary Access Application** folder and select **Add item > New project item**.



c. On the templates window select K2 SmartObject. Give the SmartObject a name of *Temporary Access Request Composite SmartObject* and click **Create**.

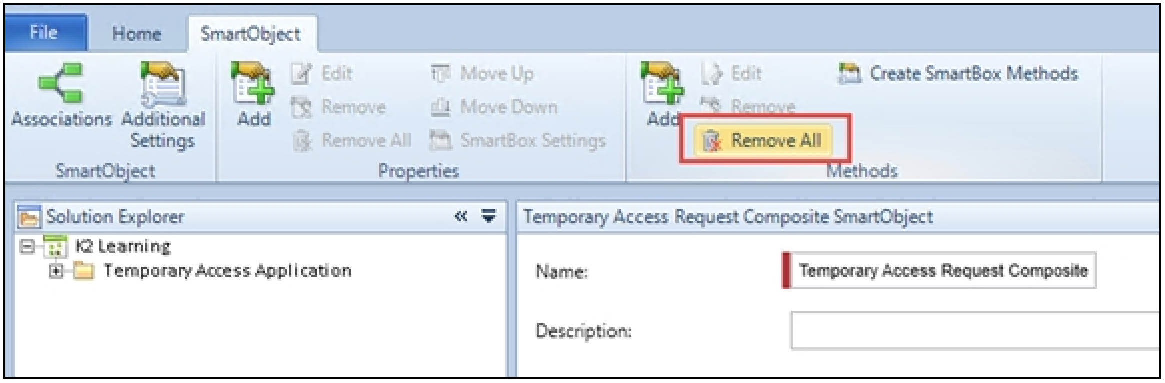


SmartObject Name: Temporary Access Request Composite SmartObject

Location: C:\K2 Learning\K2 Learning.k2proj

d. Wait for the new SmartObject design screen to appear. You will be creating an advanced SmartObject, so in the ribbon bar click on **Remove All** to remove all the methods from the SmartObject. Click **OK** on the confirmation message that appears.

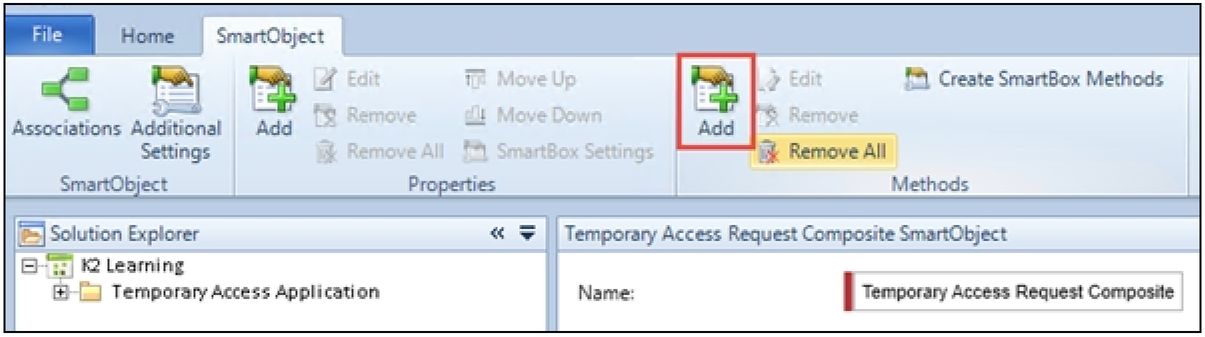
When you create an advanced SmartObject in K2 Studio, K2 will automatically add default methods (Create, Save, Delete, etc.). Since you only need one method (list), you can remove the default methods. You will add in the list method in the next few steps.



Name: Temporary Access Request Composite

Description:

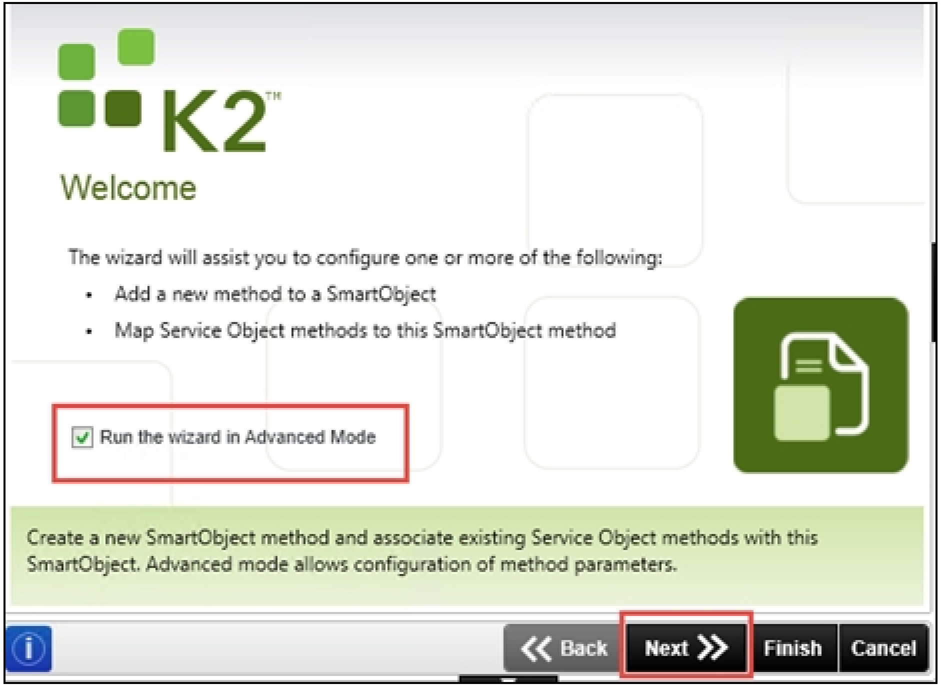
e. Click **Add** from the **Methods** group to begin.



Temporary Access Request Composite SmartObject

Name: Temporary Access Request Composite

f. On the wizard welcome screen, select the **Advanced Mode** check box and click **Next**. You must select the advanced mode option to combine results from two different systems as one method.



The wizard will assist you to configure one or more of the following:

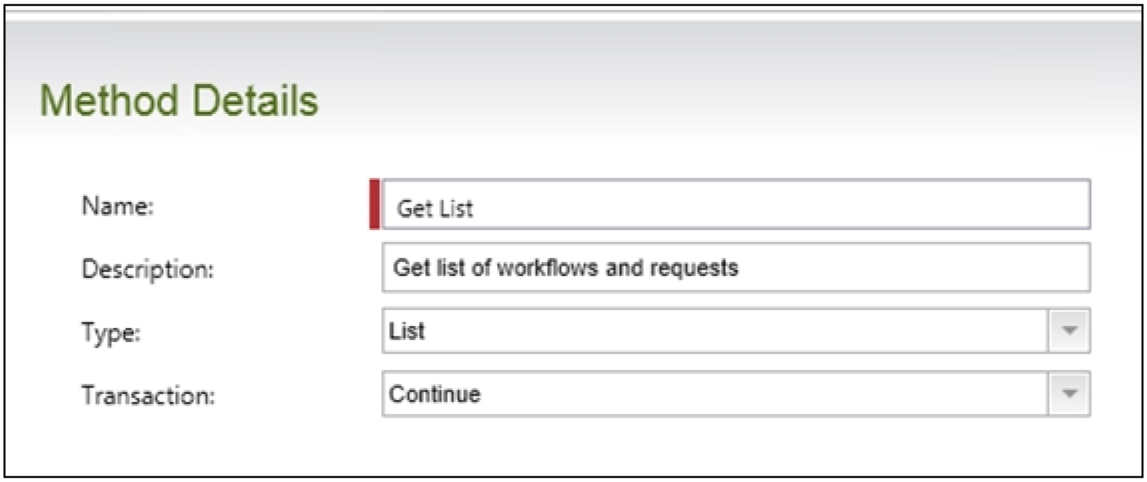
- Add a new method to a SmartObject
- Map Service Object methods to this SmartObject method

☑ Run the wizard in Advanced Mode

Create a new SmartObject method and associate existing Service Object methods with this SmartObject. Advanced mode allows configuration of method parameters.

g. Configure the **Method Details** screen as follows, then click **Next**.

Field	Value
Name	*Get List*
Description	*Get list of workflows and requests*
Type	**List** (you are returning a list of records with this method)
Transaction	**Continue** (Continue means that if one of the systems that returns data returns an error, K2 will attempt to continue calling the other system regardless.)


Method Details

Name: Get List

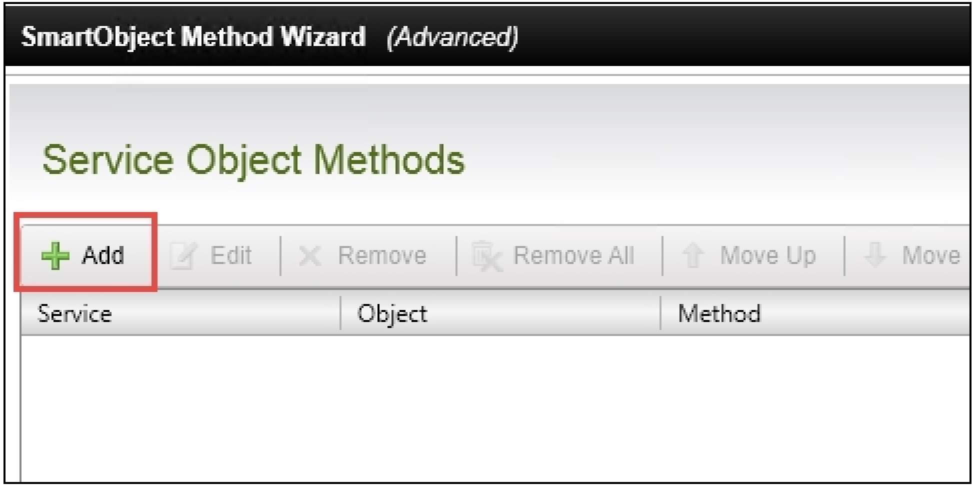
Description: Get list of workflows and requests

Type: List

Transaction: Continue

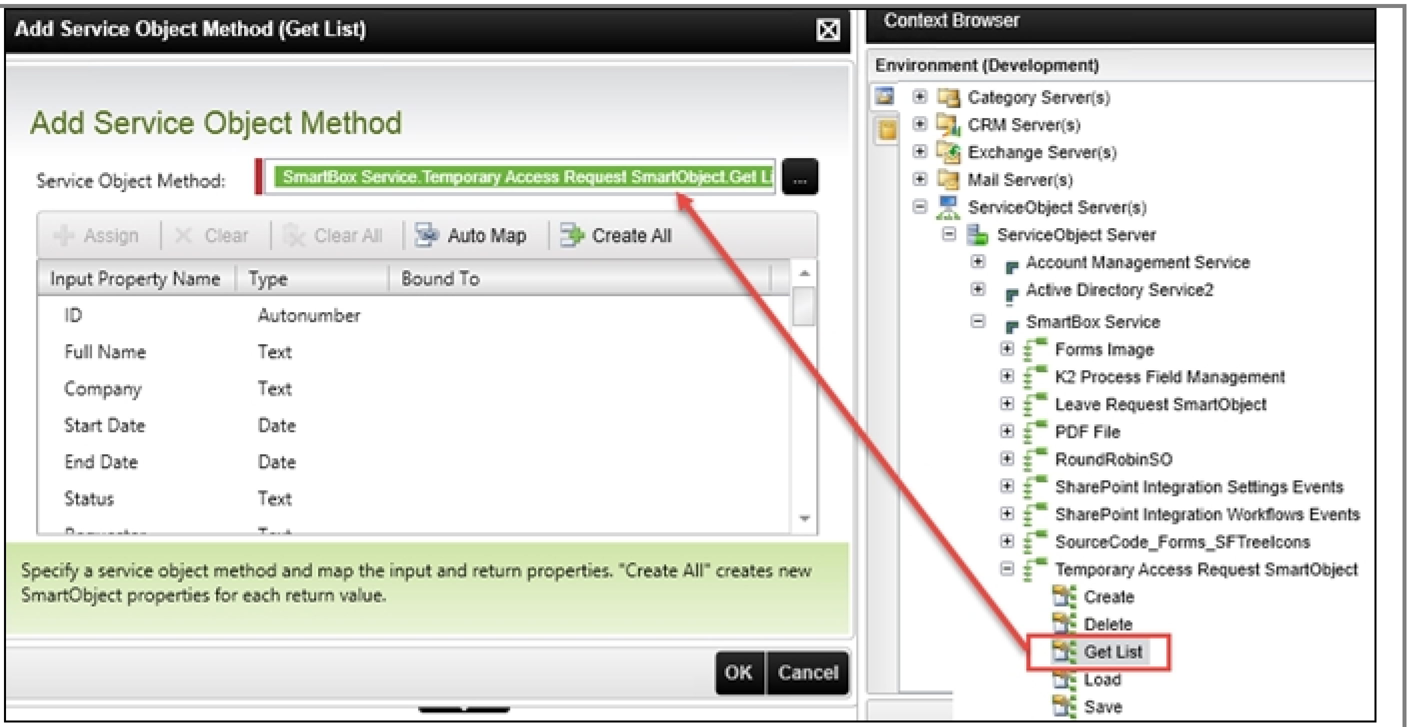
h. You don't need to add method parameters, so click **Next** to skip over the **Configure Method Parameters** screen.

i. On the **Service Object Methods** screen, click the **Add** button to add a new method.


SmartObject Method Wizard *(Advanced)*

Service Object Methods

j. Use the ellipsis button to browse for the following method and then drag it into the **Service Object Method** text box:
Service Object Server(s) > Service Object Server > SmartBox Service > Temporary Access Request SmartObject > Get List


Add Service Object Method (Get List)
Context Browser
Environment (Development)
Category Server(s)
CRM Server(s)
Exchange Server(s)
Mail Server(s)
ServiceObject Server(s)
ServiceObject Server
Account Management Service
Active Directory Service2
SmartBox Service
Forms Image
K2 Process Field Management
Leave Request SmartObject
PDF File
RoundRobinSO
SharePoint Integration Settings Events
SharePoint Integration Workflows Events
SourceCode_Forms_SFTreeIcons
Temporary Access Request SmartObject
Create
Delete
Get List
Load
Save
Assign
Clear
Clear All
Auto Map
Create All
OK
Cancel

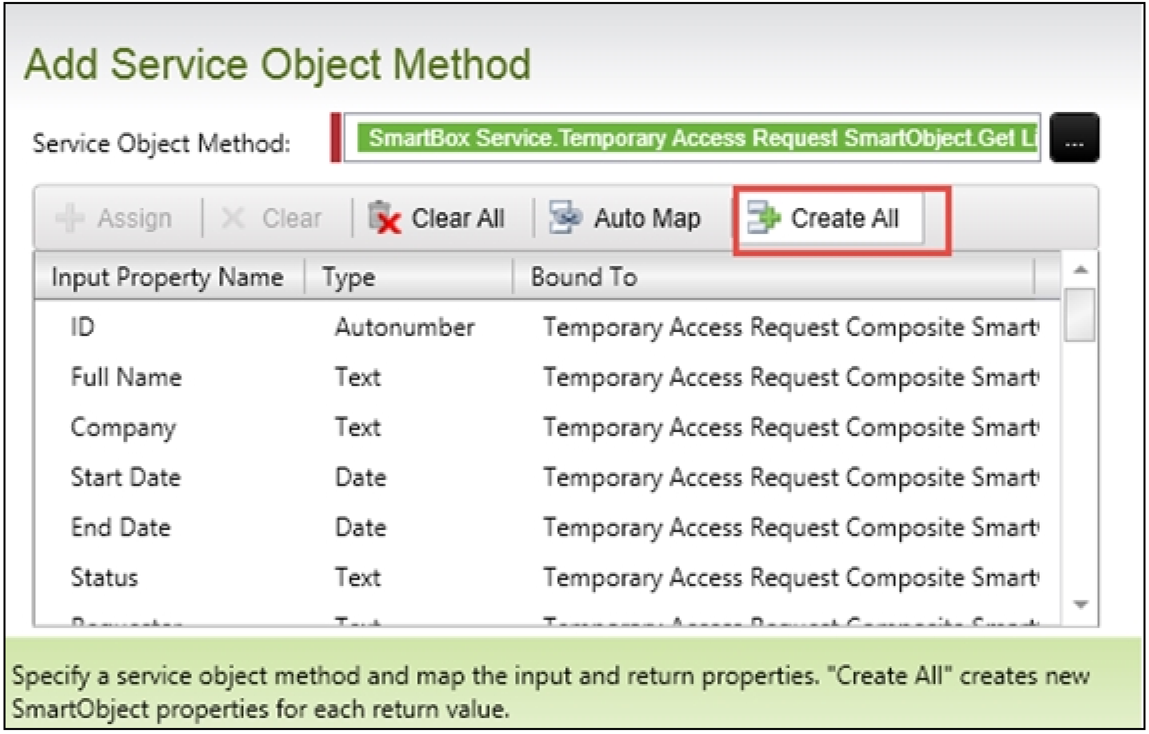
Add Service Object Method

Service Object Method: SmartBox Service.Temporary Access Request SmartObject.Get L

Input Property Name	Type	Bound To
ID	Autonumber	
Full Name	Text	
Company	Text	
Start Date	Date	
End Date	Date	
Status	Text	

Specify a service object method and map the input and return properties. "Create All" creates new SmartObject properties for each return value.

k. Click on the **Create All** button to create the properties for the SmartObject that will be bound to the properties of the service object, and then click **OK**. (You want to return all the properties from this service as properties of the new SmartObject.)



Add Service Object Method

Service Object Method: SmartBox Service.Temporary Access Request SmartObject.Get L

Input Property Name	Type	Bound To
ID	Autonumber	Temporary Access Request Composite Smart
Full Name	Text	Temporary Access Request Composite Smart
Company	Text	Temporary Access Request Composite Smart
Start Date	Date	Temporary Access Request Composite Smart
End Date	Date	Temporary Access Request Composite Smart
Status	Text	Temporary Access Request Composite Smart

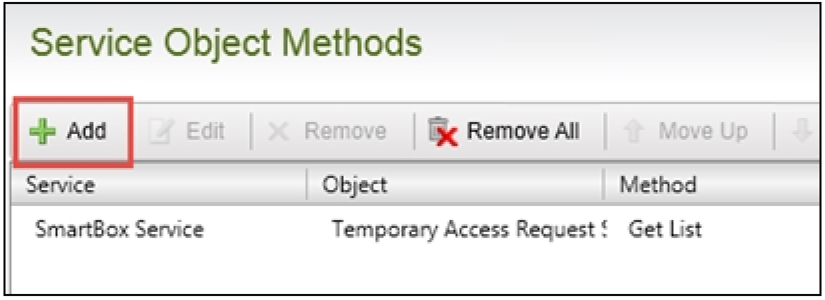
Specify a service object method and map the input and return properties. "Create All" creates new SmartObject properties for each return value.

You have just added the first service object method to your composite SmartObject. Remember that SmartObjects are associated with service objects and you can combine multiple service objects to make up a composite SmartObject.

So far, you have added the method (Get List) that returns all of the Temporary Access Request records that are stored in the K2 SmartBox database, and you configured the mappings so that all of the properties stored in the database will be exposed on the composite SmartObject.

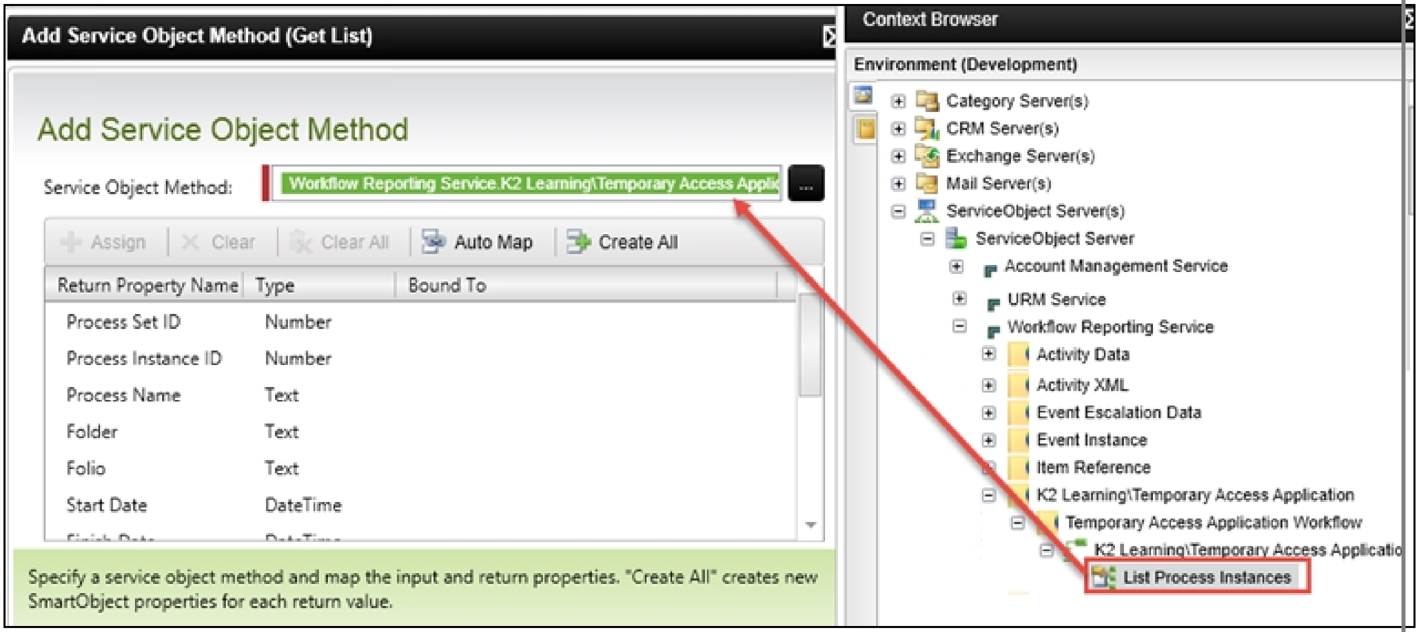
Next you will add the second service object method to your composite SmartObject, specifically a method in the **Workflow Solutions Reporting** service.

l. On the **Service Object Methods** screen, click the **Add** button again, because you want to add a second service method call to your composite SmartObject.



Service	Object	Method
SmartBox Service	Temporary Access Request S	Get List

m. This time, use the ellipsis button to browse for the following service object method, and drag it into the Service Object Method text box. **Service Object Server(s) > Service Object Server > Workflow Reporting Service > K2 Learning\Temporary Access Application > Temporary Access Application Workflow > K2 Learning\...\Temporary Access Application Workflow > List Process Instances**



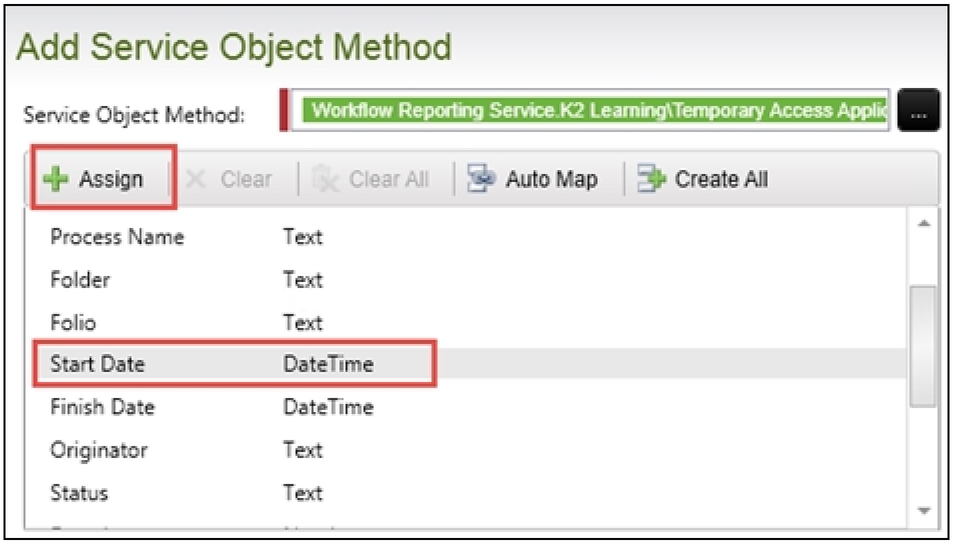
Service Object Method: Workflow Reporting Service.K2 Learning\Temporary Access Applic

Return Property Name	Type	Bound To
Process Set ID	Number	
Process Instance ID	Number	
Process Name	Text	
Folder	Text	
Folio	Text	
Start Date	DateTime	

Specify a service object method and map the input and return properties. "Create All" creates new SmartObject properties for each return value.

Tip
Remember when you selected the **Create workflow reporting SmartObjects** option when deploying the workflow? That is what created this reporting service method for you.

n. For this second service method, instead of auto-generating all properties, you only want specific properties returned from this service. Locate and select the **Start Date** property and click the **Assign** button.

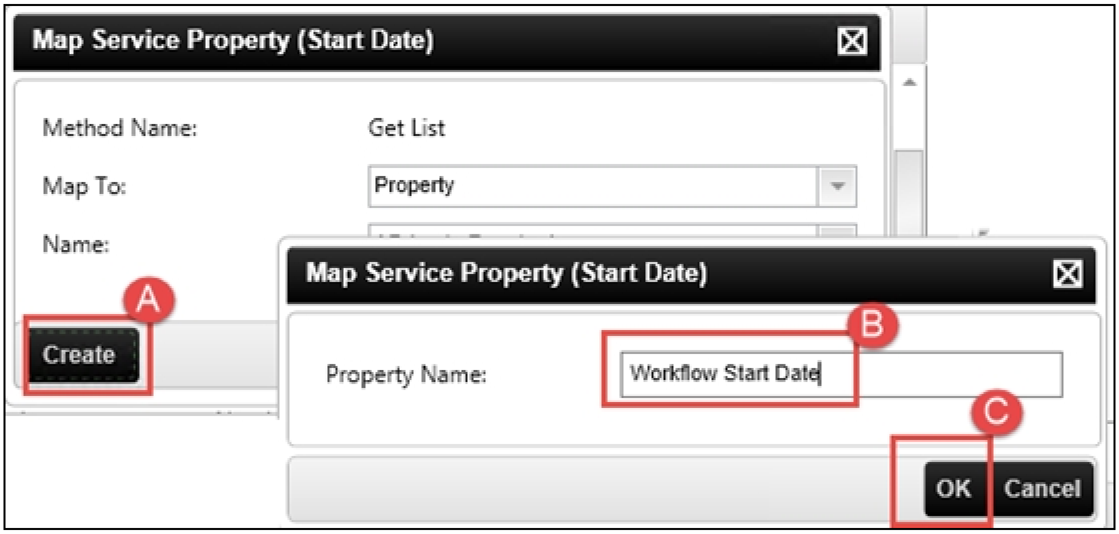


Add Service Object Method

Service Object Method: Workflow Reporting Service.K2 Learning\Temporary Access Applic

Process Name	Text
Folder	Text
Folio	Text
Start Date	DateTime
Finish Date	DateTime
Originator	Text
Status	Text

o. On the **Map Service Property** screen, click the **Create** button and then on the popup window, set the **Property Name** of the new property to
Workflow Start Date
and click **OK**.



Map Service Property (Start Date)

Method Name: Get List

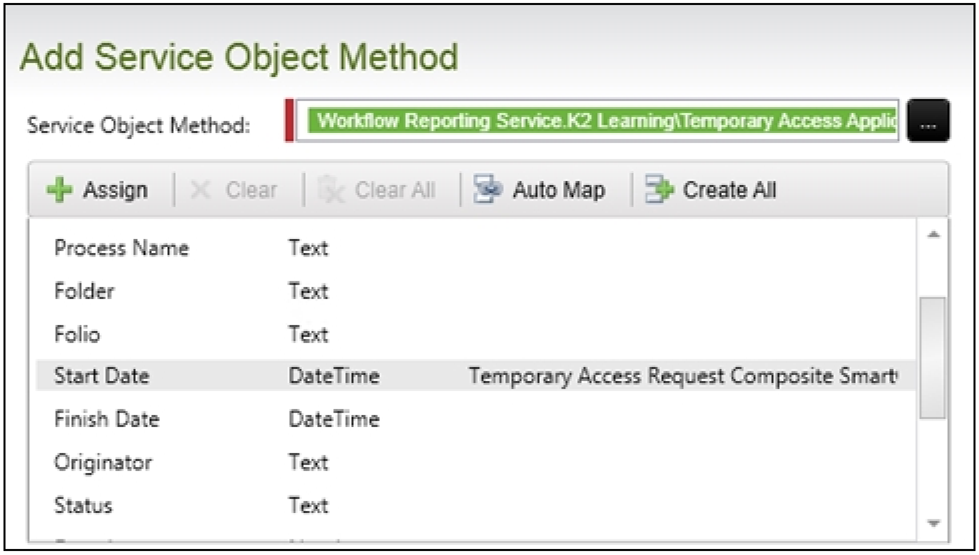
Map To: Property

Name:

Map Service Property (Start Date)

Property Name: Workflow Start Date

p. Click **OK** to close the **Map Service Property** screen and you should see that the **Start Date** has been mapped to a new property called **Workflow Start Date**.



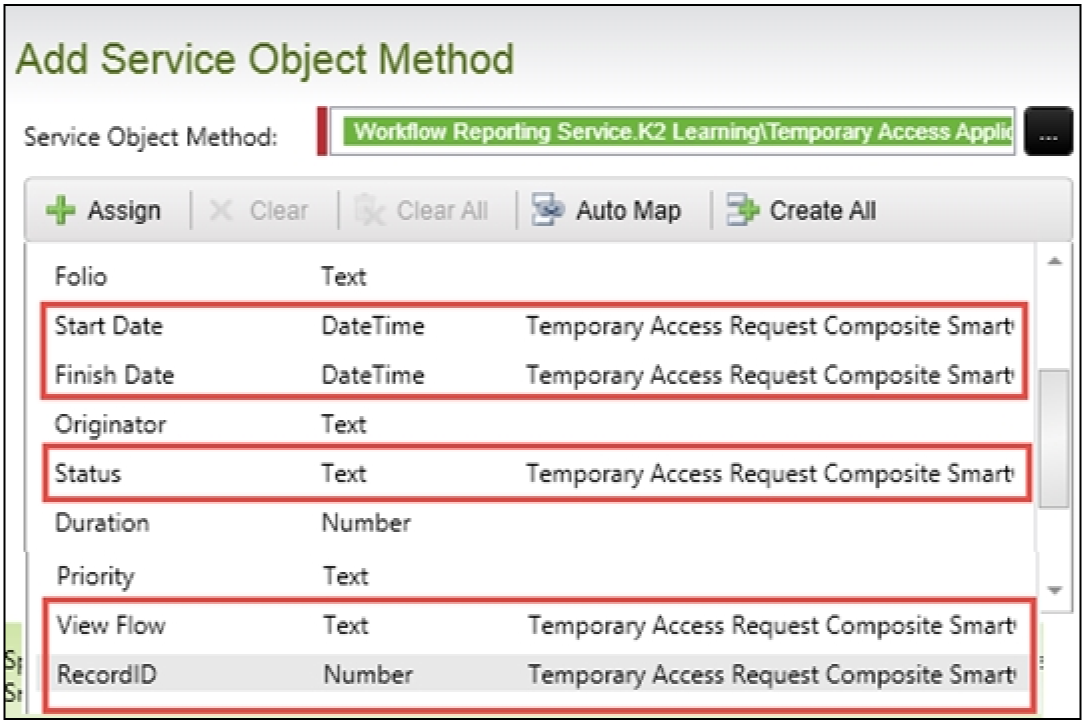
Add Service Object Method

Service Object Method: Workflow Reporting Service.K2 Learning\Temporary Access Applic

Process Name	Text	
Folder	Text	
Folio	Text	
Start Date	DateTime	Temporary Access Request Composite Smart
Finish Date	DateTime	
Originator	Text	
Status	Text	

q. Repeat the same process to **Assign** and **Create** properties for each of the following service properties. (You want to identify which properties come from the workflow service, which is why you are renaming these properties and to include "Workflow" in the property names.)

Service Property	SmartObject Property
Finish Date	*Workflow Complete Date*
Status	*Workflow Status*
View Flow	*Workflow View Flow*
RecordID	*Workflow Record ID*



Add Service Object Method

Service Object Method: Workflow Reporting Service.K2 Learning\Temporary Access Applic

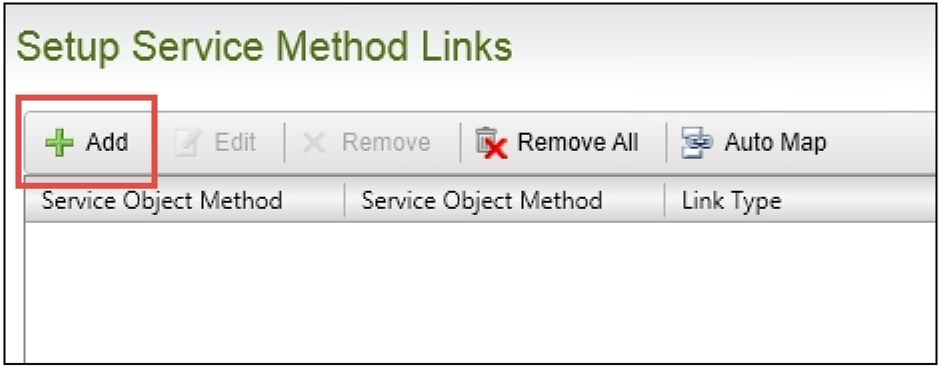
Folio	Text	
Start Date	DateTime	Temporary Access Request Composite Smart
Finish Date	DateTime	Temporary Access Request Composite Smart
Originator	Text	
Status	Text	Temporary Access Request Composite Smart
Duration	Number	
Priority	Text	
View Flow	Text	Temporary Access Request Composite Smart
RecordID	Number	Temporary Access Request Composite Smart

r. Click **OK** to close the **Add Service Object Method** screen.

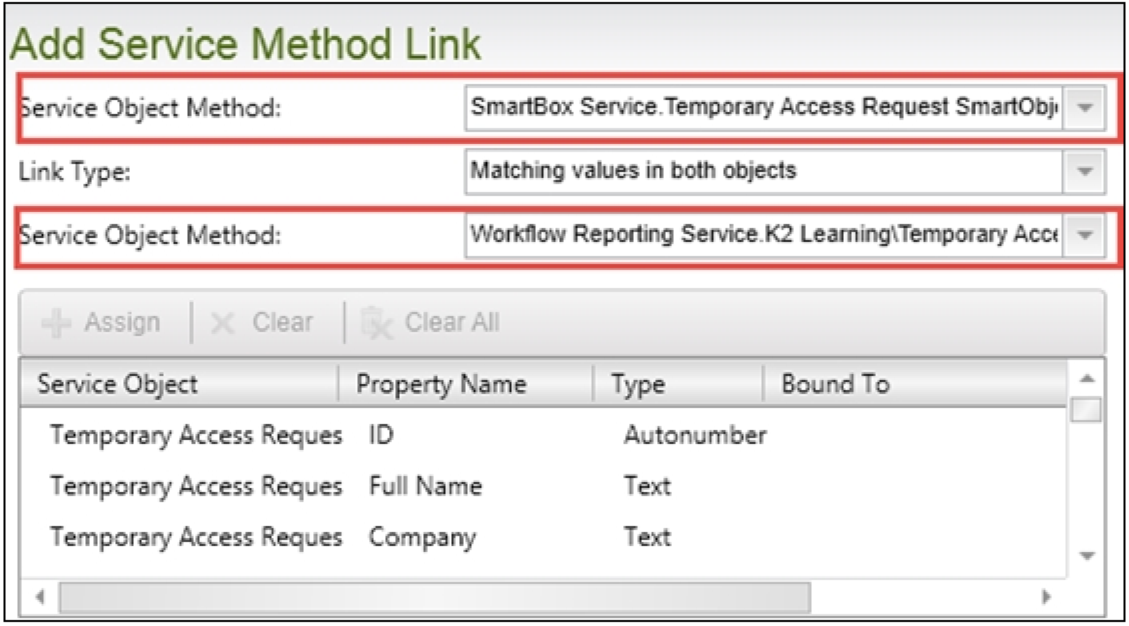
You have added the second service object method that is going to make up your composite SmartObject. In this case, you only wanted to return a few select properties from the workflow reporting service. You renamed the properties so that it is obvious that they came from the workflow service.

Importantly, you included the **RecordID** property from the workflow. This will be the common value that will allow you to join the data returned by the Workflow Reporting service with the Temporary Access Request records data returned by the SmartBox service. Remember that when you first set up the workflow, you configured it so that the form will create the record, get the record ID and then save the record ID back into the workflow. Doing this allows you to easily join the workflow reporting data with the business data.

s. Next you will configure the composite SmartObject to use a common property to join the data from the two services.
Click the **Next** button to go to the **Setup Service Method Links** screen, and then click **Add**.



t. In the **Add Service Method Link** screen, select **SmartBox...** service from the first drop-down menu and **Workflow Reporting...** service from the second drop-down menu. Select the option to return **Matching values in both objects**.



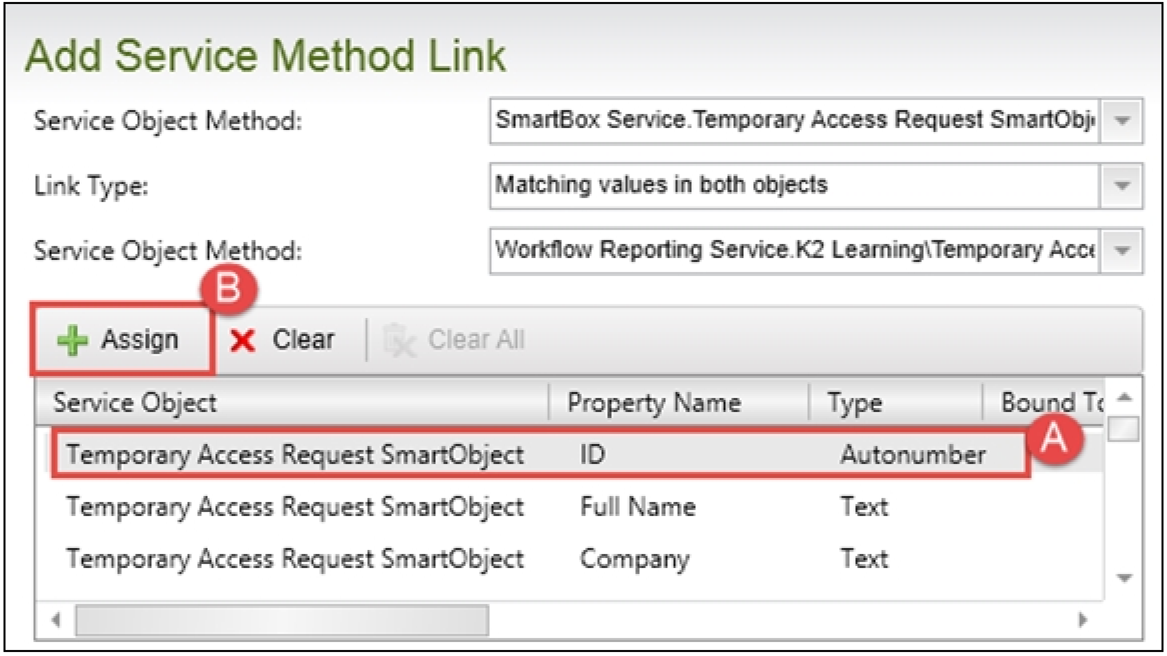
Service Object Method: SmartBox Service.Temporary Access Request SmartObj

Link Type: Matching values in both objects

Service Object Method: Workflow Reporting Service.K2 Learning\Temporary Acce

Service Object	Property Name	Type	Bound To
Temporary Access Reques	ID	Autonumber	
Temporary Access Reques	Full Name	Text	
Temporary Access Reques	Company	Text	

u. In the list of properties, locate and select the **Temporary Access Request SmartObject > ID** property, and click **Assign**.



Add Service Method Link

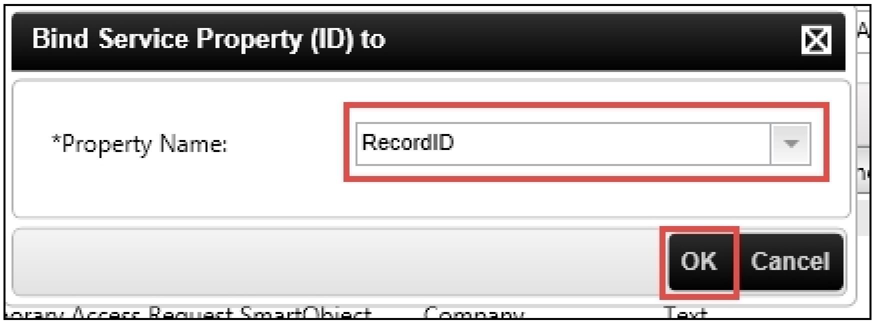
Service Object Method: SmartBox Service.Temporary Access Request SmartObj

Link Type: Matching values in both objects

Service Object Method: Workflow Reporting Service.K2 Learning\Temporary Acc

Service Object	Property Name	Type	Bound To
Temporary Access Request SmartObject	ID	Autonumber	
Temporary Access Request SmartObject	Full Name	Text	
Temporary Access Request SmartObject	Company	Text	

v. In the **Bind Service Property (ID) to** dialog box, select **RecordID** from the drop-down list, then click **OK**.



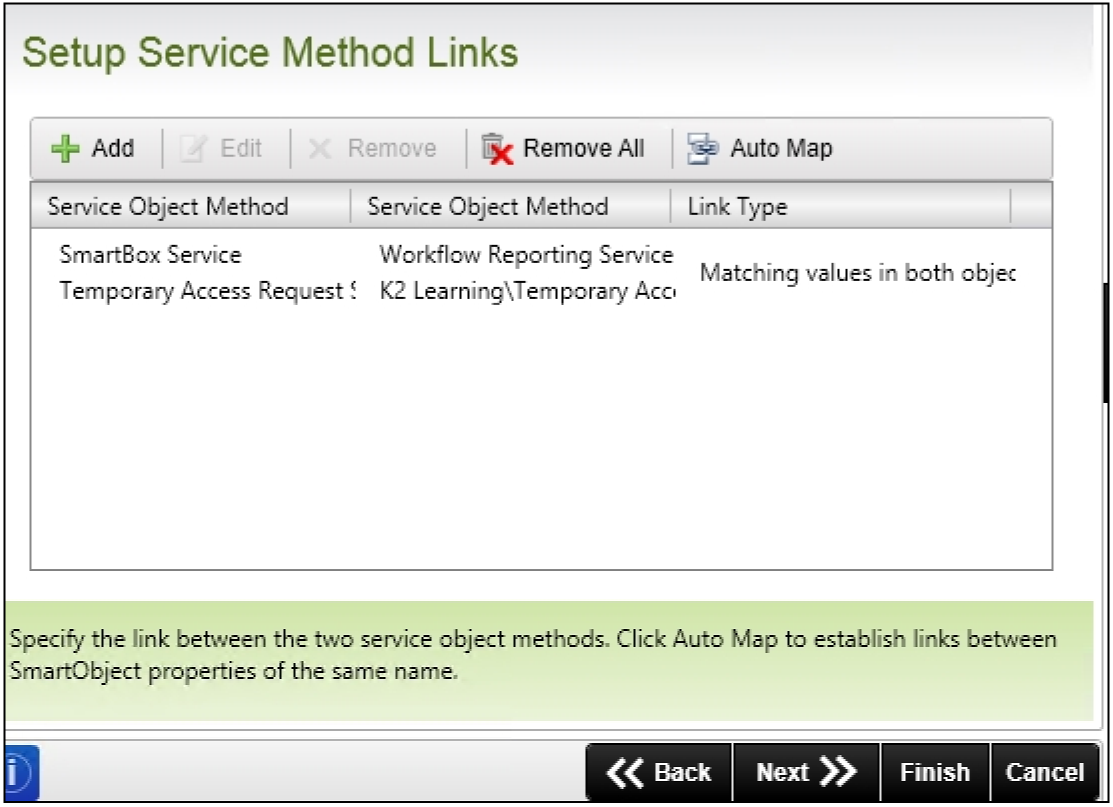
Bind Service Property (ID) to

*Property Name: RecordID

w. Click **OK** to close the **Add Service Method Link** screen.

In this step, you added the two service methods that return the data you need, selected which properties from those methods you want in your composite SmartObject, and told K2 how to join the data from the two services to one another with a common property (ID/RecordID).

x. Click **Next** on the **Setup Service Method Links** screen, and then click **Finish** to complete the wizard.



Service Object Method	Service Object Method	Link Type
SmartBox Service Temporary Access Request S	Workflow Reporting Service K2 Learning\Temporary Acc	Matching values in both objec

Specify the link between the two service object methods. Click Auto Map to establish links between SmartObject properties of the same name.

y. **Save** your work.

Step 14 Review

In this step you created a composite SmartObject that combines data returned by the SmartBox service (the Temporary Access Request records) with data returned by the Workflow Reporting Service (which is comprised of reporting data from all of the process instances of the Temporary Access Request Workflow). You selected which properties from each service to include in your SmartObject, then told K2 which common property it should use to join the two result sets to each other. (If you are familiar with the SQL language, it is almost the same as a JOIN statement in SQL: you have to tell the system which property to join the two result sets with and the tables you join are analogous to the service methods you are chaining together.)

Next you will deploy the new SmartObject.

Step 15: Deploy the new SmartObject

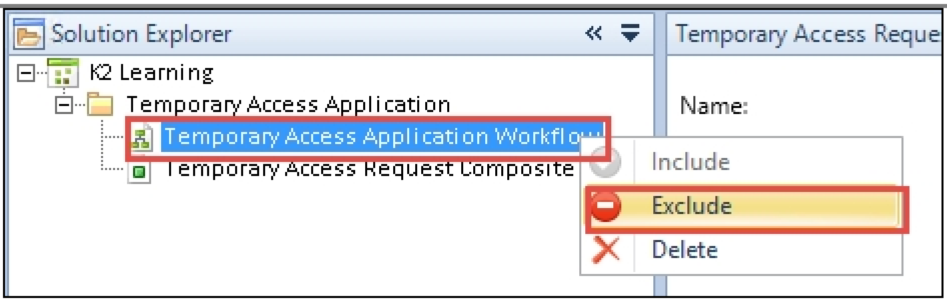
When you build workflows and SmartObjects with K2 Studio, you must deploy them before they are available in your K2 environment. In this step, you will deploy the new composite SmartObject and then test it. Because you have not made any changes to your workflow and don't need to re-deploy it, you will exclude the workflow project item from the deployment task.

Step 15 Tasks

1. Exclude the workflow from the project deployment, then deploy the project.
2. Test the new SmartObject in the K2 Management site.

Step 15 Walkthrough

a. You must deploy the new SmartObject to your K2 server, but you don't need to deploy the workflow because nothing has changed in the workflow. Open the Solution Explorer, then right-click the **Temporary Access Request Workflow**. Select the **Exclude** option.

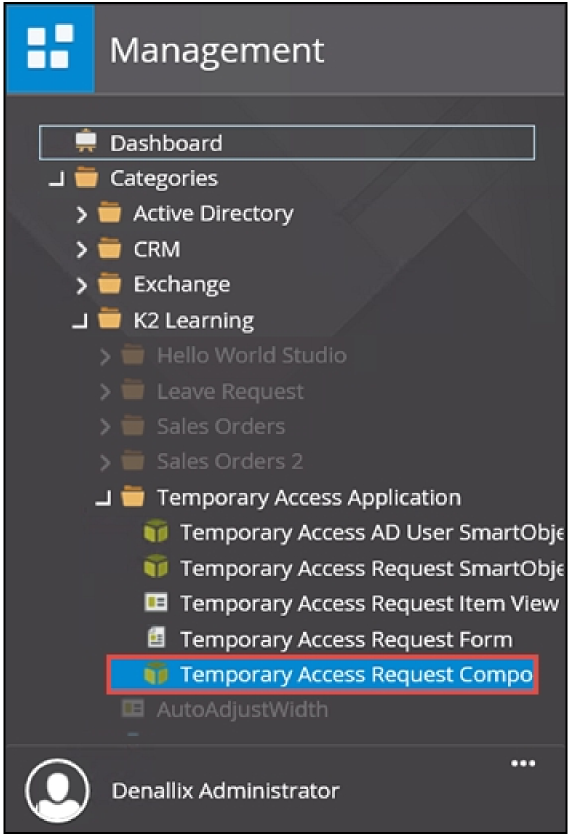

Solution Explorer
K2 Learning
Temporary Access Application
Temporary Access Application Workflow
Temporary Access Request Composite
Temporary Access Reque
Name:
Include
Exclude
Delete

b. Deploy the solution like you did before, by clicking **Next** through the deployment wizard. Because you excluded the workflow from the deployment, the deployment should complete quicker than it did the last time.

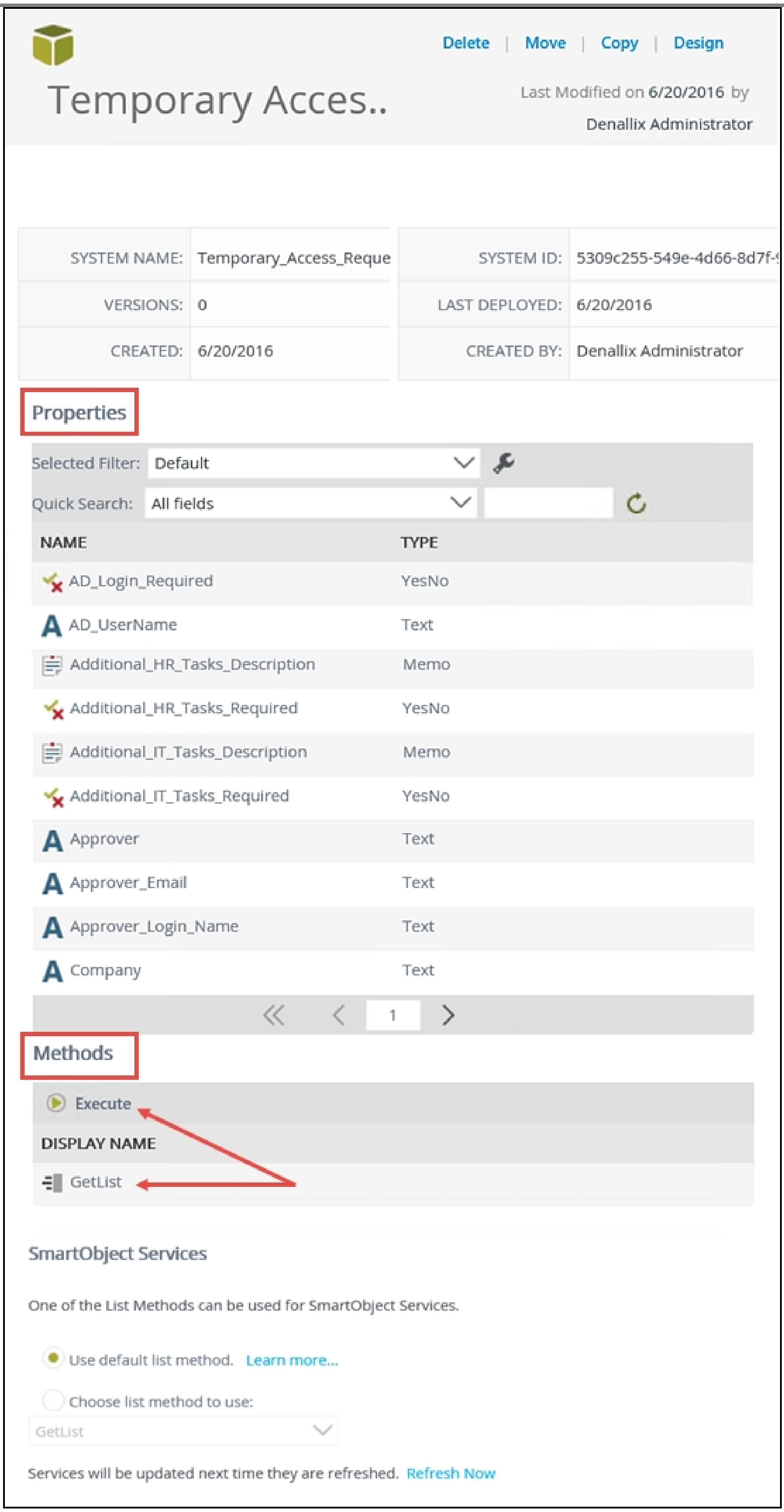
c. **Save** your work, and then close K2 Studio.

It is a good idea to test new SmartObjects before using them in forms and workflows. You can test SmartObjects directly from the K2 Management site.

d. Return to the K2 Management site in your browser. Expand the **Categories** > **K2 Learning** nodes, then **Temporary Access Application**. Click to highlight the **Temporary Access Request Composite SmartObject** (you may not be able to see the entire title).


Management
Dashboard
Categories
Active Directory
CRM
Exchange
K2 Learning
Hello World Studio
Leave Request
Sales Orders
Sales Orders 2
Temporary Access Application
Temporary Access AD User SmartObje
Temporary Access Request SmartObje
Temporary Access Request Item View
Temporary Access Request Form
Temporary Access Request Compo
AutoAdjustWidth
Denallix Administrator

e. The SmartObject's properties and methods are displayed in the central pane. Locate the **Methods** section, then highlight the **GetList** method and click **Execute**.



Temporary Acces..

Last Modified on 6/20/2016 by Denallix Administrator

SYSTEM NAME:	Temporary_Access_Reque	SYSTEM ID:	5309c255-549e-4d66-8d7f-!
VERSIONS:	0	LAST DEPLOYED:	6/20/2016
CREATED:	6/20/2016	CREATED BY:	Denallix Administrator

Properties

Selected Filter: Default

Quick Search: All fields

NAME	TYPE
AD_Login_Required	YesNo
AD_UserName	Text
Additional_HR_Tasks_Description	Memo
Additional_HR_Tasks_Required	YesNo
Additional_IT_Tasks_Description	Memo
Additional_IT_Tasks_Required	YesNo
Approver	Text
Approver_Email	Text
Approver_Login_Name	Text
Company	Text

Methods

Execute

DISPLAY NAME
GetList

SmartObject Services

One of the List Methods can be used for SmartObject Services.

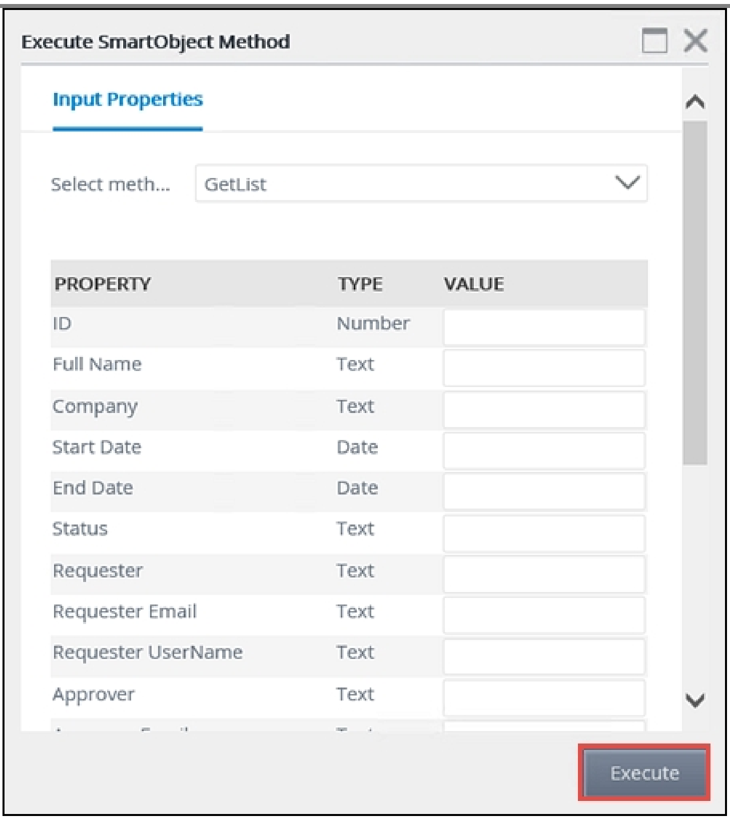
● Use default list method. Learn more...

○ Choose list method to use:

GetList

Services will be updated next time they are refreshed. Refresh Now

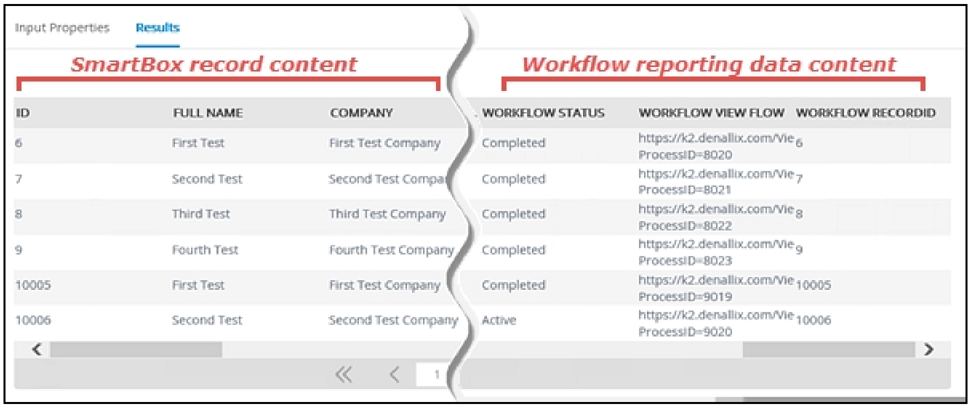
f. You want to return all of the records, so you don't have any input parameters. Click **Execute**.



Select meth... GetList

PROPERTY	TYPE	VALUE
ID	Number	
Full Name	Text	
Company	Text	
Start Date	Date	
End Date	Date	
Status	Text	
Requester	Text	
Requester Email	Text	
Requester UserName	Text	
Approver	Text	

You should now see a report of process instance data from your workflow. In particular, notice that the Full Name, Company, etc., is coming from the access request SmartBox records and the Workflow Status, etc., is coming from the workflow reporting data. (You will need to scroll to the right to see all of the returned properties.) By joining the two data sources together via a common property (ID and RecordID), you now have a report containing relevant information for your users.



ID	FULL NAME	COMPANY	WORKFLOW STATUS	WORKFLOW VIEW FLOW	WORKFLOW RECORDID
6	First Test	First Test Company	Completed	https://k2.denallix.com/Vie ProcessID=8020	6
7	Second Test	Second Test Compa	Completed	https://k2.denallix.com/Vie ProcessID=8021	7
8	Third Test	Third Test Company	Completed	https://k2.denallix.com/Vie ProcessID=8022	8
9	Fourth Test	Fourth Test Company	Completed	https://k2.denallix.com/Vie ProcessID=8023	9
10005	First Test	First Test Company	Completed	https://k2.denallix.com/Vie ProcessID=9019	10005
10006	Second Test	Second Test Company	Active	https://k2.denallix.com/Vie ProcessID=9020	10006

Step 15 Review

In this step you deployed the new SmartObject to your K2 environment. When you build artifacts with K2 Studio, you must always deploy those items to the K2 server. You can use the **Exclude** option to selectively include or exclude items from the deployment task – doing so can help to speed up your deployments. You also learned how to use the SmartObject execute function in the K2 Management site to test the new composite SmartObject.

Next you will create a view so that the data in this composite SmartObject can be shown to users.

Step 16: Create a new view for the composite SmartObject and add it to the request form as a tabbed view

In this step you will create a new SmartForm view that will display the data returned by the composite SmartObject you just deployed. You will add a rule to this view so that users can double-click a row to open the View Flow Report for a specific request. Finally, you will add the new view to the existing Temporary Access Request Form so that users can easily see all access requests from the same form used to capture new temporary access requests.

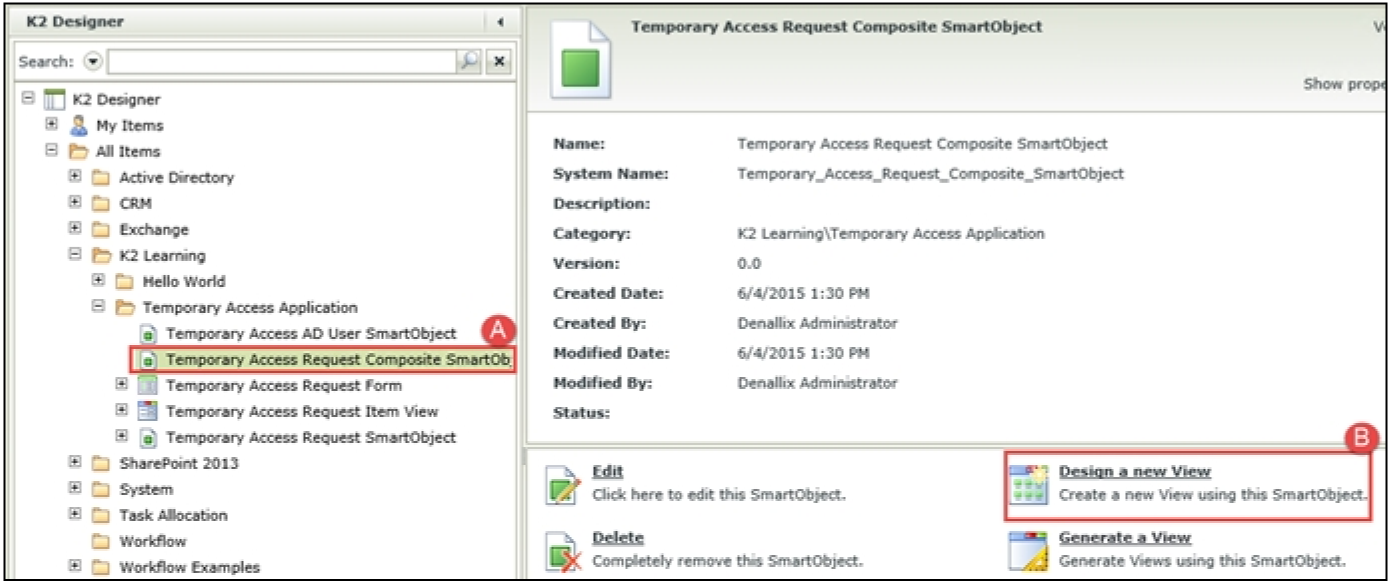
Step 16 Tasks

1. Design a new list view called
   Temporary Access Request List View
   for the composite SmartObject created in this exercise.
2. Add the following properties to the list view:
   Full Name
   Company
   Start Date
   End Date
   Status
   Requester
   Workflow Status
   Workflow View Flow
3. Add a rule that will navigate to a URL in a new window if a row is double-clicked in the view. Use the Workflow View Flow property of the SmartObject as the base URL for the navigate action.
4. Test the view.
5. Add the view to the **Access Requests Report** tab on the **Temporary Access Request Form**.
6. Check-in and then test the form to verify that the report data is returned and double-clicking a row opens the View Flow Report.

Step 16 Walkthrough

a. Open K2 Designer: **Start > All Programs > K2 blackpearl > K2 Designer**
   If the start menu link is not there, contact your K2 administrator for the URL to the K2 Designer environment.

b. Browse to the **All Items** > **K2 Learning** > **Temporary Access Application** folder where you should see the new composite SmartObject. Select the SmartObject, then select the **Design a new View** option.


K2 Designer
Search:
K2 Designer
My Items
All Items
Active Directory
CRM
Exchange
K2 Learning
Hello World
Temporary Access Application
Temporary Access AD User SmartObject
A
Temporary Access Request Composite SmartOb
Temporary Access Request Form
Temporary Access Request Item View
Temporary Access Request SmartObject
SharePoint 2013
System
Task Allocation
Workflow
Workflow Examples
Temporary Access Request Composite SmartObject
Show prope
B
Edit
Click here to edit this SmartObject.
Delete
Completely remove this SmartObject.
Design a new View
Create a new View using this SmartObject.
Generate a View
Generate Views using this SmartObject.

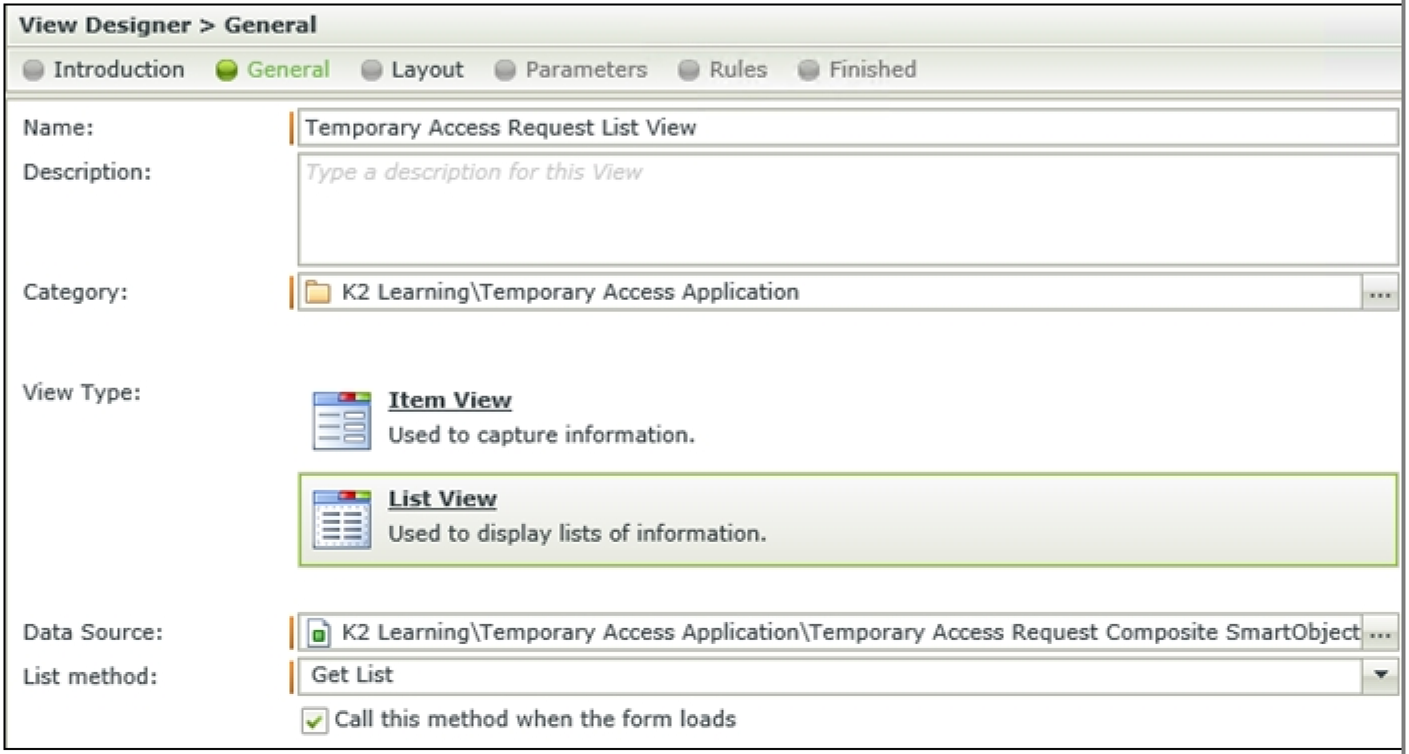
Name: Temporary Access Request Composite SmartObject
System Name: Temporary_Access_Request_Composite_SmartObject
Description:
Category: K2 Learning\Temporary Access Application
Version: 0.0
Created Date: 6/4/2015 1:30 PM
Created By: Denallix Administrator
Modified Date: 6/4/2015 1:30 PM
Modified By: Denallix Administrator
Status:

c. Click **Next** on the welcome page.

d. Configure the new view as follows, then click **Next**.

Field	Value

Name	*Temporary Access Request List View*
Category	*(leave default value)*
View Type	**List View**
Data Source	(leave default value, should automatically be set to the Composite SmartObject)
List Method	**Get List**
Call this method when the form loads	CHECKED



Name: Temporary Access Request List View

Description: Type a description for this View

Category: K2 Learning\Temporary Access Application

View Type:

Item View
Used to capture information.

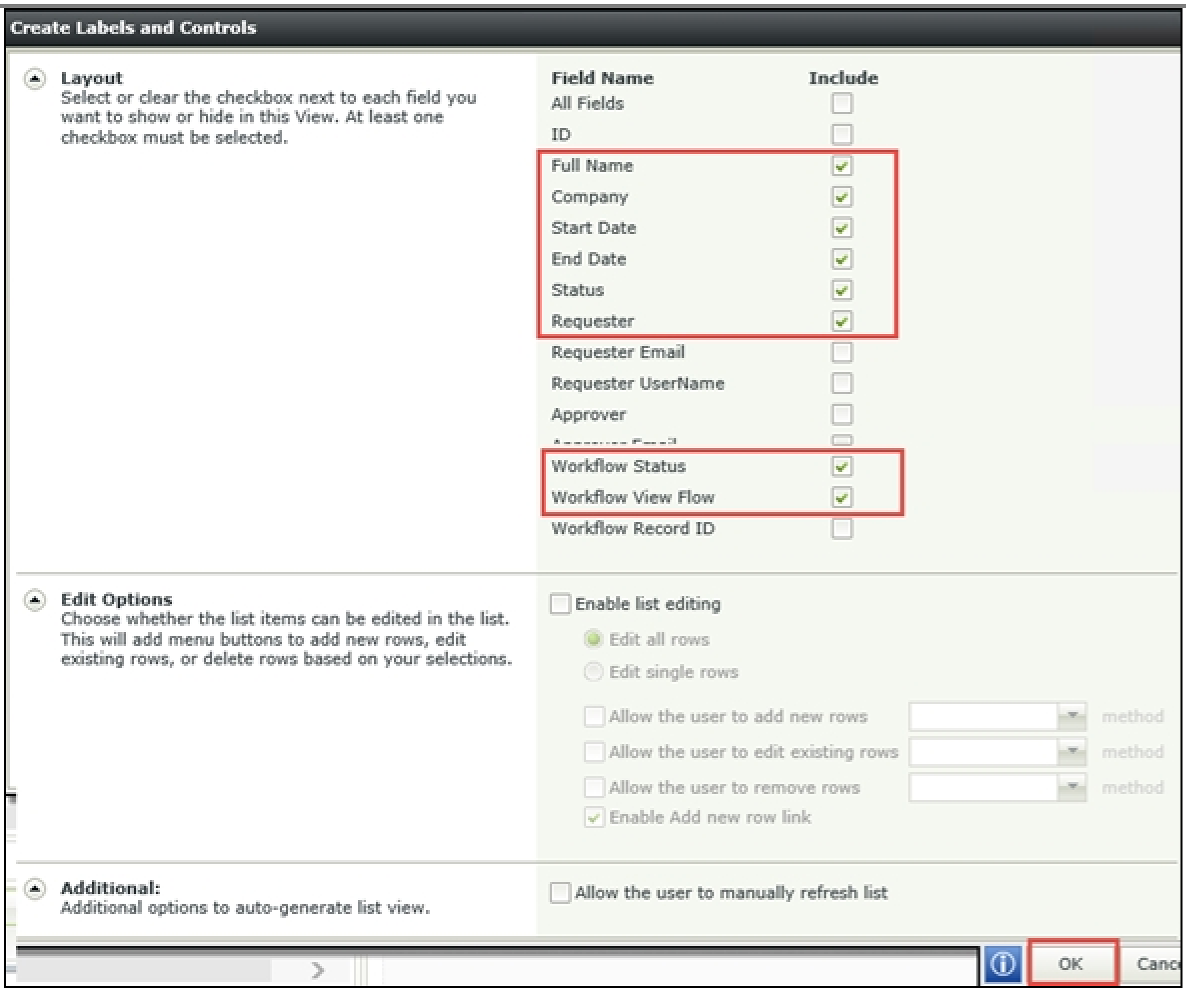
List View
Used to display lists of information.

Data Source: K2 Learning\Temporary Access Application\Temporary Access Request Composite SmartObject

List method: Get List

☑ Call this method when the form loads

e. Click the **Create Labels and Controls** link. In the dialog box that appears, select the following properties:
Full Name
Company
Start Date
End Date
Status
Requester
Workflow Status
Workflow View Flow
Leave the remaining options on this screen unchanged and click **OK**.



Create Labels and Controls

Layout
Select or clear the checkbox next to each field you want to show or hide in this View. At least one checkbox must be selected.

Field Name	Include
All Fields	□
ID	□
Full Name	☑
Company	☑
Start Date	☑
End Date	☑
Status	☑
Requester	☑
Requester Email	□
Requester UserName	□
Approver	□
Approver Email	□
Workflow Status	☑
Workflow View Flow	☑
Workflow Record ID	□

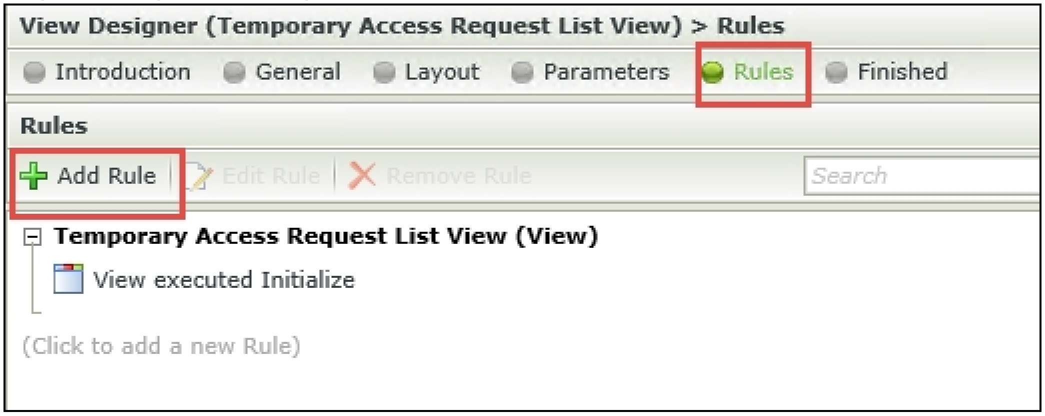
Edit Options
Choose whether the list items can be edited in the list. This will add menu buttons to add new rows, edit existing rows, or delete rows based on your selections.

□ Enable list editing
● Edit all rows
○ Edit single rows
□ Allow the user to add new rows — method
□ Allow the user to edit existing rows — method
□ Allow the user to remove rows — method
☑ Enable Add new row link

Additional:
Additional options to auto-generate list view.

□ Allow the user to manually refresh list

f. For this exercise, you will not be formatting this view, so click **Next** twice to move to the **Rules** screen. You will be adding a basic rule to this view so that when a user double-clicks a row, the View Flow Report will open for that process instance. Click on the **Add Rule** button.



View Designer (Temporary Access Request List View) > Rules

Rules

Temporary Access Request List View (View)
View executed Initialize

(Click to add a new Rule)

g. For the rule **Event**, select **When the View executes a method** and for the method, select **List Item Double Click**.

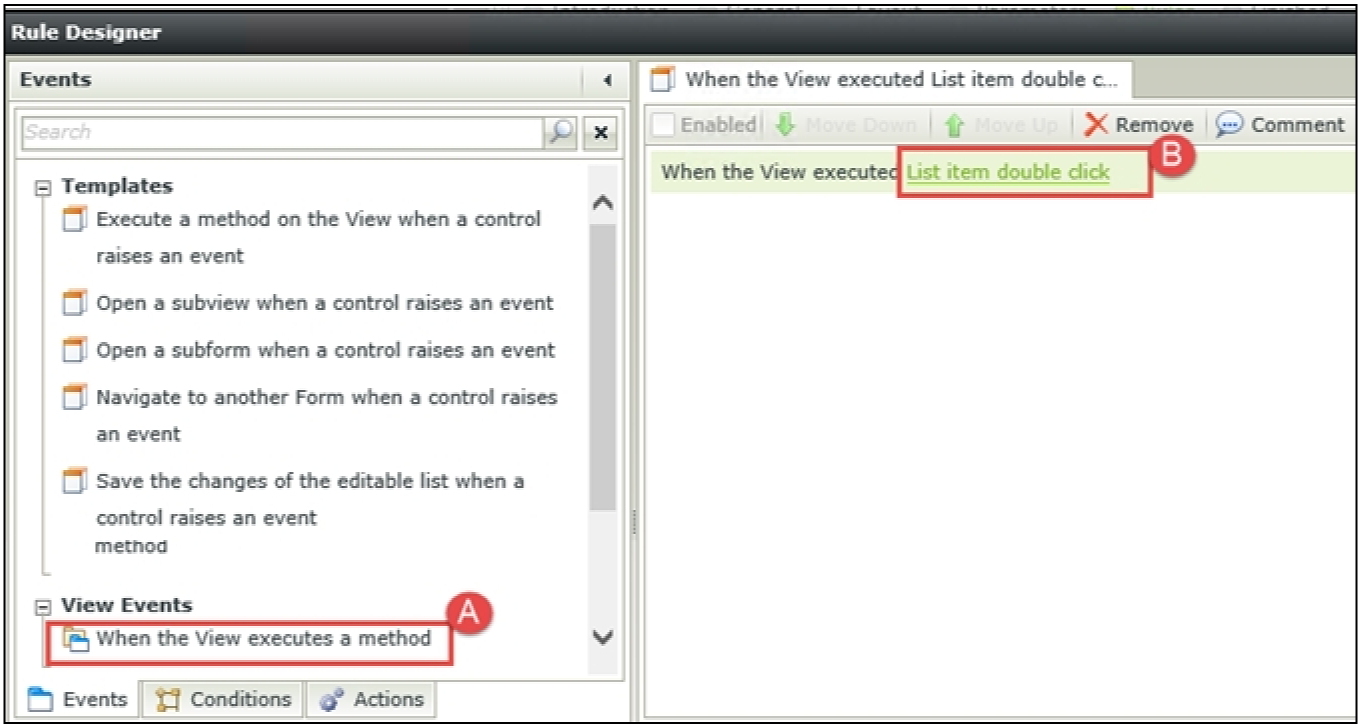
Note

Rules: Events, Conditions and Actions

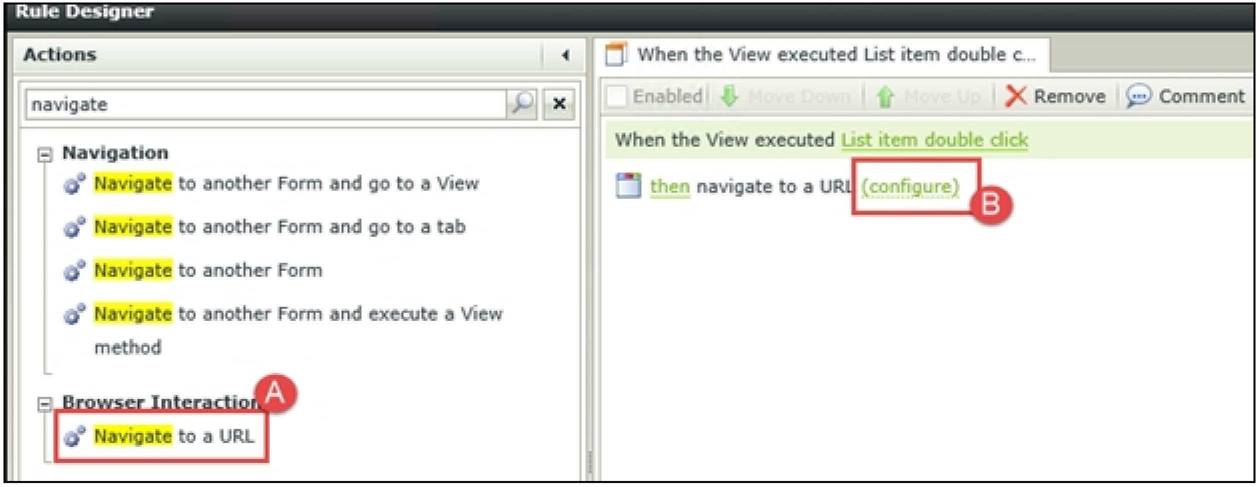
As a reminder, rules consist of events, conditions and actions.

- **Events** are *when something occurs*, such as clicking a button or when the form or view loads.
- **Conditions** evaluate *if a criteria has been met*: if the condition is true, continue. For example, evaluating a required field to determine if the field contains a value.

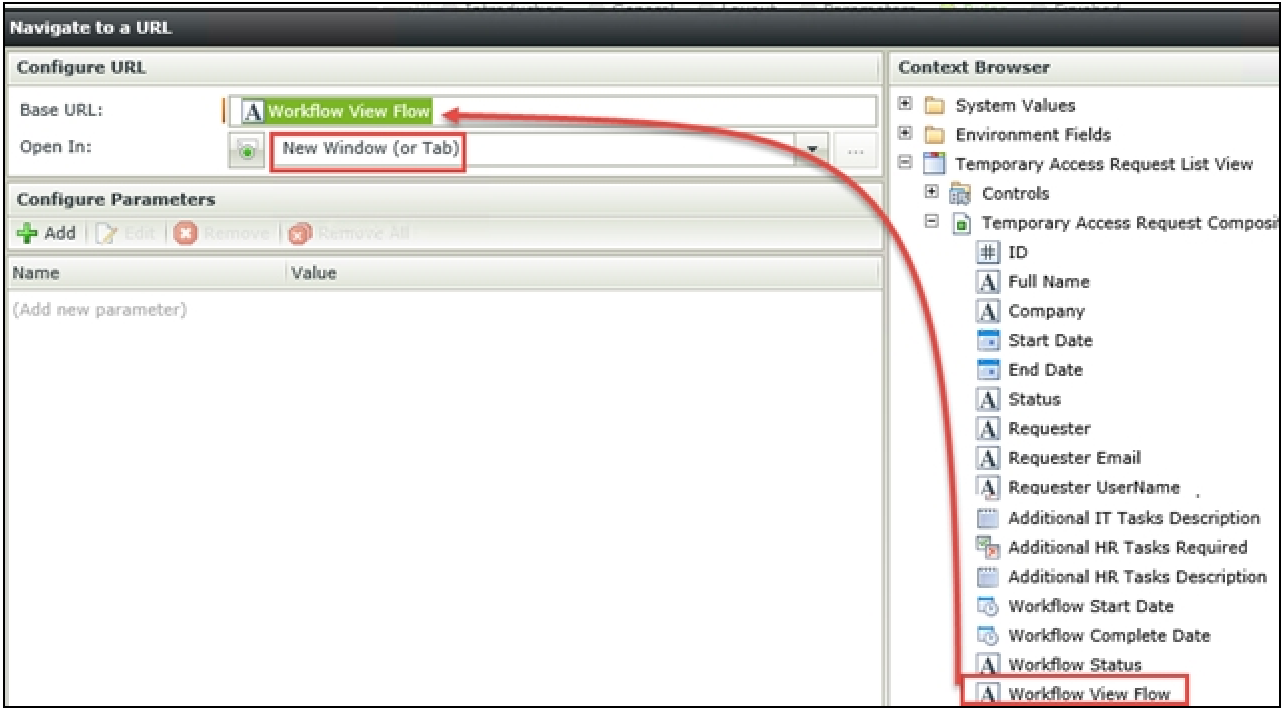
- **Actions** *do something*. If the event and condition passes, then perform the action, such as starting a workflow, or calling another rule, or enabling a form field.


Rule Designer
Events
Search
Templates
Execute a method on the View when a control raises an event
Open a subview when a control raises an event
Open a subform when a control raises an event
Navigate to another Form when a control raises an event
Save the changes of the editable list when a control raises an event
method
View Events
When the View executes a method
A
Events
Conditions
Actions
When the View executed List item double c...
Enabled
Move Down
Move Up
Remove
Comment
When the View executed List item double click
B

h. Select the **Actions** tab and then select the **Navigate to a URL** action. Click the **(configure)** link to set up the action.


Rule Designer
Actions
navigate
Navigation
Navigate to another Form and go to a View
Navigate to another Form and go to a tab
Navigate to another Form
Navigate to another Form and execute a View method
Browser Interaction
A
Navigate to a URL
When the View executed List item double c...
Enabled
Move Down
Move Up
Remove
Comment
When the View executed List item double click
then navigate to a URL (configure)
B

i. Drag the **Workflow View Flow** property into the **Base URL** text box, and select the **New Window (or Tab)** option. Click **OK** to complete the action configuration.



Navigate to a URL

Configure URL

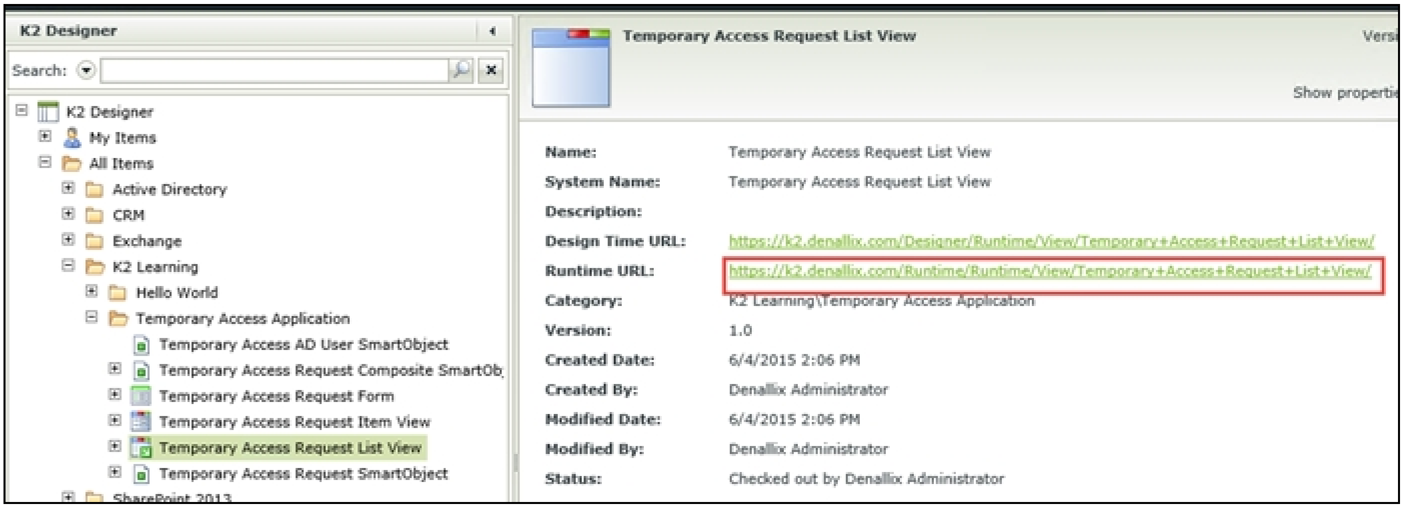
Base URL: Workflow View Flow

Open In: New Window (or Tab)

j. Click **OK** to close the rule configuration screen.

k. The view is complete, so click **Finish** to complete the wizard. (Finish is near the lower right corner of the screen, just below the properties pane.)

l. Before you add the new view to the request form, you should test it to confirm the rule configuration is functioning correctly. Make sure the view is selected in the category browser, then click on the **Runtime URL** link to open the view. You should see the temporary access requests you entered earlier, and if you double-click on a row, the View Flow Report for that request should open in a new window or tab.



Temporary Access Request List View

Name: Temporary Access Request List View

System Name: Temporary Access Request List View

Description:

Design Time URL: https://k2.denallix.com/Designer/Runtime/View/Temporary+Access+Request+List+View/

Runtime URL: https://k2.denallix.com/Runtime/Runtime/View/Temporary+Access+Request+List+View/

Category: K2 Learning\Temporary Access Application

Version: 1.0

Created Date: 6/4/2015 2:06 PM

Created By: Denallix Administrator

Modified Date: 6/4/2015 2:06 PM

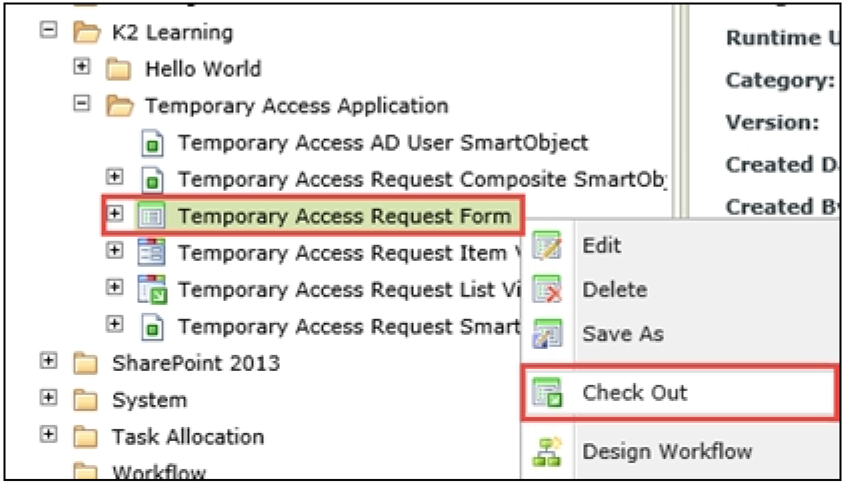
Modified By: Denallix Administrator

Status: Checked out by Denallix Administrator

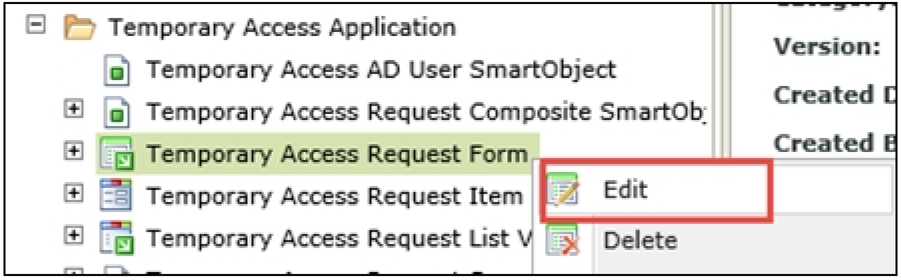
K2 Designer
Search:
K2 Designer
My Items
All Items
Active Directory
CRM
Exchange
K2 Learning
Hello World
Temporary Access Application
Temporary Access AD User SmartObject
Temporary Access Request Composite SmartOb
Temporary Access Request Form
Temporary Access Request Item View
Temporary Access Request List View
Temporary Access Request SmartObject
SharePoint 2013
Versi
Show properti

m. Next, you will add this view to the **Temporary Access Request Form**, so that users can easily see what temporary access requests were entered and add a new one. Locate the **Temporary Access**

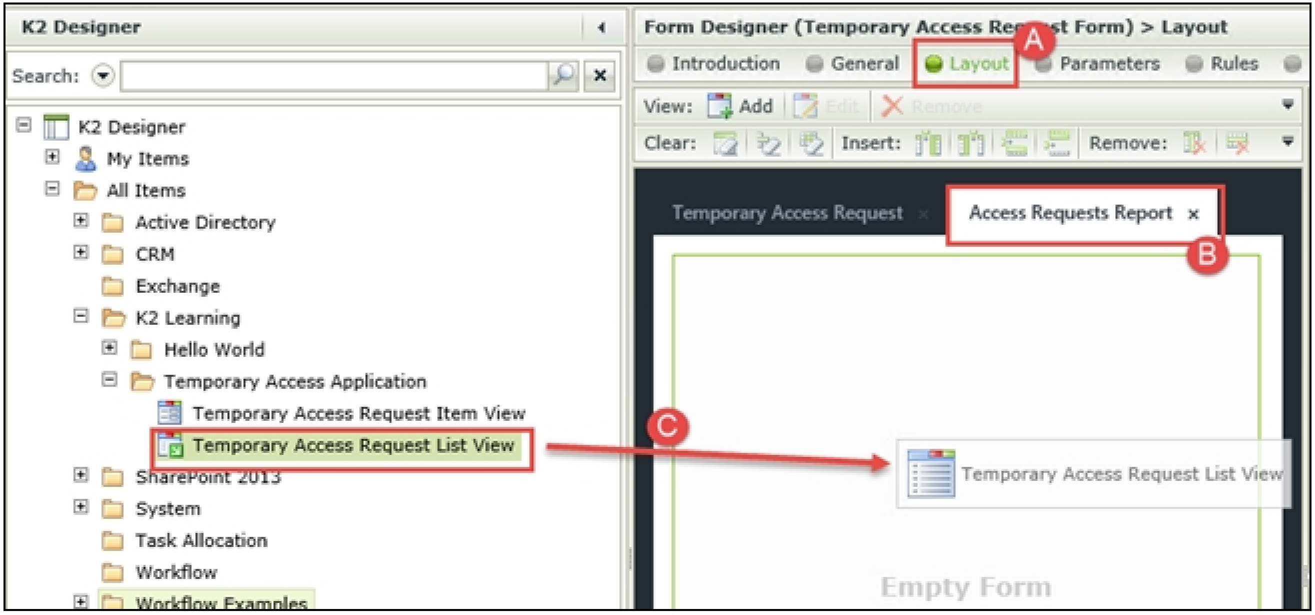
Request Form, right-click it and select **Check Out** (if it is not already).


K2 Learning
Hello World
Temporary Access Application
Temporary Access AD User SmartObject
Temporary Access Request Composite SmartOb
Temporary Access Request Form
Temporary Access Request Item V
Temporary Access Request List Vi
Temporary Access Request Smart
SharePoint 2013
System
Task Allocation
Workflow
Runtime U
Category:
Version:
Created D
Created B
Edit
Delete
Save As
Check Out
Design Workflow

n. Right-click the **Temporary Access Request Form** once again and select **Edit**.


Temporary Access Application
Temporary Access AD User SmartObject
Temporary Access Request Composite SmartOb
Temporary Access Request Form
Temporary Access Request Item
Temporary Access Request List V
Version:
Created D
Created B
Edit
Delete

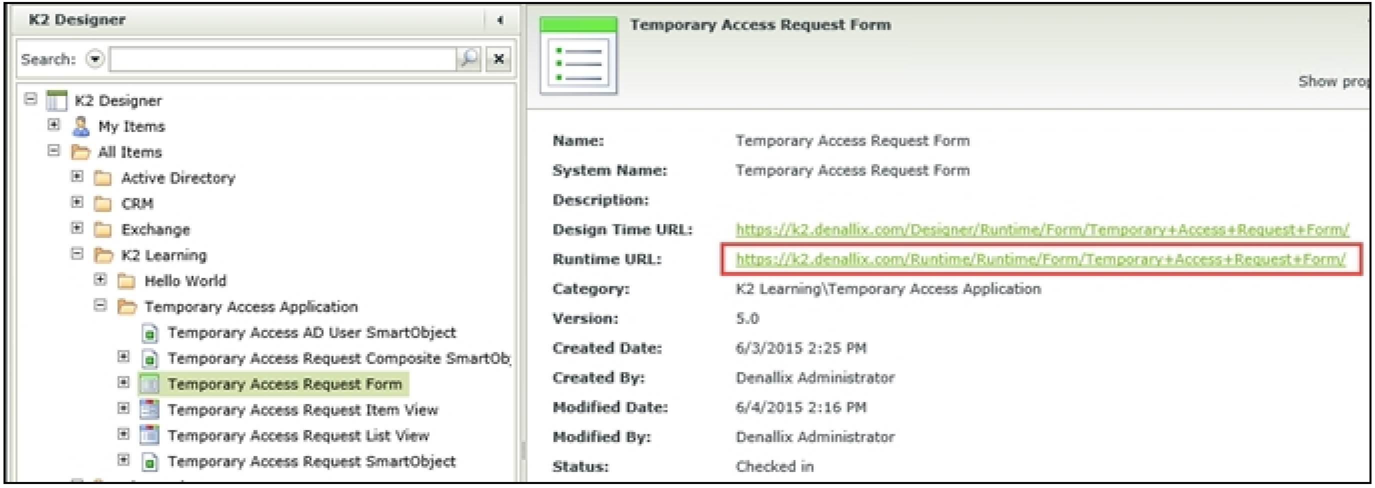
o. Skip to the **Layout** screen, then click the **Access Requests Report** tab. (The Access Request Report tab was added previously as part of the project package you deployed. See tutorials on SmartForms for instructions on how to add tabs.) From the category system, drag and drop the **Temporary Access Request List View** into the blank **Access Requests Report** tab.



p. Click **Finish** to complete the form changes.

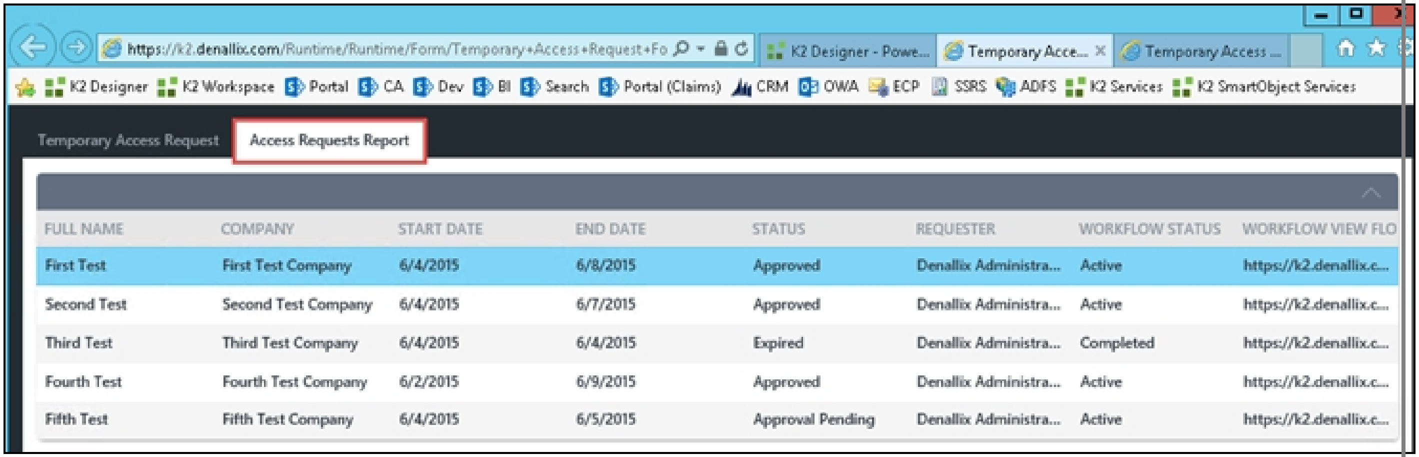
q. Right-click the **Temporary Access Request Form** and select **Check In**. Click **OK** on the warning message that appears. This message is simply stating that any associated views will be checked in as well.

r. Open the **Temporary Access Request Form** using the **Runtime URL** in the properties window, or using the browser bookmark that you saved earlier.


K2 Designer
Search:
K2 Designer
My Items
All Items
Active Directory
CRM
Exchange
K2 Learning
Hello World
Temporary Access Application
Temporary Access AD User SmartObject
Temporary Access Request Composite SmartOb
Temporary Access Request Form
Temporary Access Request Item View
Temporary Access Request List View
Temporary Access Request SmartObject
Temporary Access Request Form
Show prop

Name:	Temporary Access Request Form
System Name:	Temporary Access Request Form
Description:	
Design Time URL:	https://k2.denallix.com/Designer/Runtime/Form/Temporary+Access+Request+Form/
Runtime URL:	https://k2.denallix.com/Runtime/Runtime/Form/Temporary+Access+Request+Form/
Category:	K2 Learning\Temporary Access Application
Version:	5.0
Created Date:	6/3/2015 2:25 PM
Created By:	Denallix Administrator
Modified Date:	6/4/2015 2:16 PM
Modified By:	Denallix Administrator
Status:	Checked in

s. Click the **Access Requests Report** tab, and you should see all of the access requests listed, along with the workflow status column. Double-click any row to open the View Flow Report for that workflow instance.



FULL NAME	COMPANY	START DATE	END DATE	STATUS	REQUESTER	WORKFLOW STATUS	WORKFLOW VIEW FLO
First Test	First Test Company	6/4/2015	6/8/2015	Approved	Denallix Administra...	Active	https://k2.denallix.c...
Second Test	Second Test Company	6/4/2015	6/7/2015	Approved	Denallix Administra...	Active	https://k2.denallix.c...
Third Test	Third Test Company	6/4/2015	6/4/2015	Expired	Denallix Administra...	Completed	https://k2.denallix.c...
Fourth Test	Fourth Test Company	6/2/2015	6/9/2015	Approved	Denallix Administra...	Active	https://k2.denallix.c...
Fifth Test	Fifth Test Company	6/4/2015	6/5/2015	Approval Pending	Denallix Administra...	Active	https://k2.denallix.c...

Step 16 Review

In this step you created a view that uses the data returned by your composite SmartObject, along with a rule that will open the View Flow Report if a row is double-clicked. You then added that view to your application form so that users can easily see the temporary access requests and open the View Flow Report if they need to see more detail about the workflow.

This concludes the Temporary Access Application tutorials. You can close any open browser windows, K2 Studio or other K2 tools.

If everything worked as expected, congratulations!

Terms Of Use

ACCEPTANCE OF TERMS

THIS DOCUMENTATION IS SUBJECT TO THE FOLLOWING TERMS OF USE ("TOU"). SOURCECODE TECHNOLOGY HOLDINGS INC. ("SOURCECODE") RESERVES THE RIGHT TO UPDATE THE TOU AT ANY TIME WITHOUT NOTICE. THE MOST CURRENT VERSION OF THE TOU CAN BE REVIEWED BY CLICKING ON THE "TERMS OF USE" LINK IN THE TABLE OF CONTENTS.

COPYRIGHT

THE NAMES OF ACTUAL COMPANIES AND PRODUCTS MENTIONED HEREIN MAY BE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

THIS CONTENT AND ASSOCIATED RESOURCES ARE THE SOLE PROPERTY OF SOURCECODE TECHNOLOGY HOLDINGS, INC. AND SOURCECODE RESERVES ALL RIGHTS RELATED TO THE INFORMATION CONTAINED HEREIN. WITHOUT LIMITING RIGHTS UNDER COPYRIGHT, NO PART OF THIS CONTENT OR ITS RESOURCES, MAY BE REPRODUCED, STORED OR INTRODUCED INTO A RETRIEVAL SYSTEM, PRESENTED BY INDIVIDUALS OR ORGANIZATIONS IN TRAINING EVENTS OTHER THAN THOSE SANCTIONED BY SOURCECODE, OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, WITHOUT THE PRIOR WRITTEN CONSENT OF SOURCECODE TECHNOLOGY HOLDINGS, INC.

THIRD-PARTY INTEGRATION

SOURCECODE MAY MODIFY, ADJUST OR REMOVE FUNCTIONALITY IN THE K2 SOFTWARE IN RESPONSE TO CHANGES MADE TO VERSIONS OF THE RELEVANT MICROSOFT OR OTHER THIRD PARTY PRODUCTS. THIS DOCUMENTATION MAY PROVIDE ACCESS TO OR INFORMATION ON CONTENT, PRODUCTS, AND SERVICES FROM THIRD PARTIES. SOURCECODE TECHNOLOGY HOLDINGS INC. IS NOT RESPONSIBLE FOR AND EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND WITH RESPECT TO THIRD-PARTY CONTENT, PRODUCTS, AND SERVICES.

NOTICE REGARDING CONTENT OF THIS DOCUMENTATION

THE INFORMATION CONTAINED IN THIS DOCUMENT AND ITS ASSOCIATED RESOURCES, INCLUDING UNIFORM RESOURCE LOCATORS AND IDENTIFIERS, IS SUBJECT TO CHANGE WITHOUT NOTICE.

UNLESS EXPLICITLY STATED OTHERWISE, THE PEOPLE, ORGANIZATIONS, COMPANIES, PLACES, DOMAIN NAMES, E-MAIL ADDRESSES, PRODUCTS, LOGOS, AND EVENTS DEPICTED ARE FICTITIOUS AND NO ASSOCIATION WITH ANY ACTUAL PEOPLE, ORGANIZATIONS, COMPANIES, PLACES, DOMAIN NAMES, E-MAIL ADDRESSES, PRODUCTS, LOGOS, AND EVENTS IS INTENDED, OR SHOULD BE INFERRED UNDER ANY CIRCUMSTANCE.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS DOCUMENTATION, THE DOCUMENTS AND RELATED GRAPHICS CONTAINED IN THIS DOCUMENTATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. SOURCECODE TECHNOLOGY HOLDINGS INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE DOCUMENTATION AT ANY TIME AND WITHOUT NOTICE. ERRORS AND INACCURACIES MAY BE REPORTED IN WRITING TO DOCUMENTATION@K2.COM