

K2 Business Analysis

Hands-On Exercises Guide

Version: Version 3 Revision 1

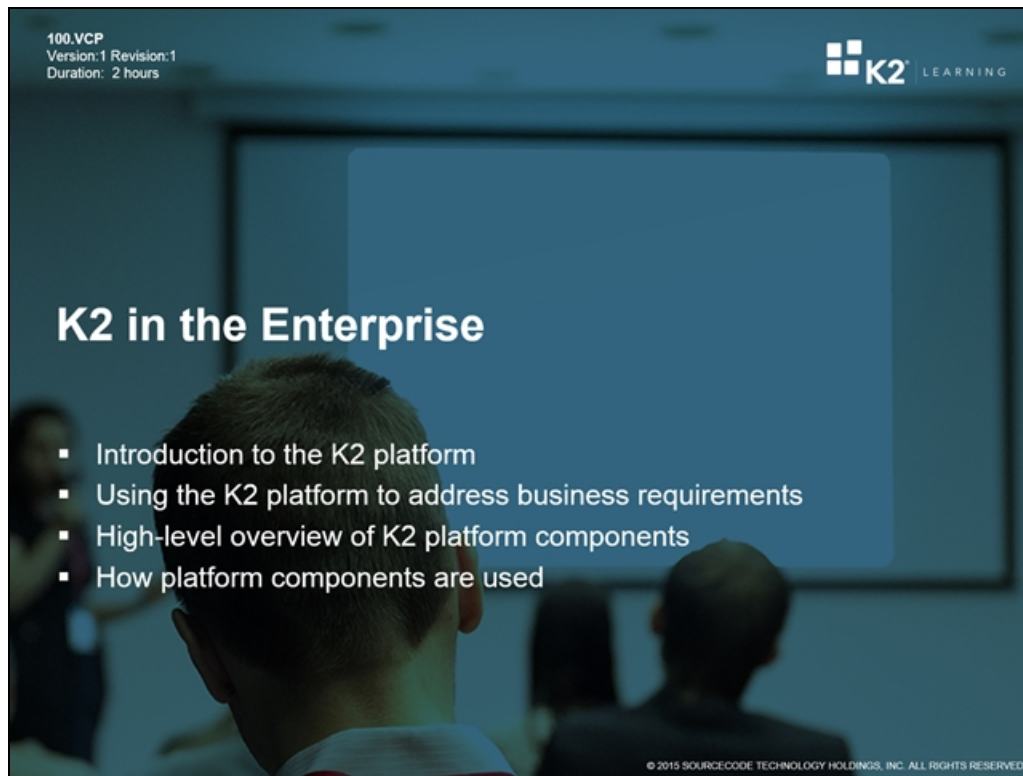
Date: January 2016

"Terms Of Use" on page 186

Table of Contents

Table of Contents	2
K2 in the Enterprise	3
PART 1: What K2 Means for the Enterprise	4
DEMO 1: A Sample K2 Application	5
PART 2: The K2 Platform: A High-Level Overview	21
DEMO 2: The Components of the Sample Expense Claim Application	22
100.BHX: Introduction to K2 Applications with K2 Designer	30
Part 1: K2 Application Basics	31
EXERCISE 1: Leave Application (Basic Version)	32
Part 2: Diving Deeper into Data, Forms and Workflows	82
EXERCISE 2: SmartObjects	83
EXERCISE 3: SmartForms	101
EXERCISE 4: Workflows	109
100.CWL: Reporting in K2	140
Part 1: Standard Reports in K2	141
EXERCISE 1: Standard K2 Reports	142
Part 2: Custom reporting in K2	163
EXERCISE 2: Building custom reports in K2 Workspace (Optional)	164
EXERCISE 3: Building Custom Reports with Microsoft Excel (Optional)	174
Terms Of Use	186
ACCEPTANCE OF TERMS	186
COPYRIGHT	186
THIRD-PARTY INTEGRATION	186
NOTICE REGARDING CONTENT OF THIS DOCUMENTATION	186

K2 in the Enterprise

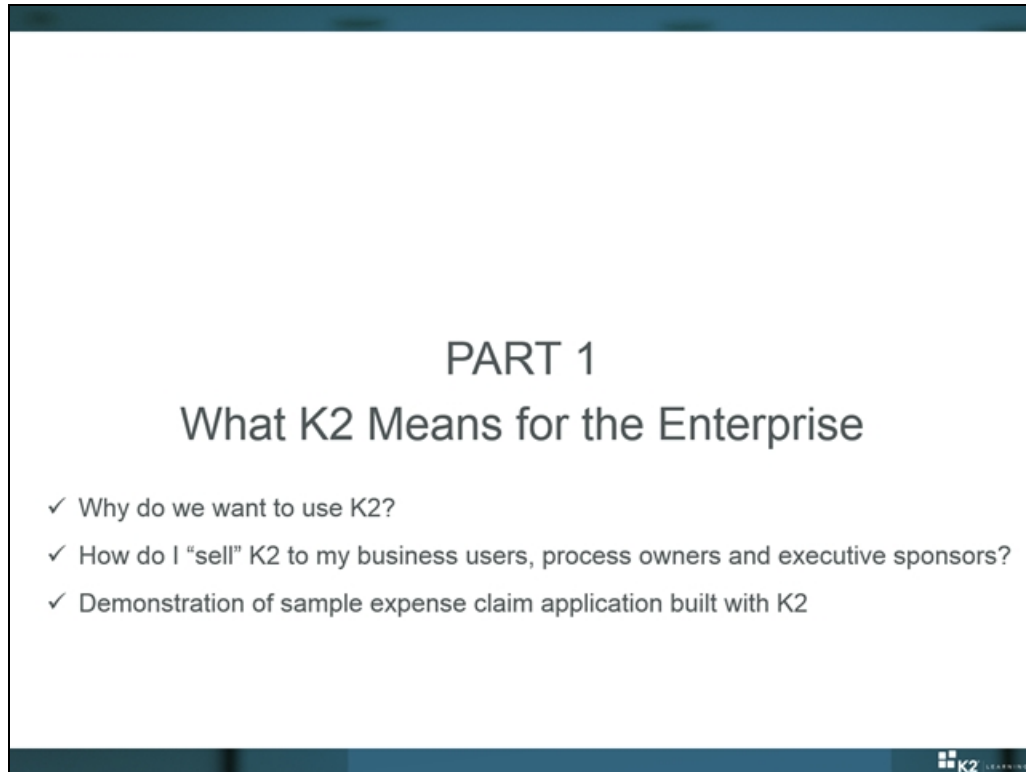


Welcome to K2 in the Enterprise. This module builds on concepts introduced superficially in the module 100.ABZ - Introduction to K2 and should take about 2 hours to complete. Roughly 50% of time will be spent on lecture and 50% on activities and exercises.

By the end of this module, you should be able to:

- Explain how the K2 platform addresses business requirements
- Describe K2 platform components in general
- Define BPA, BPM, and BPMS
- Explain why organizations use BPM and BPMS
- Explain why organizations use K2 for BPM and BPMS
- Explain what SmartObjects are and what they do
- Identify the five phases of BPM maturity development in organizations
- Identify the six stages in the BPM life-cycle


PART 1: What K2 Means for the Enterprise



PART 1

What K2 Means for the Enterprise

- ✓ Why do we want to use K2?
- ✓ How do I “sell” K2 to my business users, process owners and executive sponsors?
- ✓ Demonstration of sample expense claim application built with K2



In Part 1 we will look at the business benefits of K2. Next we will practice selling the business benefits of K2 to business users, process owners, and executive sponsors. At the end of Part 1, we will see a demonstration of a sample expense claim application.

DEMO 1: A Sample K2 Application

DEMO 1: A Sample K2 Application

Expense Claim Approval Process (Demo)

This flowchart illustrates the Denali Expense Claim Approval process

In this demonstration the instructor will demonstrate a sample K2 solution: an Expense Claim Approval application

[Video](#)

Note: the sample is not a comprehensive demonstration of all the features in K2. It is just an example solution for a particular requirement.

This demo shows a sample expense claim approval process. While this solution is not intended to cover everything possible in K2, it is a good example of how Data, Workflows, Forms, and Reports all work together to build a business application. This specific solution uses integration into various data sources, uses external business rules maintained in a SharePoint list, and features custom reporting to give the organization some BI into their expense claims.

Tip
A recording of this demonstration is available at <http://www.youtube.com/watch?v=Tx0jyPACUc>

Later on in this module we will look at the individual parts of the application to see how it was assembled.

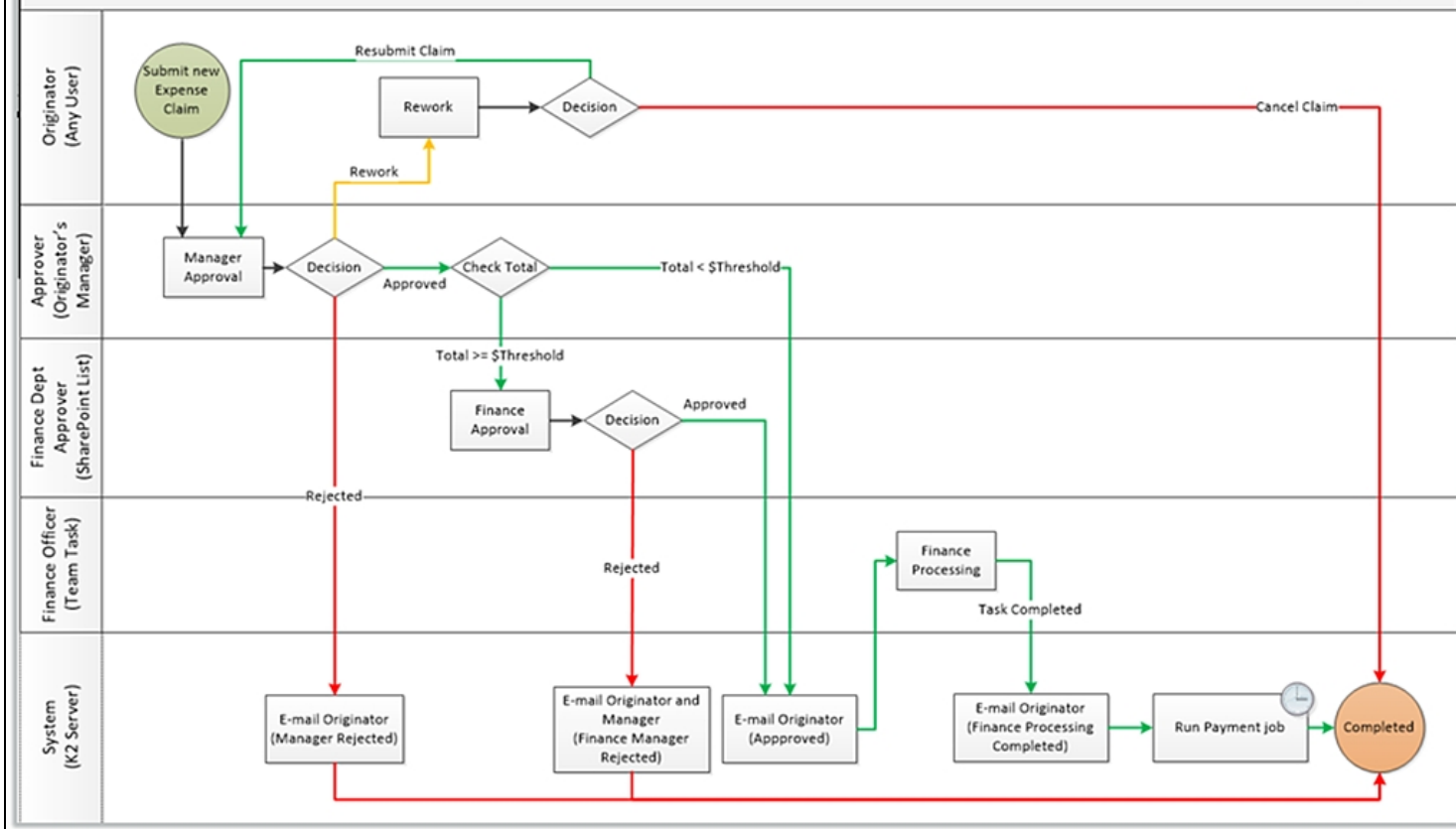
In the **200.BEL** K2 design methodology module, we will discover how this application was designed and a spec written so that it could be implemented. Don't worry about how the solution was built just yet. In Demo 2, you will dive into the components and show how the solution was assembled.

The diagram below shows the workflow components of the demonstration application. The instructor will talk you through the workflow. Remember there are many different ways that this application may have been implemented on K2, and many different applications that could be implemented on K2 using the same approach.

Expense Claim Expense Process

Expense Claim Approval Process (Demo)

This flowchart illustrates the Denallix Expense Claim Approval process



Demonstration 1: A sample K2 Application

This section covers the step-by-step instructions for completing the first of two demonstrations of a sample K2 Application. In this demonstration, we will submit, review and action a sample expense claim request. We will take a look at some Rules that determine the path the workflow will take and finally, we will see what happens when we reject a request.

Note

The demonstrations in this learning module use a pre-built Expense Claim application. This application must be downloaded and installed on a K2 Core 5.x (SharePoint 2010) virtual machine.

If you have not done so already, please download the following file from K2's help site:

<http://help.k2.com/files/10710>.

Once the zip file is downloaded, please extract the contents of the zip file to the virtual K2 environment, and then run the script `[Extracted Directory]\Expense Claim Application\Setup\SetupPreRequisites.bat` to install the necessary prerequisites and application components.

The application only needs to be installed and deployed on the instructor's virtual K2 environment, since it will only be used as a demonstration. It is not necessary nor recommended to install the application on participant's virtual environments.

Tip

If anyone is interested in learning how to build the application themselves, they can refer to the **K2 for SharePoint Application Builder** course or the [Expense Claim Approval application tutorial](#), where they will actually build a very similar solution as the one being demonstrated here.

Tip

If you are unable to install or live-demo the sample application, you can play a video of the demo, located at

<http://www.youtube.com/watch?v=Txe0jyPACUc>

Duration

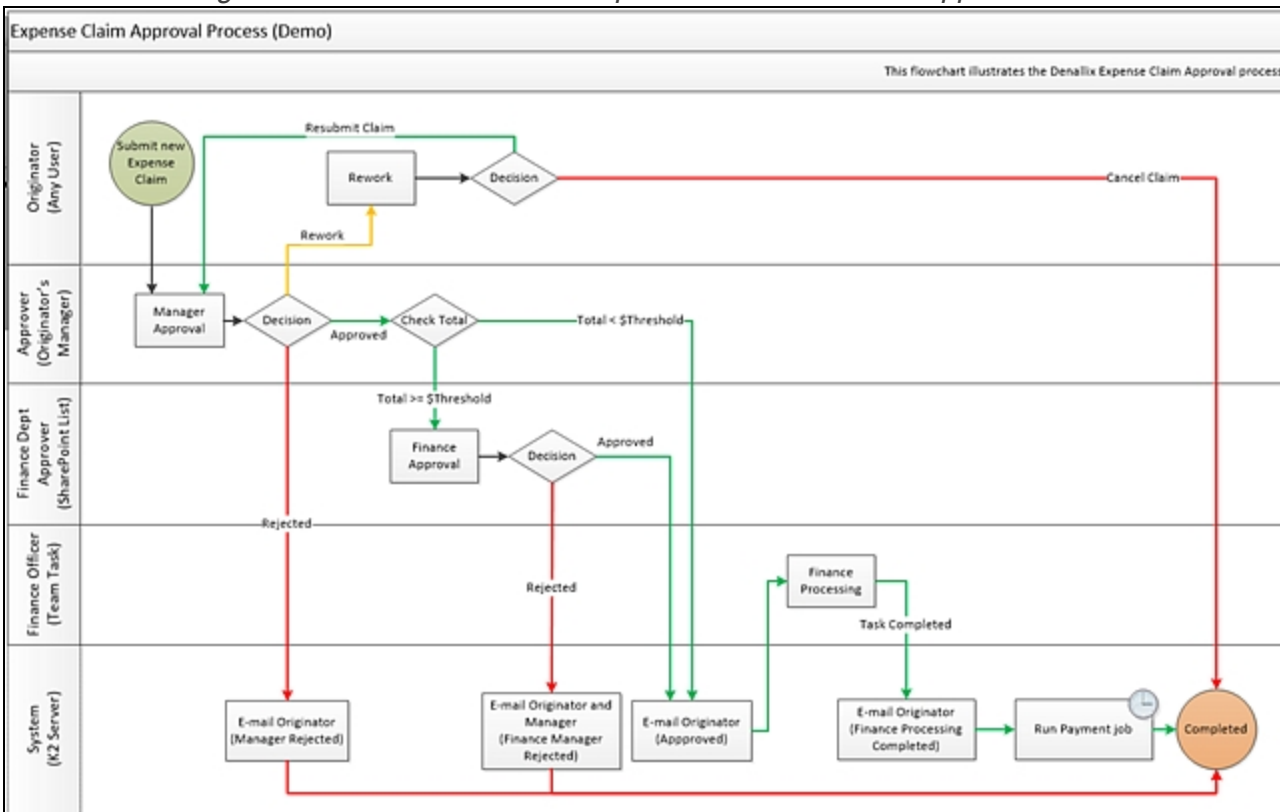
This demonstration should take around 15 minutes to complete.

Context

To demonstrate some of the concepts of K2 business applications, we have created a sample expense claim solution. The instructor will run through the application in the first demonstration, then explain how the solution was built with components of the K2 platform in the second demonstration.

Note: This solution is not intended to be a comprehensive demonstration of all the features of the K2 platform; it would just take too long to demonstrate everything in K2. Rather, think of this application as a sample of something you could build with K2.

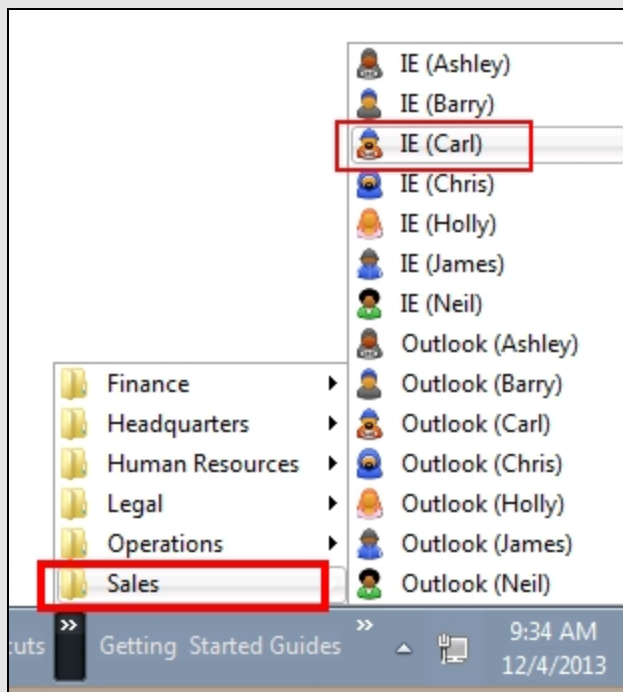
The workflow diagram below illustrates the basic process for the business application we are demonstrating



Step 1: Start a new expense claim on behalf of Carl

Step 1 Tasks

1. Close all Internet Explorer windows and all open instances of Microsoft Outlook.
2. Open Carl's IE browser.

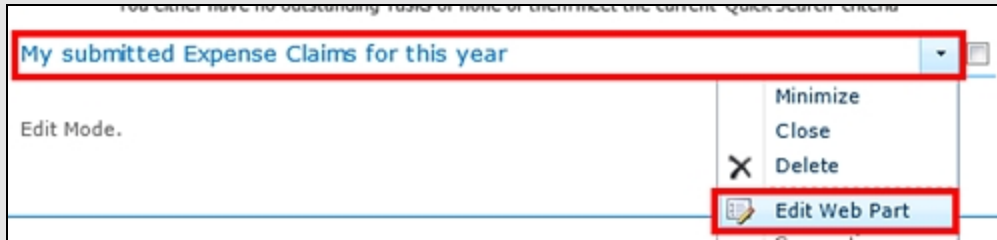


3. As Carl, browse to the SharePoint site for this demonstration: <http://portal.denallix.com/K2Learning/K2LearningExpenseClaimDemo>

If you do not use the direct URL link above, you may browse to the demo site by the following: Site Actions > All Site Content > K2 Learning > K2 Learning Expense Claim Demo.

4. Before we submit a new expense claim, we need to side-track for just a second to configure the **My submitted Expense Claims for this year** web part to show the ExpenseClaimDemo workflow instances. **Edit** the web part. CHECK the option for **ExpenseClaimDemo Expense Claim Approval** under the Processes filter. Confirm the options for **Processes I've started** and **This year** are also CHECKED.

Click OK to apply the update and close the edit window. After the expense claim is submitted, this web part will list the user's submitted expense claims, the data submitted and the current status. There will also be a View Flow link so that the process report can be conveniently opened and viewed as well.



← My submitted Expense Claims for this year

Settings

Processes

Filter

Select All

ExpenseClaimDemo Expense Claim Approval

TestK2EmailConnection

Denallix Expense Approval

Denallix leave request approval

Customer onboarding

Finance - Denallix Expense Approval

Show completed process instances

Processes I've started

Processes I've been involved in

Date Range

All

Started today

This week

This month

This quarter

This year

Sorting

Sort by default

Sort by

Start Date

Paging

Use paging

Number of items on the page

5. Click on the **Submit new Expense Claim** link to start a new expense claim.
6. The **Requester Info** view should be auto-populated with Carl's details. Make a note of the **Requester Manager** name, since this person will need to approve the expense claim. (It will probably be Holly.)

Expense Claim

Expense Claim Title:

Total Amount: Status:

Date Submitted: DATE

Date Processed:

Date Paid:

Requestor Comments:

Approver Comments:

Finance Comments:

Requester Info

Requestor: Carl Townsend

Email: Carl@denallix.com

Telephone Number: 206-555-1261

Department: Sales

Title: Sales Officer

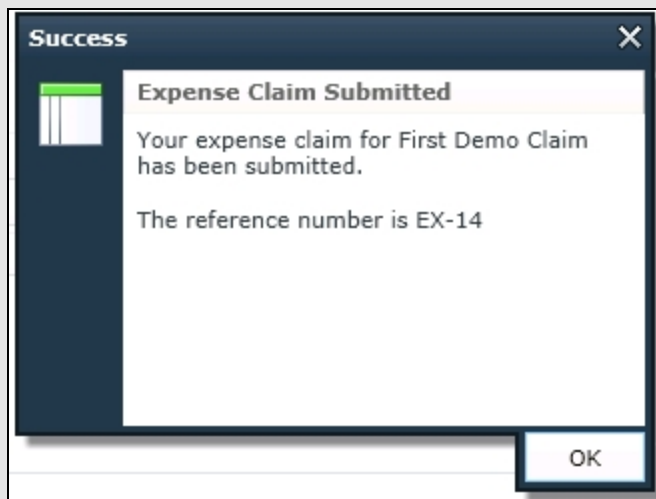
Requester Manager: → Holly Anderson

Expense Claim Items

+ Add Edit Delete

CATEGORY	PAYEE	DATE	AMOUNT	CURRENCY	EX. RATE	USD AMT	BILLABLE	COMMENT	ATTACHMENT
Select an it...	Type a value	Select a d... <small>DATE</small>	Type a value	Select an...		\$0.00	<input type="checkbox"/>	Type a value	Click here to...

7. Create a new expense claim with a title and comments of your choice. Add two expense claim line items to the expense claim, and attach receipts from the location (make sure the total amount is less than \$1000). **C:\K2 Learning\100.VCP\Sample Receipts**
8. **Submit** the new expense claim.



9. You should be redirected back to the **K2 Learning Expense Claim Demo** SharePoint site. (You may or may not see an item appear in the **My submitted Expense Claims...** web part, depending on the workflow timing). If necessary, wait a few seconds and refresh the page, you should see the new instance appear in the **My submitted Expense Claims...** web part.
10. Click on the **View Flow** link to show the progress of the submitted expense claim. Show that the approval task is sitting with Carl's manager (Holly).

Submit new Expense Claim

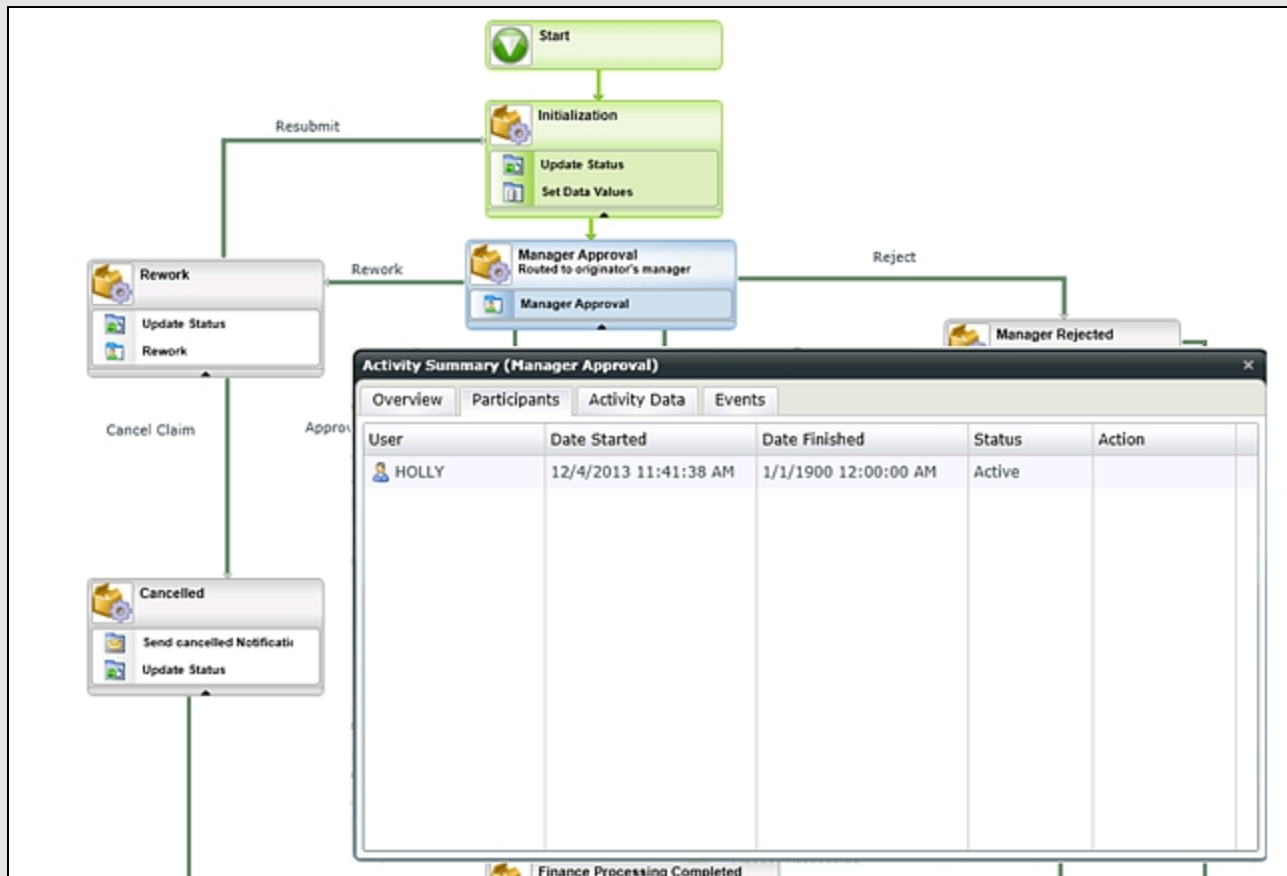
My K2 Worklist Tasks

Selected Filter: **Default** Quick Search: **All**

You either have no outstanding Tasks or none of them meet the current 'Quick Search' criteria

My submitted Expense Claims for this year

	Folio	Start Date	Current Duration	Current Activity	Status	View Flow
Expense Claim Approval	First Demo Claim	12/4/2013 11:41:32 AM	00:00:37	Manager Approval	Active	View Flow



11. Close all open browser windows and any open instances of Outlook.

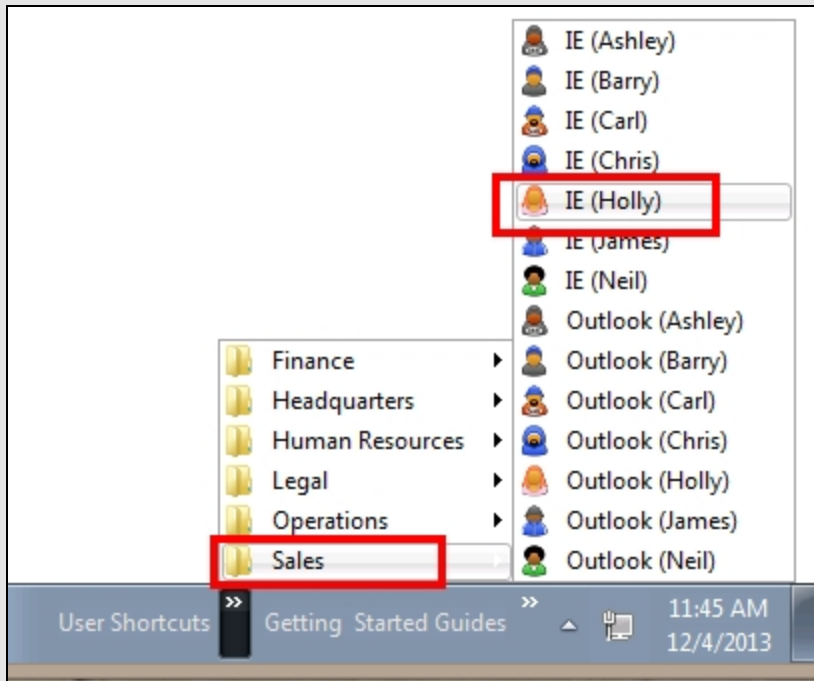
Step 1 Walkthrough

There is no Walkthrough for this step.

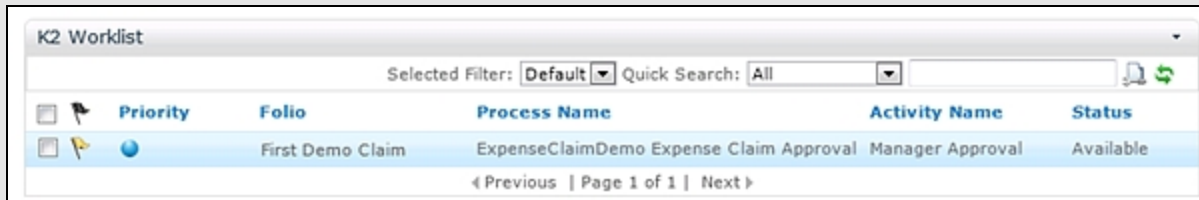
Step 2: Approve the new expense claim as Holly

Step 2 Tasks

1. Launch **Holly's** Internet Explorer.
If needed, Holly's login credentials are as follows:
Domain Name\Username: *denallix\holly*
Password: *K2pass!*



2. You should see a new task notification in the **K2 Worklist** task list. Click the link to open the expense approval Form.



3. When the form opens, review the expense claim details.

Notice that the attachment links are available and you can open the attached items for the expense claim.

Notice that all the controls apart from "Approver comments" are disabled, so the approver cannot make any changes on the form apart from adding comments and selecting the approval decision.

4. Add a comment and **Approve** the expense claim.

The screenshot shows the 'Demonstration: Expense Claim Solution' form, specifically the 'Manager Approval' section. The form displays the following details:

- Expense Claim Title:** First Demo Claim
- Total Amount:** \$600.00
- Status:** Submitted
- Date Submitted:** 12/4/2013
- Date Processed:**
- Date Paid:**
- Requestor Comments:** This is the first demo
- Approver Comments:** Holly's comments (highlighted with a red box)
- Finance Comments:**
- Decision:** Approve (highlighted with a red box) and a Submit button.

The 'Requestor' information is also visible on the right side of the form:

- Requestor:** Carl Townsend
- Email:** Carl@denallix.com
- Telephone Number:** 206-555-1261
- Department:** Sales
- Title:** Sales Officer
- Requestor's Manager:** Holly Anderson

At this point, Carl (as the form originator) should receive an email notification that his expense claim was approved. If time allows, open Carl's Outlook (again, from the User Shortcuts) and confirm the confirmation emails.

Demonstration Expense Claim Approval: Expense Claim Sample Expense Claim from Carl approved

k2service@denallix.com

Sent: Mon 1/4/2016 12:42 PM

To: Carl Townsend

Dear Carl Townsend

Your expense claim for Sample Expense Claim from Carl submitted on 04 Jan 2016 has been approved and will be sent to finance for processing.

Demonstration Expense Claim Approval: Sample Expense Claim from Carl processed

k2service@denallix.com

Sent: Mon 1/4/2016 12:43 PM

To: Carl Townsend

Dear Carl Townsend

Your expense claim for Sample Expense Claim from Carl submitted on 04 Jan 2016 has been processed and will be paid on 05 Jan 2016

You can review your expense claim using the following link:

<http://k2.denallix.com/Runtime/Runtime/Form/ExpenseClaimDemo+Expense+Claim+ReadOnly/?ExpenseClaimHeaderId=11>

5. Close all open Internet Explorer windows and any instances of Outlook.

Step 2 Walkthrough

There is no Walkthrough for this step.

Step 3: Review the completed expense claim View Flow report

Step 3 Tasks

1. Launch **Carl's** Internet Explorer.
2. Once again, browse to the SharePoint site for this demonstration: **http://portal.denallix.com/K2Learning/K2LearningExpenseClaimDemo**
3. In the **My submitted Expense Claims for this year** web part, the expense claim should have proceeded to the **Completed** stage. (This will be indicated in the Status column.)
4. Click on the **View Flow** link to show the progression of the expense claim workflow.
5. Leave Carl's browser open.

Step 3 Walkthrough

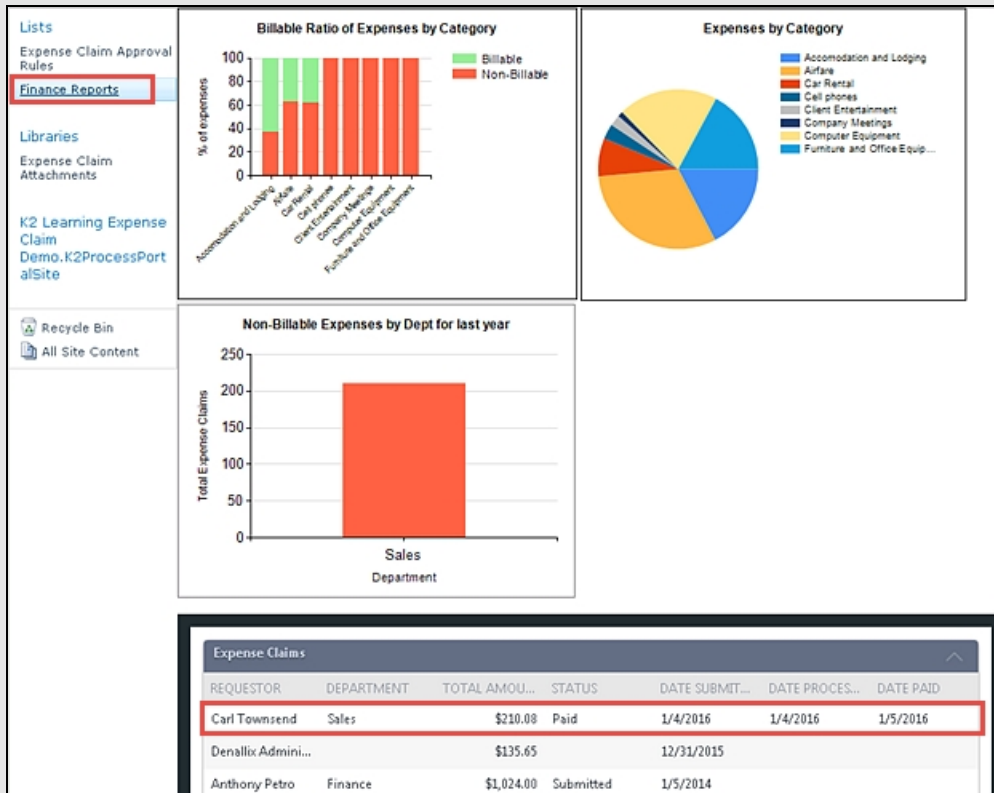
There is no walkthrough for this step.

Step 4: Review the expense claim reports

Step 4 Tasks

1. Open the **Finance Reports** link and locate the new expense claim that was submitted in the list of submitted expense claims.

Click the new expense claim just submitted on Carl's behalf (from the list of Expense Claims below the charts), and review the list of expense claim items for the expense claim. It should only show those items associated with the selected expense claim.



Expense Claim Information

Demonstration: Expense Claim Solution

Expense Claim

Expense Clai... Sample Expense Claim from Carl

Total Amount: \$210.08 Stat... Paid

Date Submitt... 1/4/2016

Date Processed: 1/4/2016

Date Paid: 1/5/2016

Requestor Co...

Approver Co... Holly's comments for approval.

Finance Com...

Requestor

Requestor: Carl Townsend

Email: Carl@denallix.com

Telephone Number: 206-555-1261

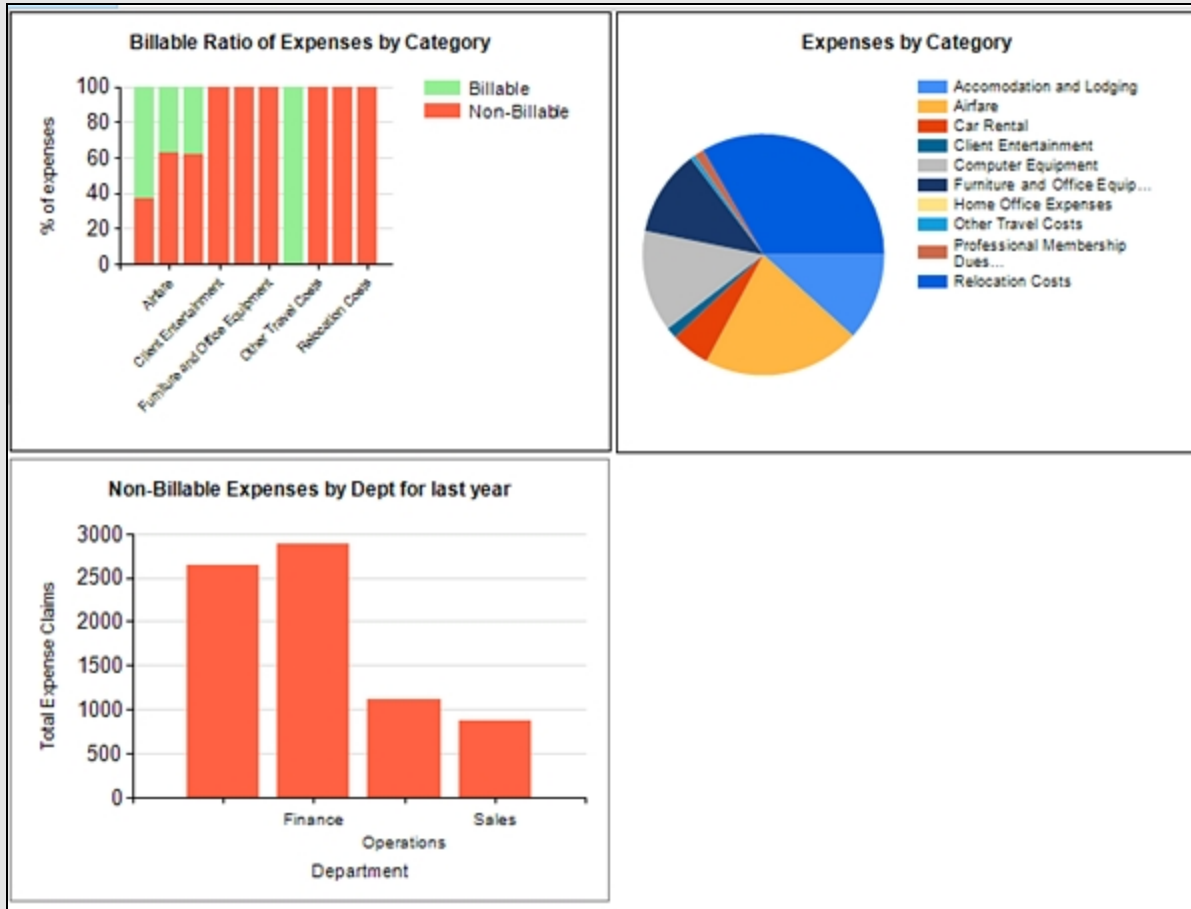
Department: Sales

Title: Sales Officer

Requestor Manager: Holly Anderson

CATEGO...	PAYEE	DATE	AMOUNT	CURREN...	EX. RA...	USD AMT	BILL...	COMMENT	ATTACH...
Company...	Payee one	1/1/2016	55.00	US Dollar	1.00	\$55.00	false	Comment sample one.	http://porta
Cell phones	Payee two	1/1/2016	155.08	US Dollar	1.00	\$155.08	false	Comment sample two	http://porta

2. Briefly show the custom reports that display expense claim Business Intelligence Information



3. Leave Carl's browser open.

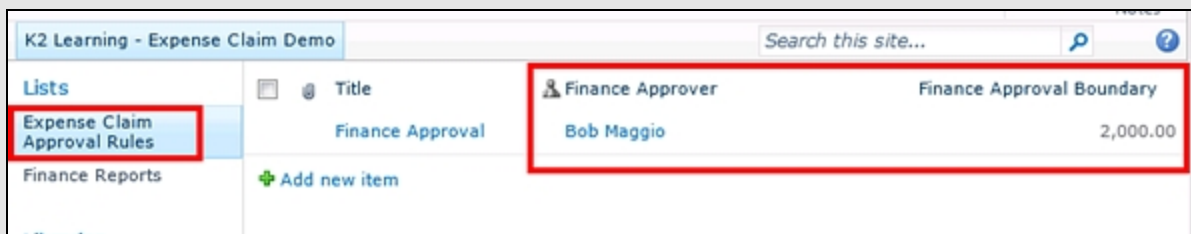
Step 4 Walkthrough

There is no Walkthrough for this step.

Step 5: Show the expense claim Rules list

Step 5 Tasks

1. Open the **Expense Claim Approval Rules** list, and show that there is an external Business Rule that determines whether or not finance approval is required, and who the approver is.



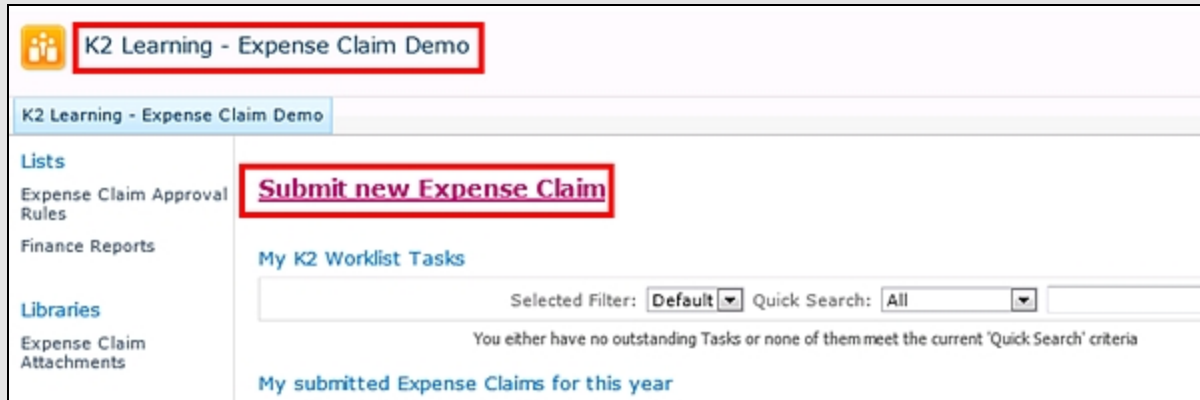
Step 5 Walkthrough

There is no Walkthrough for this step.

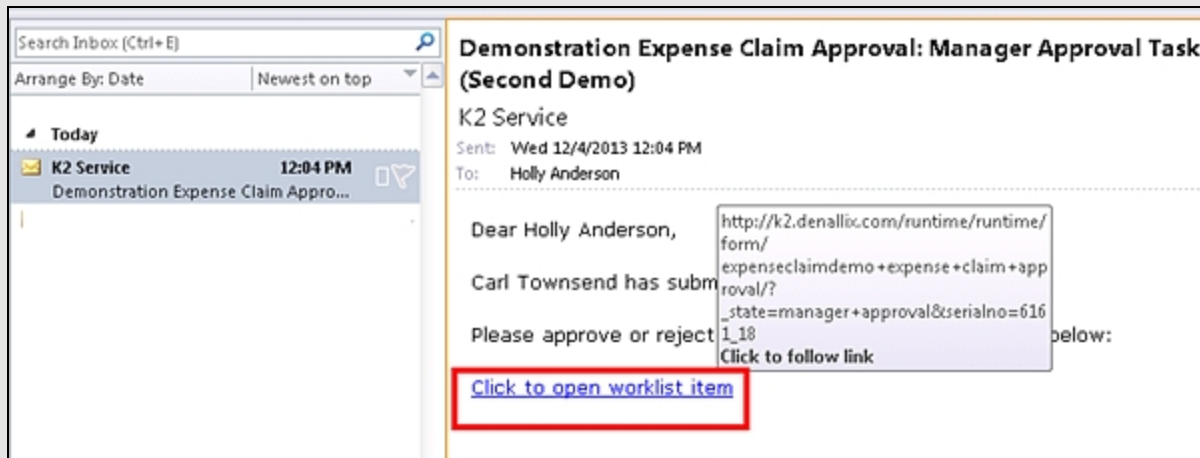
Step 6: Submit a new expense claim for MORE than \$2000 as Carl, then approve the request as Holly

Step 6 Tasks

- Return to the main page for the Expense Claim Demo SharePoint site and click the link to **Submit new Expense Claim**.



- Submit the new expense claim with two or more items, but ensure that the total is MORE than \$2000, so that it will go through to Finance for approval.
- Close all open Internet Explorer windows and any instances of Outlook.
- Open Holly's outlook and click the open task list in the notification email for the second expense claim that you just submitted



- As Holly, enter some comments in the Approver Comments box, then select '**Approve**' for the Decision and submit the form.

Approver Comments:

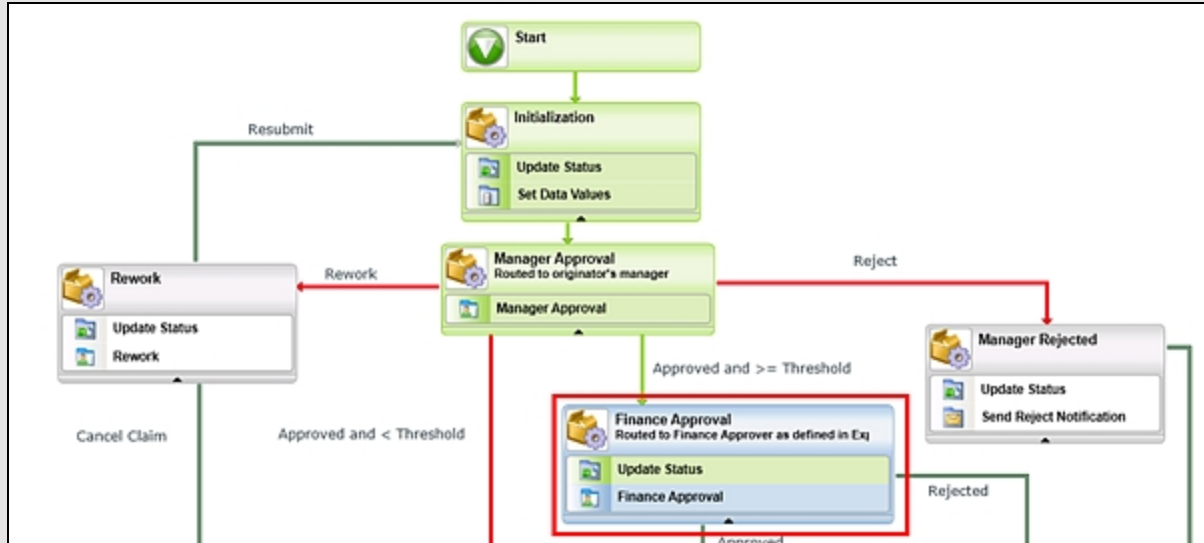
Finance Comments:

Decision:

- Close all open Internet Explorer windows and any instances of Outlook.
- Launch Internet Explorer as Administrator (current user). Use K2 Workspace to launch the **Process Overview** report for the second expense claim.

Process Folio	Originator	Status	Priority	Start Date	Finish Date	Duration
Sample Expense Claim from Carl	K2:DENALLIX\CARL	Completed	Medium	1/4/2016 10:29:57 AM	1/4/2016 12:43:09 PM	00:02:13:12
Expense Claim test more than \$2000	K2:DENALLIX\CARL	Active	Medium	1/4/2016 1:54:29 PM		00:00:11:59

- Open the View Flow report and show that the workflow went to Finance for approval, because the total amount was over the limit defined in the SharePoint list for Expense Claim 'Rules'.



- Open the active Activity and verify that the task is assigned to Bob.

Activity Summary (Finance Approval)

Activity Summary (Finance Approval)			
Overview	Participants	Activity Data	Events
User	Date Started	Date Finished	Status
BOB ←	1/4/2016 2:05:24 PM	1/1/1900 12:00:00 AM	Active

- Close all open Internet Explorer windows and any instances of Outlook.

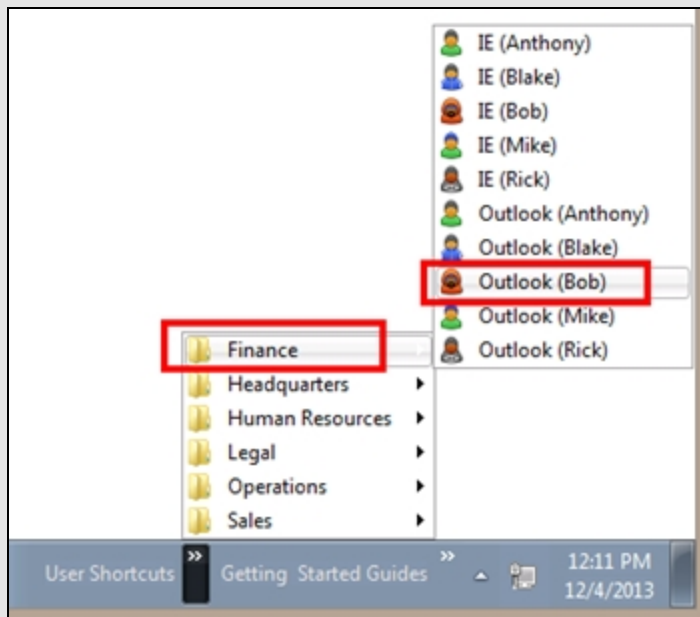
Step 6 Walkthrough

There is no Walkthrough for this step.

Step 7: Reject the expense claim request

Step 7 Tasks

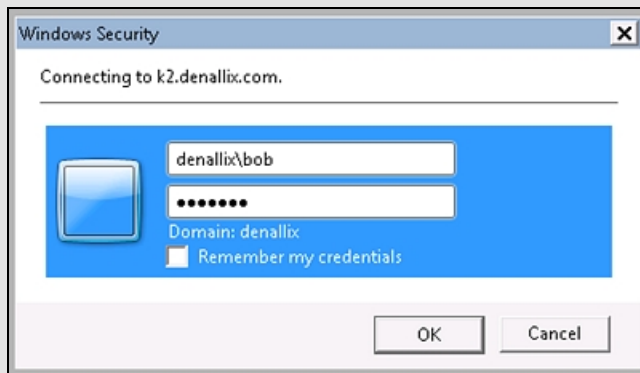
- Open Bob's Outlook and review the notification E-mail that was sent. Open the Worklist task item.



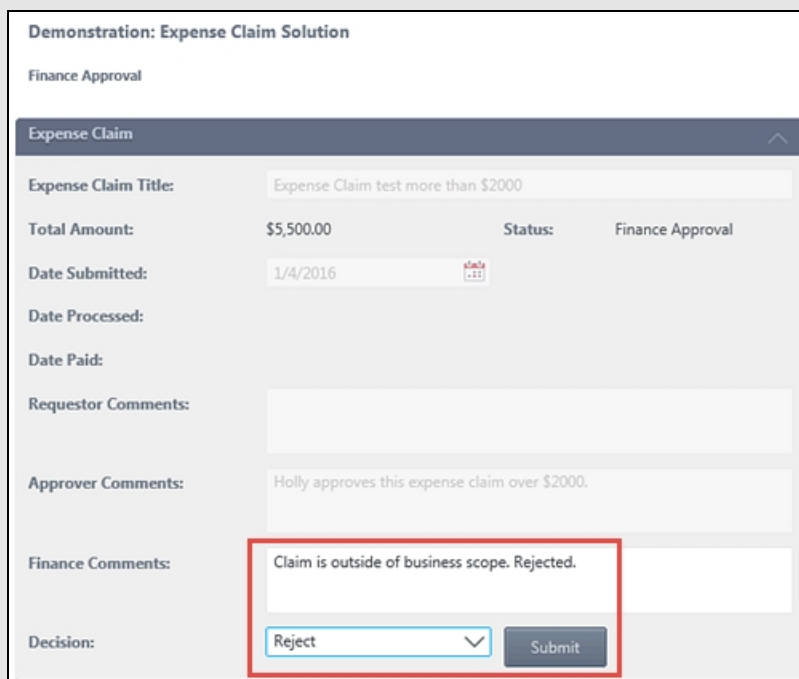
2. If you are prompted for login credentials, enter the following:

Domain\Username: *denallix\bob*

Password: *K2pass!*



3. Enter some comments, then **Reject** the expense claim.



4. Close all Internet Explorer windows and any instances of Outlook.

5. Open Carl's Outlook, and review the rejection notification email that was sent after Bob rejected the claim.



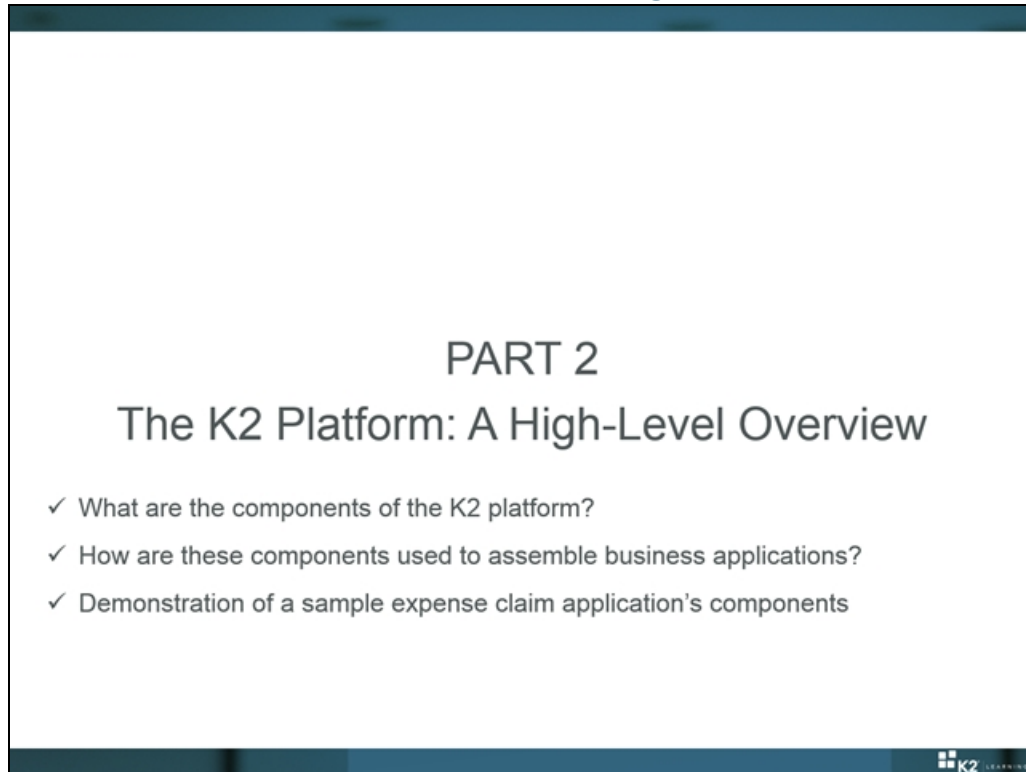
Step 7 Walkthrough

There is no Walkthrough for this step.

Demonstration 1 Review

This completes the first demonstration for this learning module. In the [second demonstration](#), we will review the components used in creating the Expense Claim solution.

PART 2: The K2 Platform: A High-Level Overview



PART 2

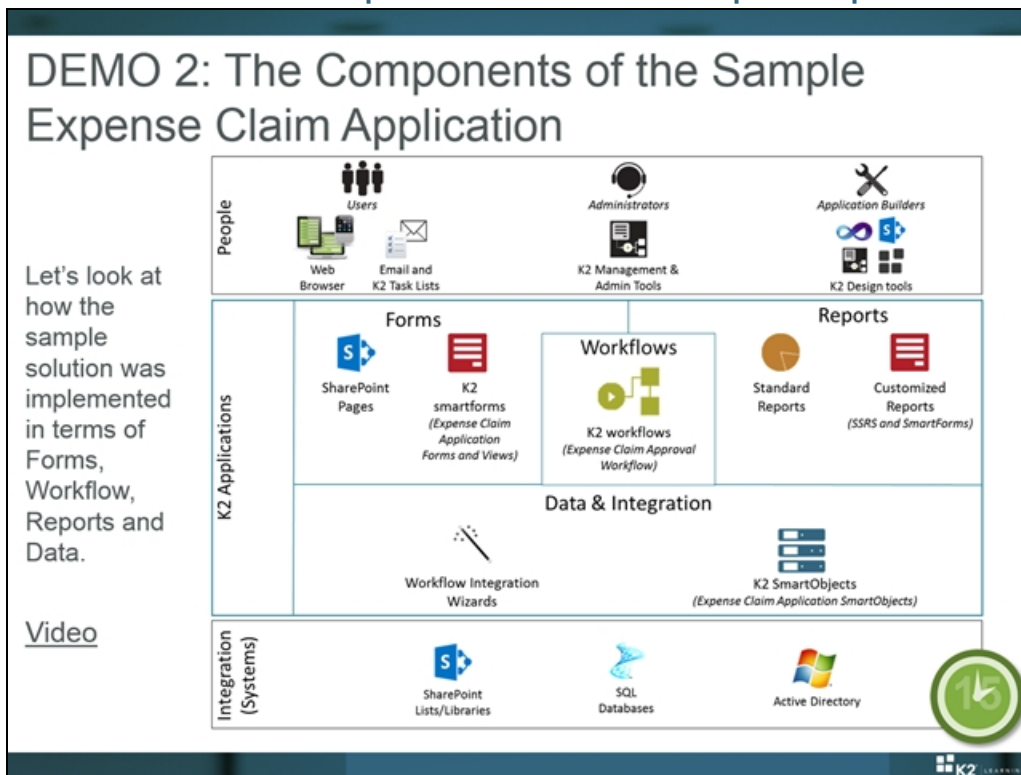
The K2 Platform: A High-Level Overview

- ✓ What are the components of the K2 platform?
- ✓ How are these components used to assemble business applications?
- ✓ Demonstration of a sample expense claim application's components

K2 LEARNING

In Part 2 we will look at the components of the K2 platform. We consider how these components are used to assemble business applications. At the end of Part 2, we will see a demonstration of a sample expense claim's components.

DEMO 2: The Components of the Sample Expense Claim Application



In this demo the instructor will explain how the sample expense claim application was implemented by going through and showing each design artifact. He or she will start with the integration layer, showing the data sources, then show the SmartObjects, then the workflow, then the smartforms, and finally the reports for the solution.

Don't get bogged down in the technical detail of the implementation. Consider this demonstration a “peek” at how a typical application was assembled. In the 200.BEL K2 Business Analysis module we will look at the design and specification of this sample application to understand how the requirements led to this implementation.

Tip

A recording of this demonstration is available at <https://www.youtube.com/watch?v=O4Uz-FjG7HE>

Demonstration 2: The Components of an example K2 application

In this demonstration, the instructor will briefly cover the various components of the sample Expense Claim solution in terms of Data, Workflow, Forms and Reports, then explain how each was implemented. This module does not focus on building solutions, so we will not go through the detailed build of the entire solution.

Note

The demonstrations in this learning module use a pre-built Expense Claim application. This application must be downloaded and installed on a K2 Core 5.x (SharePoint 2010) virtual machine.

If you have not done so already, please download the following file from K2's help site:

<http://help.k2.com/files/10710>.

Once the zip file is downloaded, please extract the contents of the zip file to the virtual K2 environment, and then run the script [Extracted Directory]\Expense Claim Application\Setup\SetupPreRequisites.bat to install the necessary prerequisites and application components.

The application only needs to be installed and deployed on the instructor's virtual K2 environment, since it will only be used as a demonstration. It is not necessary nor recommended to install the application on participant's virtual environments.

Tip

If you are unable to install or live-demo the sample application, you can play a video of the demo, located at

<https://www.youtube.com/watch?v=O4Uz-FjG7HE>

Duration

This demonstration should take around 15 minutes to complete.

Context

In this section, the instructor will open each of the components of the Expense Claim solution seen in the first demonstration.

Step 1: Integration - review the data sources (AD, SQL and SharePoint)

Step 1 Tasks

1. Use SQL Management Studio and open the **K2Learning** SQL database on localhost.

Show the Finance tables, and show the records for the demonstration you did in Demo 1 in the **Expense Claim Header** and **Expense Claim Details** tables. Explain that this database is exposed through K2 SmartObjects.

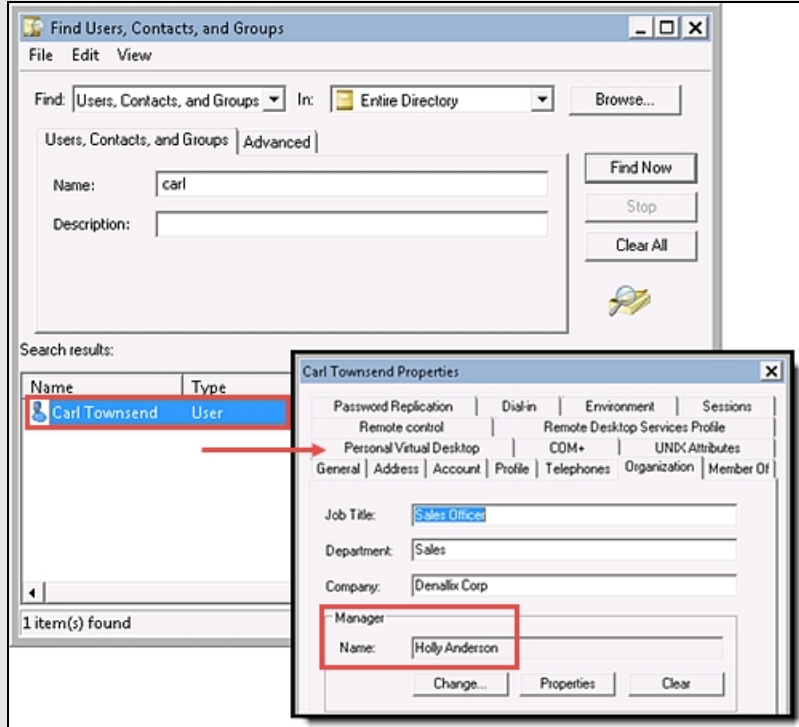
The screenshot shows Microsoft SQL Server Management Studio with the K2Learning database selected in the Object Explorer. The Finance folder is expanded, showing the ExpenseClaimHeader table. A query window displays the following SQL query:

```
select * from K2Learning.Finance.ExpenseClaimHeader
```

The Results pane shows the following data:

ExpenseClaimHeaderId	EmployeeId	DateSubmitted	ExpenseClaimTitle	EmployeeComments	ApproverComments
1	4	2014-01-04	Customer Visit	NULL	NULL
2	5	2	2014-01-02	London Office visit	NULL
3	6	23	2014-01-03	Implementation Project for Customer B	NULL
4	7	29	2014-01-04	Home Fax Machine and Printer	NULL
5	8	1	2014-01-05	New Laptop	NULL
6	9	31	2015-12-31	test 3	test 3
7	10	26	2015-01-04	Sample Expense Claim from Carl	NULL
8	11	26	2015-01-04	Sample Expense Claim from Carl	Holly's comments for approval.
9	12	26	2015-01-04	Expense Claim test more than \$2000	Holly approves this expense claim over \$2000.

- Open **Active Directory Users and Computers** and show the properties for the user Carl. Show that K2 is retrieving the User's manager (Holly) from Active Directory. Explain that K2 integrates natively with AD and that we also created composite SmartObjects that combine data from the HR database tables and Active Directory.



- Open the SharePoint site for this solution.
<http://portal.denallix.com/K2Learning/K2LearningExpenseClaimDemo>
 Show the **Expense Claim Approval Rules** list, and explain that this list is exposed as a SmartObject which will be used by the workflow.
- Open the **Expense Claim Attachments** list. There should be two document sets from the demonstrations you did earlier. Open one of the sets and show the receipts that were attached to the expense claim were saved into SharePoint with information about the associated expense claim header. This was all done with K2 SmartObjects.



Step 1 Walkthrough

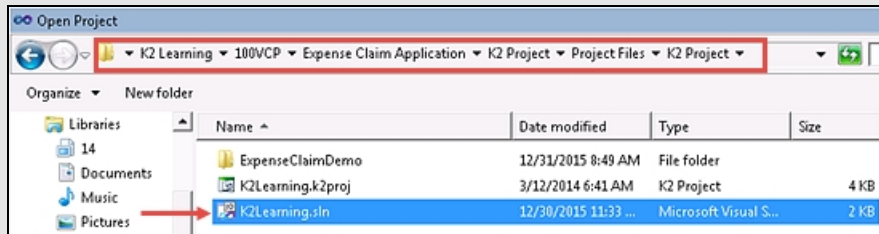
There is no Walkthrough for this step.

Step 2: Data - show the SmartObjects used in the solution

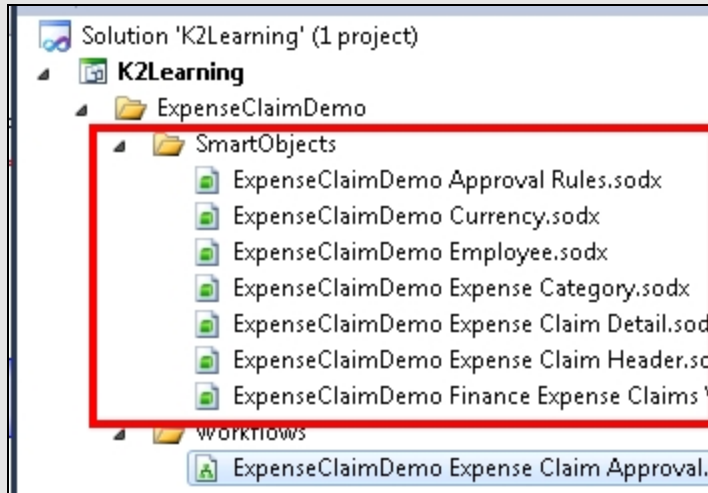
Step 2 Tasks

- Use **Visual Studio** to open the K2 solution for this project, located at
C:\K2 Learning\100VCP\Expense Claim Application\K2 Project\Project Files\K2Learning.sln

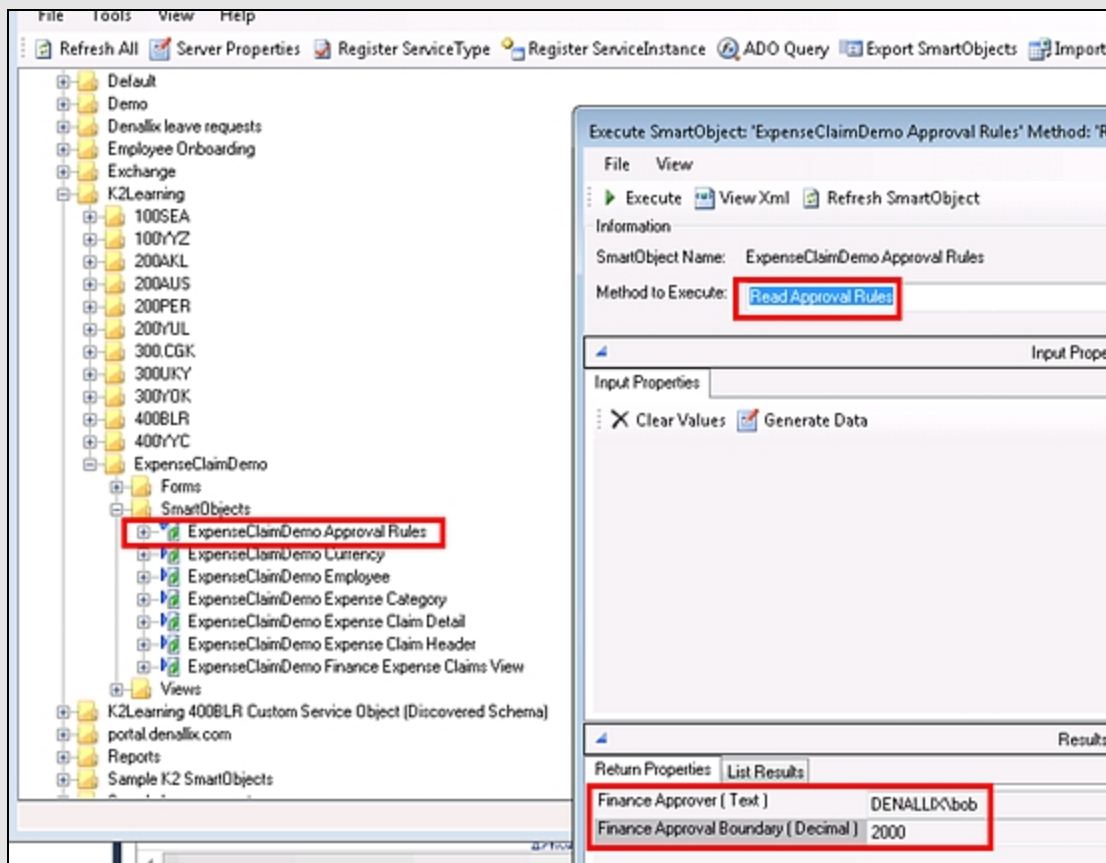
(Depending on the package installation, the project file path may be slightly different.)



2. Show the SmartObjects in the solution.



3. Open the **ExpenseClaimDemo Approval Rules** SmartObject and show the properties and methods for the SmartObject. This is the SmartObject that interacts with the expense claim approval rules list in SharePoint.
4. Open the **ExpenseClaimDemo Expense Claim Header** SmartObject. Explain that this SmartObject connects to the SQL database table for the expense claim header information. Notice that there are several methods for this SmartObject that can perform CRUD methods as well.
5. Open the **ExpenseClaimDemo Employee** SmartObject. Show that this SmartObject is a composite SmartObject that combines data from different data sources (Active Directory and SQL)
6. Launch the SmartObject Service Tester utility, and execute the **Read Approval Rules** method for the **ExpenseClaimDemo Approval Rules** SmartObject. Show the properties that are returned from SharePoint.



- Execute the **List** method for the **ExpenseClaimDemo Employee** SmartObject. Explain that the properties returned are a combination of properties from SQL and AD, but as the consumer you don't need to know where to retrieve the properties from.

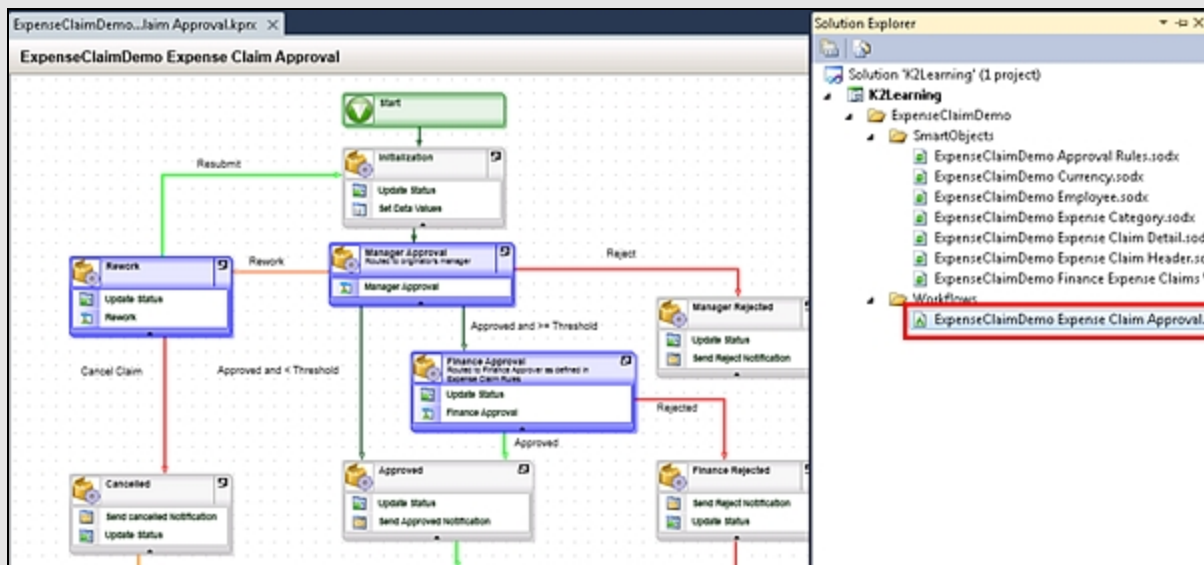
Step 2 Walkthrough

There is no Walkthrough for this step.

Step 3: Workflow - show the workflow used in the application

Step 3 Tasks

- Open the workflow for this solution (**ExpenseClaimDemo Expense Claim Approval**)



Explain that the workflow has a combination of system tasks for K2 (light grey activities) and users' tasks for users (blue activities). We also used color-coding to "document" the flow of the process.

2. Run through the e-mail wizard for one of the **Send Approved Notification** emails. Explain that the developer would use a wizard like this to configure the events in the workflow.
3. Run through one of the **Update status** event wizards. Explain that the developer can integrate with back-end systems using SmartObjects.
4. Open the Destination Rule for the **Finance Approval** activity. Show that K2 is using a SmartObject that points to the Expense Approval Rules list in SharePoint to determine who in Finance should approve the expense request if the amount is larger than the threshold.
5. If time permits, open the Line Rule for the **Approved and >= Threshold** line, and show that K2 is using a SmartObject that points to the expense approval rules list to determine whether or not to route the claim for finance approval.

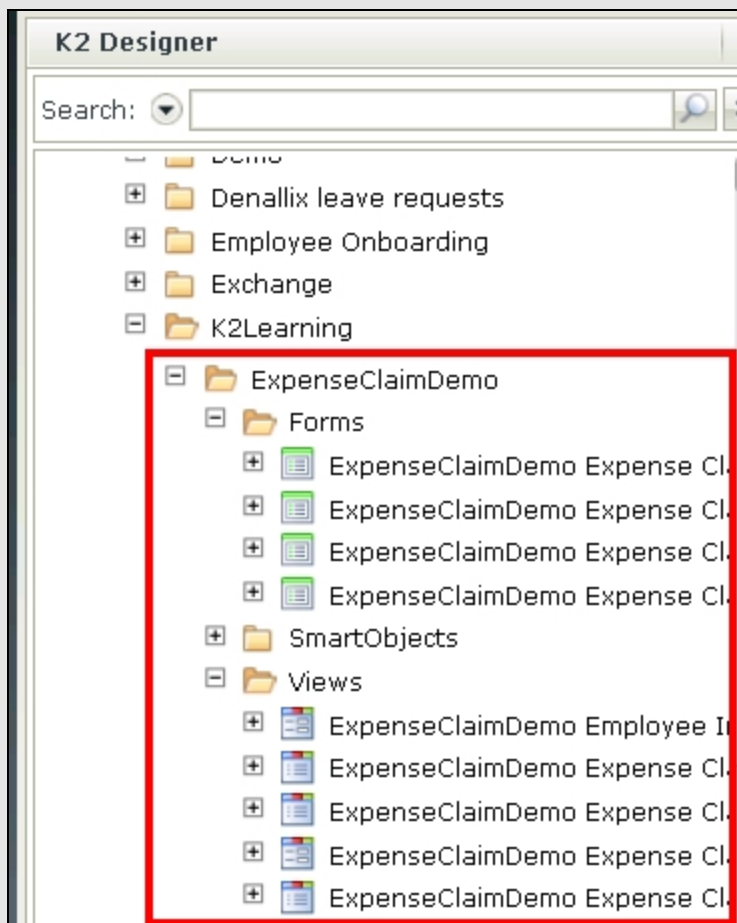
Step 3 Walkthrough

There is no Walkthrough for this step.

Step 4: Forms - show the SmartForms used for the user interfaces

Step 4 Tasks

1. Open K2 Designer and locate the folder for the solution's Forms and Views.



2. Check-out the **ExpenseClaimDemo Employee Info** View and edit it.
3. Show the layout of the View and explain that the View retrieves its information from the Employee SmartObject.

Show the list of fields that are exposed by the SmartObject.

Open the Controls tab and show that the designer can drag and drop controls onto the canvas.

4. Check out and edit the Form **ExpenseClaimDemo Expense Claim Submission**. Show that the Form consists of different Views. Explain that Views can be re-used across different Forms (in this solution, the Views are all used by the Approval, Read-Only and Report Forms as well).

Show the **Rules** designer for the Form. Run through the configuration of the **When btnSubmitNew is clicked** rule. Explain how a designer would configure the Rule using wizards and drag-and-drop.

If time permits, add an Action to show a message box and show how a designer would configure the action. You do not need to save changes to the Form design.

Step 4 Walkthrough

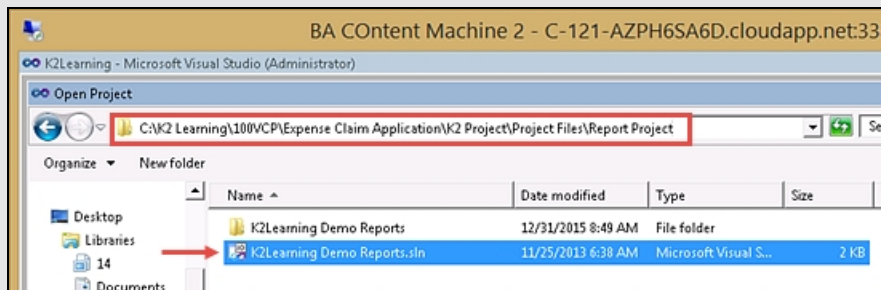
There is no Walkthrough for this step.

Step 5: Reports - show the custom reports created for the solution, as well as the standard K2 reports in K2 Workspace.

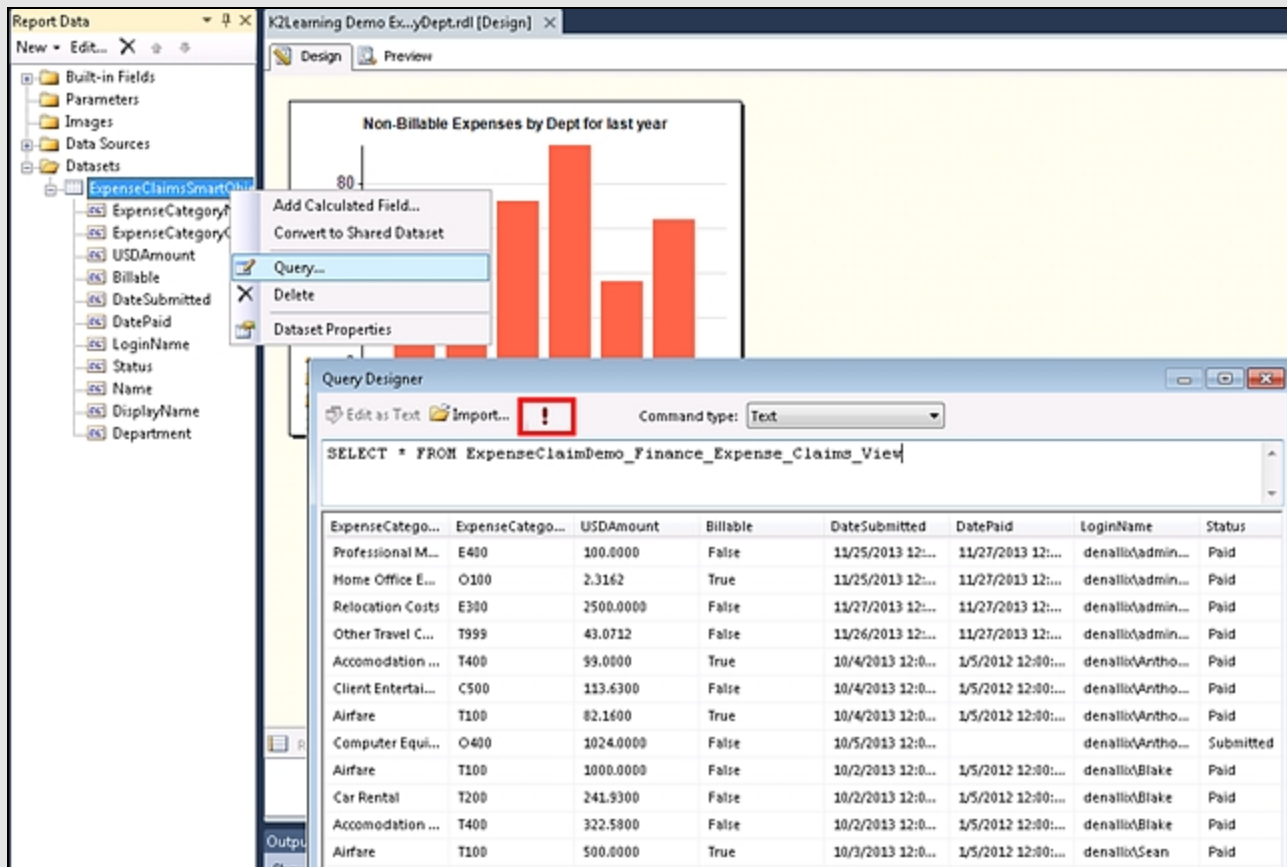
Step 5 Tasks

1. Use Visual Studio to open the reports project for this solution, located at
C:\K2 Learning\100VCP\Expense Claim Application\K2 Project\Project Files\Report Project\K2Learning Demo Reports.sln

(Depending on the package installation, the project file path may be slightly different.)



2. Open the report **K2Learning Demo ExpensesByDept** report. Show that the report uses SmartObjects as its data source, and run the query to show that the SmartObject query syntax is SQL-like.



- Open K2 workspace and run the Process Overview report. Drill down into one of the completed Expense Claim demos you did earlier, and explain how K2 automatically tracks Process Metrics. This data is exposed on the standard K2 reports and can also be accessed through K2 SmartObjects to create custom reports.

Step 5 Walkthrough

There is no Walkthrough for this step.

Step 6: Administration (optional) - show the Management Console in K2 Workspace

Step 6 Tasks

(If time permits)

- Open the K2 workspace.
- Open the Management console in K2 Workspace.
- Navigate to the Process for this demonstration (**K2 Learning > ExpenseClaimDemo > Expense Claim Approval**).
- Open the Instances node and explain the Stop/Start behavior for any running instances.
- Open the Versions node and explain the concept of workflow versions.

Step 6 Walkthrough

There is no Walkthrough for this step.

Demonstration 2 Review

This concludes the second demonstration for this learning module.

100.BHX: Introduction to K2 Applications with K2 Designer



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.

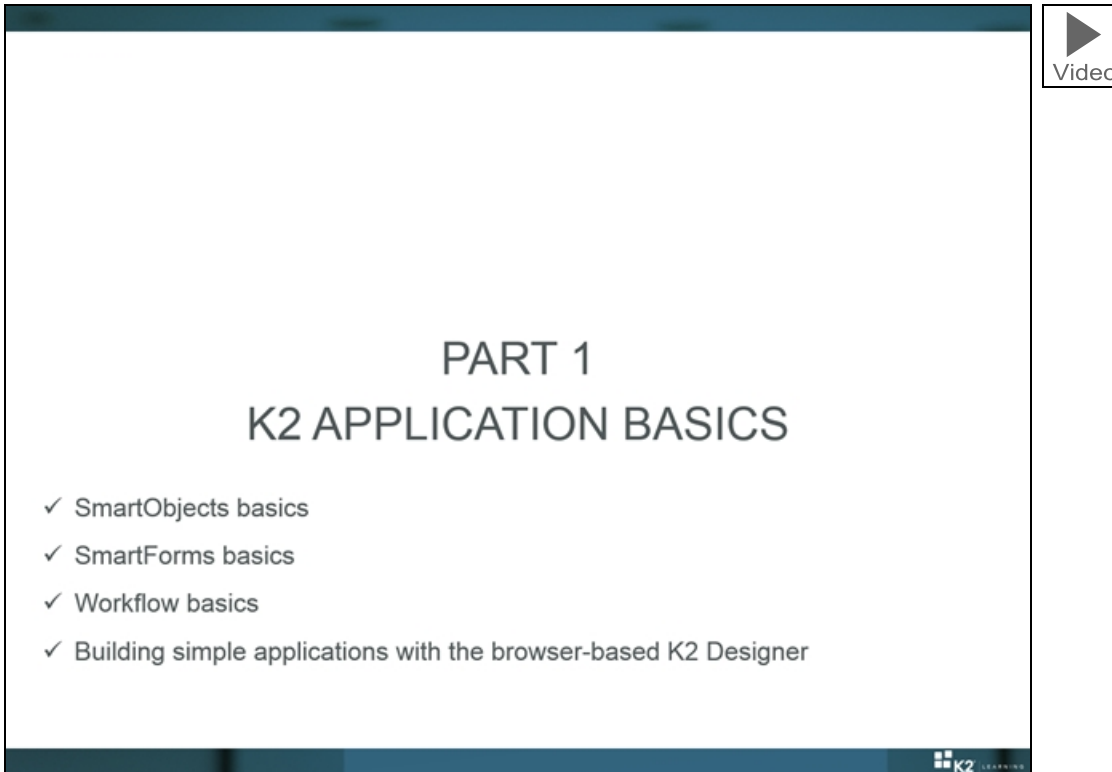
This module covers the following concepts:

- 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



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

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

Name	Description	Type	Key	Required	Unique
ID	The key used to identify a specific record.	Autonumber	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Leave Request Title		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Employee Name		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Employee Email		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Leave Start Date		Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Leave End Date		Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Leave Type		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requester Comments		Memo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Request Status		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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

Create

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

Submit

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

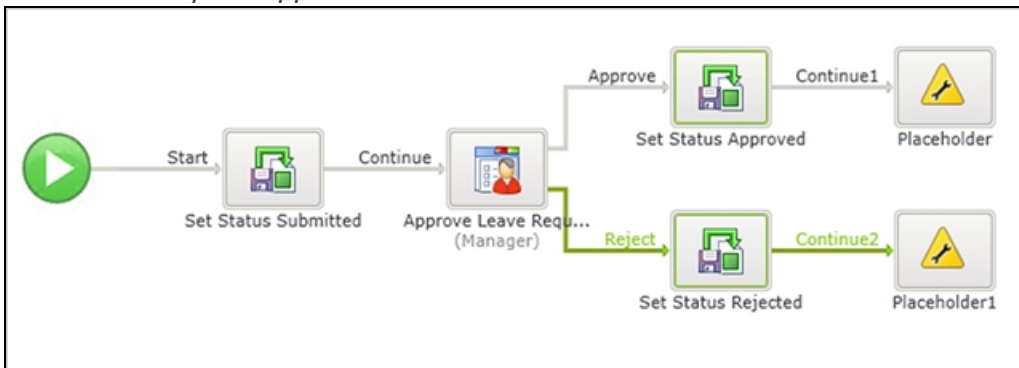
Requester Comments: Test 3 comments

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 38.

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

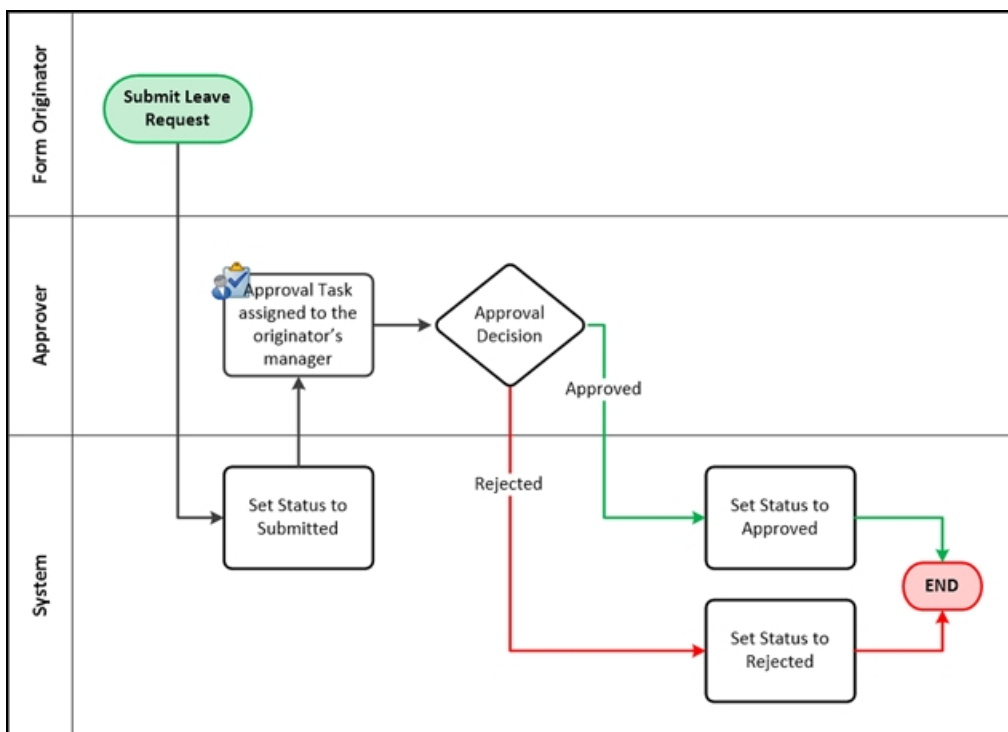
Before starting to build an application, it is important to perform some preliminary design to decide exactly *what* we need our application to accomplish. If your application has a workflow component, a good place to start is to create a flowchart to determine what tasks need to take place and who (or 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.



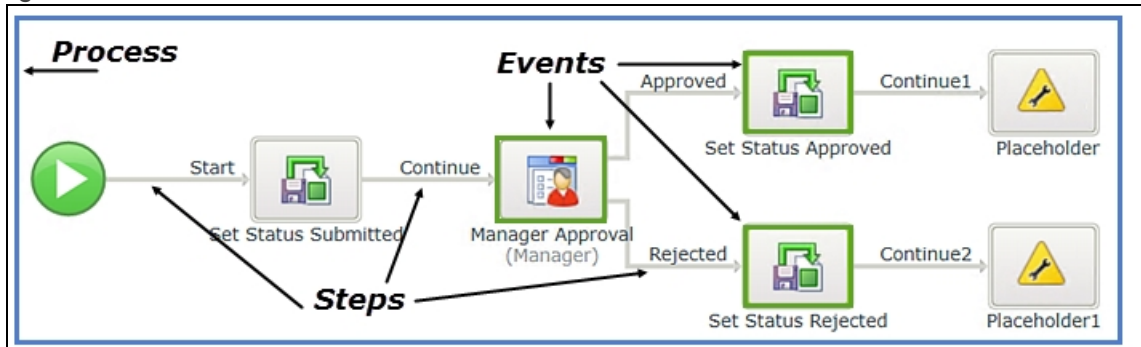
There are two swim-lanes for User Tasks. The first is the Form Originator, or the person submitting the form for the first time. The second swim-lane represents the Approver lane, in our case this will be the form originator's manager. The third lane represents System Tasks, or tasks that K2 will manage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

The screenshot shows a 'Leave Request' form with the following fields and callouts:

- Leave Request Title:** Vacation Leave for April. Callout: Automatically populate these values for the current user.
- Employee Name:** Denallix Administrator.
- Employee Email:** Administrator@denallix.com.
- Leave Start Date:** 4/23/2015.
- Leave End Date:** 4/30/2015. Callout: Select from a predefined list of leave types.
- Leave Type:** Vacation Leave.
- Requester Comments:** My April Time off. Callout: User should be able to provide comments for their leave request.
- Buttons:** A 'Create' button is visible.
- Callouts:**
 - Show all the previous leave entries for this user (pointing to the 'Previous Leave Requests' section).
 - Show whether the request were approved or rejected (pointing to the 'REQUEST STATUS' column in the table).

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:

The image shows a 'Leave Request' form and a 'Previous Leave Requests' table. The form is annotated with 'Form', 'Item View', and 'Controls'. The table is annotated with 'List View'.

Form Annotations:

- Form:** Points to the entire form container.
- Item View:** Points to the form fields.
- Controls:** Points to individual form fields and the 'Create' button.

Table Annotations:

- List View:** Points to the table of 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
<i>Leave Request Title</i>	Text	Allows the user to enter a unique title for their leave request
<i>Employee Name</i>	Text	The name of the employee who submitted the leave request
<i>Employee Email</i>	Text	The email of the employee who submitted the leave request
<i>Leave Start Date</i>	Date	The date on which the leave is due to start
<i>Leave End Date</i>	Date	The date on which the leave is due to end
<i>Leave Type</i>	Text	The type of leave. To keep the data clean, we want users to select from a predefined list of leave types.
<i>Requester Comments</i>	Long Text	Allow the requester to add comments about their leave request
<i>Request Status</i>	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*

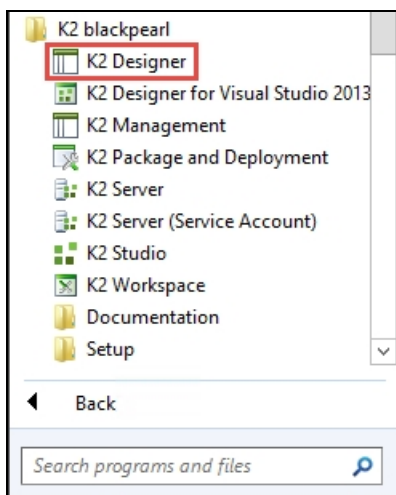
Note

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

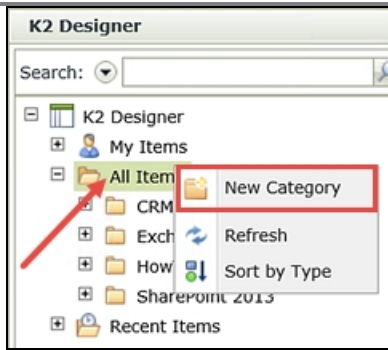
- a. Launch K2 Designer (**Start > All Programs > K2 blackpearl > K2 Designer**).
(If the shortcut is not on your desktop or if you are not using a K2-provided virtual machine for this tutorial, check with your K2 administrator what the web address (URL) is for your K2 Designer environment, then open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



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**.

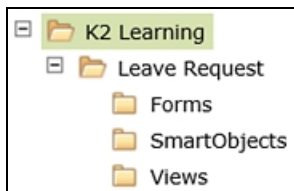


Note
 If you are using an environment that is shared with other users, you may want to add your name to the folder name, so that your application does not conflict with another user's applications.

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* and *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

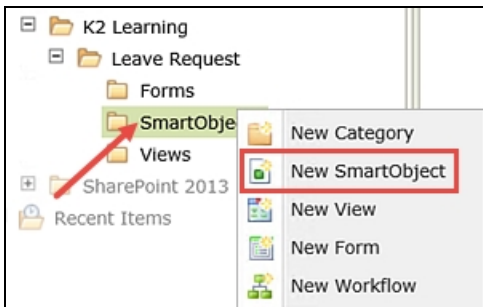
1. Create a New SmartBox SmartObject under the SmartObjects category and name it *Leave Request SmartObject* and allow it to be used in workflows.
2. Add the following properties with their corresponding data types to the SmartObject.

Name	Type	Notes
<i>ID</i>	Autonumber	This field should be added already by default
<i>Leave Request Title</i>	Text	
<i>Employee Name</i>	Text	
<i>Employee Email</i>	Text	
<i>Leave Start Date</i>	Date	
<i>Leave End Date</i>	Date	
<i>Leave Type</i>	Text	
<i>Requester Comments</i>	Memo	
<i>Request Status</i>	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

Introduction **General** Define Properties Configure Associations (Optional) Finished

Name:

Description:

Category:

Type: **SmartObject**
Create a SmartObject using the K2 SmartBox Service.

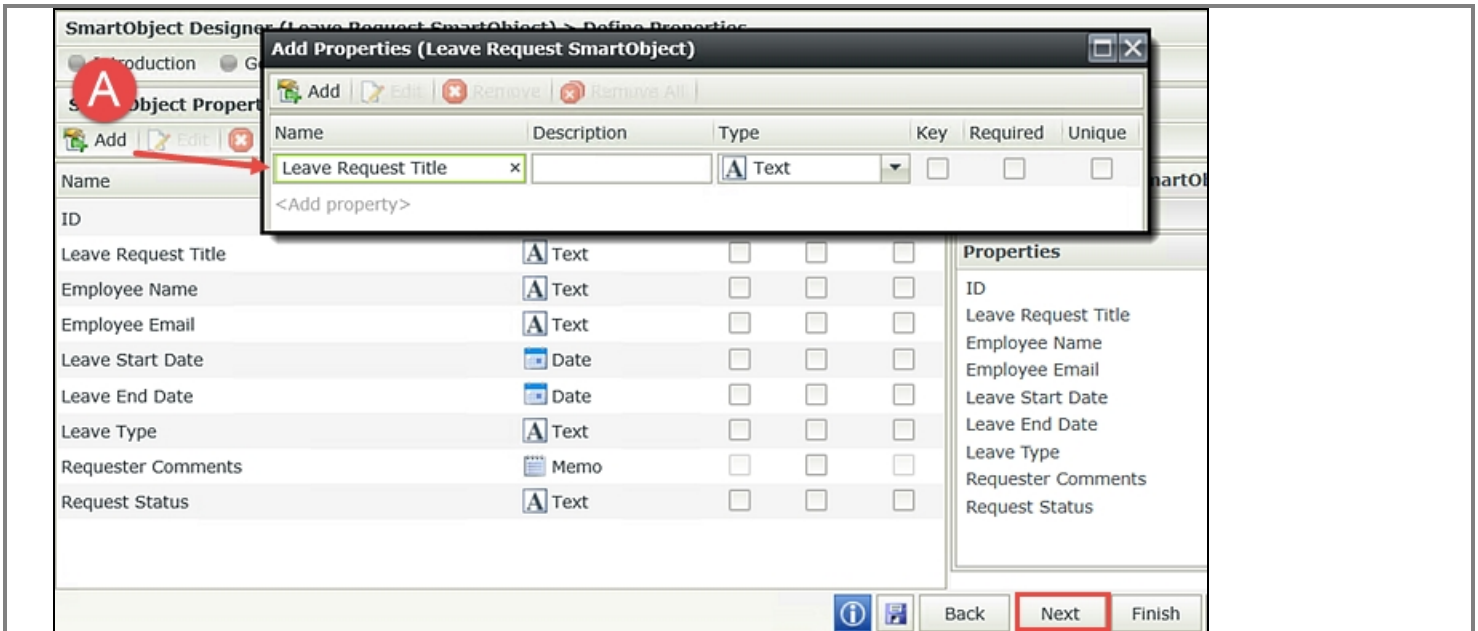
Advanced SmartObject
Create a SmartObject using one or more SmartObject Services.

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



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

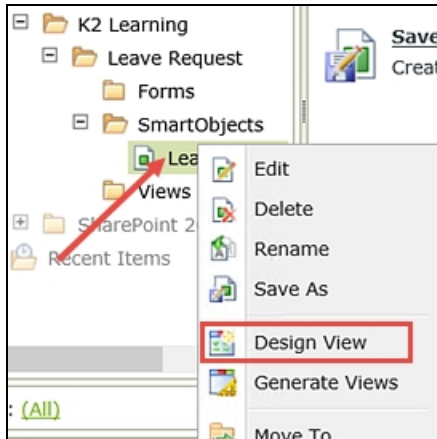
Family Responsibility

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

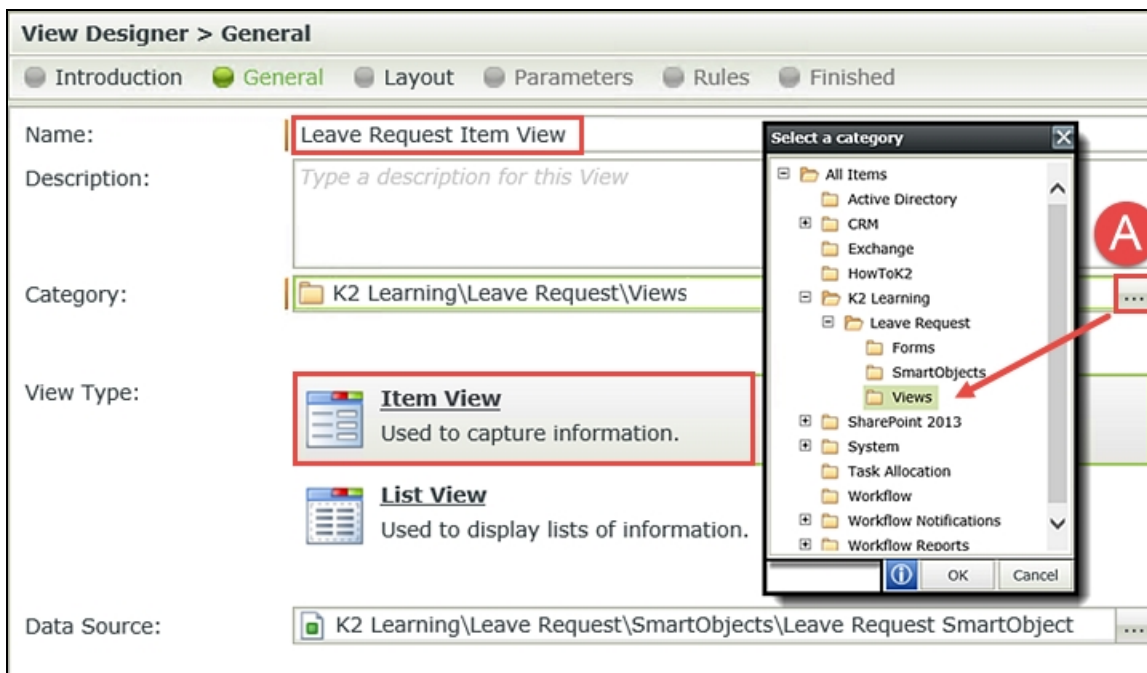
6. Move the **Create** button to the second cell and align it to the right.

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.



- 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.)



- 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:

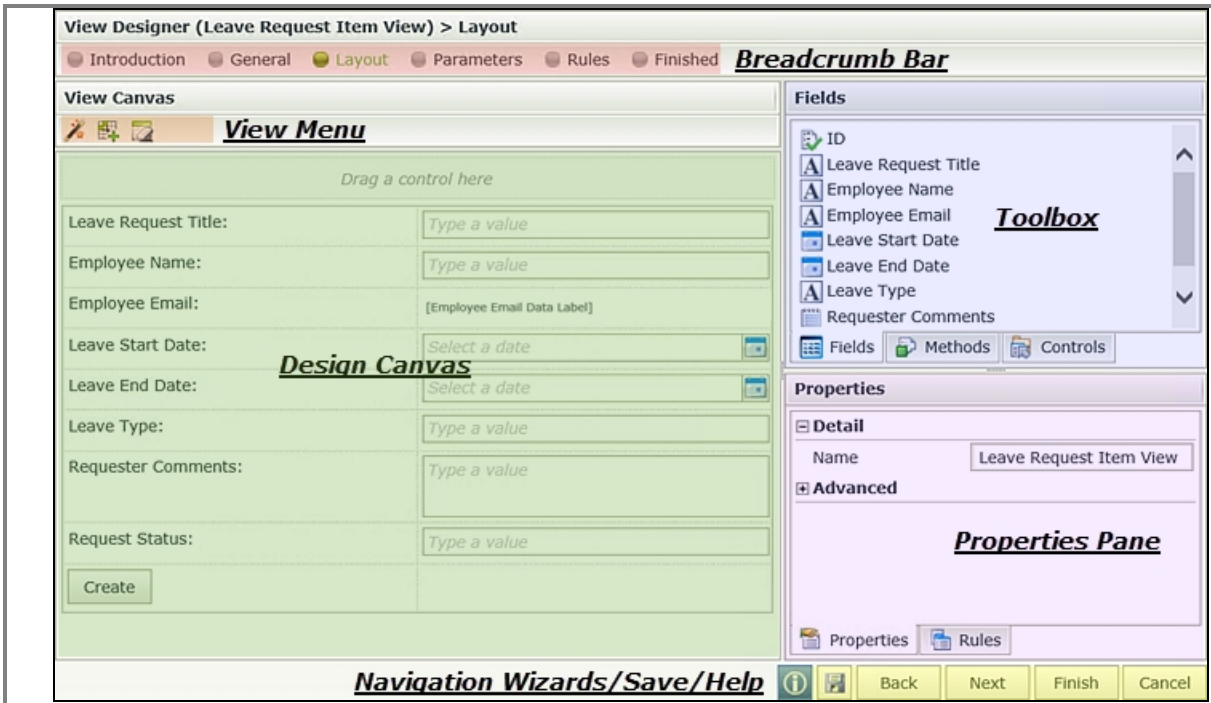
Field Name	Include	Display Only
All Fields	<input type="checkbox"/>	<input type="checkbox"/>
ID	<input type="checkbox"/>	<input type="checkbox"/>
Leave Request Title	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Employee Name	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Employee Email	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Leave Start Date	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Leave End Date	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Leave Type	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Requester Comments	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Request Status	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Label Position:

Colon suffix (":"):

	Standard	Toolbar
All methods	<input type="checkbox"/>	<input type="checkbox"/>
Create	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Save	<input type="checkbox"/>	<input type="checkbox"/>
Delete	<input type="checkbox"/>	<input type="checkbox"/>
Load	<input type="checkbox"/>	<input type="checkbox"/>

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.

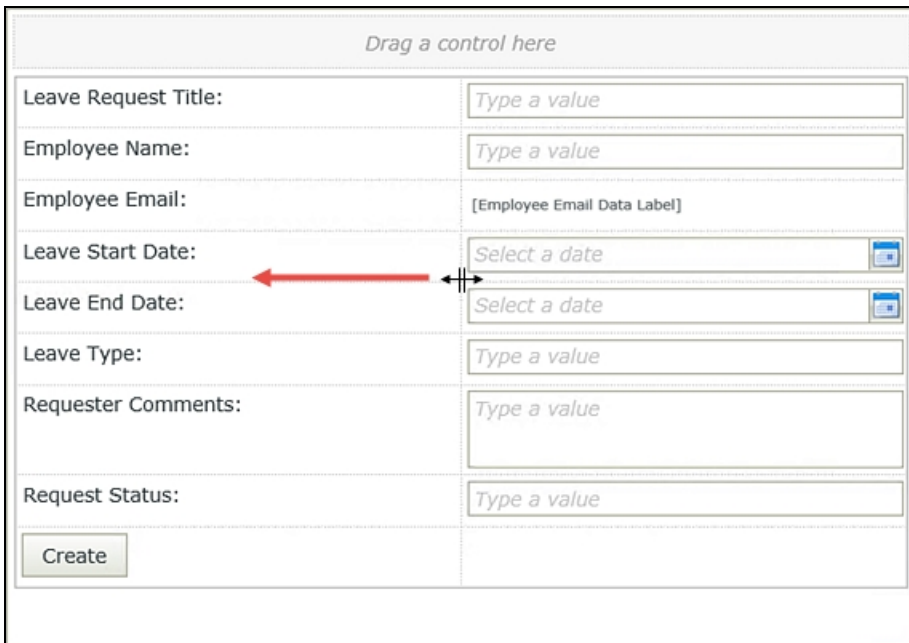


Now that you have the basic view designed, you'll add a few customizations that will make the view a little more user friendly. 99% of view and form design are preference and you will learn what works for your users and what doesn't as you build more applications. First, you'll adjust the width of the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

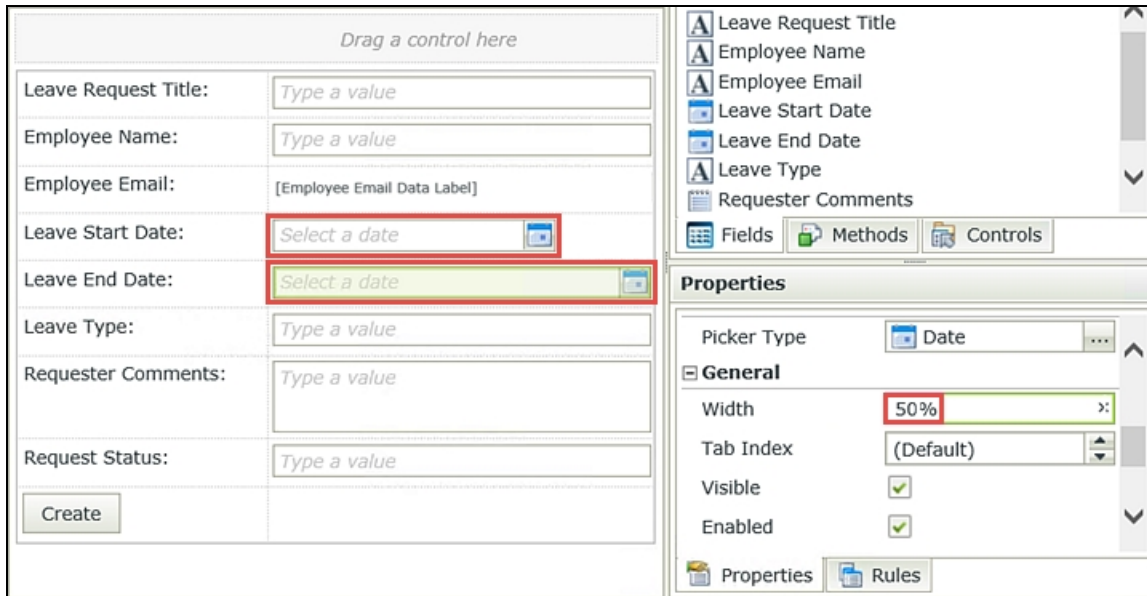
Note

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.

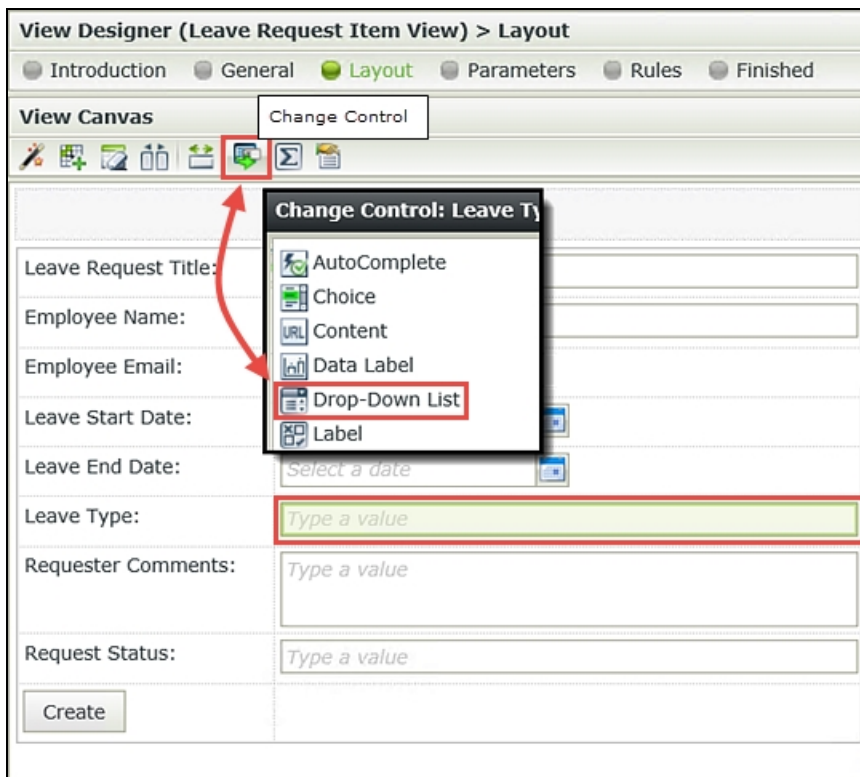


- g. Highlight the **Leave Start Date Calendar** box by clicking it. In the **Properties** Pane, change the **Width** to **50%** then repeat this same step for the **Leave End Date Calendar** box.



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.

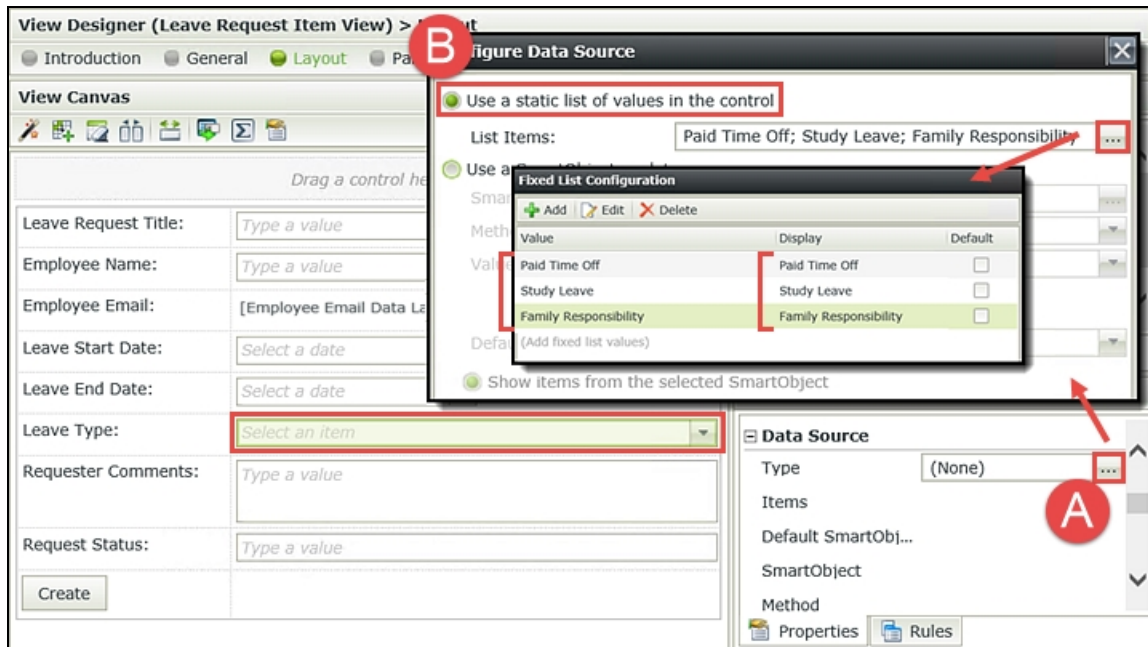
- h. Highlight the **Leave Type Text** box. Select the **Change Control** icon found in the View Menu. Change the control to a **Drop-Down List**.



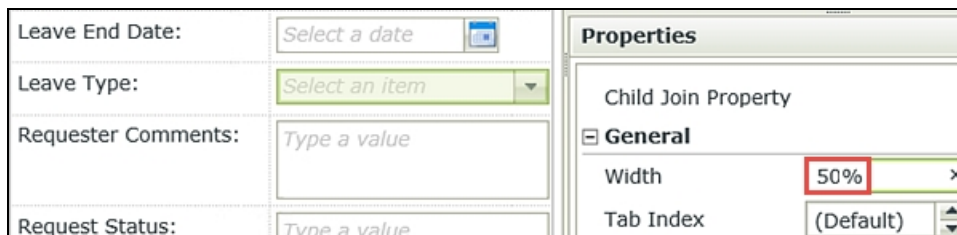
- i. With the Leave Type control still highlighted, open the Data Source **Type** editor (**A** below) found in the Properties Pane. Select the option to **Use a static list of values in the control**. (**B** below) Open the list editor and **Add** the following options for *both* the Value *and* the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

you are done.

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

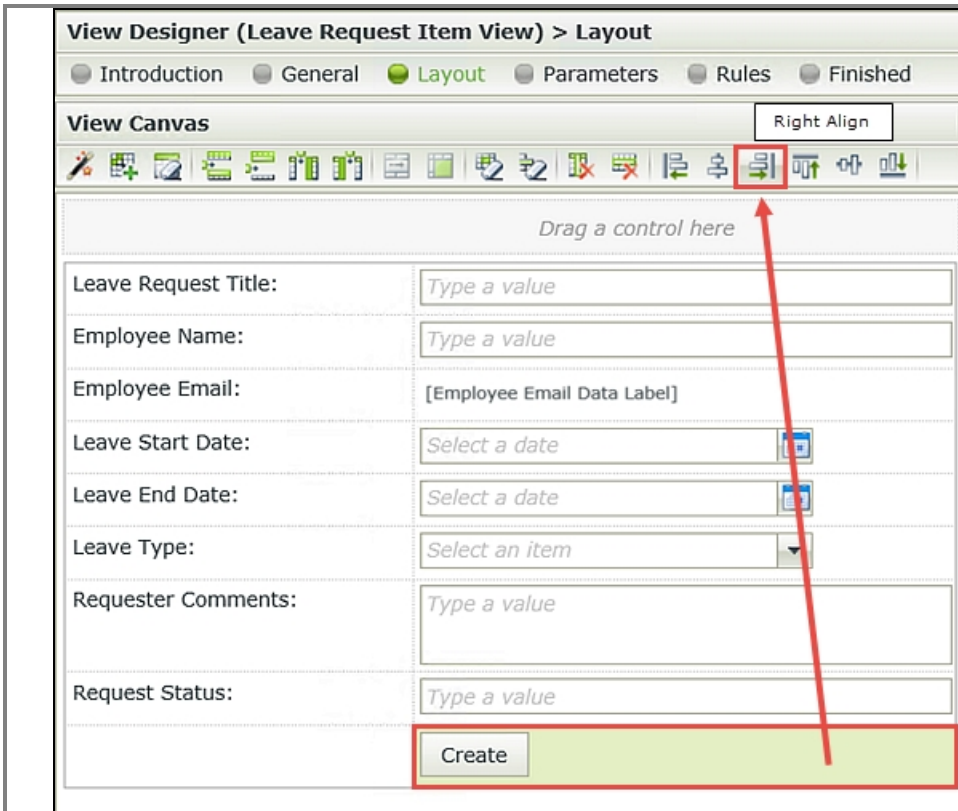


- j. Change the **Width** property of the Leave Type drop-down control to 50% (Remember to click and highlight the drop-down control so that you see its properties represented in the Properties pane.)

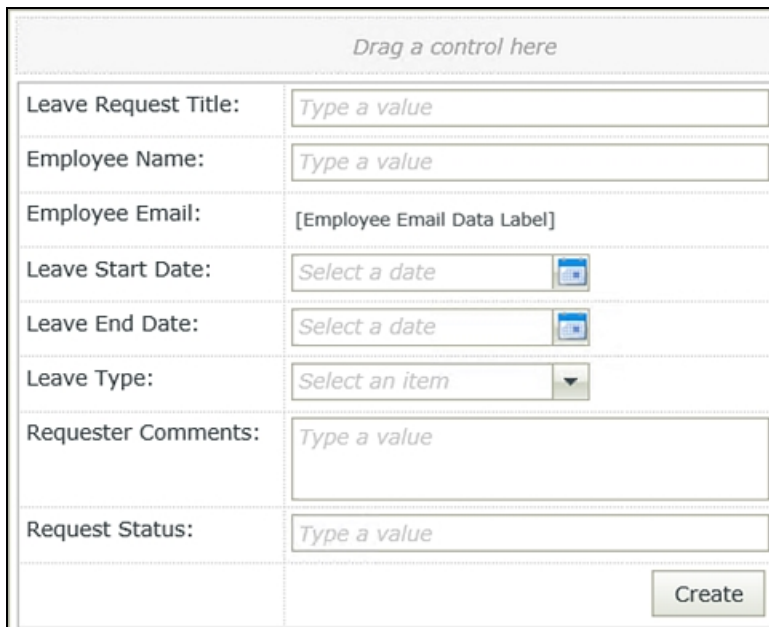


You have one last adjustment to make to your view layout. You'll move the **Create** button to the right side of the view, which will make the view flow smoother for your users.

- 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.



Your view should now look like the image below.



- 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.

The screenshot shows the 'View Designer > General' window. The 'Name' field is 'Leave Request List View'. The 'Category' is 'K2 Learning\Leave Request\Views'. The 'View Type' is 'List View'. The 'Data Source' is 'K2 Learning\Leave Request\SmartObjects\Leave Request SmartObject'. The 'List method' is 'Get List'. The checkbox 'Call this method when the form loads' is unchecked. A file explorer window is open over the 'Views' folder, and a red arrow points from the '...' button in the 'Category' field to the 'Views' folder in the explorer.

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	<input type="checkbox"/>
ID	<input type="checkbox"/>
Leave Request Title	<input checked="" type="checkbox"/>
Employee Name	<input checked="" type="checkbox"/>
Employee Email	<input type="checkbox"/>
Leave Start Date	<input checked="" type="checkbox"/>
Leave End Date	<input checked="" type="checkbox"/>
Leave Type	<input checked="" type="checkbox"/>
Requester Comments	<input type="checkbox"/>
Request Status	<input checked="" type="checkbox"/>

Enable list editing

- Edit all rows
- Edit single rows

Allow the user to add new rows Create

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 disabled 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

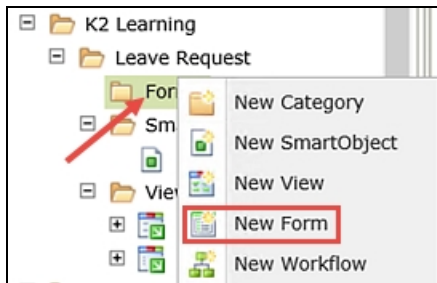
Recall that a form is the main container for views and additional controls. In this next step, you will create a Form and add the two views created in the previous steps. You will then apply a new theme to change the form's look and feel. Oftentimes, forms will contain multiple views. And views can be 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

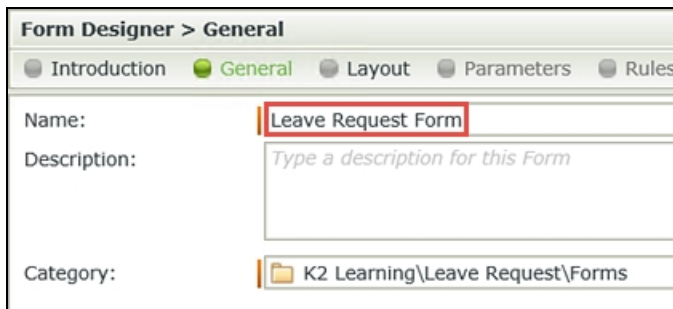
1. Create a **New Form** from the **Forms** category.
2. Add the **Leave Request Item View** and change the view title to *Leave Request*
3. Add the **Leave Request List View** to just below the Item View and change the view title to *Previous Leave Requests*
4. Change the default form Theme to the **Lithium**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

Step 5 Walkthrough

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

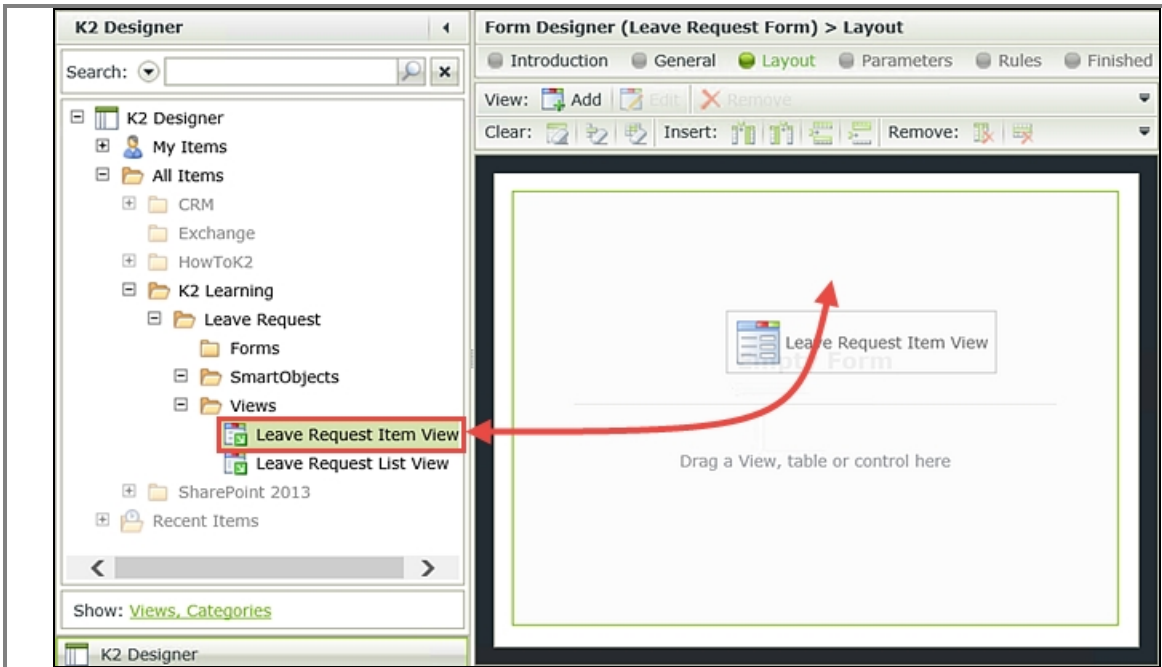


- b. Name the new form *Leave Request Form* then click **Next**. Because you created the form directly from the Forms category, it will be created within that category, so you don't need to move it.

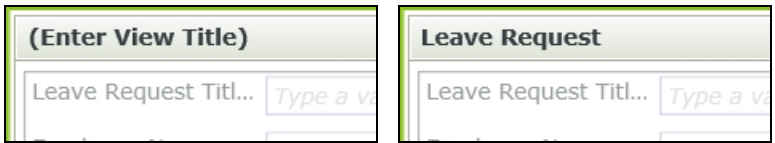


You should now be on the Form Designer screen. The Form Designer has a design canvas and many of the same toolbars and panes as the View Designer.

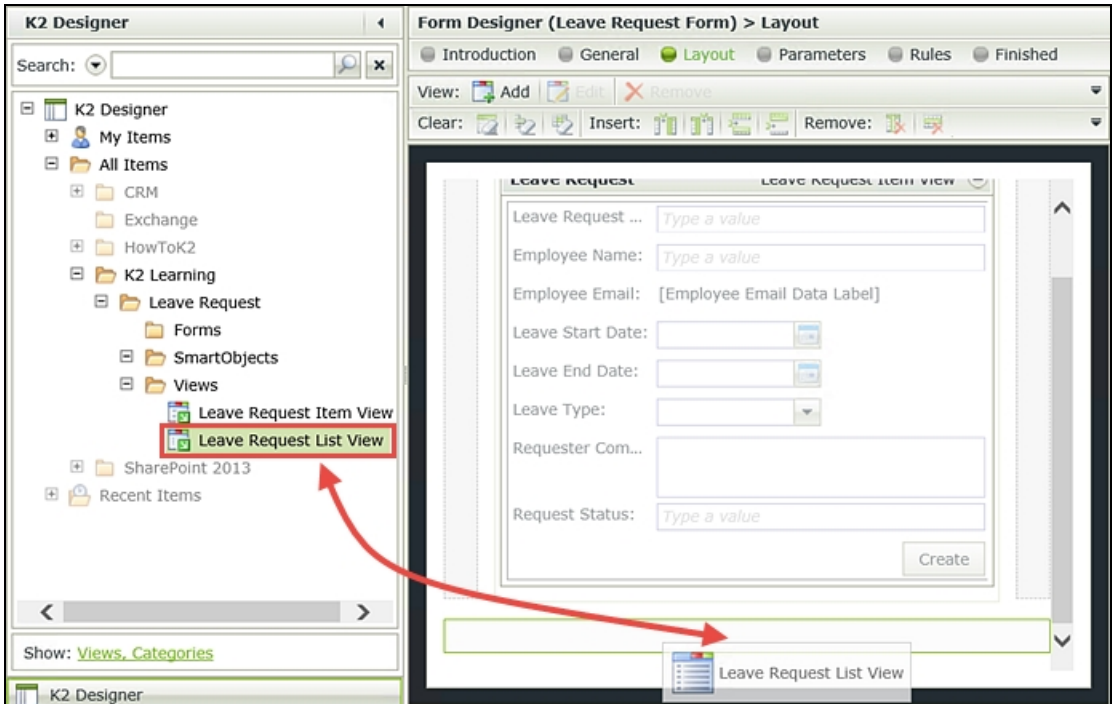
- c. Drag the **Leave Request Item View** onto the form design canvas.



- 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.		

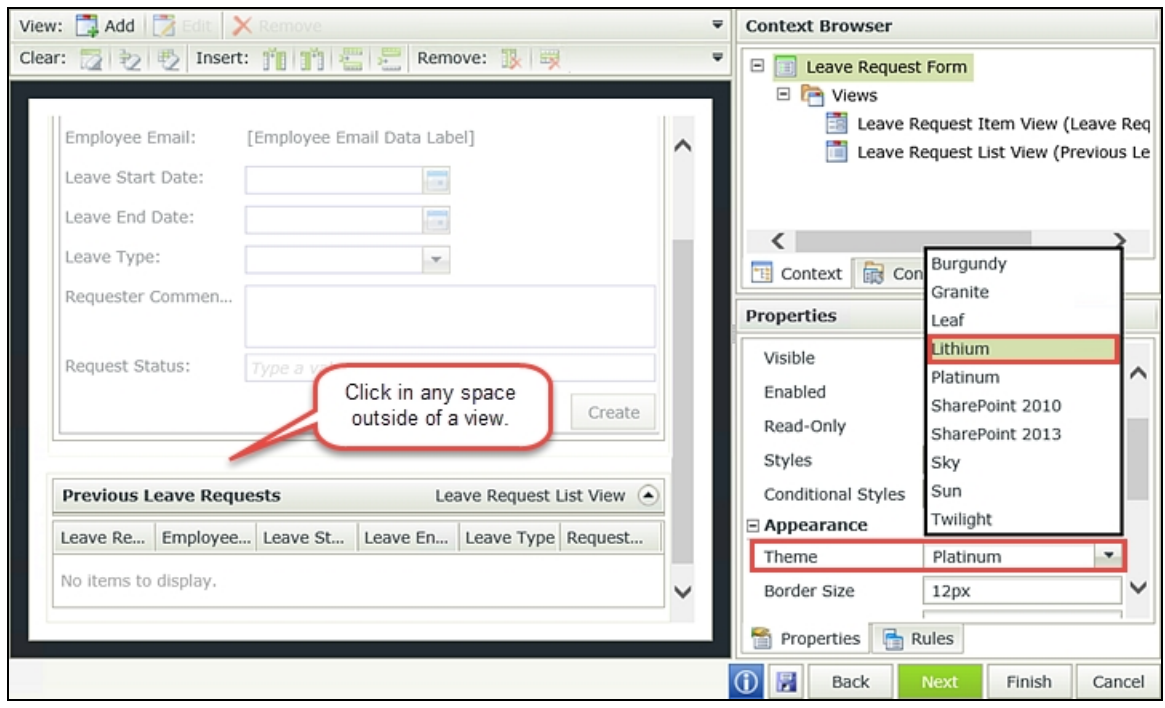
The last step in creating your form will be to customize the look and feel by changing the default theme to the Lithium theme.

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.)



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

- 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.

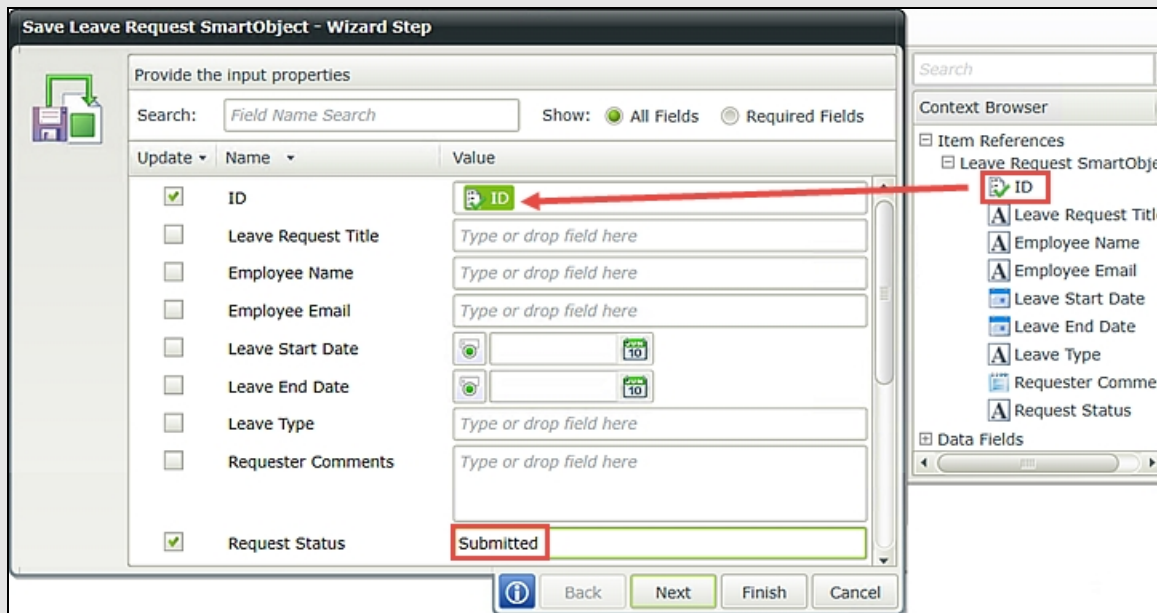
Step 6: Create a new workflow, assign permissions and add a system task to update the request status property

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*

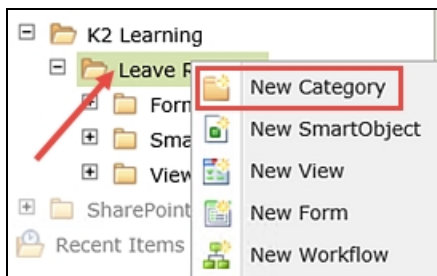
2. From the Workflows category, create a **New Workflow**. On the Workflow setting screen, make the following adjustments:
 - **Name:** *Leave Request Workflow*
 - **Form:** *Leave Request Form*
 - **Folio:** Use the *Leave Request Title Text Box* control found in the Context Browser (**Leave Request 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.



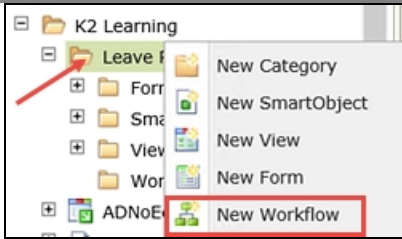
6. Change the Save step title to *Set Status Submitted*

Step 6 Walkthrough

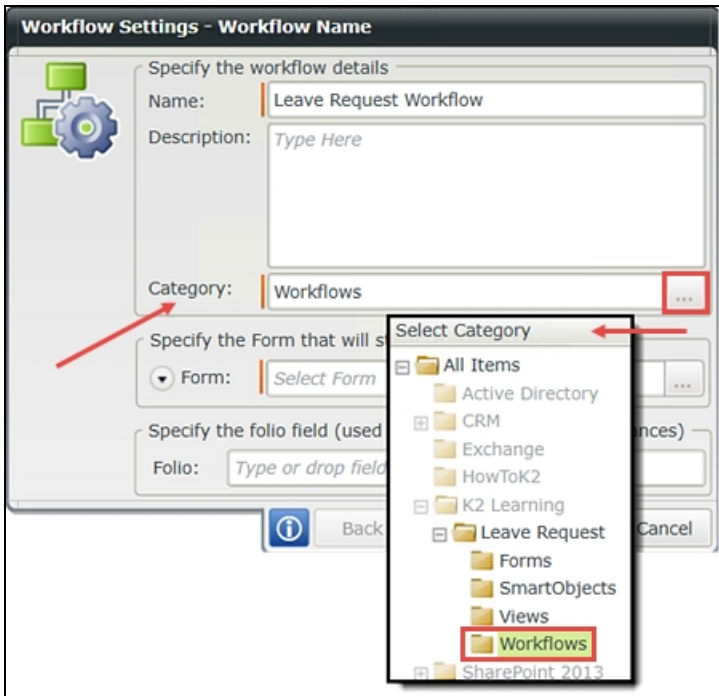
- a. Create a **New Category** under Leave Request and name it *Workflows*



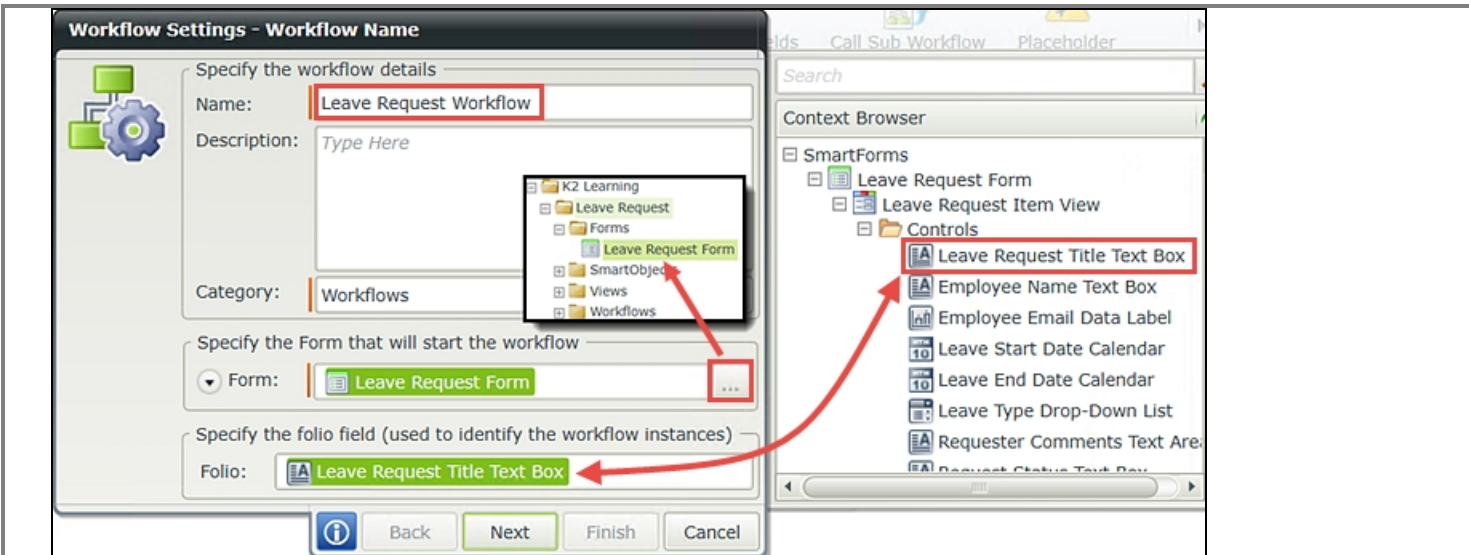
- b. Right-click the **Leave Request** category and select **New Workflow**.



- 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**.



- 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.



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

Instances

Restart Stop Delete Goto Activity View Flow Start New Refresh

Retry

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

<< < 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.

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

Note

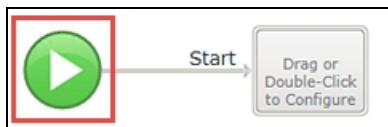
Item References

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**.

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.

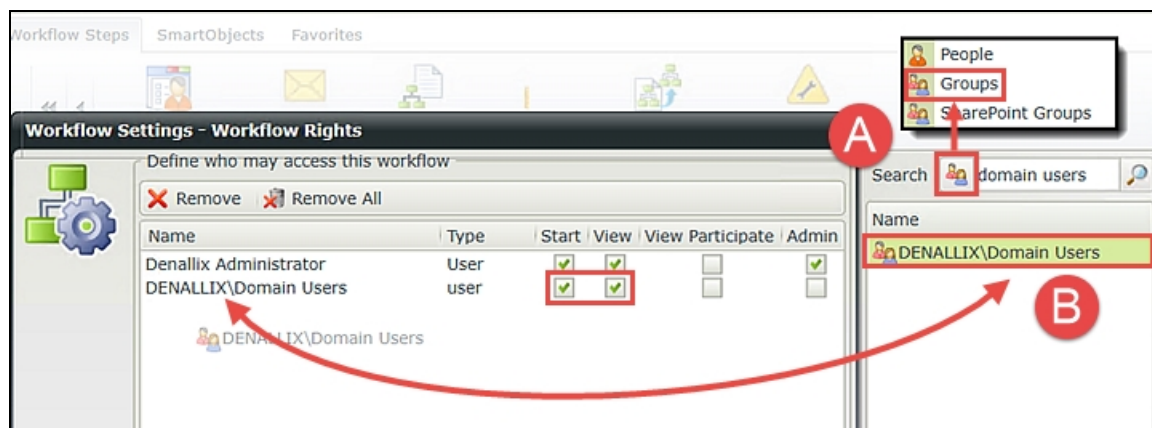


- j. This is the same wizard you just worked through, however the advanced version contains the permissions settings. Click **Next** until you reach the **Workflow Rights** screen (five times).
- k. Begin by changing the search scope to **Groups**. (A below) Enter *domain users* into the search box and 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**



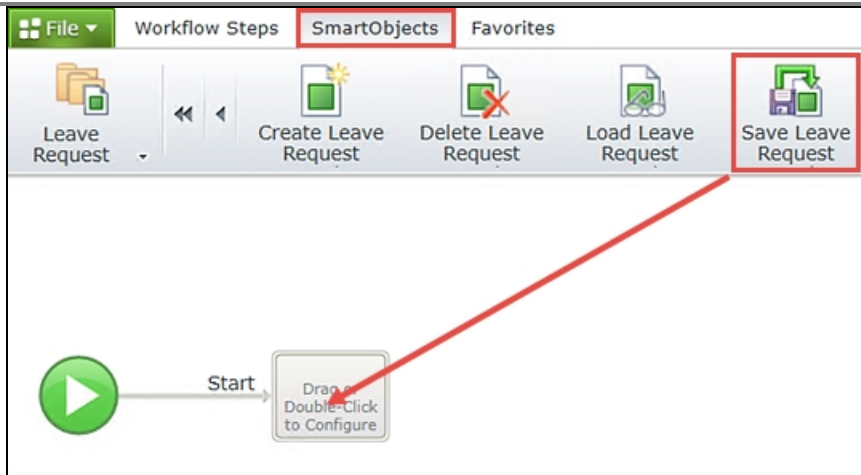
Now that you have your workflow settings complete, you can move on to adding steps to the workflow. recall that the workflow steps are basically the lines that connect one event to another. The first event you are going to add will be a System Event using the SmartObject Event wizard. (System events 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".)

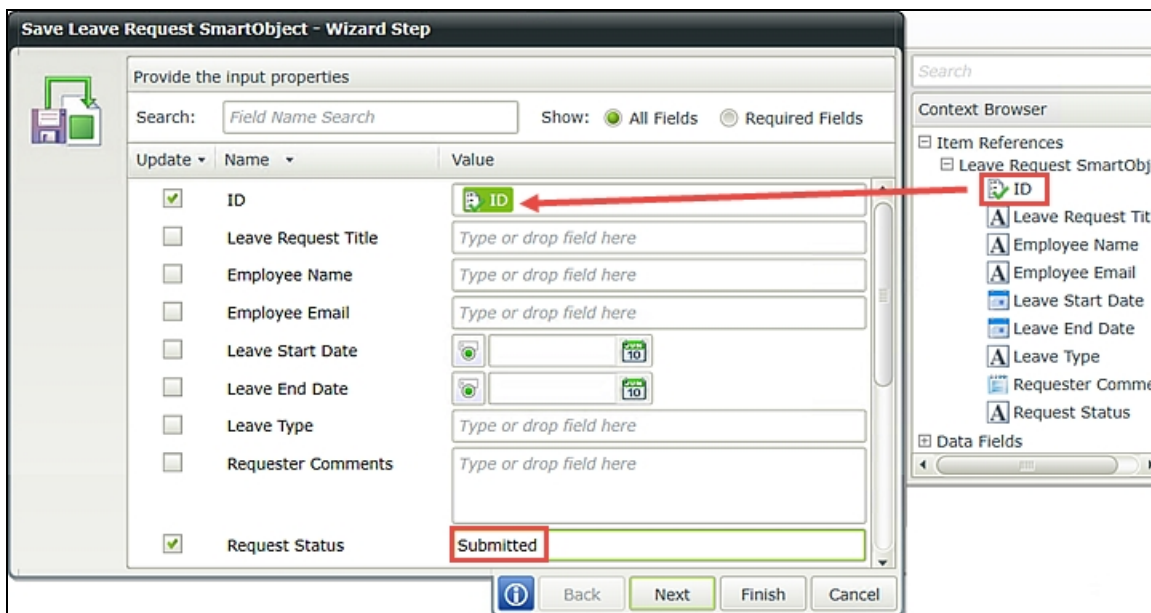
Note

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.

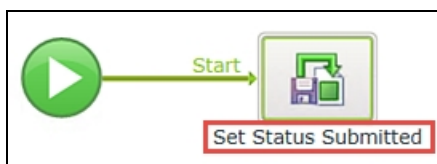
- m. Recall that when you created the Leave Request SmartObject, you *allowed* the SmartObject to be used in workflows. This is where that setting comes into play. Had you not checked the 'Allow' option, none of the Leave Request SmartObject methods would appear as they do below. (The SmartObject events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



- 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* and *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.

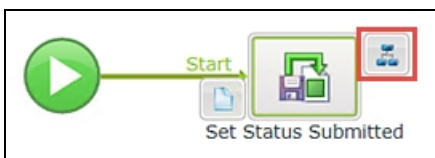
Note

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.

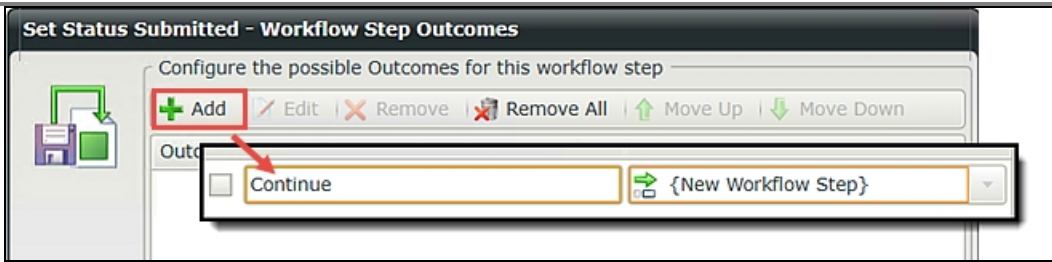
4. Save your work.

Step 7 Walkthrough

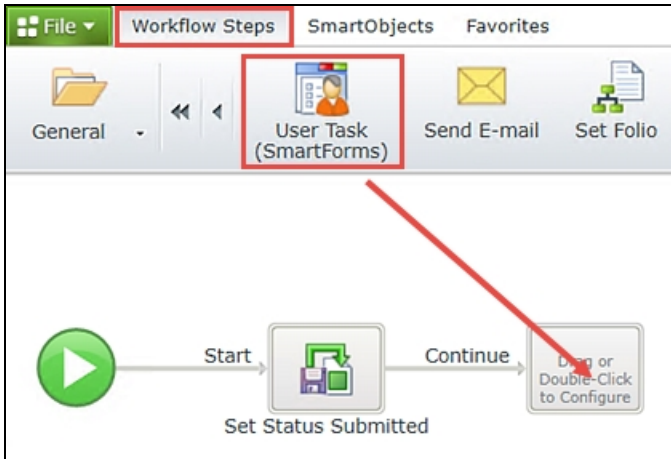
- a. Hover over the **Set Status Submitted** event until you can see the **Workflow Step Outcome icon**. (The icon looks like a blue mini-flowchart. If you don't see the icon when hovering, click once on the workflow canvas to refocus your mouse to the canvas. Then try hovering again.) Click the **Outcome**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



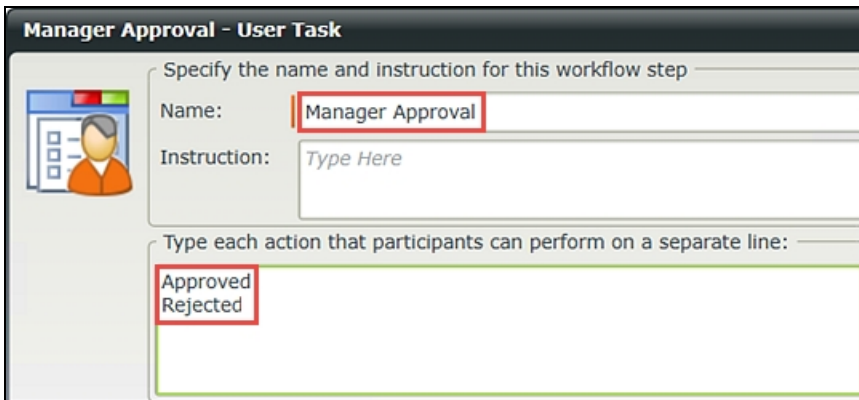
- b. On the configuration screen, click **Add**. Replace New Outcome with *Continue* then click **OK**. Here, you are simply giving your step, or outcome line, a label so that it is easy for us to tell exactly what that step does. In this case, it is very straightforward. However, as you build more complex 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.



- 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.



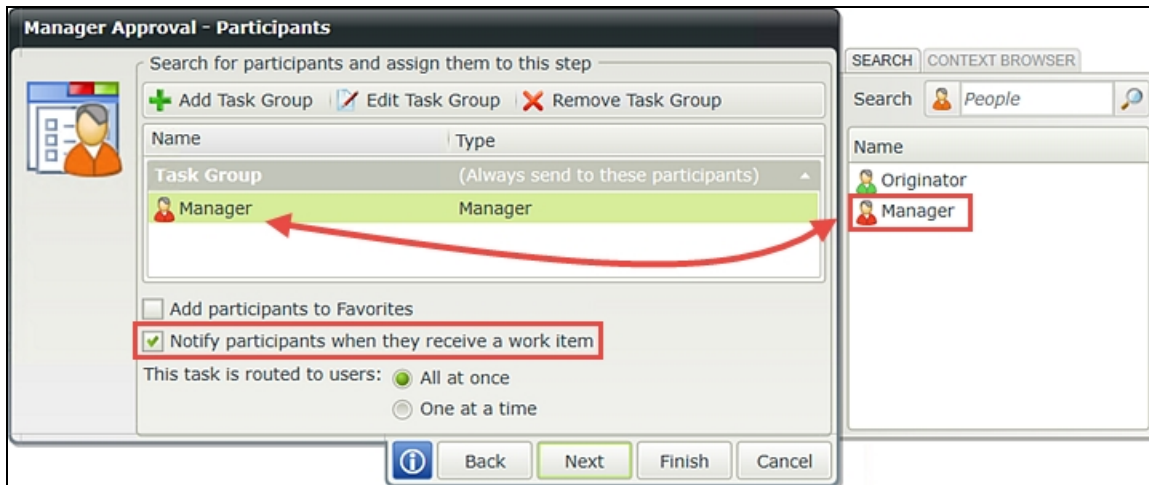
- d. The User Task wizard opens. Name the task *Manager Approval*
- e. On this screen you can will also configure the actions. Actions are the decision options that the user assigned the task can make. You can have many actions, but you need at least one. For your workflow, you will have two. Enter the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



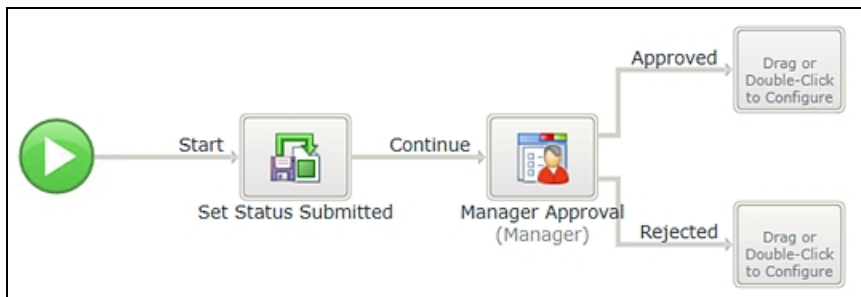
- f. The User Form setting is fine, so click **Next**. The work item rule is fine, so click **Next**.
- g. On the Participants screen, drag the **Manager** name into the Task Group window. CHECK the box to **Notify participants when they receive a work item**. Here, you are assigning this task to the 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

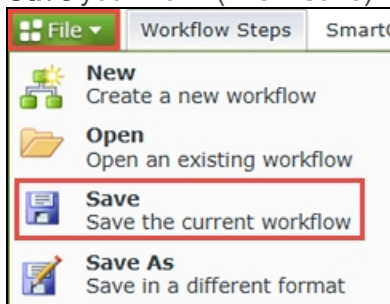
a user where you have access to their Outlook. You might consider assigning this task to yourself for access/testing purposes. If you are building this application on a K2-provided VM, you will have access to the manager's Outlook.



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)



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

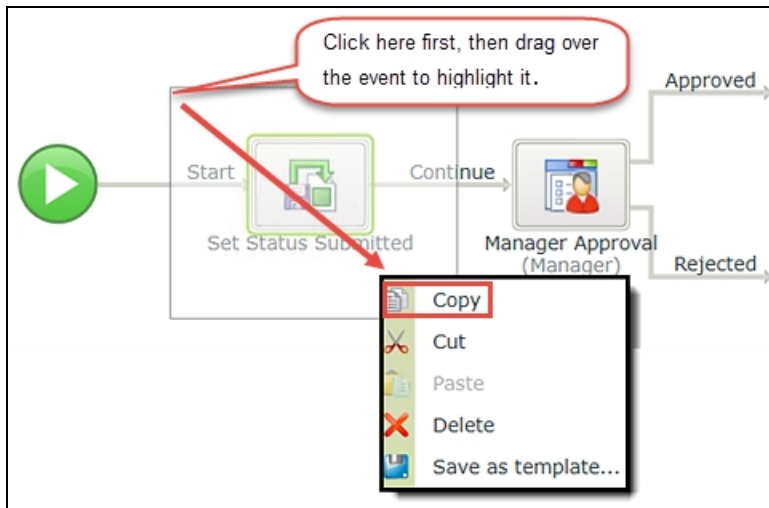
Step 8 continues with the addition of two System Tasks that will update the Request Status property in the Leave Request SmartObject depending on the decision made by the approving manager. K2 Designer makes reusing events very simple by allowing you to copy and paste existing events. With a few minor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

Step 8 Tasks

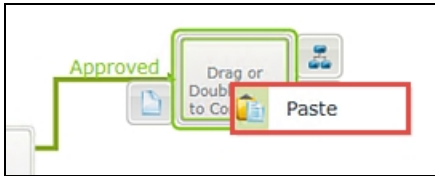
1. Copy the **Set Status Submitted** event and paste it into the event box for the Approved outcome line, then once again for the Rejected outcome line.
2. Edit both events and change the **Request Status** property to the appropriate outcome (*Approved* or *Rejected*). Change the event titles to match 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.

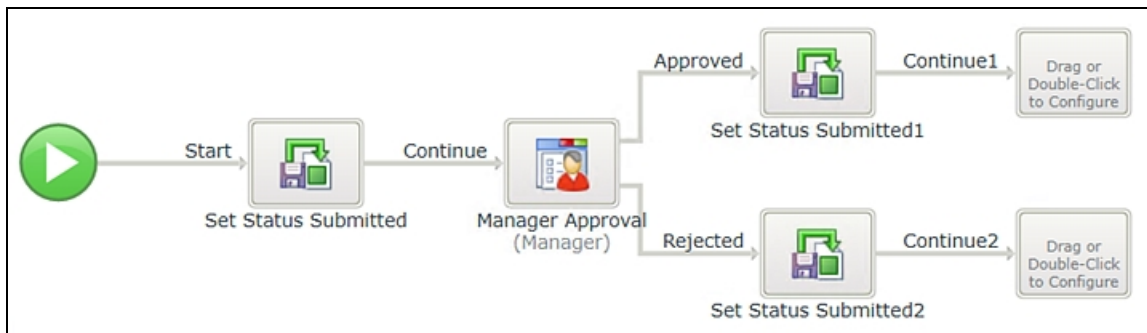


- b. Hover over the Approved outcome event, right-click, then click **Paste**.



- 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:

Description:

- 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.

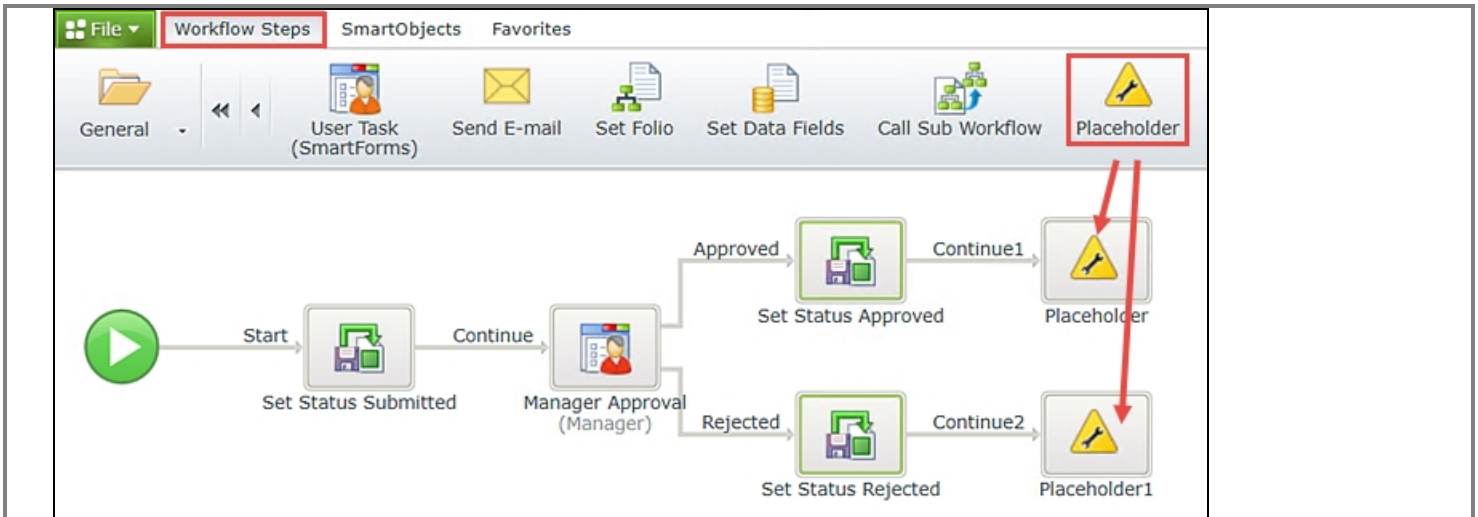
Set Status Submitted1 - Wizard Step

Provide the input properties

Search: Show: All Fields Required Fields

Update	Name	Value
<input checked="" type="checkbox"/>	ID	<input type="text" value="ID"/>
<input type="checkbox"/>	Leave Request Title	<input type="text" value="Type or drop field here"/>
<input type="checkbox"/>	Employee Name	<input type="text" value="Type or drop field here"/>
<input type="checkbox"/>	Employee Email	<input type="text" value="Type or drop field here"/>
<input type="checkbox"/>	Leave Start Date	<input type="text" value="10"/>
<input type="checkbox"/>	Leave End Date	<input type="text" value="10"/>
<input type="checkbox"/>	Leave Type	<input type="text" value="Type or drop field here"/>
<input type="checkbox"/>	Requester Comments	<input type="text" value="Type or drop field here"/>
<input checked="" type="checkbox"/>	Request Status	<input type="text" value="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

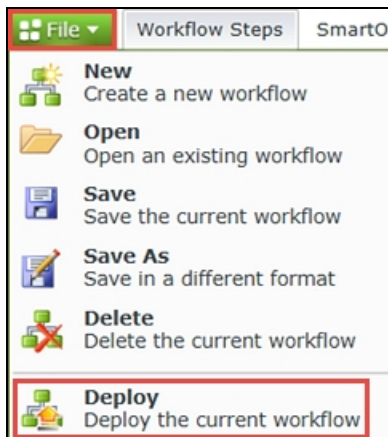
Next you will deploy your workflow. Deploying a workflow publishes the workflow to the K2 server which then allows it to be exposed to the forms configured for that workflow. If you make changes to your workflow, you must redeploy it before the changes are published to the K2 server.

Step 9 Tasks

1. Deploy the workflow.

Step 9 Walkthrough

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



- 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**.

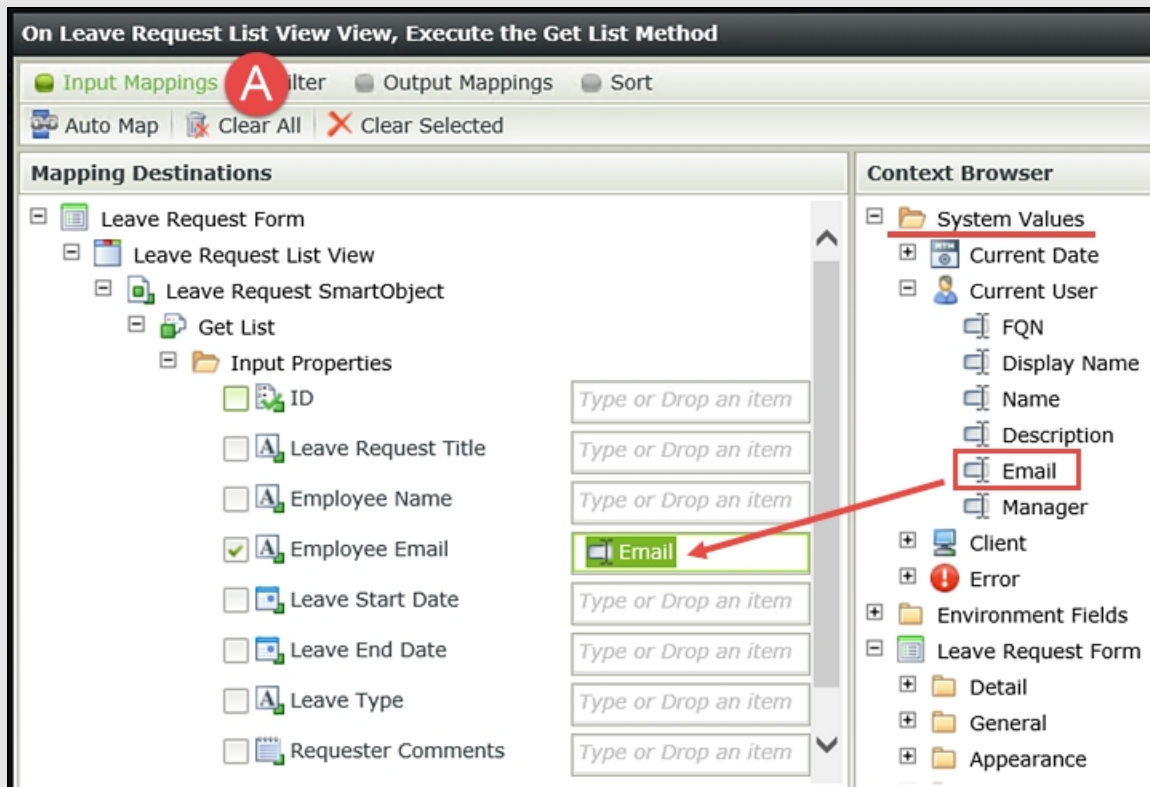
Step 10: Edit the Form Initializing rule to display the current user's employee details and previous leave requests

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

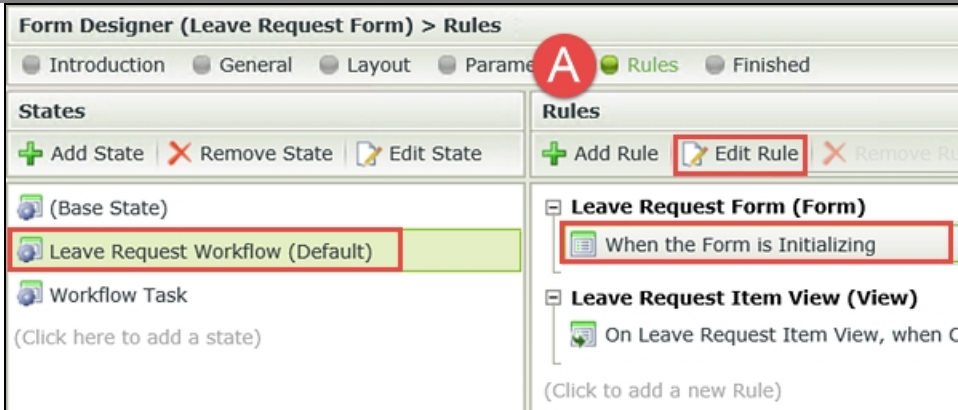
1. Edit the Leave Request Form, **When the Form is Initializing** rule on the Leave Request Workflow (Default) state and add an execute a view method action to retrieve the previous leave requests for the current user. Use the system value **Email** as the input property, then map all of the return properties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



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.

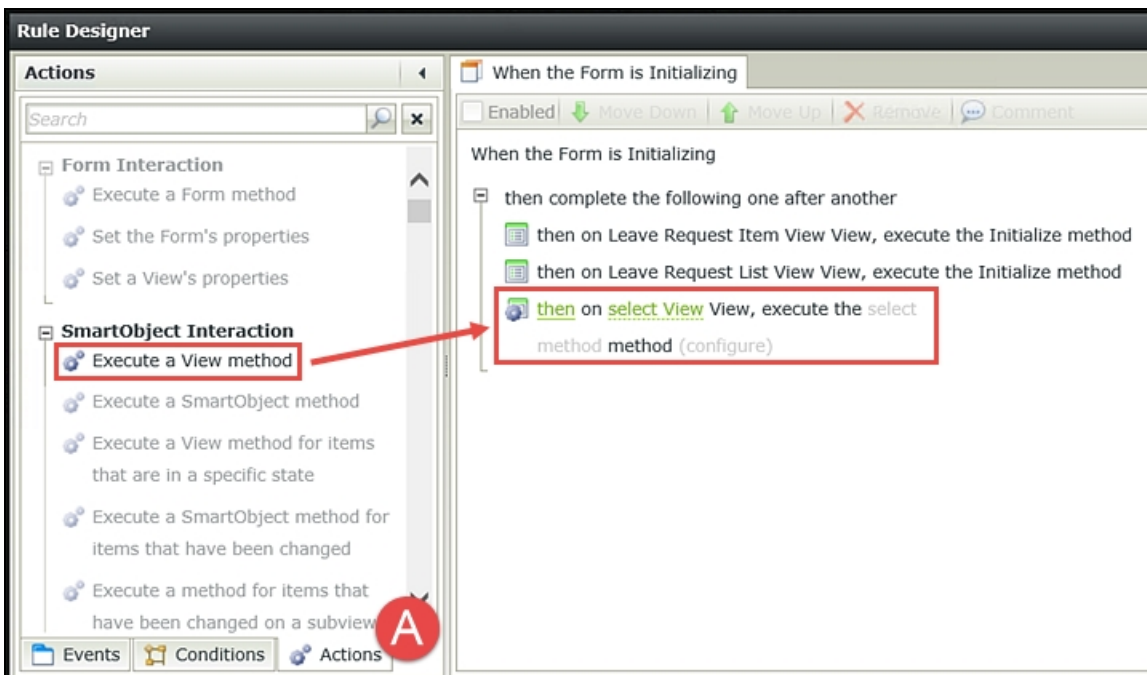
Step 10 Walkthrough

- 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**.

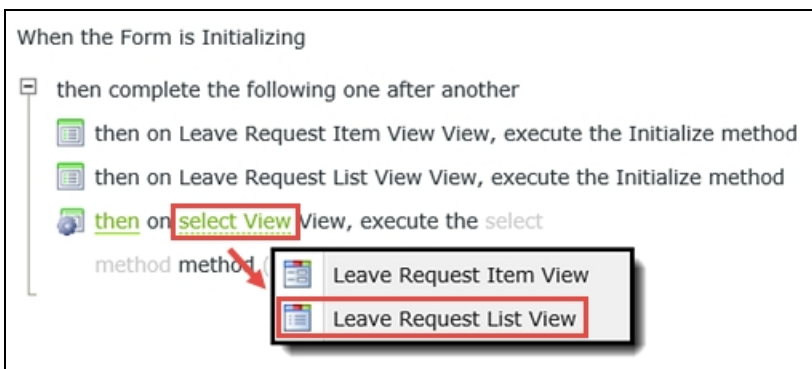


First, you want to populate the Leave Request List View. Recall that this is the view you created to output a list of all leave requests previously submitted by the form originator. To do this, you want to execute (call) a SmartObject method (Get List) for the list view. You want K2 to populate the previous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

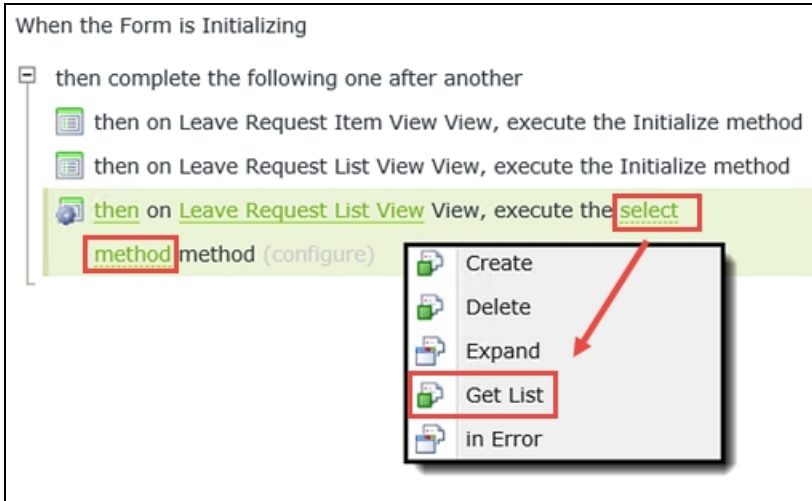
- b. From the Rule Designer **Actions** pane (A below), click once on the action **Execute a View method** to add it to the Rule Definition pane.



- 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**.

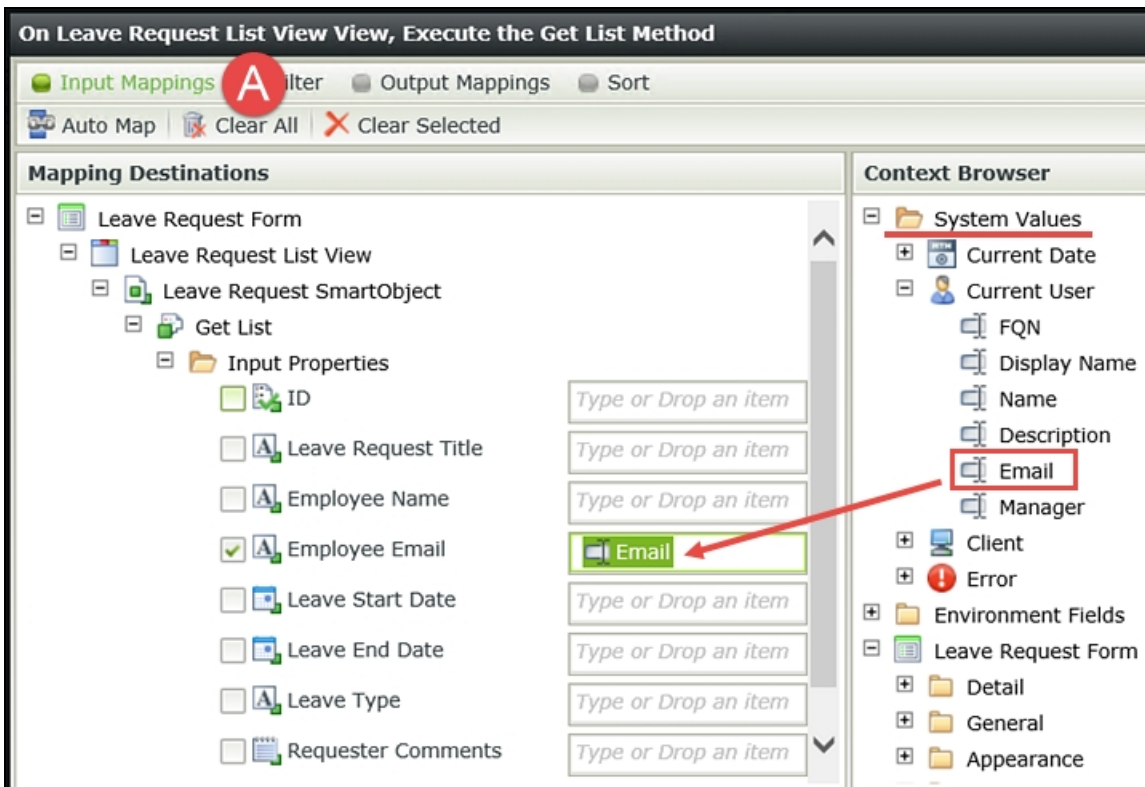


- d. Click the **select method** link and choose **Get List**. The Get List method retrieves content from a data source.

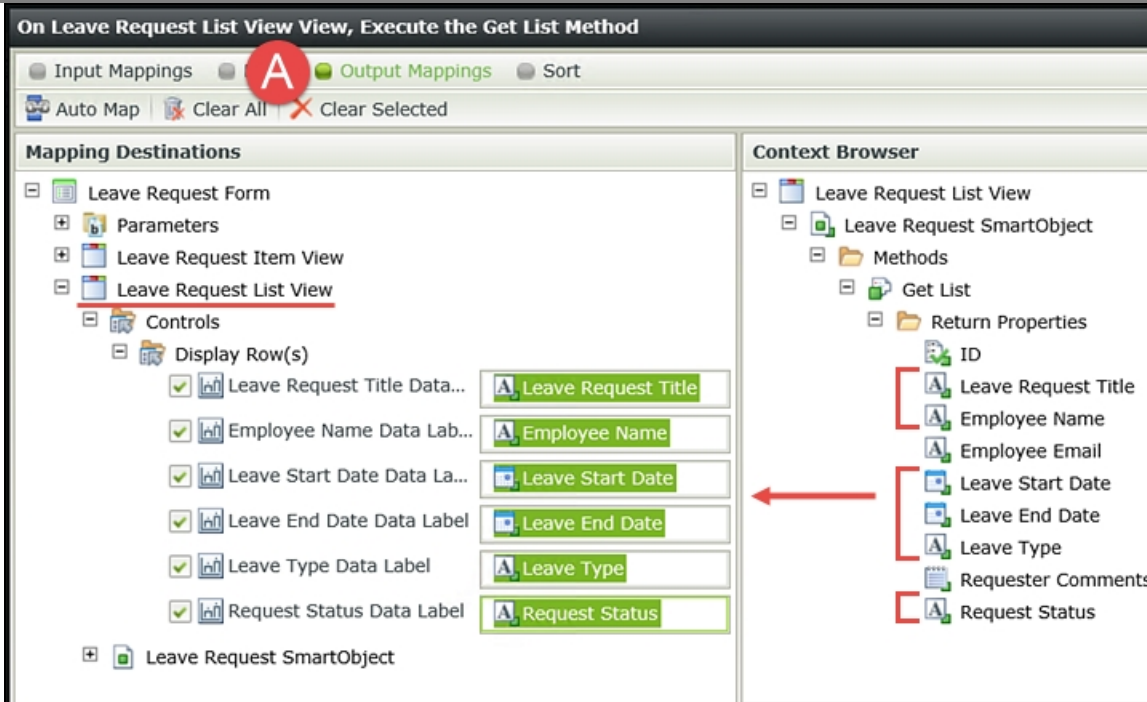


Now click the **(configure)** link. There are two primary sections to the editor that opens. First, you need to tell K2 *which* user records you want to retrieve (this is the input value) and second, you need to tell K2 which SmartObject properties you want to return (these are the output values).

- 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.

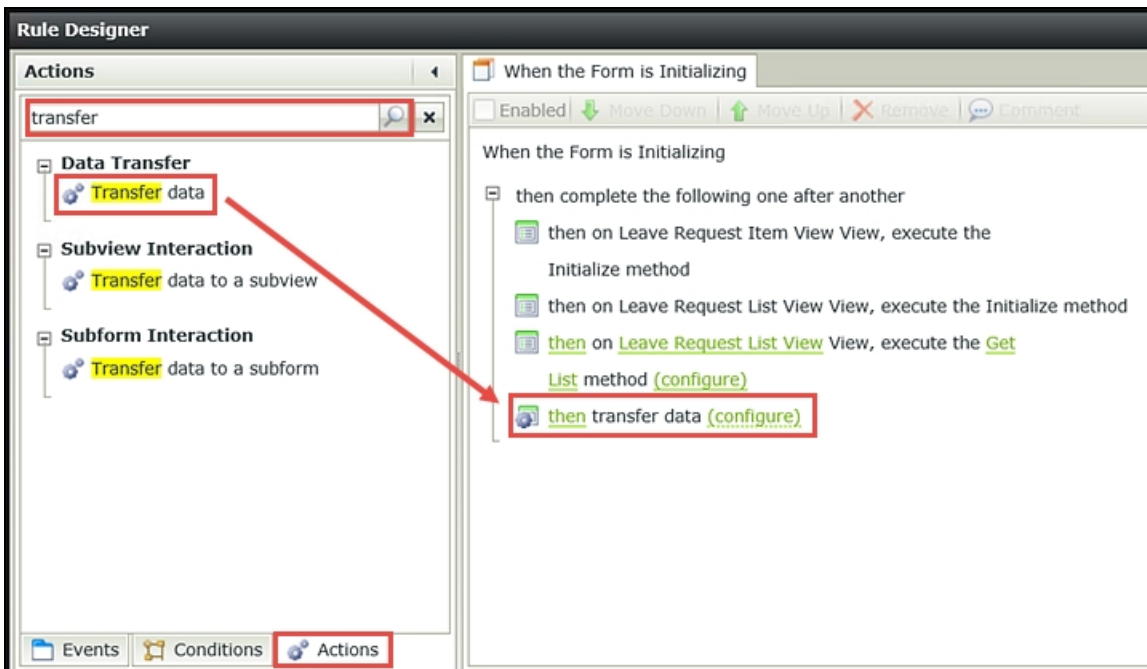


- f. Now click the **Output Mappings** button. (A below). On this screen, you want to tell K2 which properties you want returned to your form. Using the image below as a guide, drag the highlighted **Return Properties** into the Leave Request List View fields. (You may have to collapse the Item View 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.



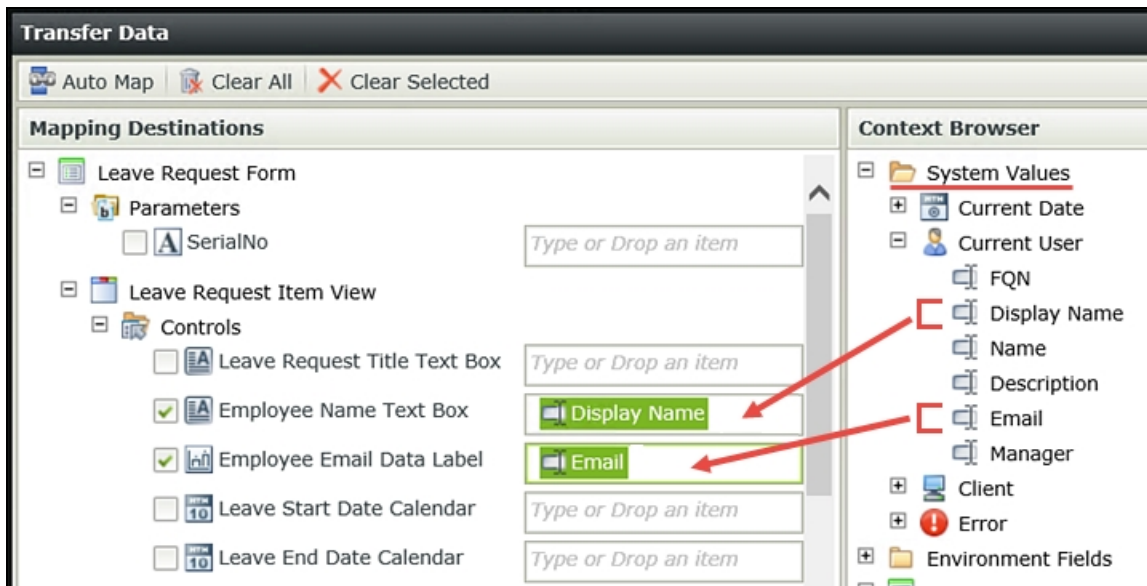
Next, you want to display the current user's name and email on the Leave Request Item View. K2 will automatically populate these fields based on who is logged in. These values will then be saved to the Leave Request SmartObject when the form is submitted, along with the leave request details. This step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

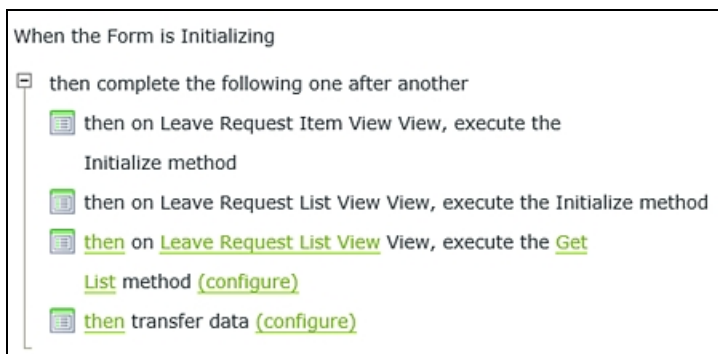


- h. Click the **(configure)** link. In the Context Browser, expand the **System Values** node, then **Current User**. Drag the **Display Name** into the **Employee Name Text Box** field. Drag the **Email** into the **Employee Email Data Label** field. Click **OK**.

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.



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



STEP 10 REVIEW

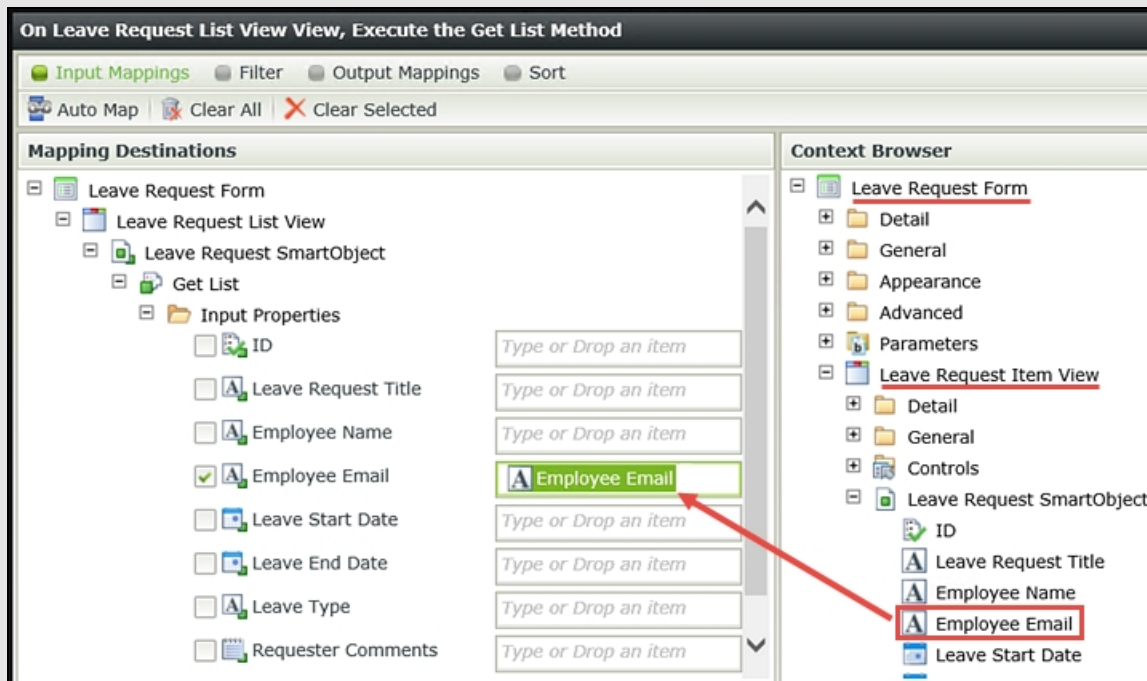
In this step, you edited the Form Initializing rule on the Leave Request Workflow (Default) state, so that the current user's previous leave requests will populate the Leave Request List View when the form is first loaded. The Default state is the form configuration that the form originator will see. 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

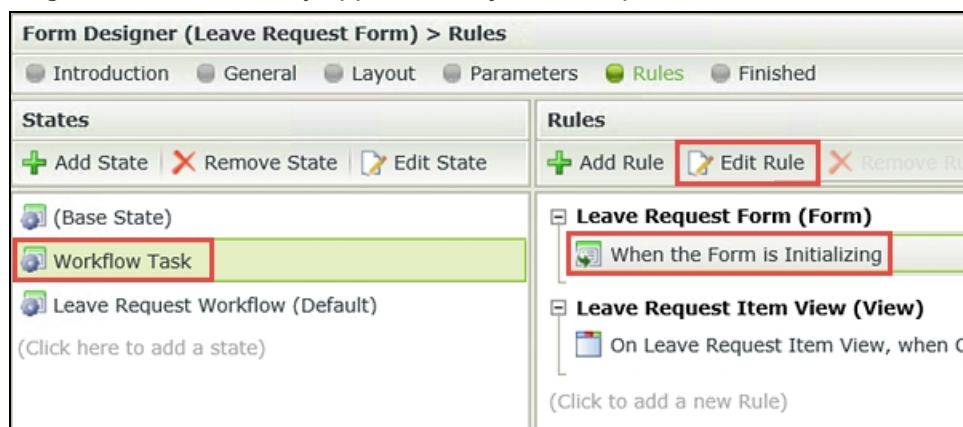
1. Edit the **Form Initializing** rule for the **Workflow Task** state. Execute the **Get List** method for the **Leave Request List View**.
2. For the **Input Property**, use the Leave Request SmartObject **Employee Email** property for the **Employee Email** input property.



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

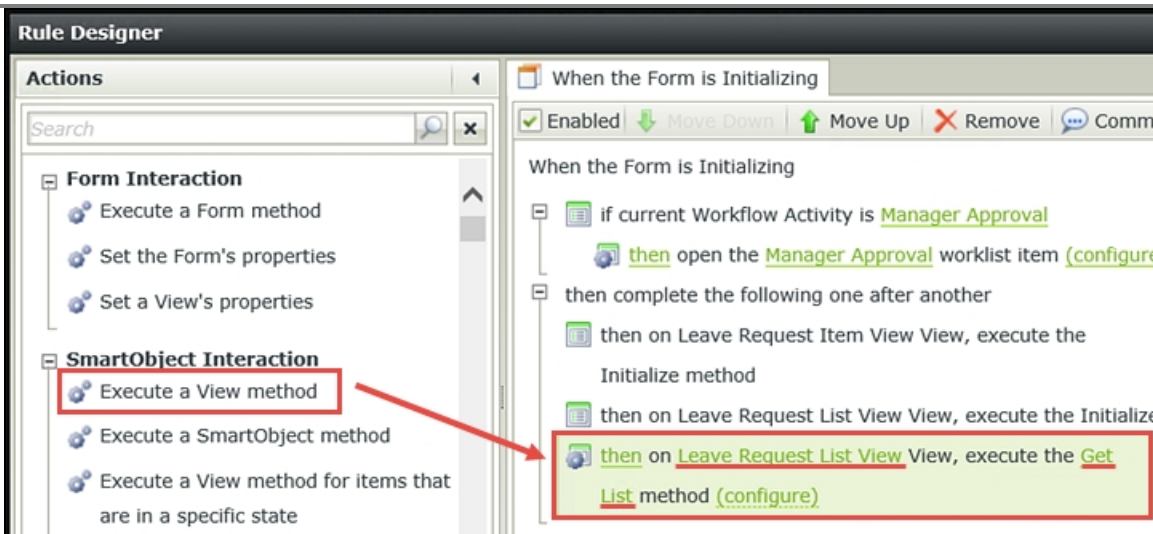
Step 11 Walkthrough

- 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.



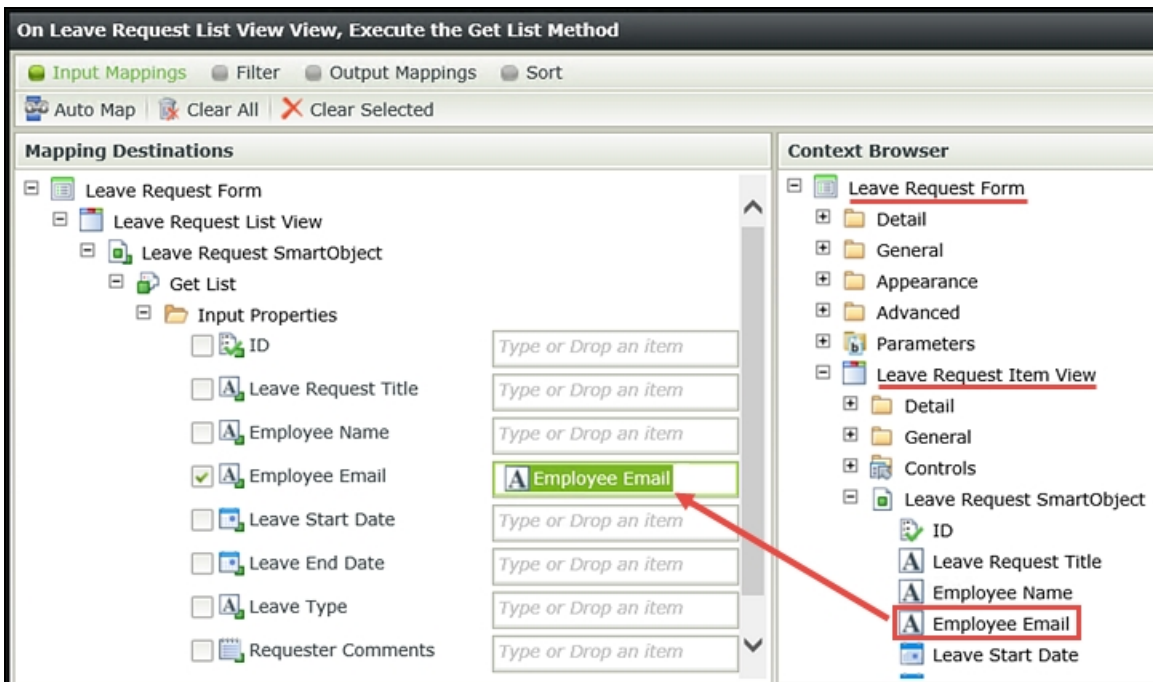
First, you want to configure an action so that the originator's previous leave requests are loaded. Recall in previous steps, you told K2 to retrieve the previous requests of the current user. On the manager's form, you don't want the current user (that would be the manager), instead you need K2 to retrieve 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**.

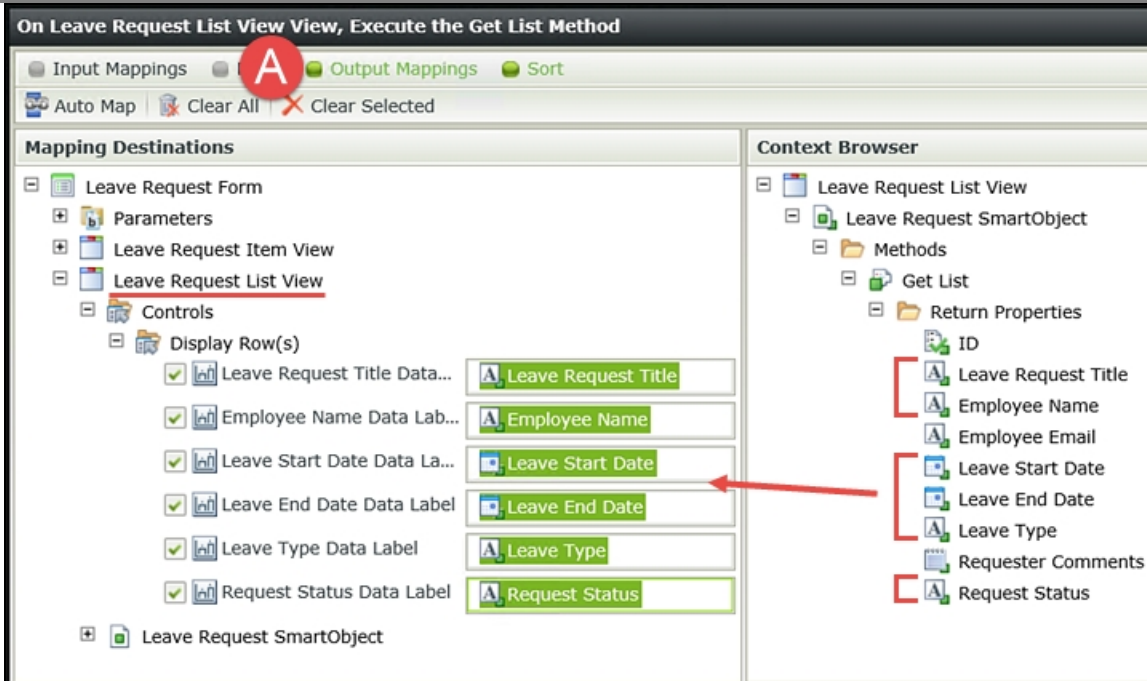


- c. Click the **(configure)** link. Using steps very similar to those in Step 10, you will map the Input Properties and the Return Properties. But, instead of using the System Value for the email input, you will use the SmartObject value for the email input because you don't want the email of the current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

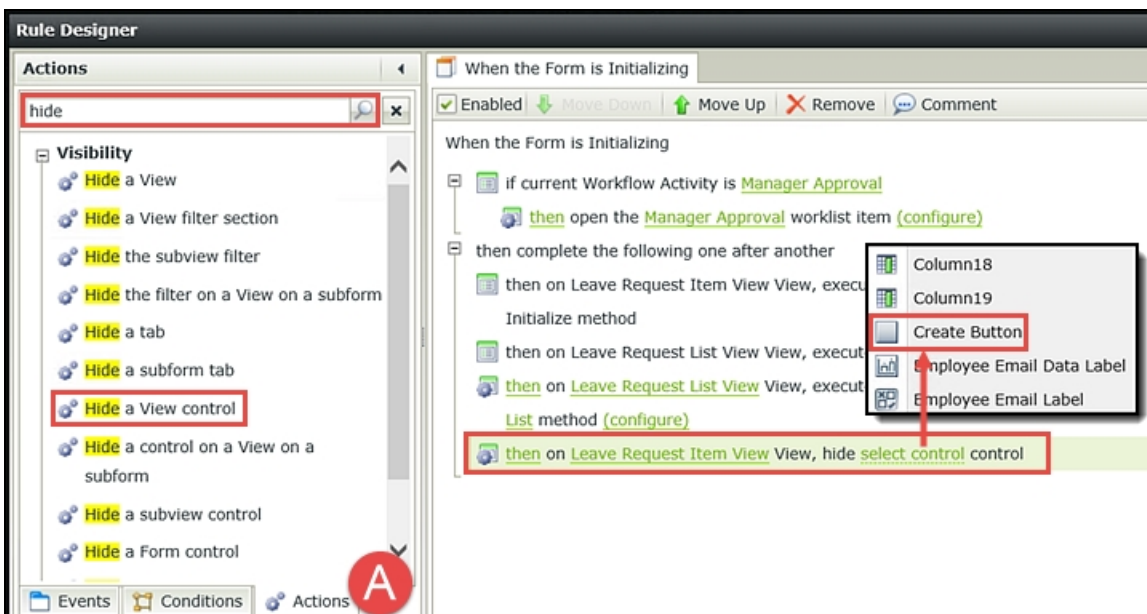
In the Context Browser expand the **Leave Request Item View** node, then the **Leave Request** SmartObject. Drag the **Employee Email** into the Employee Email input property.



- 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.



- 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

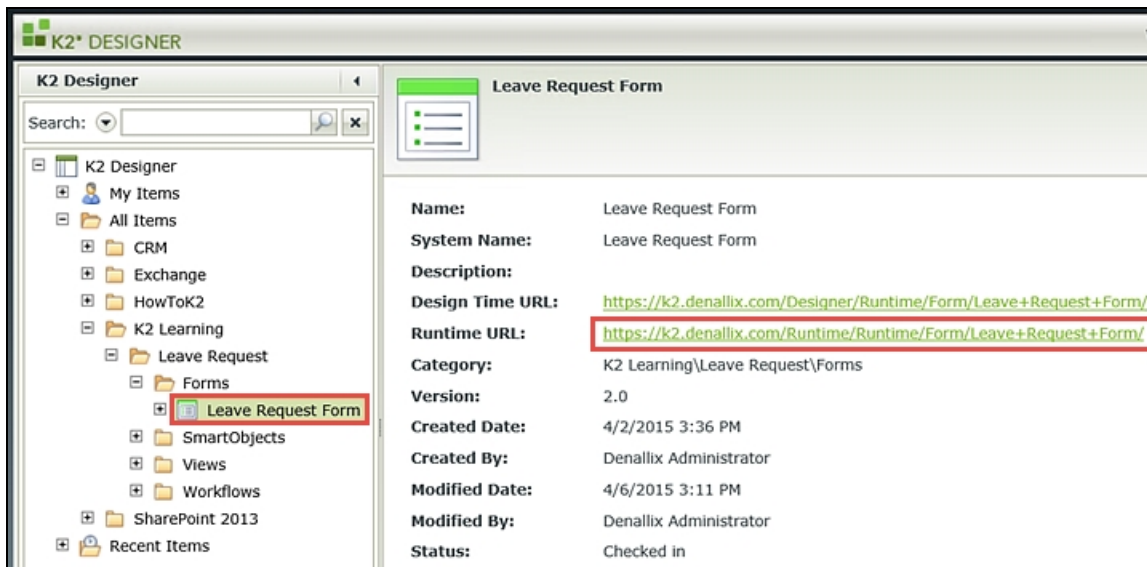
You will submit two Leave Request Forms. Using the link found in one of the task notification emails, you will open the worklist item and action it (mark it as "Approved" or "Rejected"). You will then open the Leave Request Form again to confirm that the previous two leave request records are appearing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



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

Leave Request
⤴

Leave Request Title:

Employee Name:

Employee Email:

Leave Start Date:

Leave End Date:

Leave Type: ⌵

Requester Comments:

Request Status:

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:

Employee Name:

Employee Email:

Leave Start Date:

Leave End Date:

Leave Type: ⌵

Requester Comments:

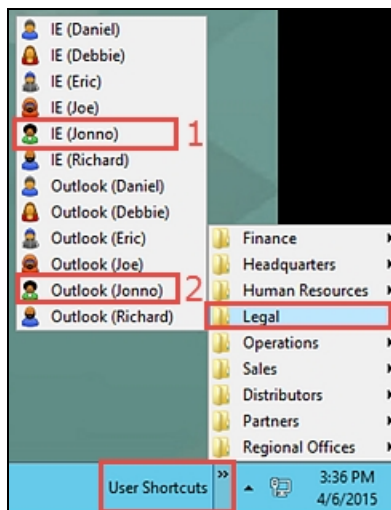
Request Status:

Previous Leave Requests
⤴

LEAVE REQUEST...	EMPLOYEE NAME	LEAVE START DA...	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
No items to display.					

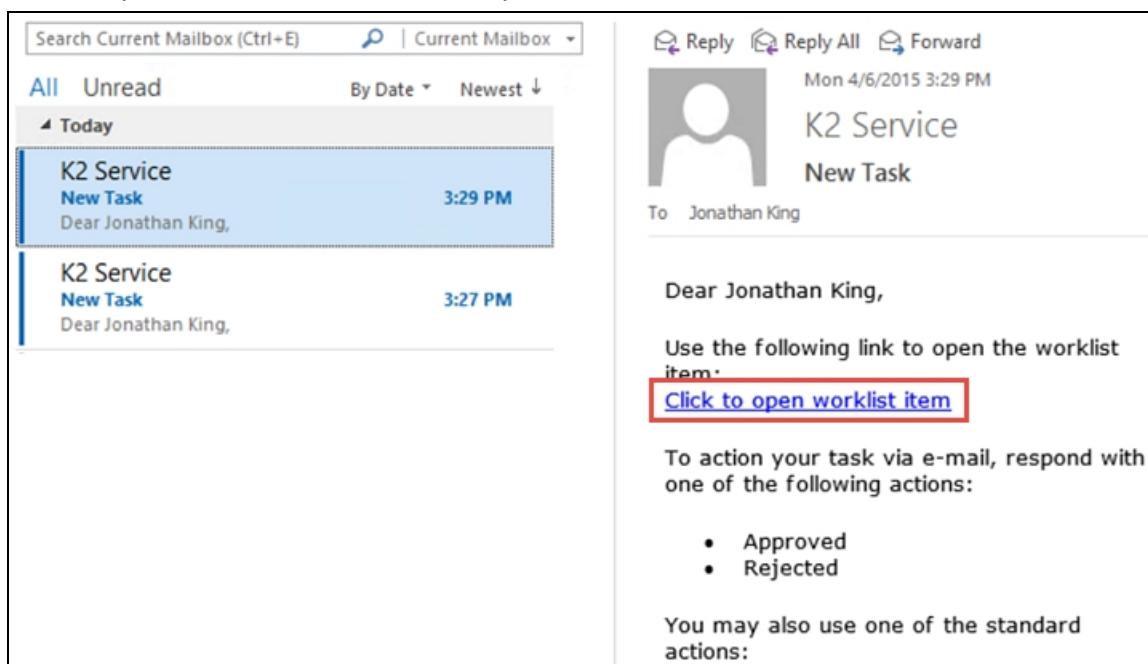
Now you need to open the approving manager's Outlook and confirm the task notification emails were received. Then you will action one of the tasks. If you are working on a K2-provided virtual environment (VM), you are likely logged in as Denallix Administrator. Administrator's manager is Jonno, so you'll 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.



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:

Activity Name:

Instruction:

Select Action: ▼

Leave Request

Leave Request Title:

Employee Name:

Employee Email:

Leave Start Date: K2

Leave End Date: K2

Leave Type: ▼

Requester Comments:

Request Status:

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

The K2 Learning logo, consisting of a small grid icon followed by the text "K2 LEARNING".

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 an 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

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).

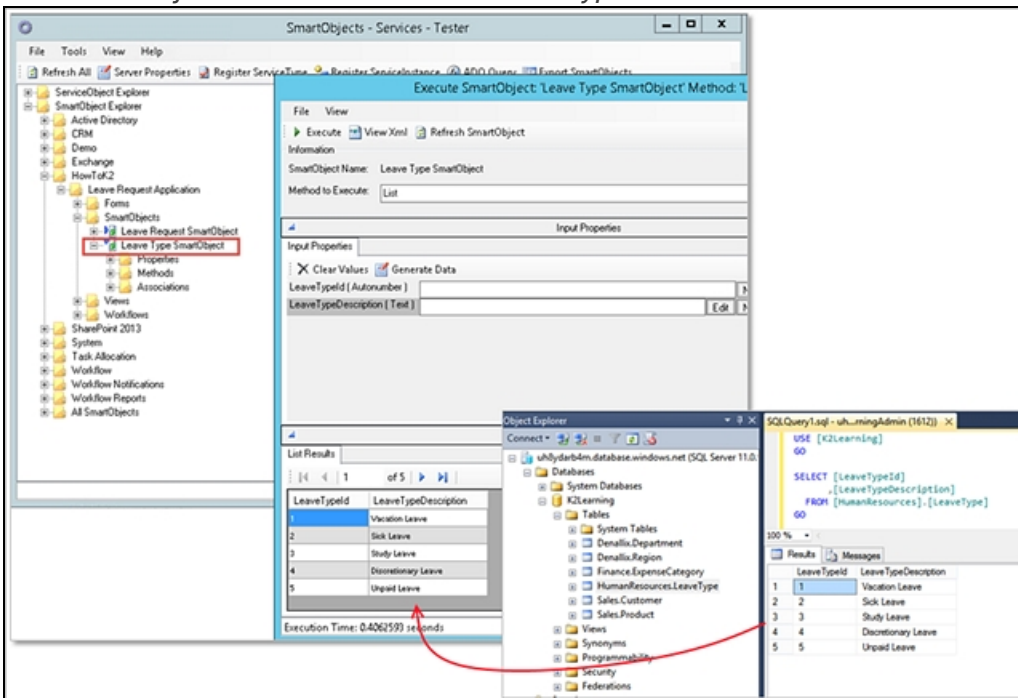
You will also learn how to register a Service Instance, auto-generate SmartObjects and move SmartObjects in the Category System.

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

Adding an "Approver Comments" property to the existing SmartObject

Name	Description	Type	Key	Required	Unique
ID	The key used to identify a specific record.	Autonumber	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Leave Request Title		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Employee Name		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Employee Email		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Leave Start Date		Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Leave End Date		Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Leave Type		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requester Comments		Memo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Request Status		Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Approver Comments		Memo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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



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 87 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**.

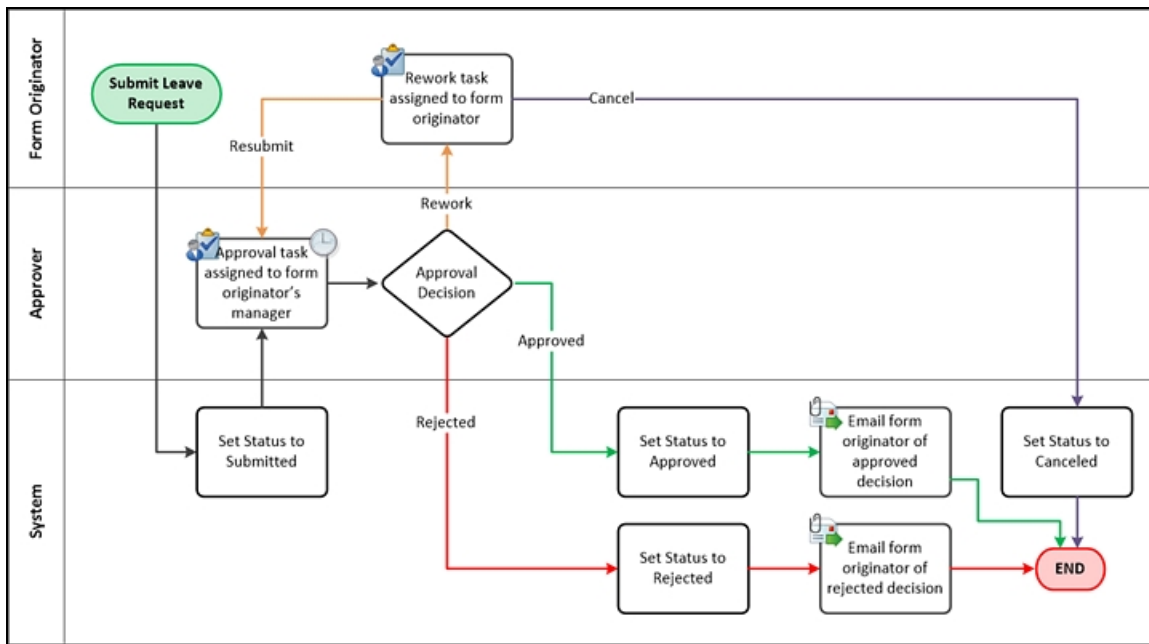
In Part 1, you expand upon the **Data** component by editing the existing Leave Request SmartObject and adding an additional property. You'll open the K2 Management site and build a service instance based on the SQL Server Service Type. You will then auto-generate a new SmartObject called Leave Types, 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.

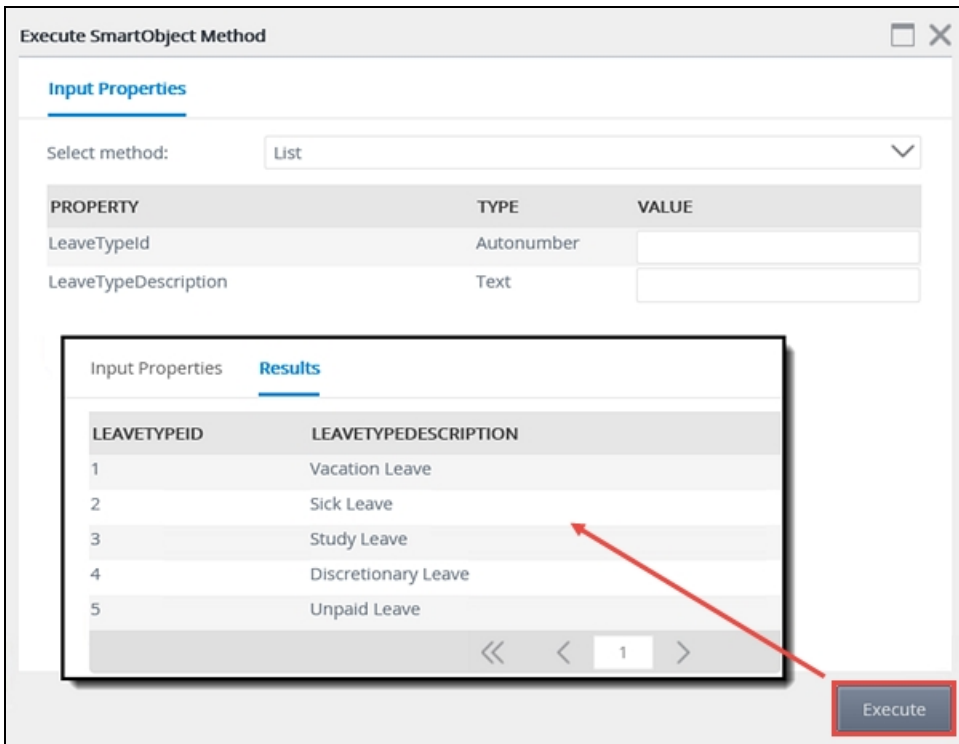


Leave Request Data

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 an 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.



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:

Employee Name:

Employee Email:

Leave Start Date:

Leave End Date:

Leave Type:

Requester Comments:

Approver Comments:

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

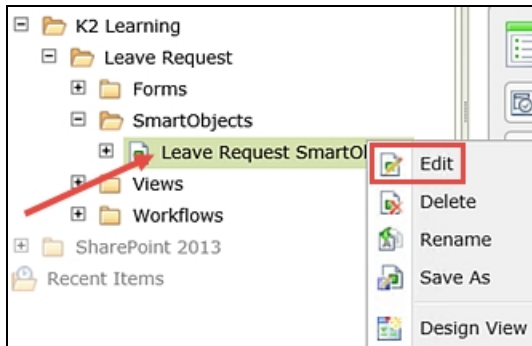
- In K2 Designer, edit the **Leave Request SmartObject** and add a **Memo** type property called *Approver Comments*

Step 1 Walkthrough

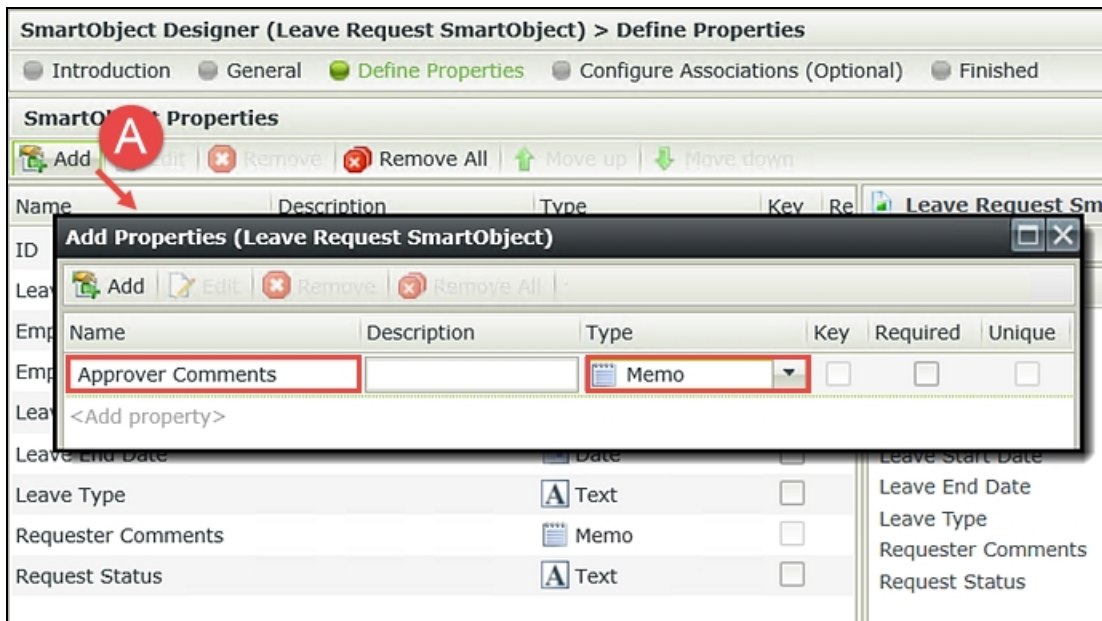
- Launch K2 Designer (**Start Button > All Programs > K2 blackpearl > K2 smartforms > K2**

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

Configure Service Instance

Service Instance

Display Name: **A** Leave Types

Description: **B** Returns a list of leave types.

Service Type: SQL Server Service

Service Authentication

Authentication Mode: **C** Static

Security Provider:

OAuth Resource Name:

OAuth Resource Audience:

User Name: **D** K2LearningUser

Password: **E** ●●●●●●●●●●

Extra: Type a value

Enforce Impersonation

Service Keys

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

SmartObjects

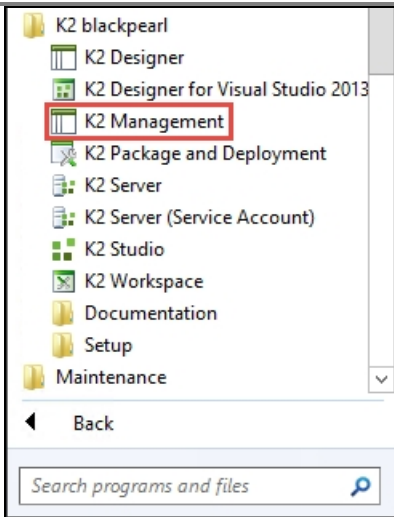
K Generate SmartObjects for this Service Instance

OK Cancel

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

Step 2 Walkthrough

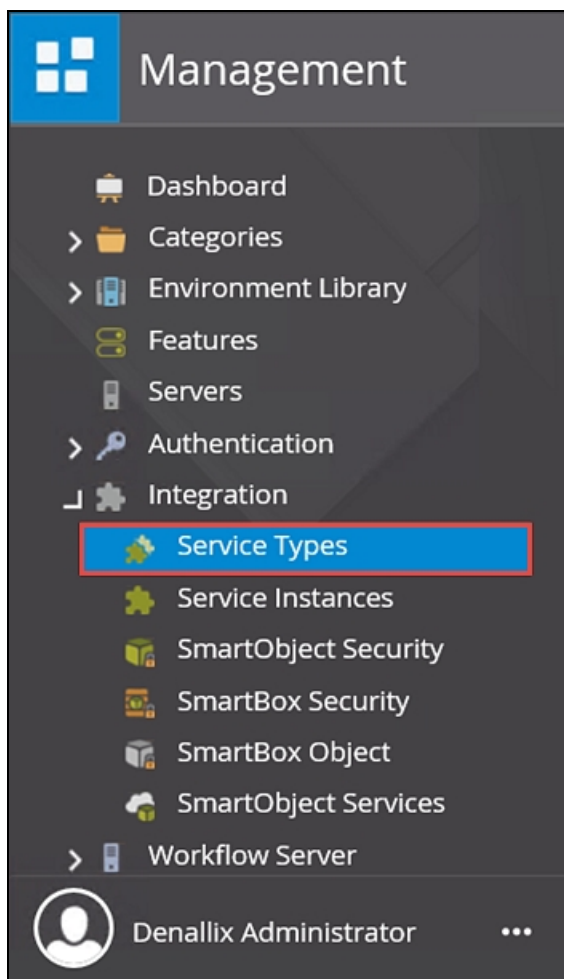
- a. Launch the **K2 Management** site. (**Start > All Programs > K2 blackpearl > K2 Management**)



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.

Service Types

+ New Edit Delete + New Instance

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

Current Service Types

- 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.

Service Types

+ New Edit Delete + New Instance

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

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

Configure Service Instance □

Service Instance

Display Name: A Leave Types

Description: B Returns a list of leave types.

Service Type: SQL Server Service ▼

Service Authentication

Authentication Mode: C Static ▼

Security Provider: ▼

OAuth Resource Name: ▼

OAuth Resource Audience:

User Name: D K2LearningUser

Password: E ●●●●●●●●

Extra:

Enforce Impersonation

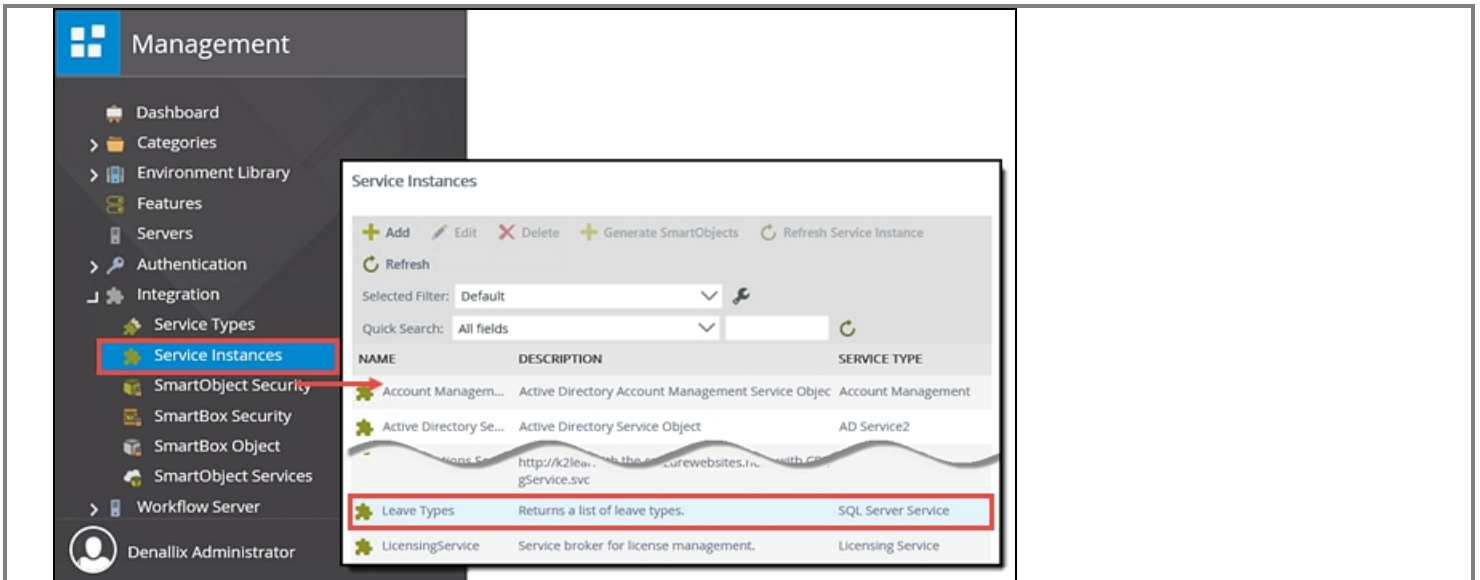
Service Keys

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

SmartObjects

K 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.



STEP 2 REVIEW

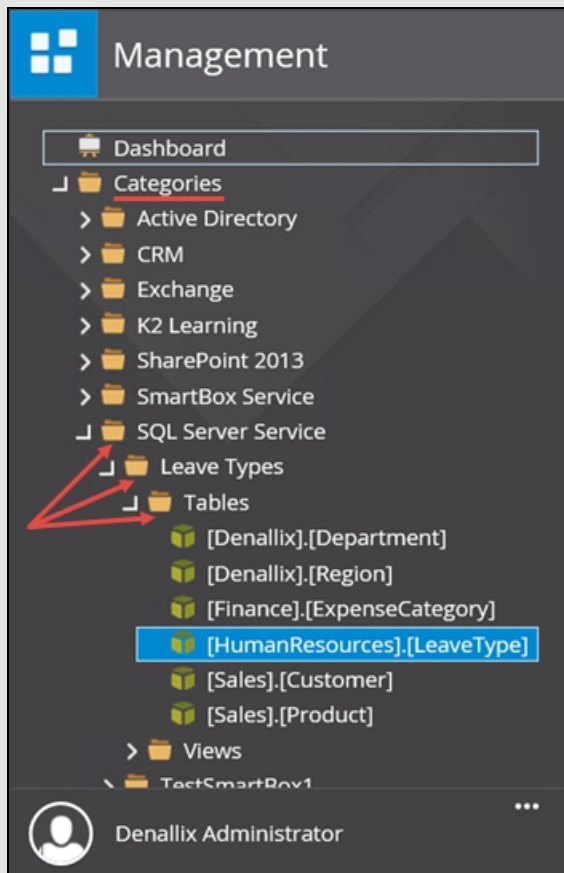
In Step 2 you created a new service instance based on the SQL Server Service Type. The new service instance contains connection settings that are specific to an external SQL database. As part of the configuration, you selected the option for K2 to automatically generate SmartObjects for the data objects 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**.



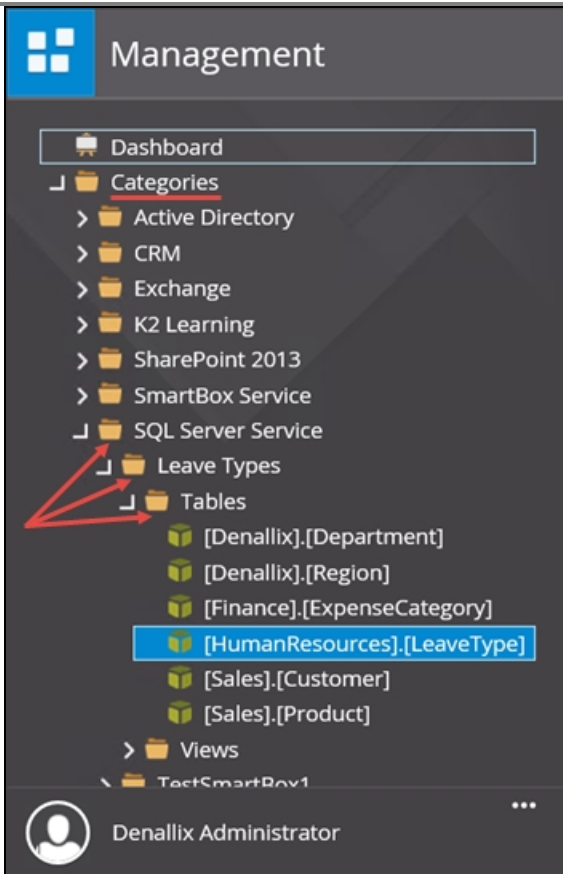
2. Click the Design link to edit the SmartObject. Rename the SmartObject to *Leave Types* then move the **Leave Types** SmartObject to the **K2 Learning > Leave Request > SmartObjects** category.

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-click 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.

Delete | Move | Copy | Design

[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
- CREATE NAME
- Create
- Delete
- List
- Read
- Update

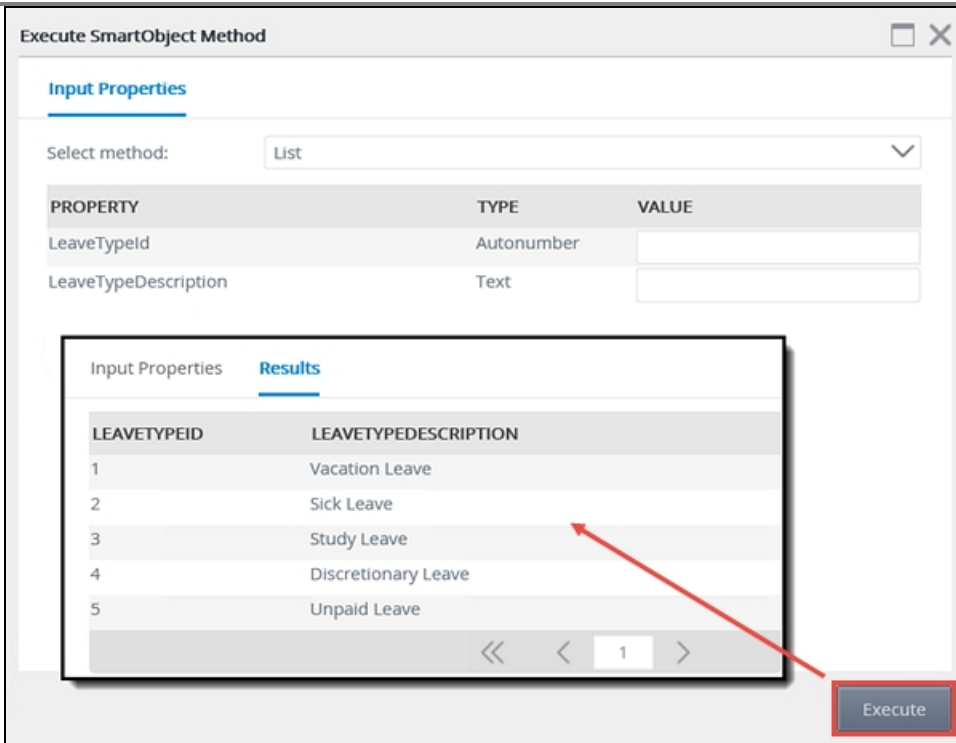
Next, you will test the SmartObject connection by executing the List method.

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

Methods

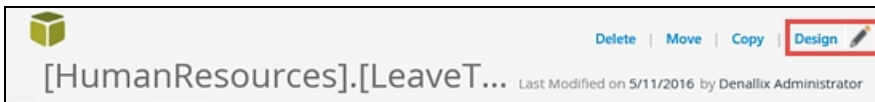
- Execute
- DISPLAY NAME
- Create
- Delete
- List
- Read
- Update

- 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.

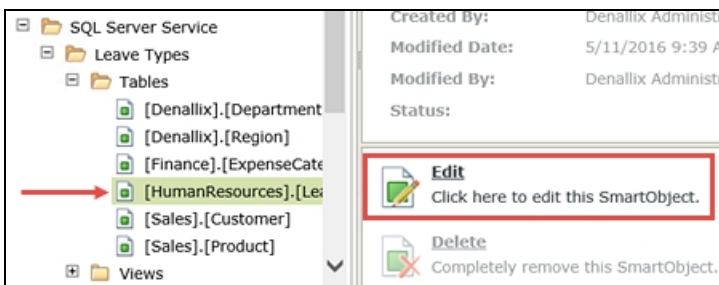


The last step in this series is to rename, then move the new SmartObject to the Leave Request category that you created using K2 Designer in the basic tutorial. This will keep all of your K2 artifacts organized and easy to locate as you move through the next steps.

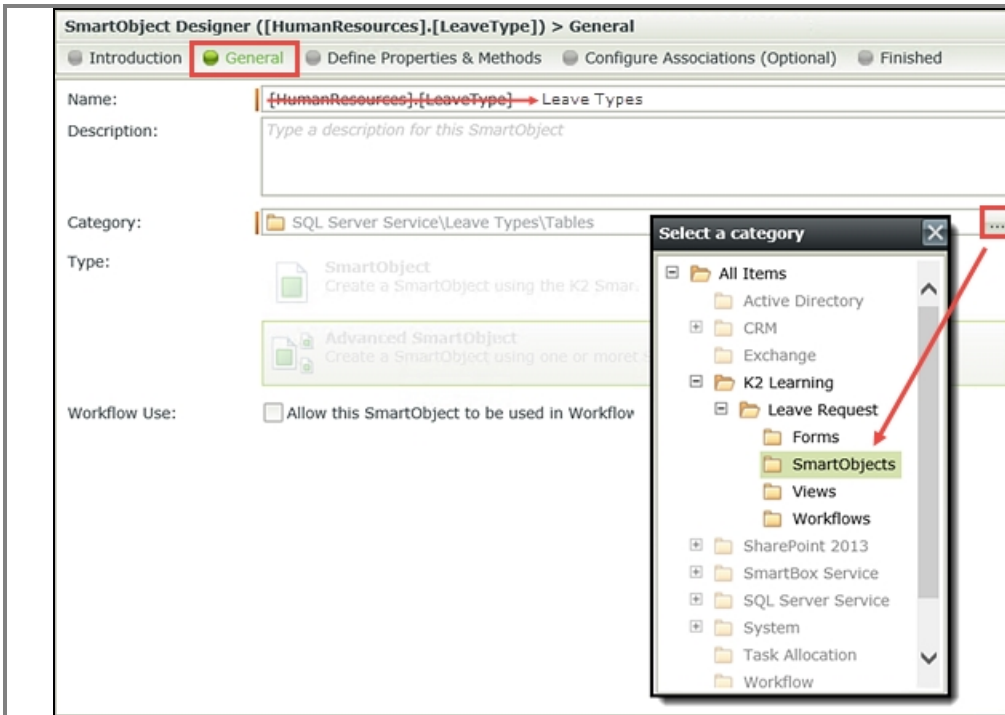
- e. Click to highlight the **[HumanResources].[LeaveType]** table to expose its properties (if they are not already). Click the **Design** link in the upper right corner of the screen.



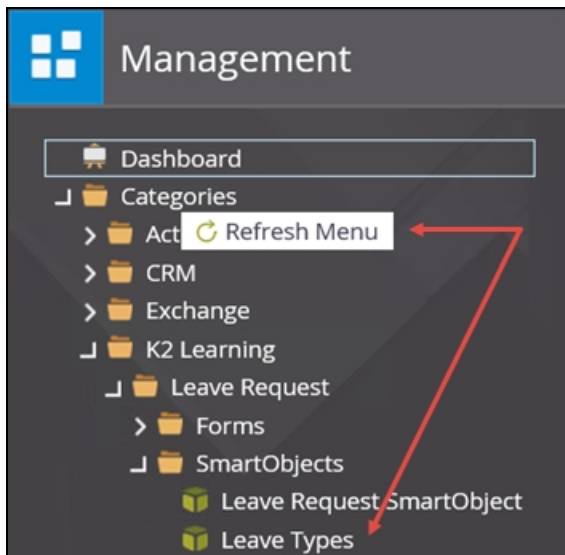
- f. You will be redirected to the K2 Designer. (The K2 Designer will open in a new tab on your browser.) Confirm the **[HumanResources].[LeaveType]** table is highlighted, then click **Edit**.



- 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.



- 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

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)

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.

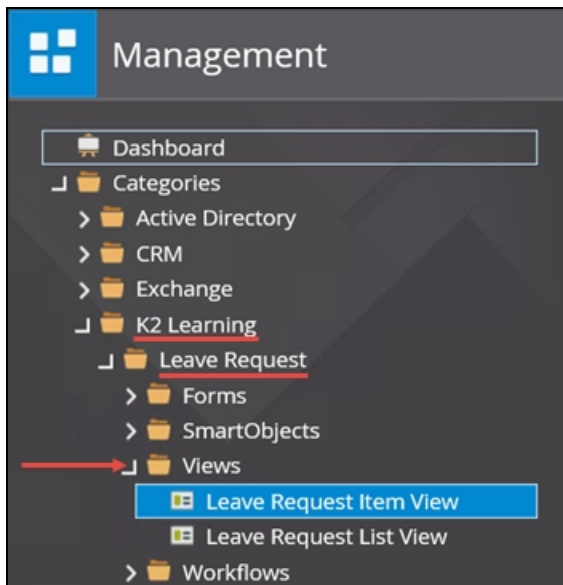
Step 4: Add the Approver Comments to the Leave Request item view and configure the Leave Types SmartObject as the data source for the Leave Type drop-down list

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.

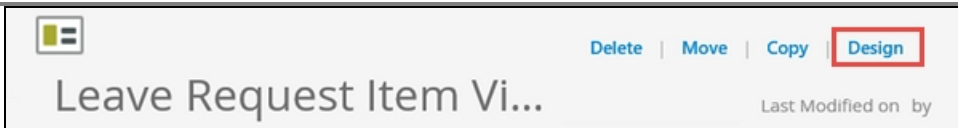
Step 4 Walkthrough

- 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.

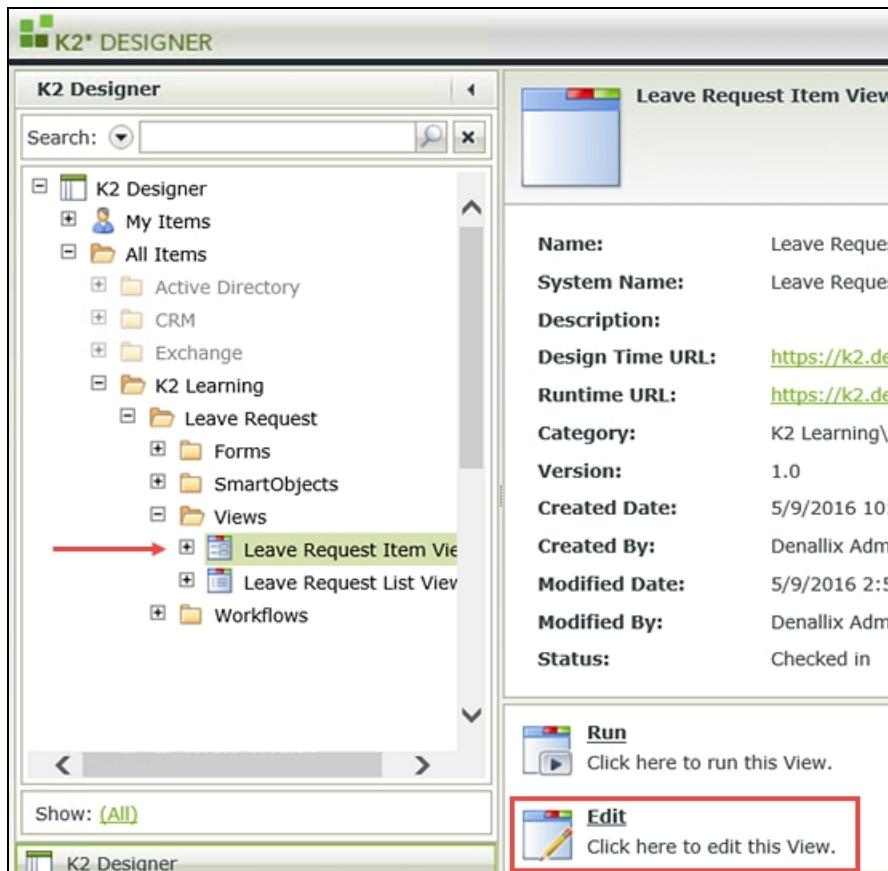


In the next few steps, you are going to replace the static values for the Leave Type drop-down list to the Leave Types SmartObject data source values.

- 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.

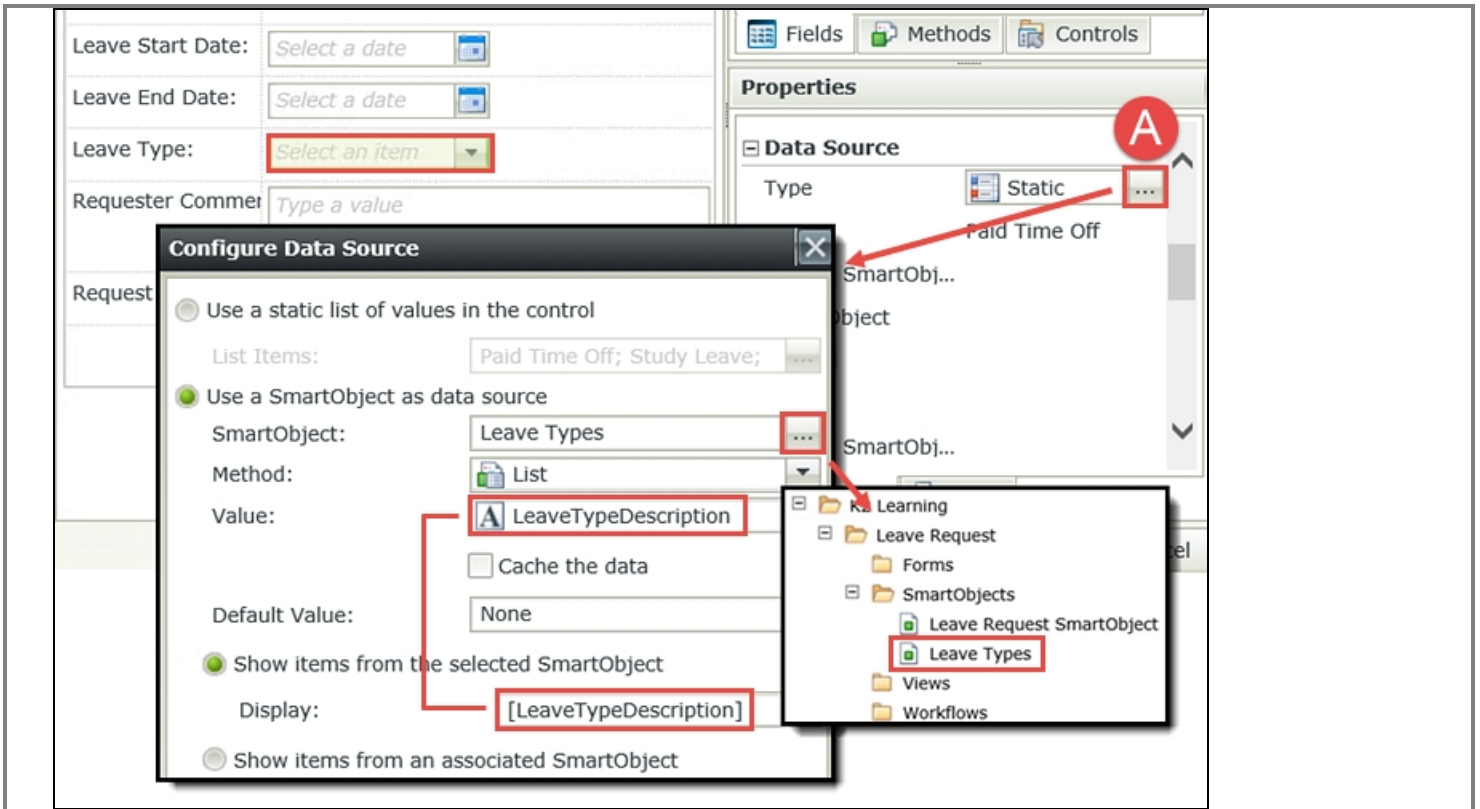


d. The K2 Designer will open in a new tab on your browser. With the item view highlighted, click **Edit**.



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

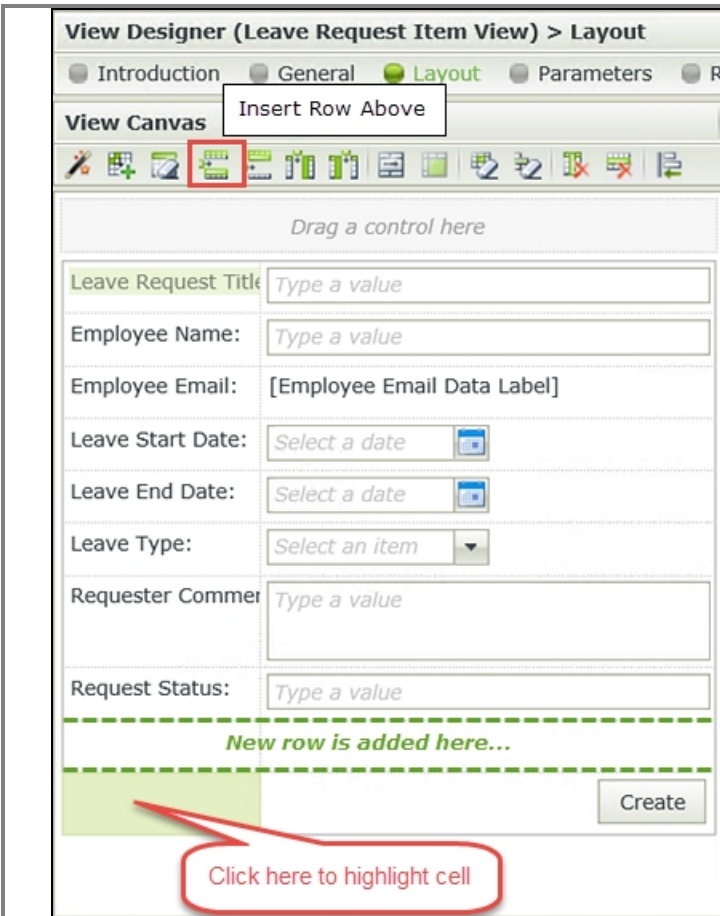


Tip

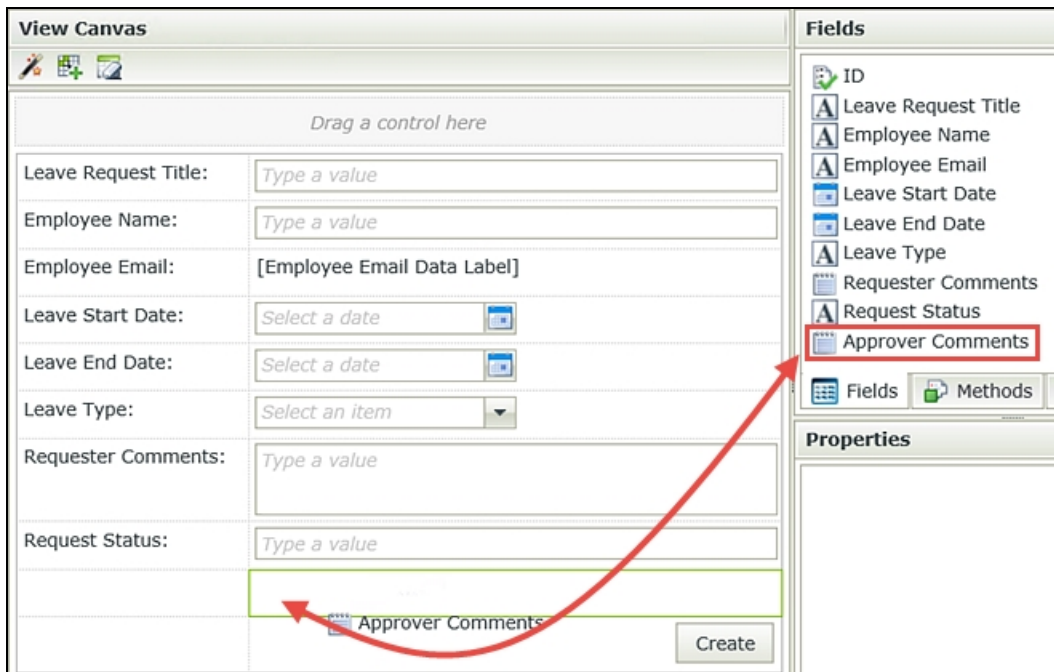
One advantage to using a SmartObject as a data source for the values in a drop-down list is that you simply update the SmartObject itself if you need to change the list values. For example, if you have several forms that all contain a Leave Type field, you would want to keep the values consistent across 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.

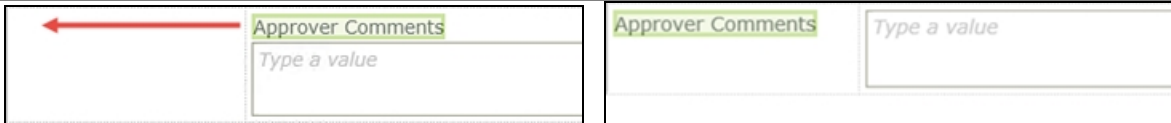
- f. First, you will add a new row to your view layout table so that you have a location for the new Approver Comments field and label. Click once in the bottom row, first cell, to highlight the cell. Click the **Insert Row Above** icon found in the View Menu to add a new row.



- 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.



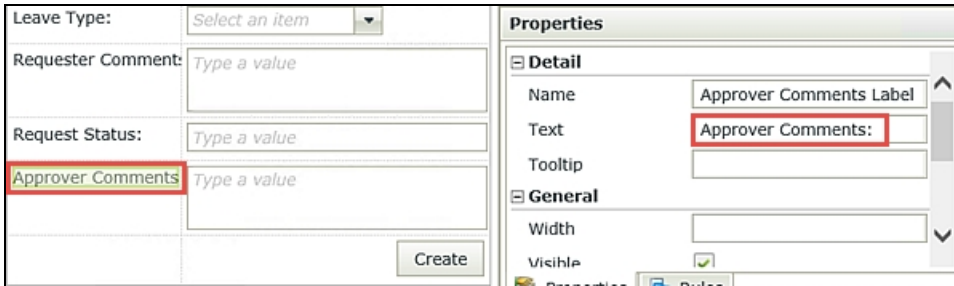
- h. Notice that K2 has placed the Approver Comments label to just above the field. Click and drag the **label** into the empty cell to the left.



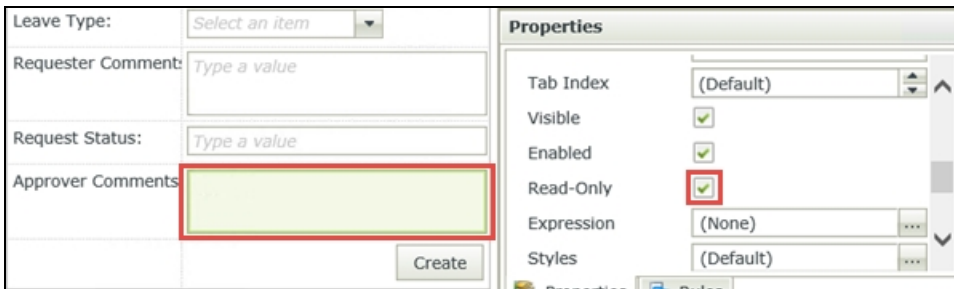
- i. You have two minor adjustments to make to the new label and field. First, highlight the label, then locate the **Text** value in the Properties pane. Add a colon (:) to the end of the text value so that it matches the other labels in the view.

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.



- 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.)



- 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

- I. 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.

K2 Designer

Search: []

- K2 Designer
 - My Items
 - All Items
 - CRM
 - Exchange
 - HowToK2
 - K2 Learning
 - Leave Request
 - Forms
 - SmartObject
 - Views
 - Leave Request Item View
 - Workflows
 - SharePoint 2013
 - Recent Items

Leave Request Item View

Name: Leave Request Title: Type a value

System Employee Name: Type a value

Description Employee Email: [Employee Email Data Label]

Design Leave Start Date: Select a date

Runtime Leave End Date: Select a date

Category Leave Type: Select an item

Version Requester Comments: []

Created Approver Comments: []

Created

Modified

Modified

Status: []

Run Click here to run this View.

Check In Check this View in.

- 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

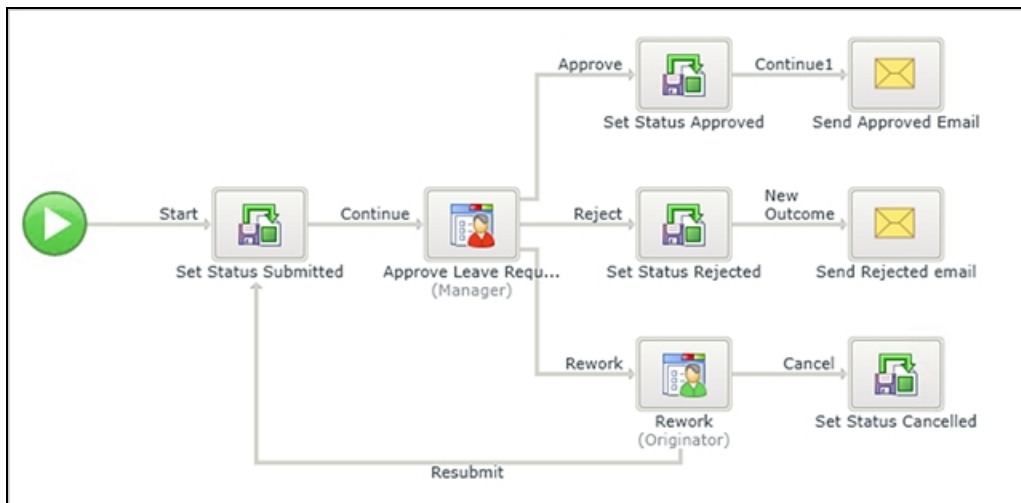
45-60 mins

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.

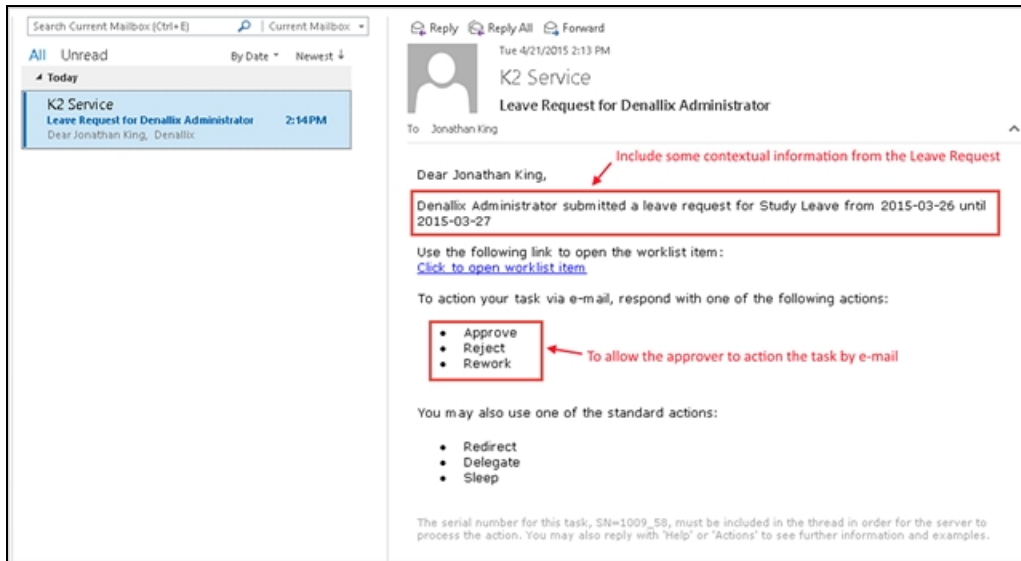
In this exercise you will learn about the following workflow functionality:

- 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.



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

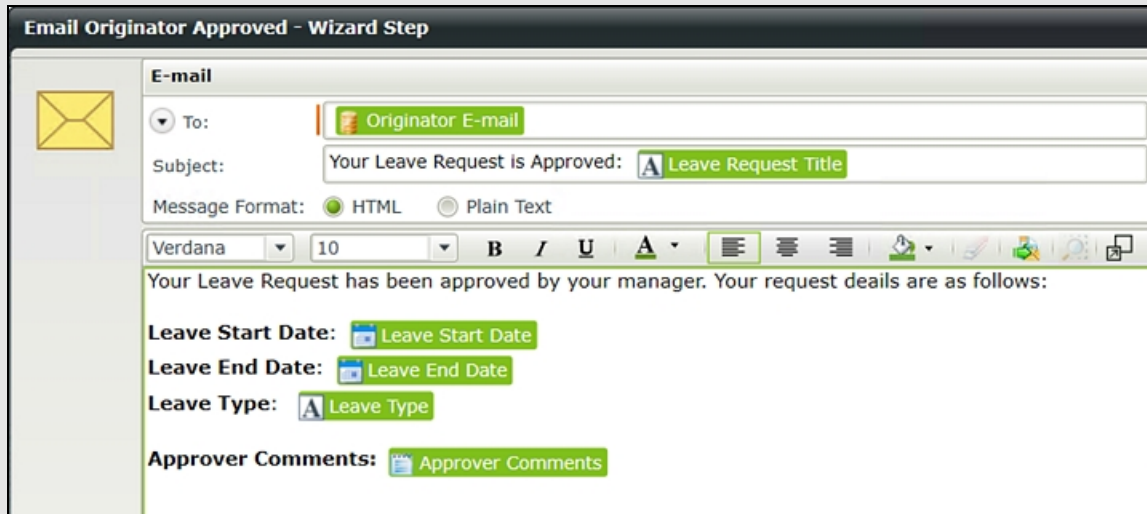
(If you have not completed [Part 2: Forms](#) of this tutorial yet, please do so before continuing with this part.)

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

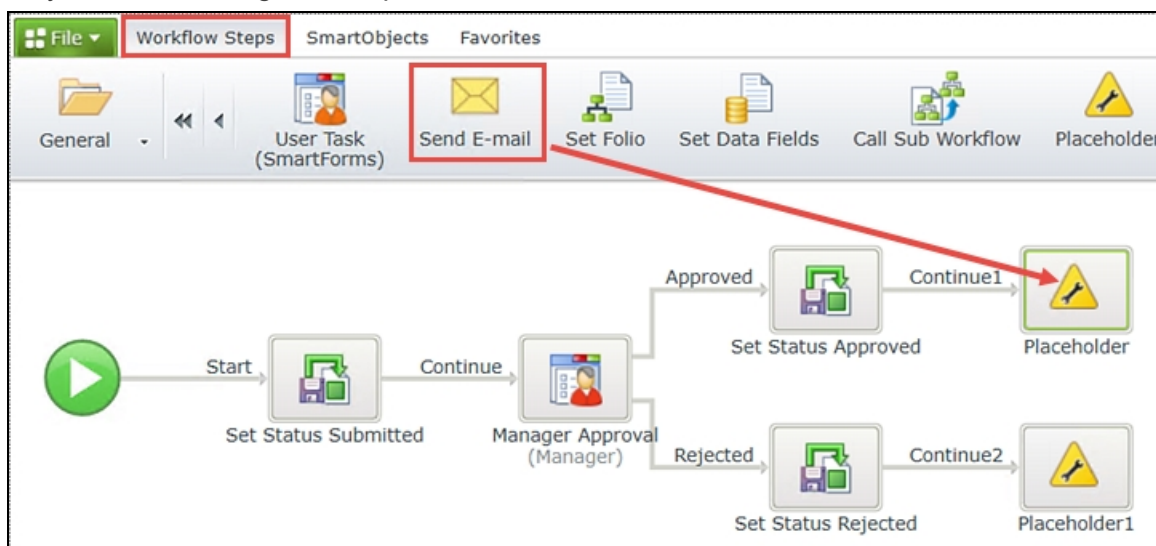
1. Add a **Send E-mail** event to the approved outcome event, replacing the existing Placeholder. Route the email to the **Originator Email**.
2. Customize the subject line and message body using Item References properties. You may structure the content in any manner you like; the image below is an example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



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

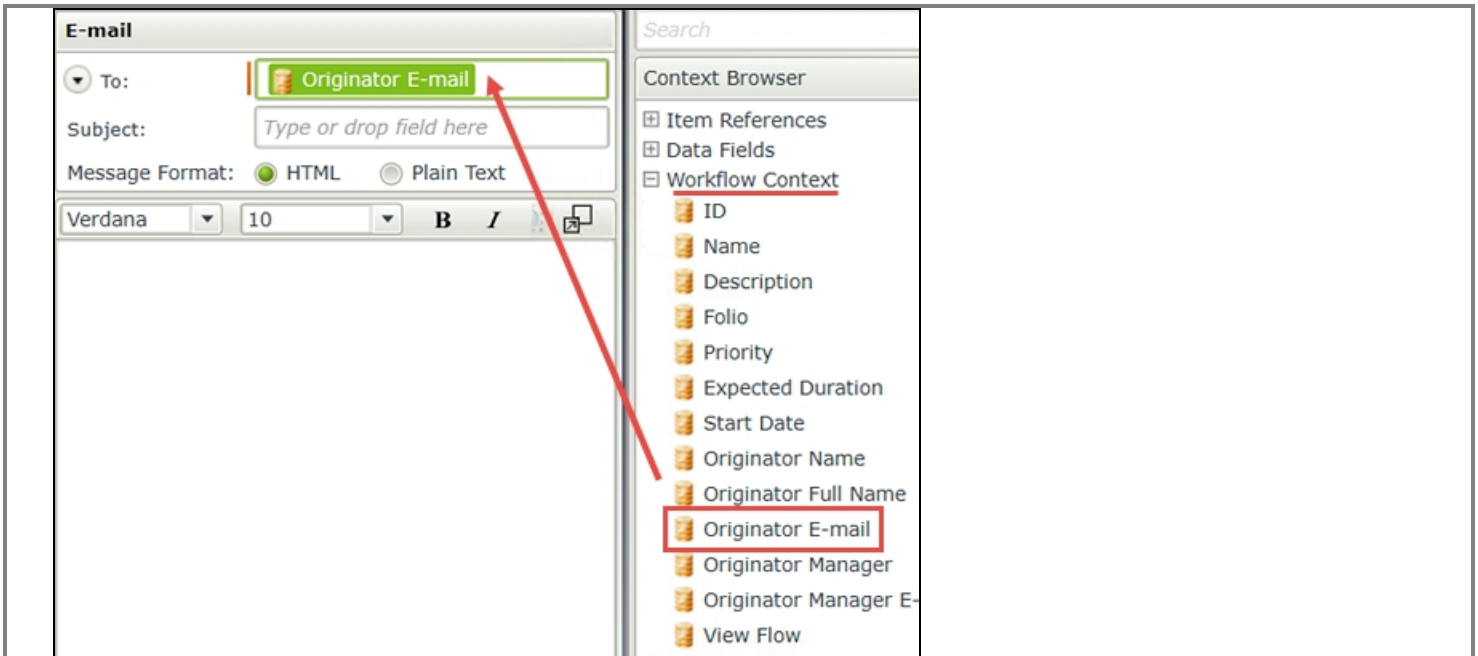
- a. In K2 Designer, edit the **Leave Request Workflow**. (Right-click the workflow name in the category tree, then select **Edit**.)
- b. Click the **Workflow Steps** tab to expose the wizards if it is not already open. Drag a **Send E-mail** event into the approved outcome Placeholder event. Click **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

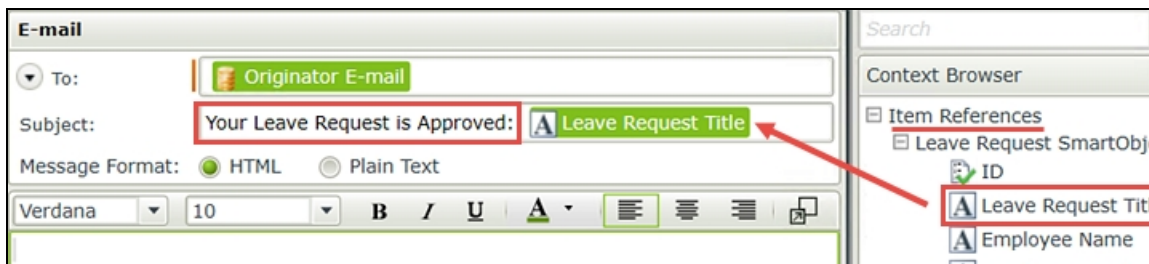
Recall that the context browser provides dynamic values. Context browser properties are displayed as green blocks of text, so that you can easily distinguish them from other properties. At runtime, the context browser properties are replaced by "real" values. For example, in the image below, the Originator 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.



- 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. 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:

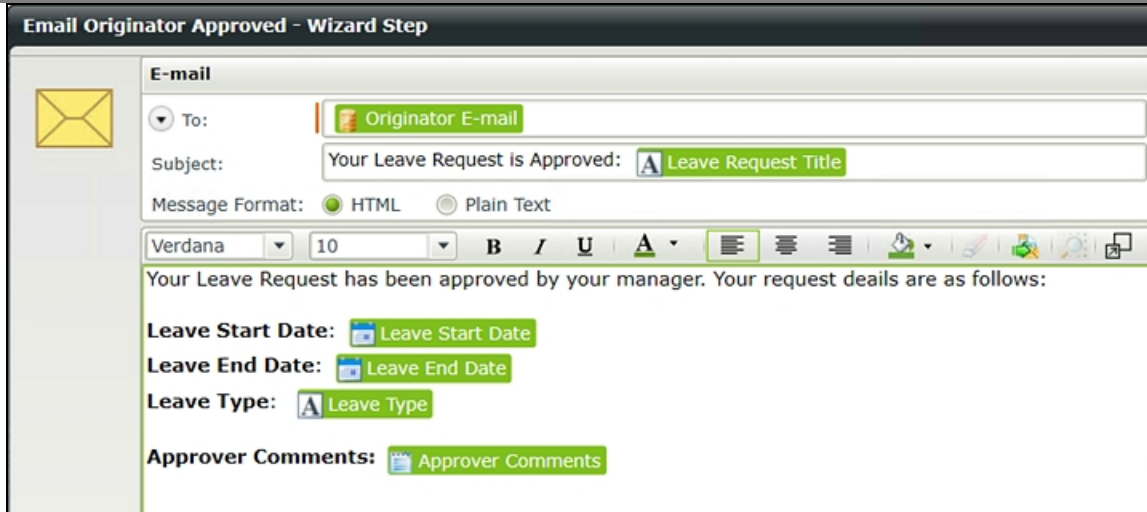
Leave Start Date: [Leave Start Date]

Leave End Date: [Leave End Date]

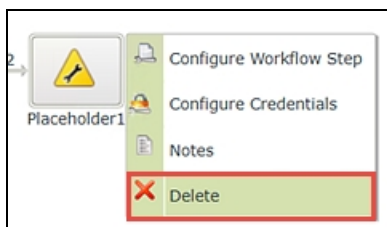
Leave Type: [Leave Type]

Approver Comments: [Approver Comments]

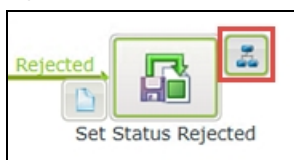
Your email should look like the image below. Click **OK** when ready.



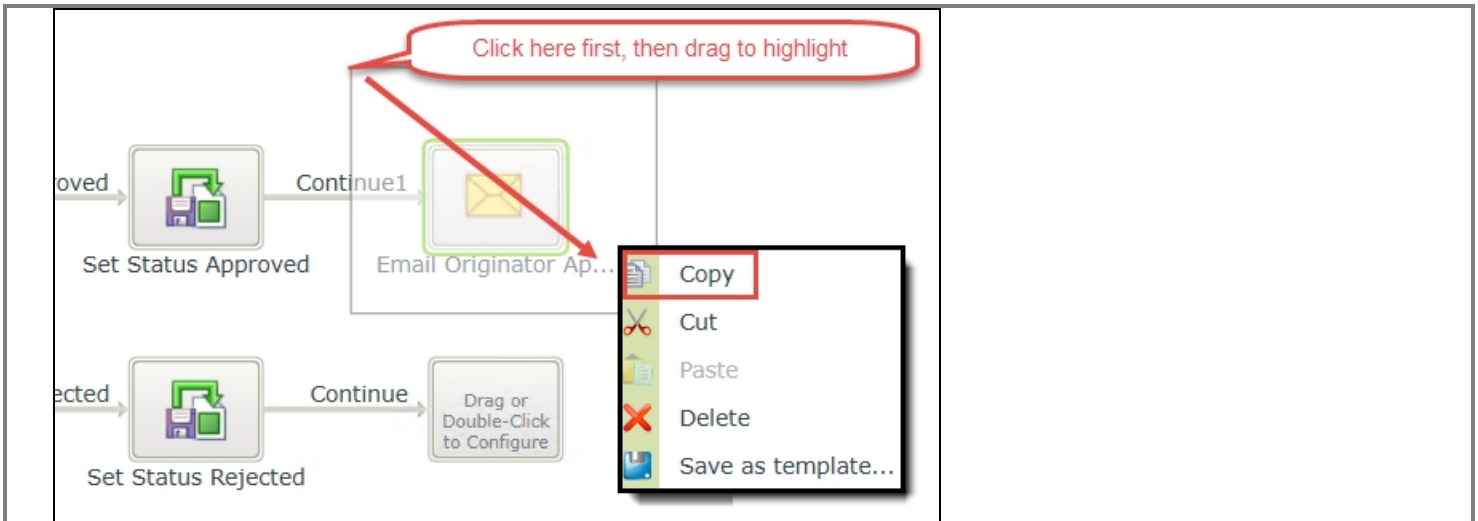
- f. Double-click the Placeholder title of the step and change it to *Email Originator Approved*
- g. In the basic Leave Request tutorial, you copied and pasted events to speed up the build process. You will do the same here. Begin by deleting the rejected side Placeholder so that you can paste the e-mail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



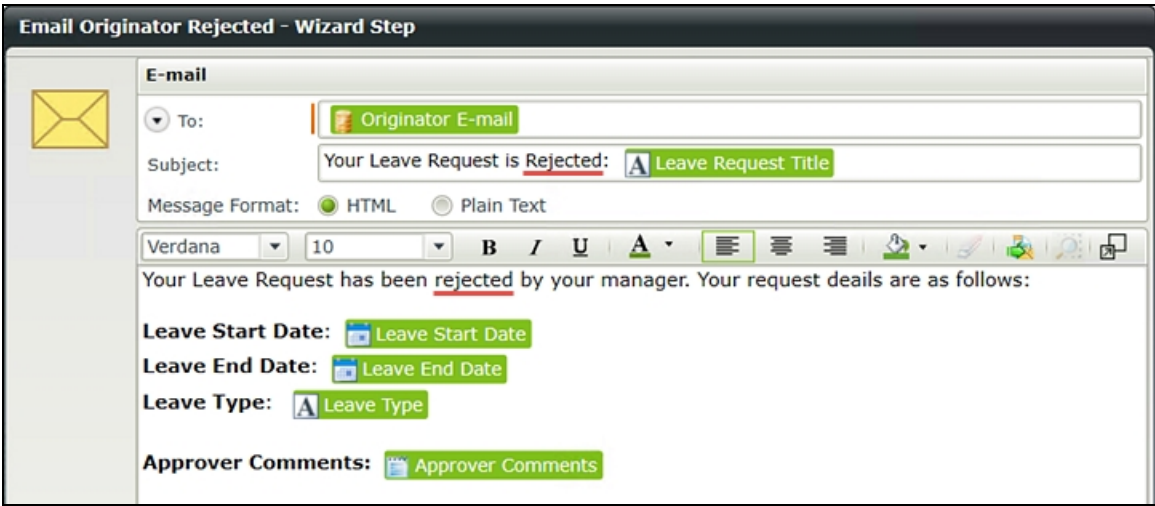
- h. Next, you need to add back in the Continue outcome line, which will add a corresponding event holder. Hover over the Set Status Rejected event until you see the **Workflow Step Outcomes** icon (it looks like a blue mini-flowchart). (If you don't see the icon when hovering, click anywhere on the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



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.

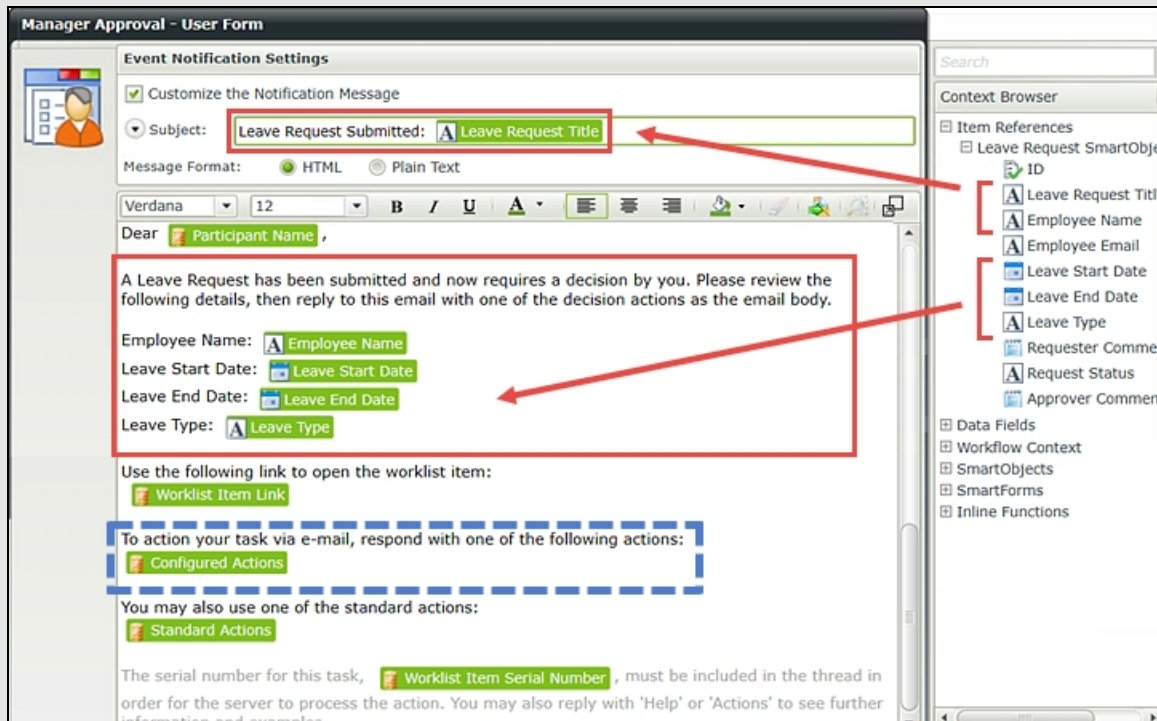
Step 6: Add a third manager outcome to rework the request and configure a new user rework task

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

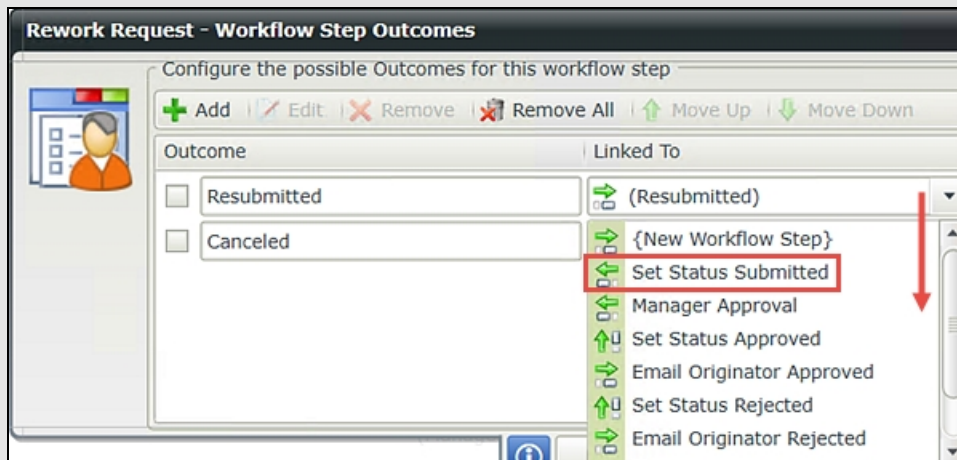
user task for the form originator to action the rework task, then add events to complete the rework decision outcomes.

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.



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.



- 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

- 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:

Instruction:

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

Allow user to action the task without opening the Form
 Automatically generate Outcomes for the above Actions

- 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**.
- 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]

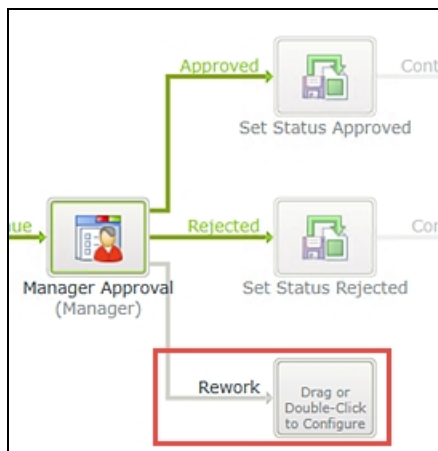
Leave Start Date: [Leave Start Date]

Leave End Date: [Leave End Date]

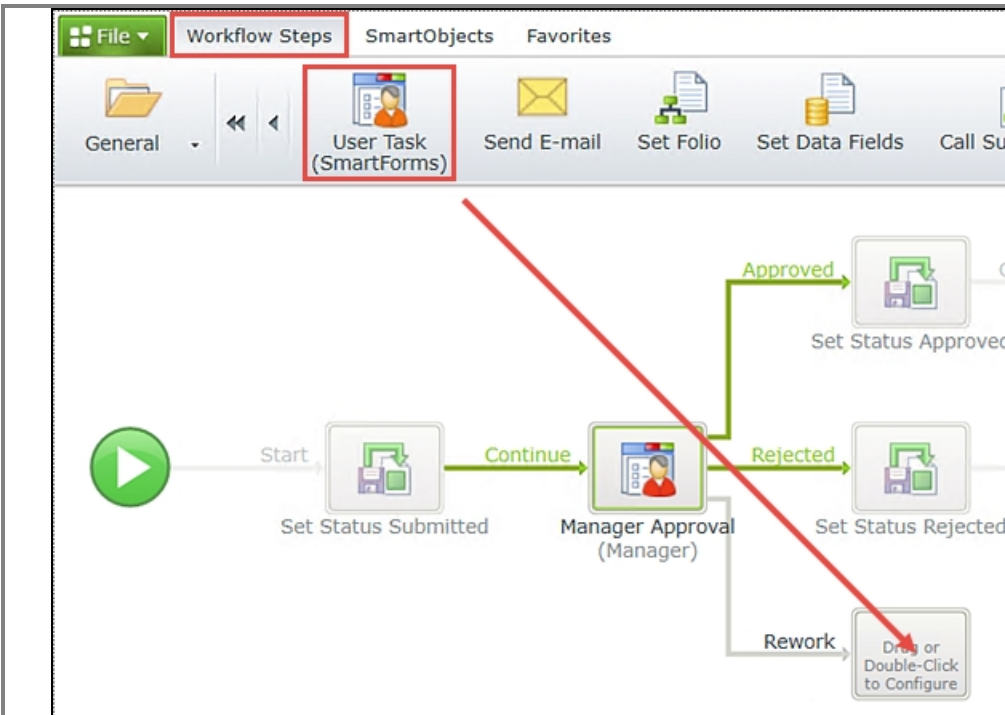
Leave Type: [Leave Type]

(Click **Finish** when ready.)

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.

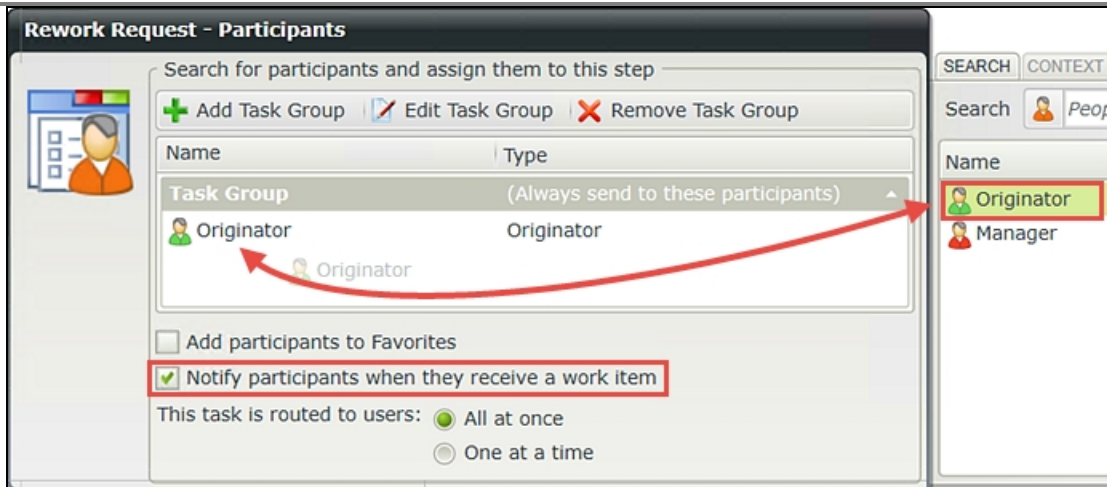


- e. The User Task wizard opens. Name the new event *Rework Request* and enter the following for the Instruction:

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

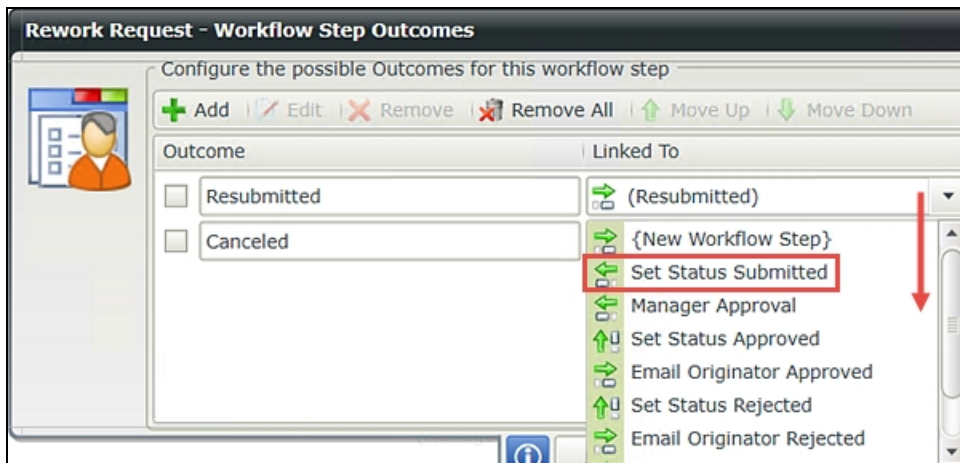
For the actions, enter *Resubmitted*
Canceled
then click **Next**.

- 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**.

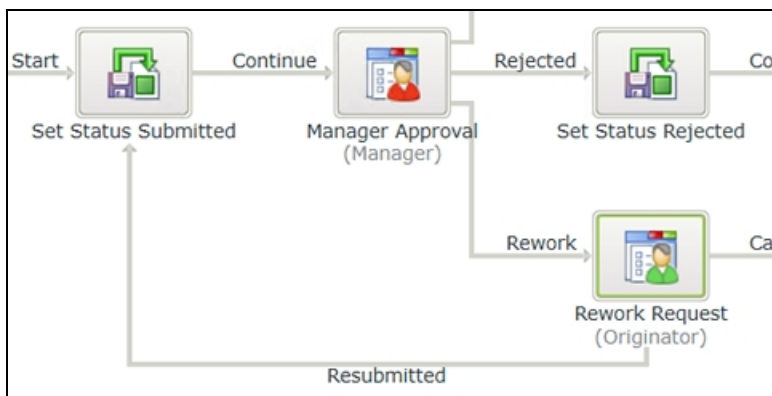


Now you want to configure the events for the two rework outcomes (**Resubmitted** and **Canceled**). First, you will route the resubmitted outcome back to the **Set Status Submitted** event so that the workflow returns to a previous step in the workflow. For the canceled outcome, you will copy and paste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



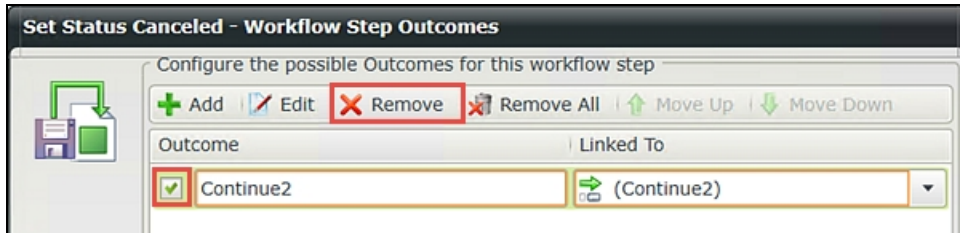
Notice now that K2 routes the **Resubmitted** outcome back to the **Set Status Submitted** event. This will create a loop effect in the workflow between the rework and resubmitted outcomes until a different action is selected.



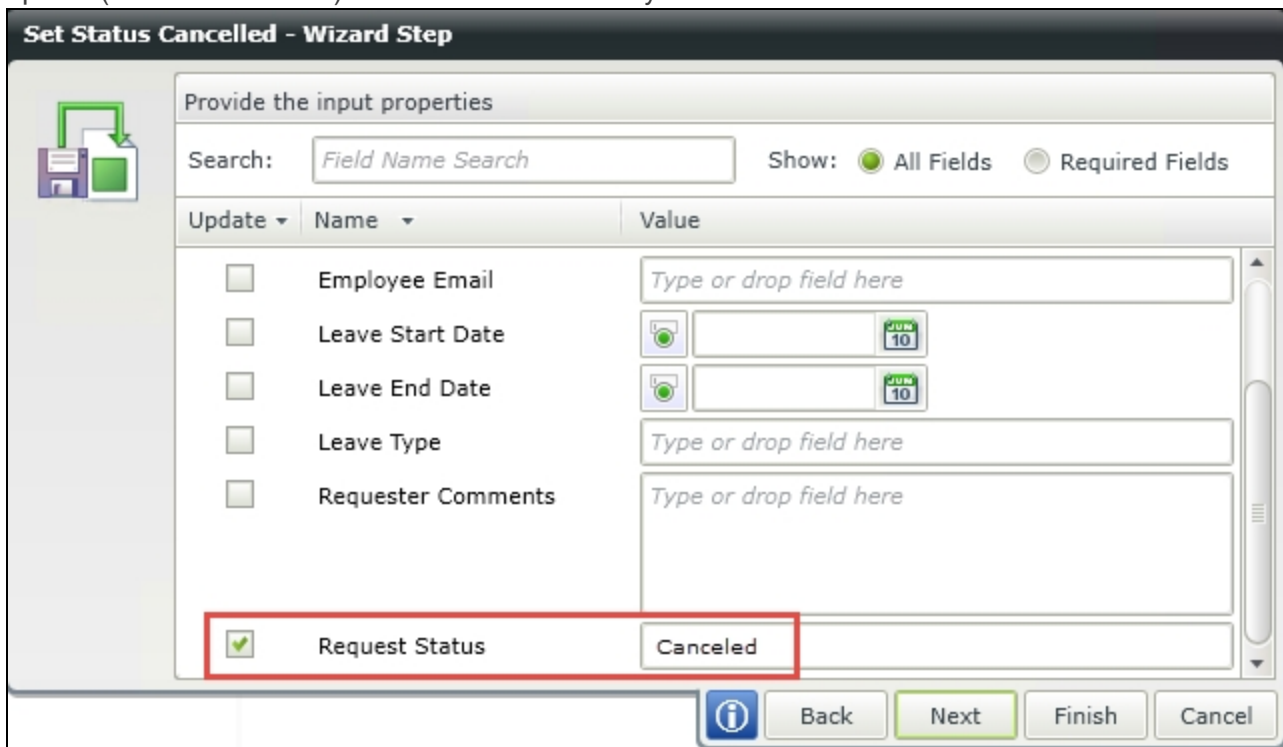
- i. Copy and paste the **Set Status Submitted** event into the Canceled event. (See Step 5 (j) if you need a reminder on how to copy and paste events in K2 Designer.)

- 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**.

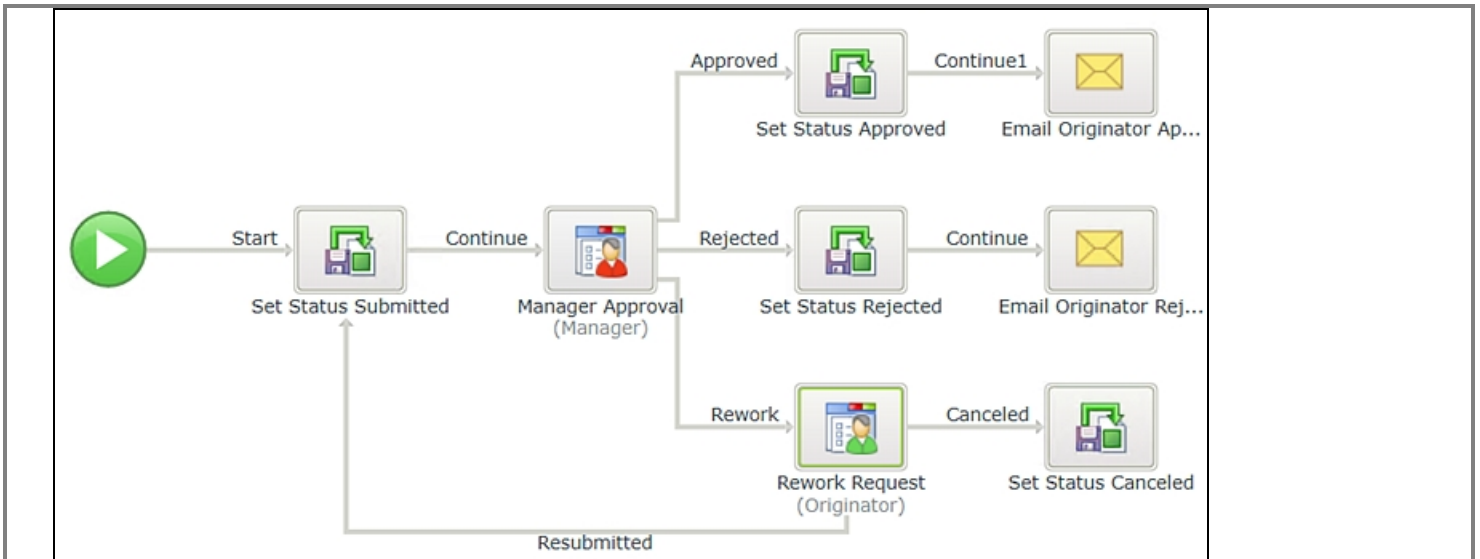
You are removing this outcome because you don't need to continue the workflow after this step. If the decision is to cancel the request, then after the request status has been updated, the workflow will be complete.



- 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.



- 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).

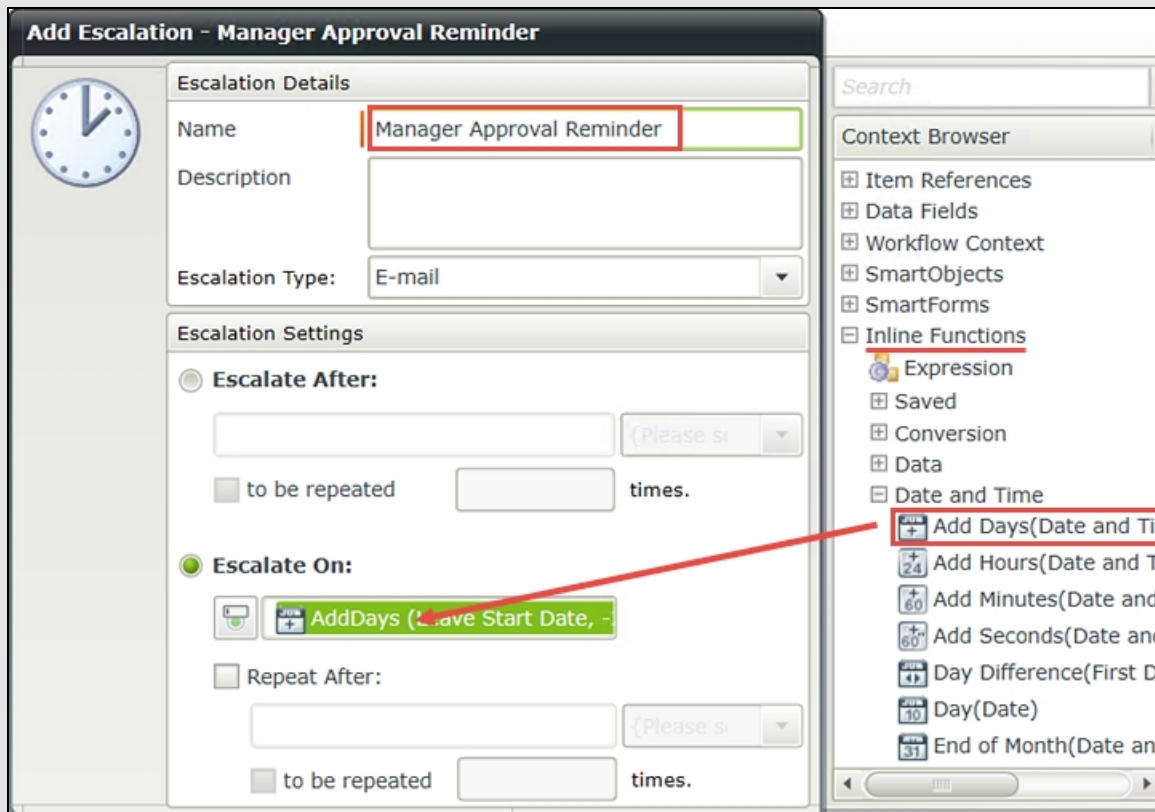
Caution

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

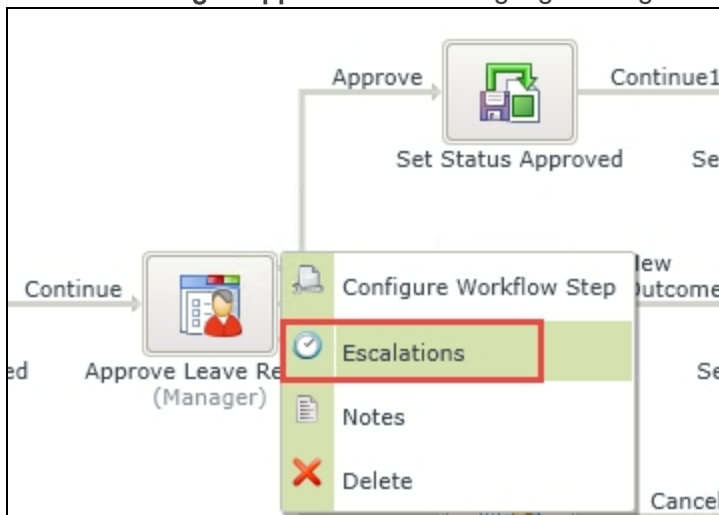
1. **Add** a new escalation to the manager approval event and name it *Manager Approval Reminder* then keep the default **E-mail** escalation type.
2. Select the **Escalate On** option then using the Inline Functions to subtract two days from the Leave Start Date value.



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

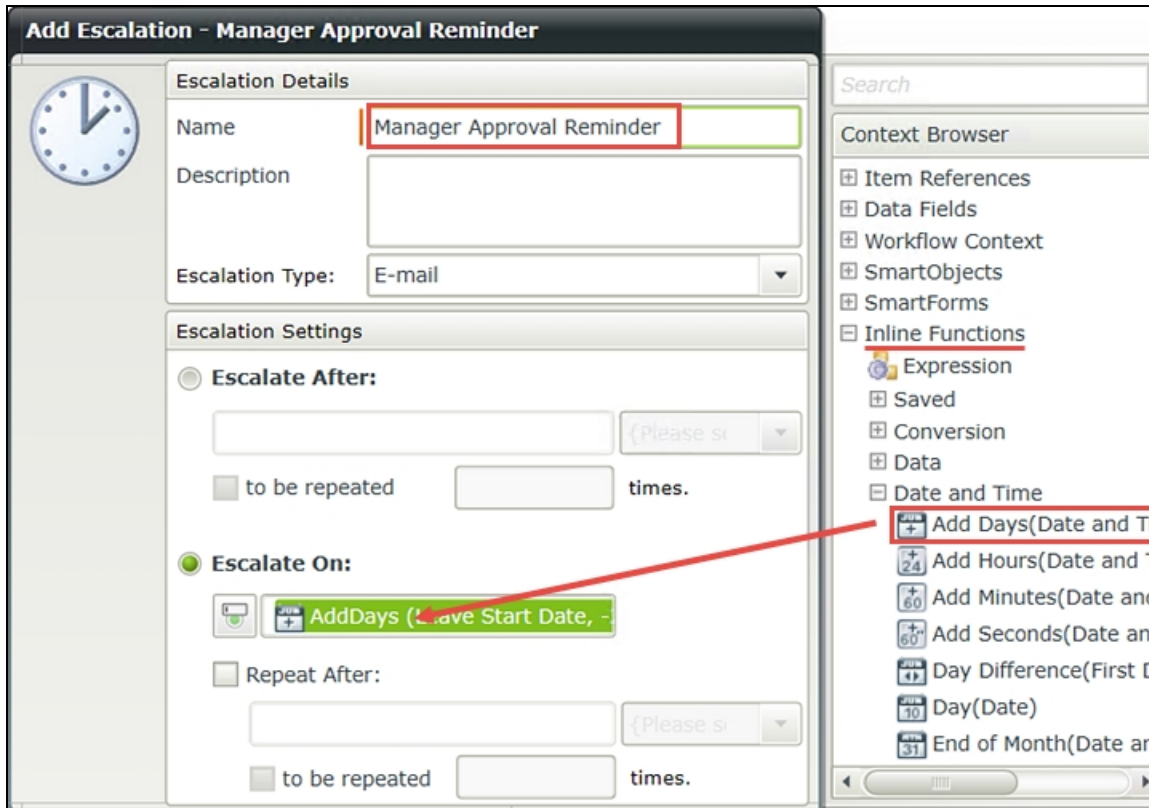
- a. Click the **Manager Approval** event to highlight it. Right-click and select **Escalations**.



- 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.

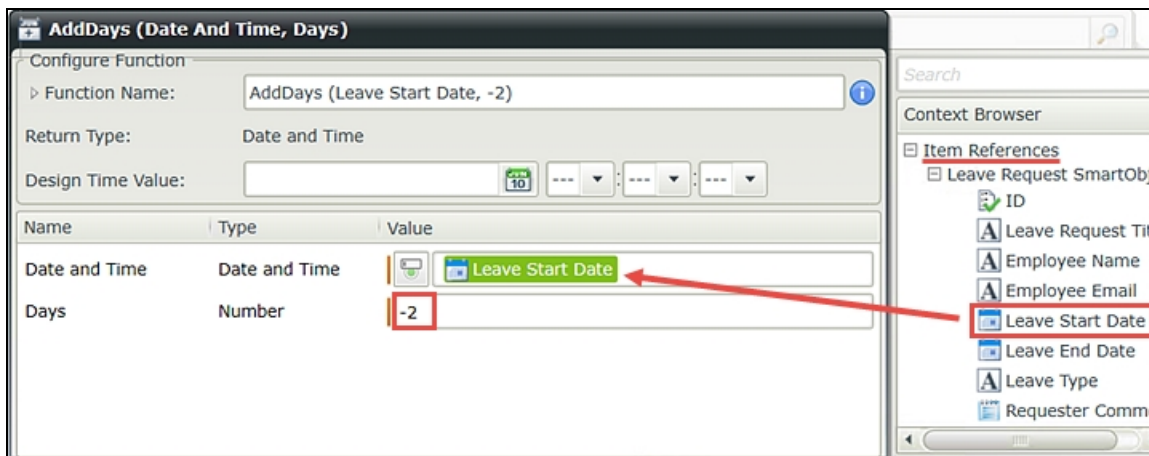
Note

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.



- 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.



- 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:

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

Subject: REMINDER: Leave Request for **Originator Full Name**

Message Format: HTML Plain Text

Verdana 12 B I U A

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

+ Add Edit Remove Remove All

Name	Rule	Action
<input type="checkbox"/> 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

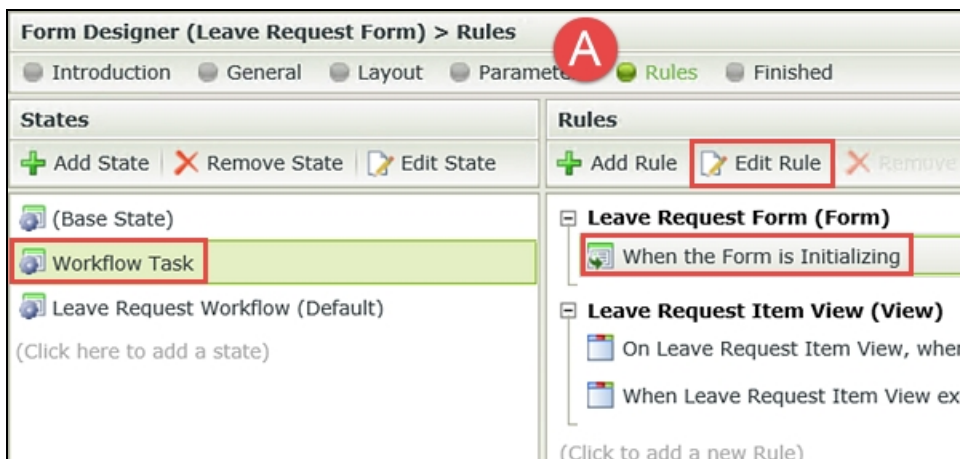
A *state* represents the configuration and behavior of a form at a given point in the workflow. States are most commonly configured at the user task event level. For example, you might want the form configuration to contain certain fields for the manager approval task, then contain different fields for 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.)



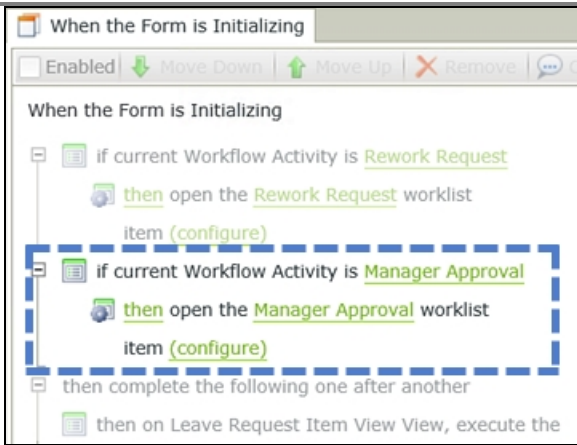
Note

C. Rules: Events, Conditions and Actions

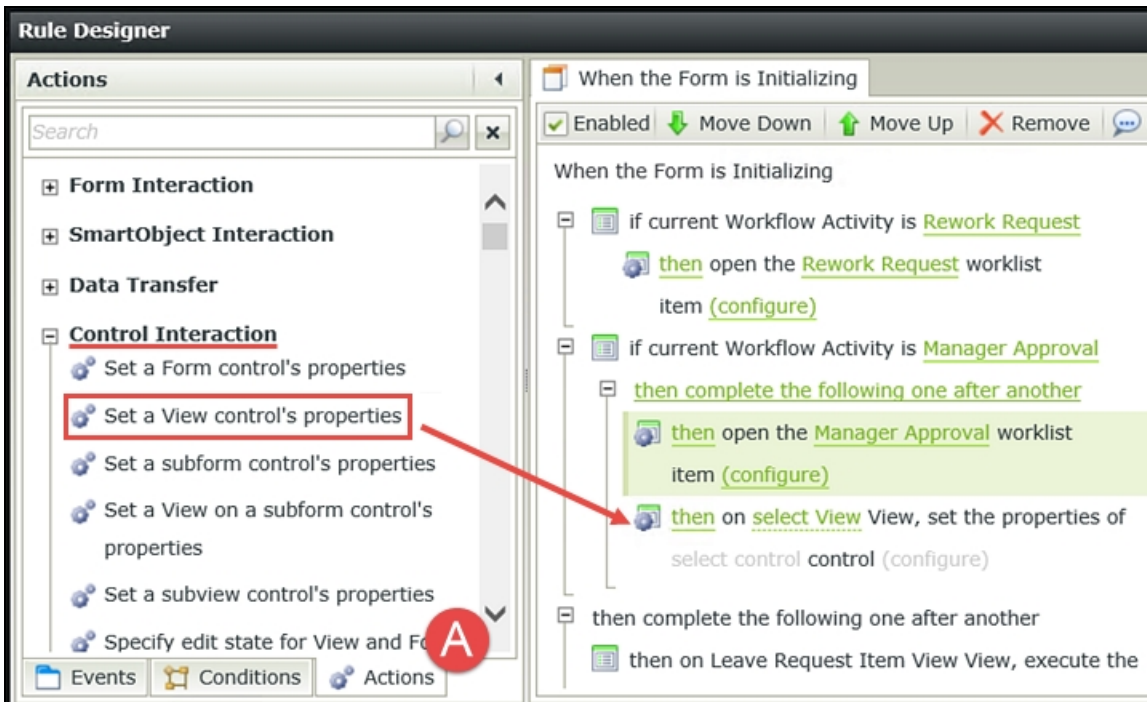
As a reminder, rules consist of events, conditions and actions.

- **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.



- 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.



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



- 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 set-

ting, the manager will be able to add comments when they open the form. Click **OK**. Click **OK** to return to the Rules home screen.

The screenshot shows the 'Controls' configuration window for 'Approver Comments Text Area'. The window is organized into several sections: 'Detail', 'Settings', and 'General'. The 'Read-Only' checkbox is checked and highlighted with a red box. Other visible options include 'Name', 'Text', 'Field', 'Data Type', 'Display Time Zone', 'Watermark', 'Tooltip', 'Rows', 'Width', 'Tab Index', 'Visible', 'Enabled', and 'Expression'. The 'Read-Only' option has radio buttons for 'Yes' and 'No', with 'No' selected.

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.

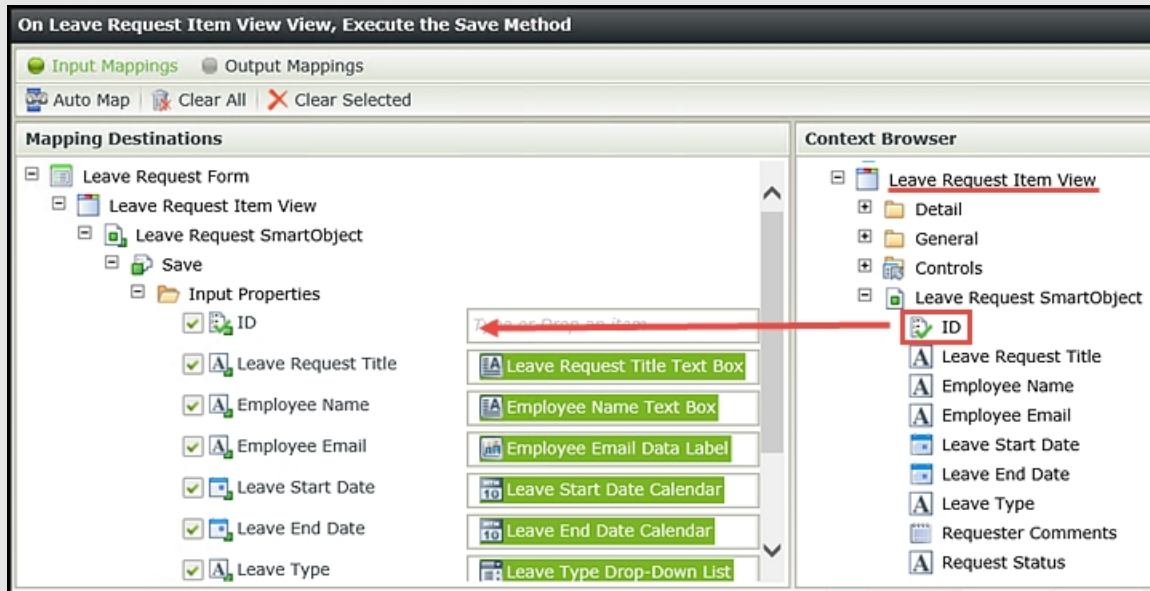
Step 10: Add a save method to update the SmartObject with changes made by the manager

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

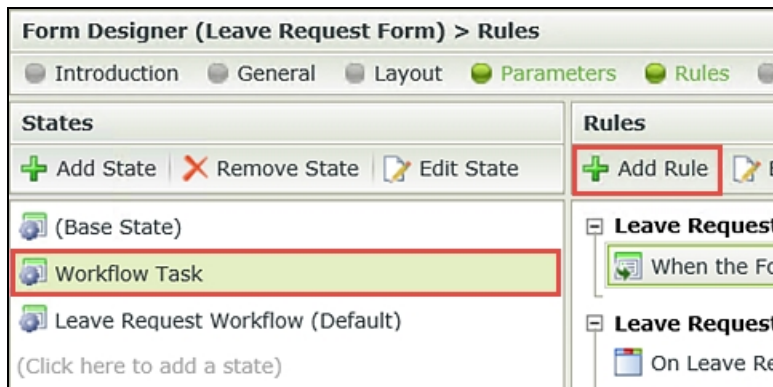
1. Add a new rule to the Workflow Task state. Add the event **After the Workflow action was submitted**.
2. Add a **Execute a View method** action to call the **Save** method of the Leave Request item view.
3. You may **Auto-Map** the Input Properties, but you **MUST** delete the auto-mapped **ID** value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

referencing the current record.

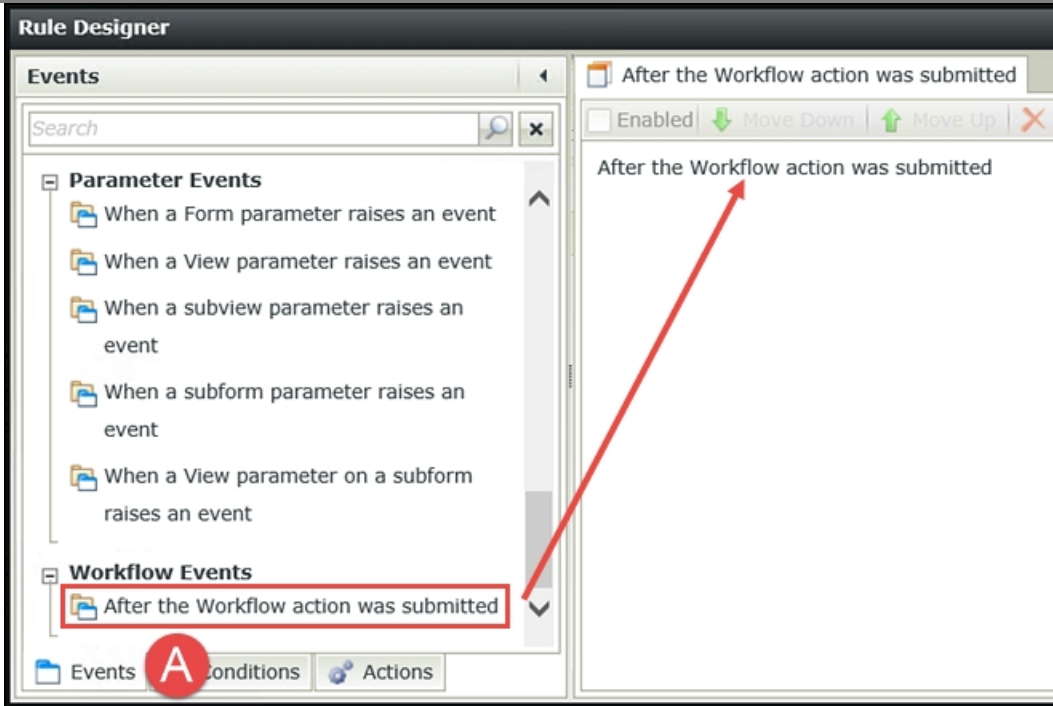


Step 10 Walkthrough

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



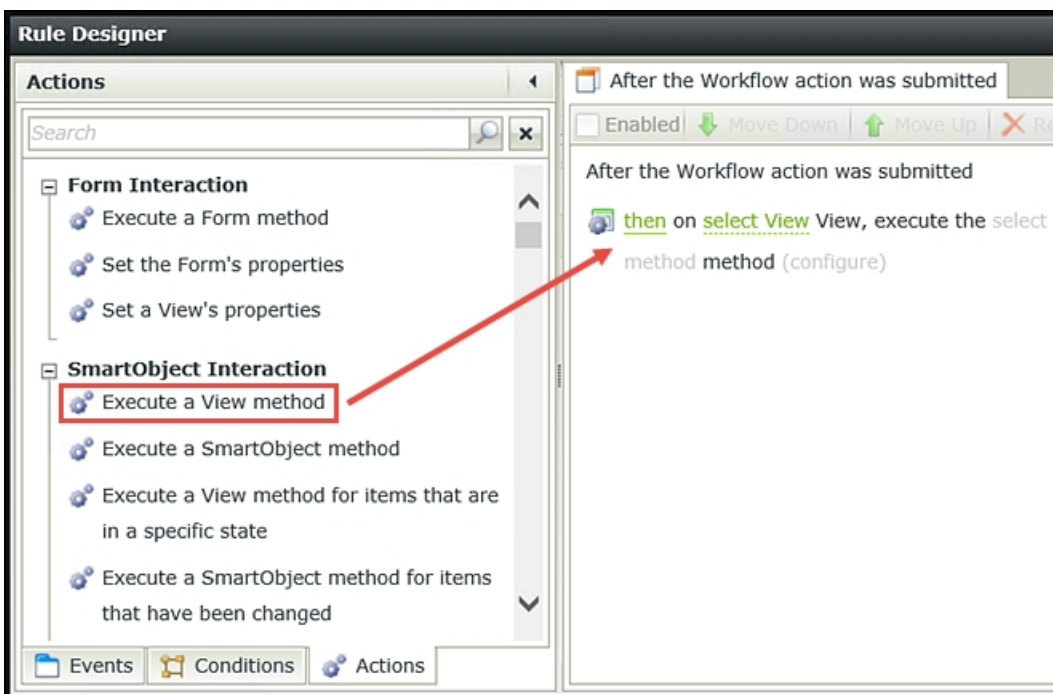
- Click the **Events** tab. (A below) Scroll down until you see the Workflow Events heading. Locate and click the **After the Workflow action was submitted** event to add it to the Rule Definition pane.



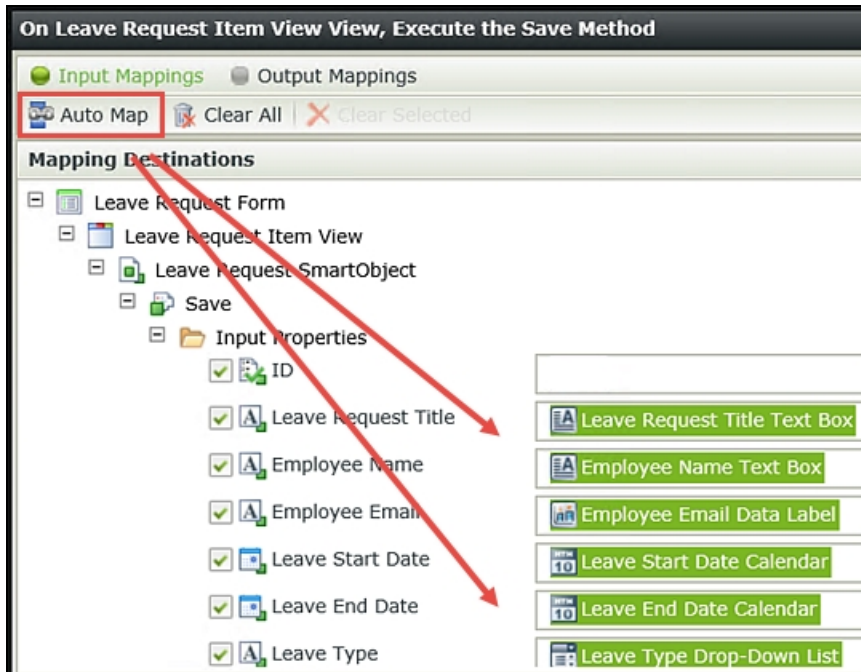
- c. Now click the **Actions** tab and click to add the **Execute a View method** action to the Rule Definition pane. Click the **select View** link and select **Leave Request Item View**. Click the **select method** link and select **Save**.

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.



- 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

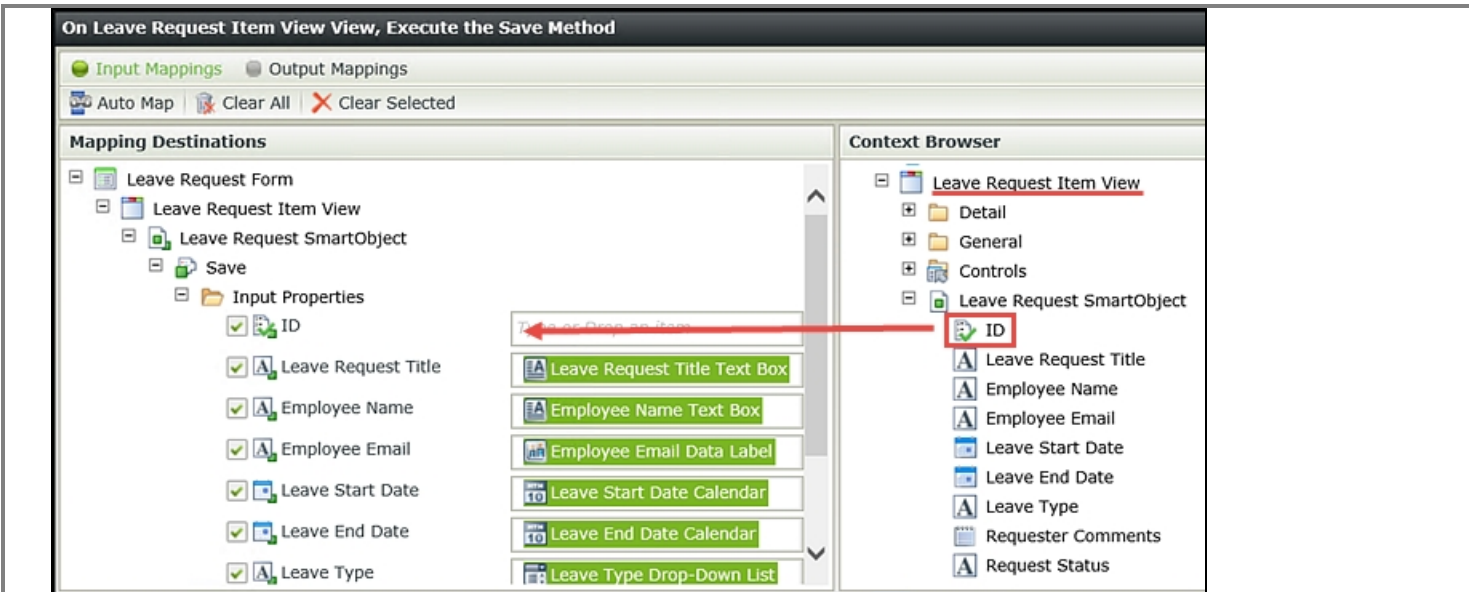
When you auto-map the input properties, K2 creates a new record in the SmartObject and saves the form field entries to their corresponding SmartObject properties. While we want to save the form field entries, we don't necessarily want to create a new SmartObject record. Why is this? When you started 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.



STEP 10 REVIEW

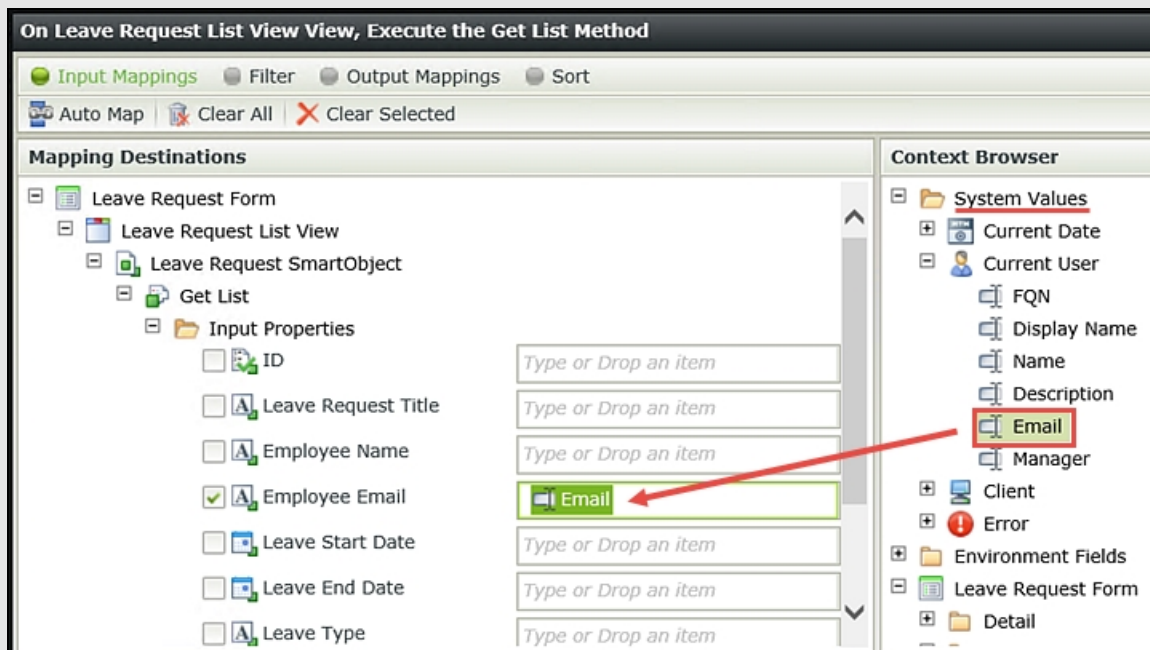
In this step, you added a rule that will fire off the save method (which is the equivalent of update) for the Leave Request SmartObject with any changes the approving manager might have made.

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

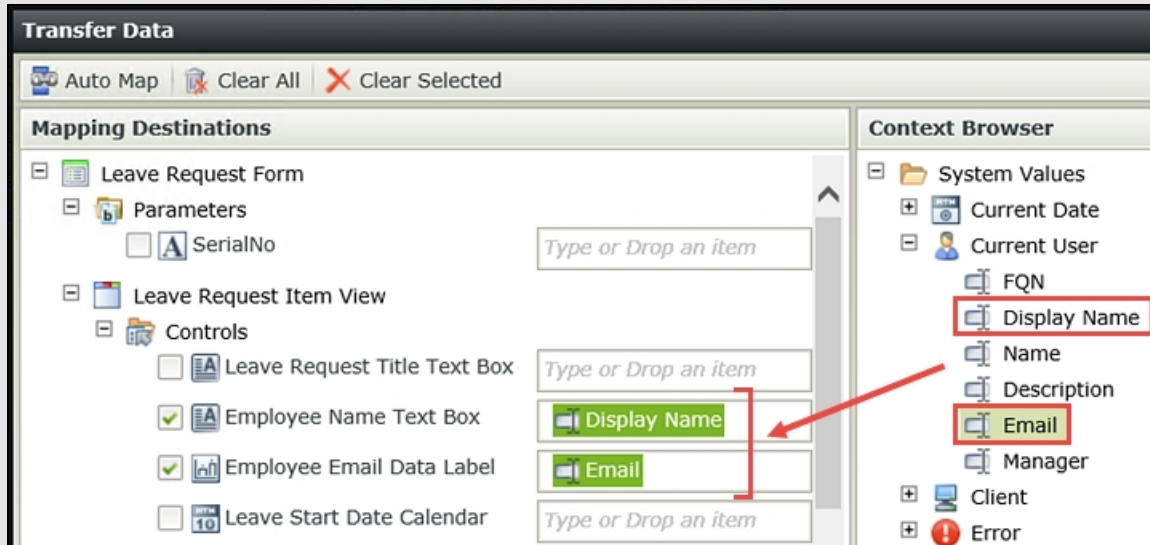
The last few edits you are going to make to the rules will be to clear the form entries after the form originator has clicked the Create button, then add the new entry to the list view. (The list view contains all of previous leave requests.) Recall in the basic tutorial, when the form originator submitted 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.



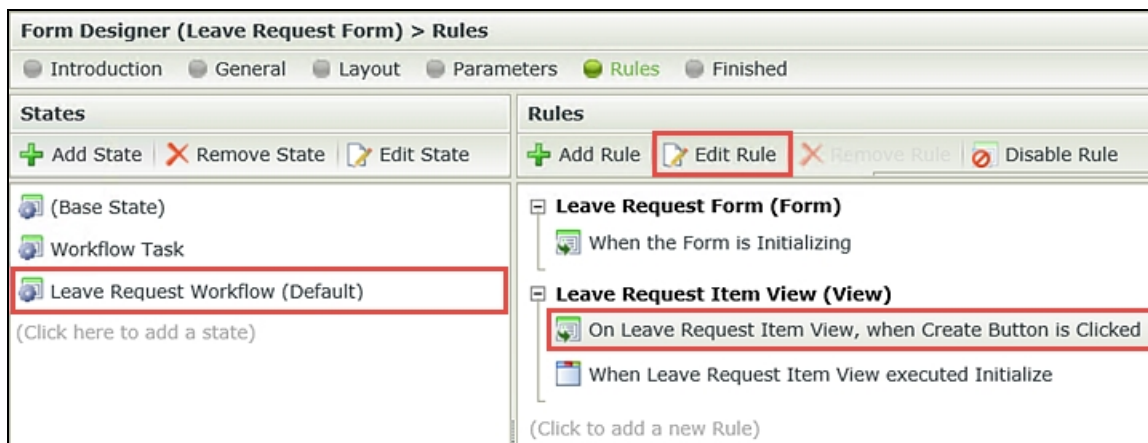
3. Add a **Transfer data** action and map the Current User Display Name and Email to their respective form controls.



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

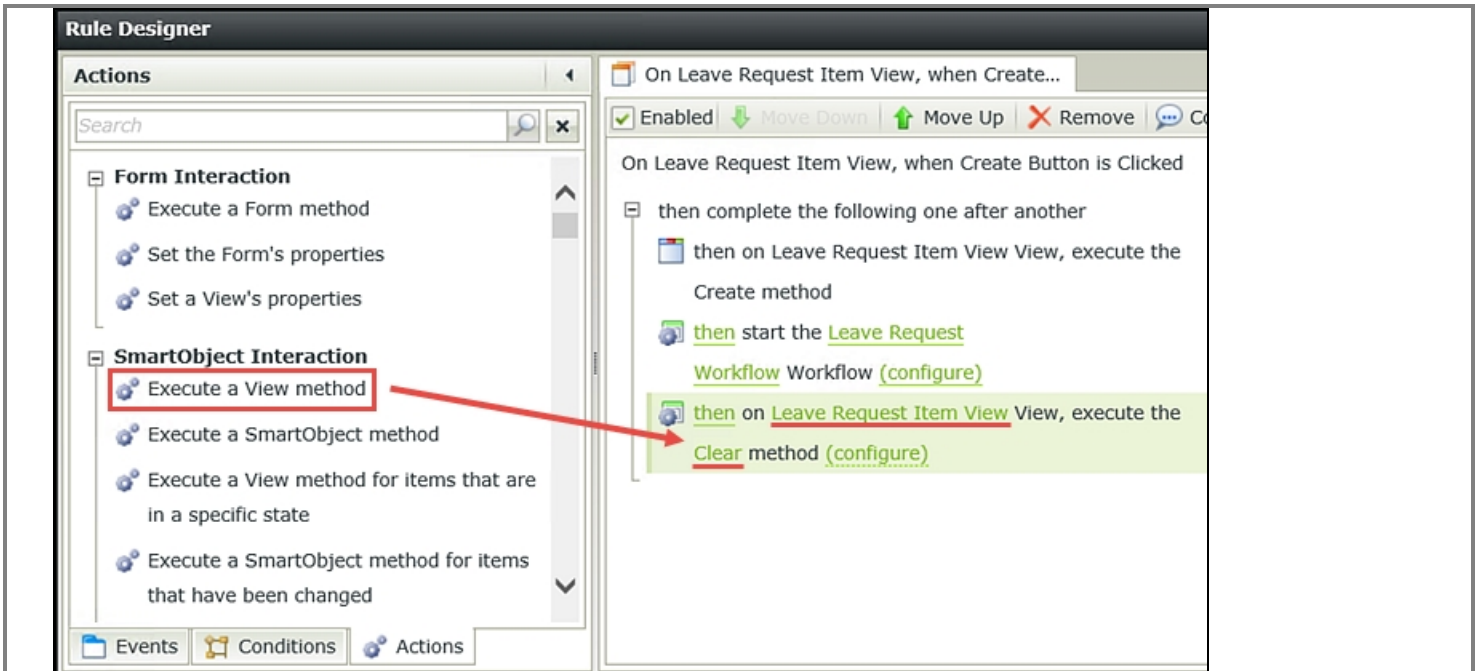
Step 11 Walkthrough

- 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**.



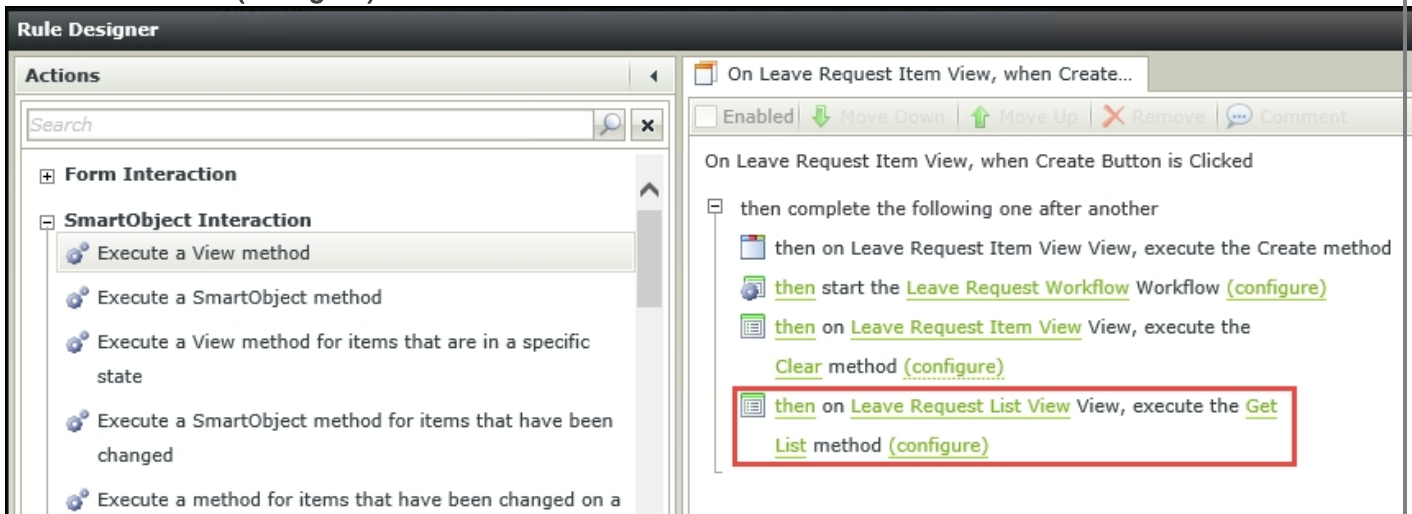
- b. Confirm the **Actions** tab is highlighted, then click the **Execute a View method** action to add it to the Rule Definition pane. Click the **select View** link and select **Leave Request Item View**. Click the **select method** link and select **Clear**.

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.

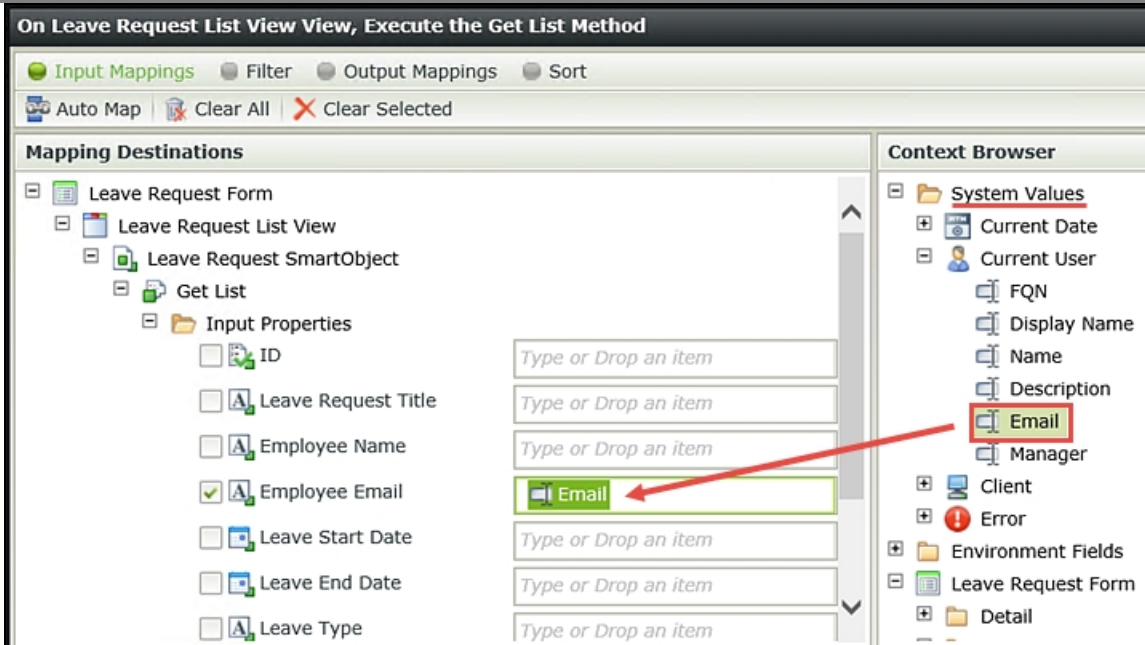


Now you will update the Leave Request list view so that the new entry is displayed.

- 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.

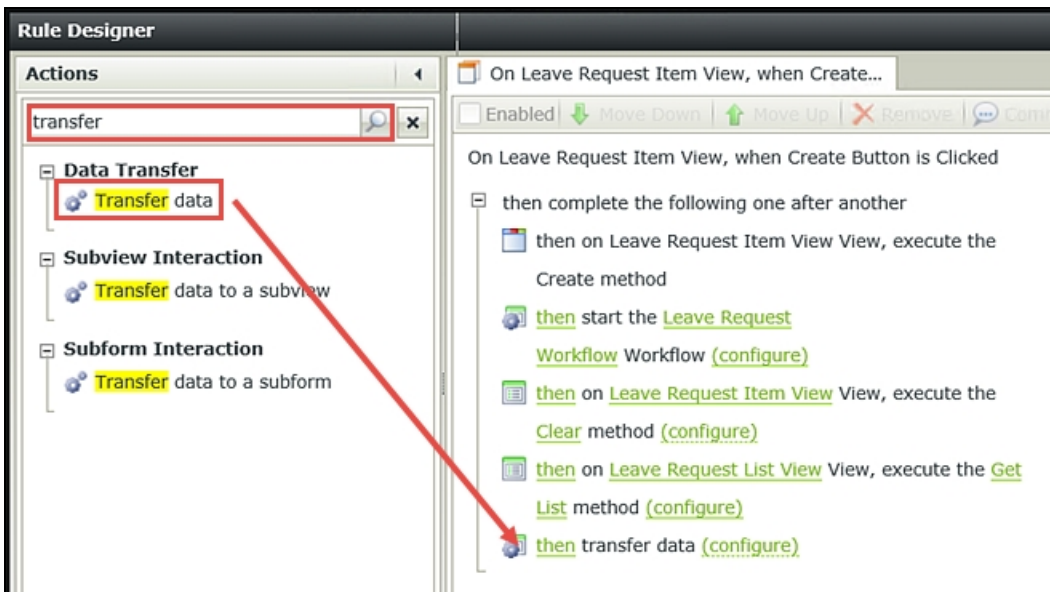


- d. In this step, you will map the current user's email as the input property so that K2 knows to search the SmartBox records for those that have the same email value as the current user. In the context browser, expand **System Values**, then **Current User** and drag the **Email** property into the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

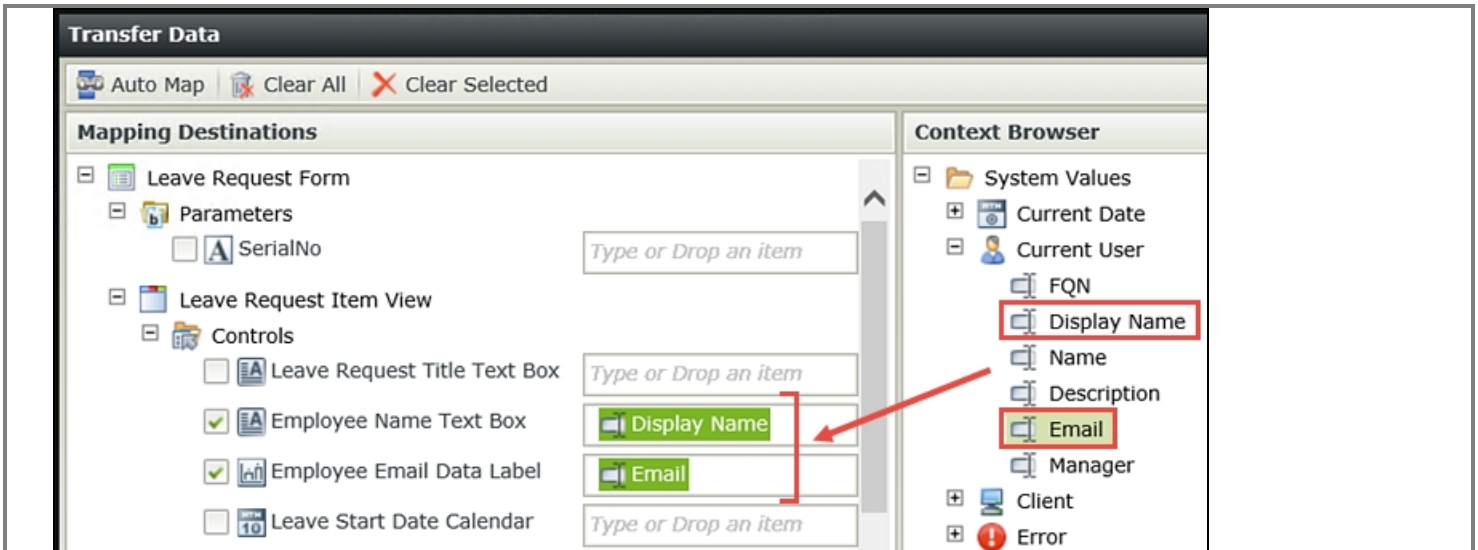


You have one additional step you need before you close out of your form. In the previous step, you cleared the form entries when the user clicked the Create button. Now you need to add back in the Employee Name and Employee Email values so the refreshed form appears exactly the same as a brand new form. 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.



- 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

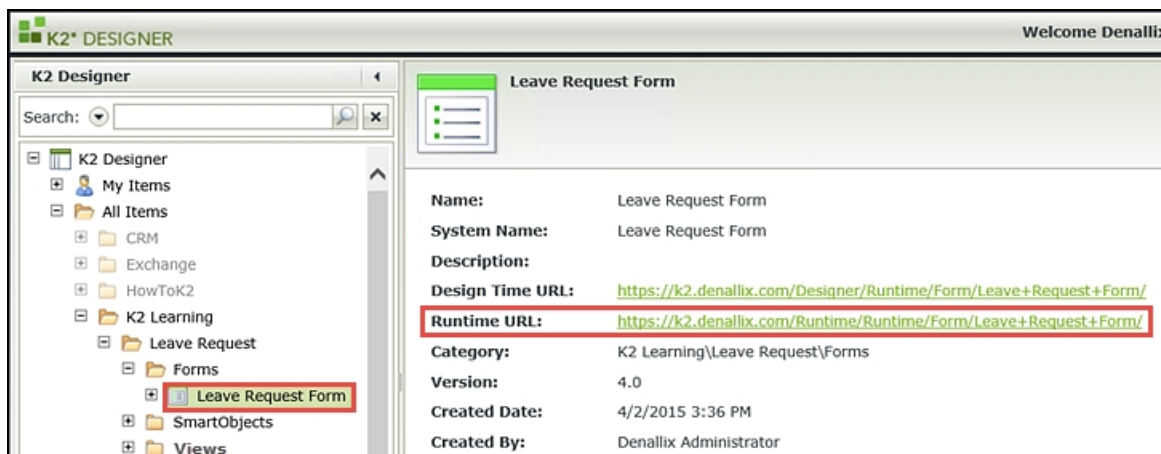
1. Using the Runtime URL, submit two Leave Request forms. Make the **Leave Start Date** for one of the requests today's date. This will fire off the escalation since you have it configured to start within two days of the Leave Start Date.
2. Access the approving manager's Outlook and using **SmartActions**, 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.

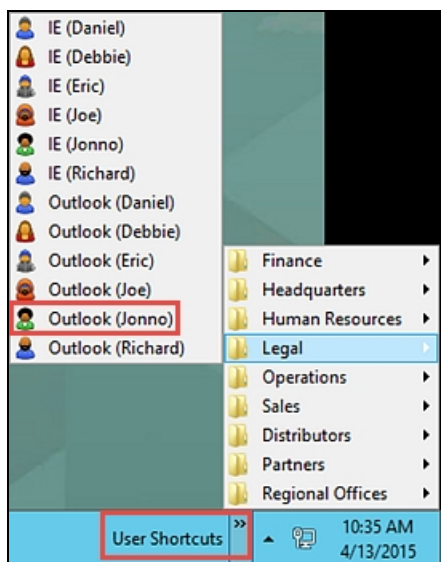
If you have time and would like to submit more leave requests so that you have additional content when you work through the Reports tutorial, feel free to do so! (The Reports tutorial uses the workflow process instances from the Leave Request tutorials for content.)



After you have submitted a form, confirm that the field entries are cleared (except the Employee 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.

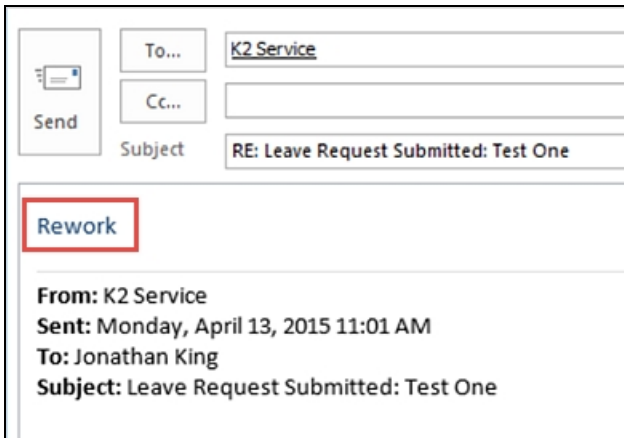
At this point, you need to open the approving manager's Outlook. If you are building this application on a K2-provided VM, you are most likely logged in as Denallix Administrator. Administrator's manager is Jonno, so you'll open Jonno's Outlook. If you are building this application within your own environment, 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.

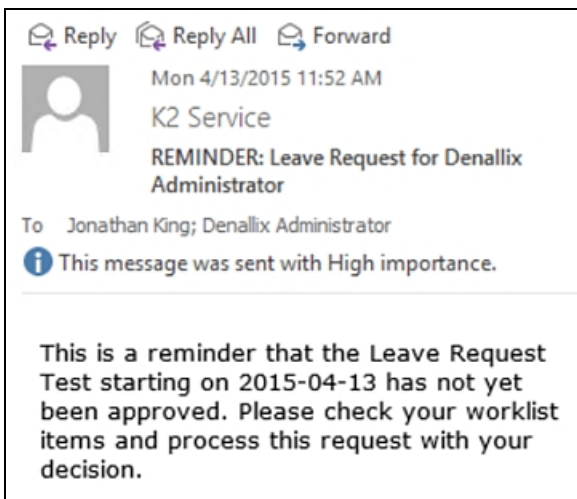


- c. Confirm there are two task notification emails for Jonno. Open one of the emails and reply with *Rework* as the email message body. Send the email.

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.

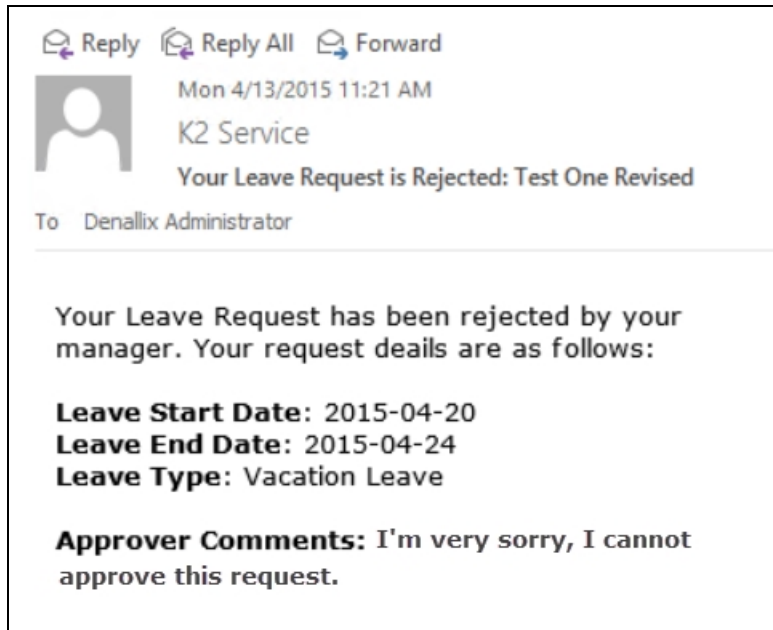


- d. Confirm there is an escalation email for the task with today's date as the Leave Start Date.



- 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.



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

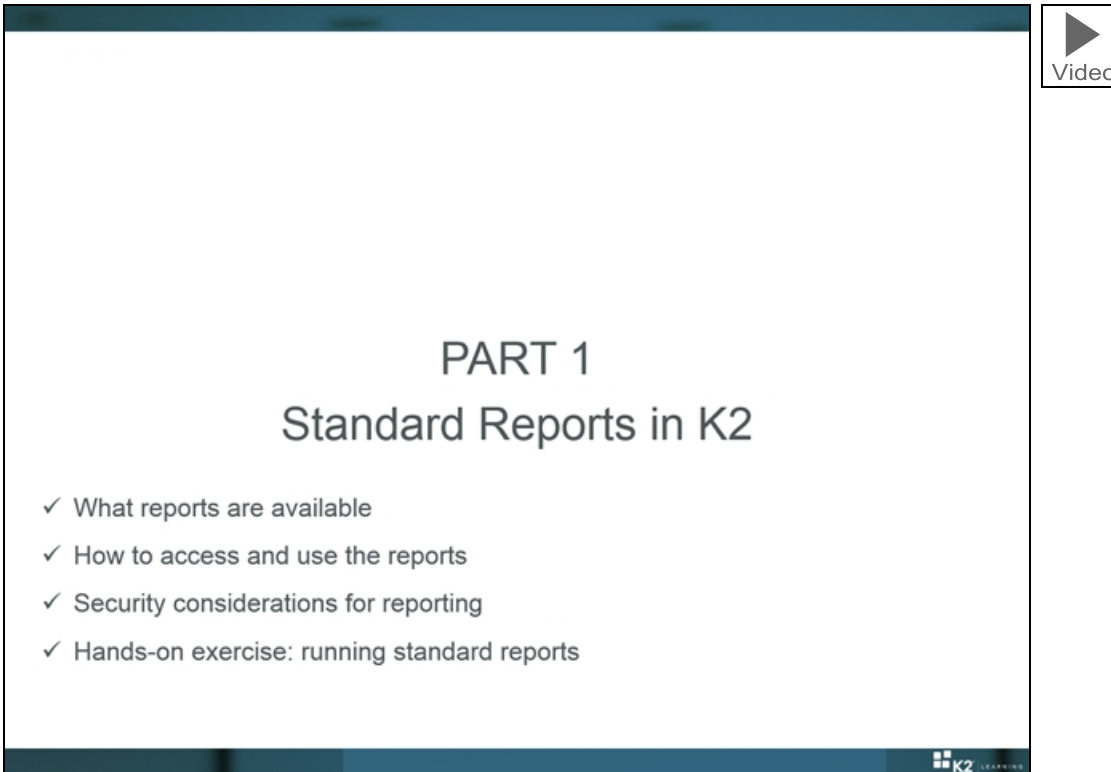


The *100.CWL: Reporting in K2* training module explains how to use the available standard and custom reporting in K2 to report on workflows. The module should take around 1 to 1.5 hours to complete, depending on how many of the hands-on exercises you choose to do.

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



▶
Video

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

 K2 LEARNING


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



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

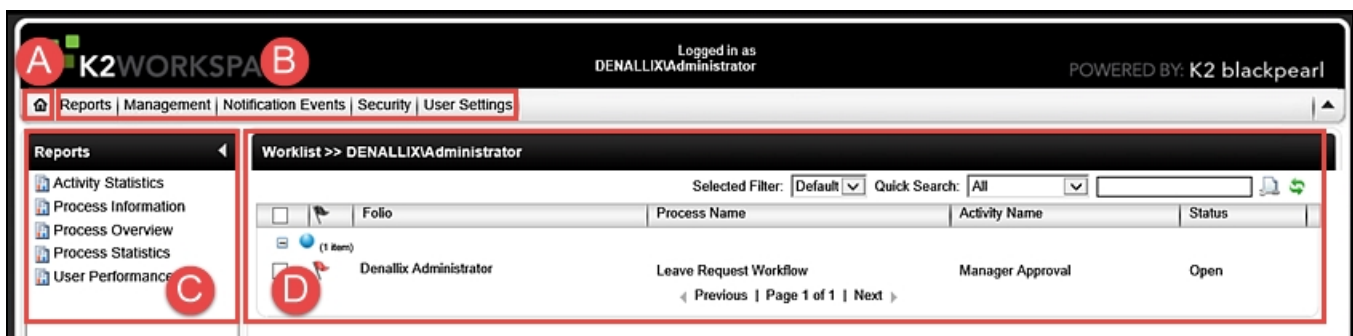
In this step, we will take a brief tour of the Management Console in K2 Workspace. We will look at the Process Rights for the Leave Request Workflow.

Step 1 Tasks

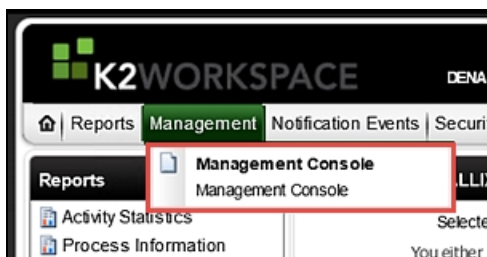
1. In this step, we are taking a brief tour of the Management Console, specifically looking at the **Process Rights** for the **Leave Request Workflow** Process. If you are familiar with the Management Console and Process Rights, feel free to move on to the next step.

Step 1 Walkthrough

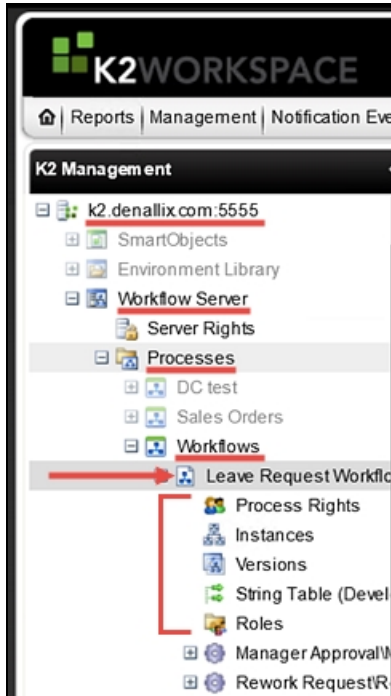
- a. Launch K2 Workspace. (**Start > All Programs > K2 blackpearl > K2 blackpearl Workspace**). If you do not see this link in your start menu, contact your K2 Administrator to obtain the URL of the K2 Workspace in your environment.
- b. Notice the following sections of the landing page:
 - A. Home Page 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.)



- 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

+ Add | Save

Selected Filter: (None) Quick Search: All

User/Group	Admin	Start	View	View Participate
K2.DENALLIXadministrator	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
K2.DENALLIXDomain Users	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

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.

Parameters Filters Settings

Process Version: 1

Date Range: Specific Date Range

From: 2015/04/04 **To:** 2015/05/06

Options: Exclude Server Events

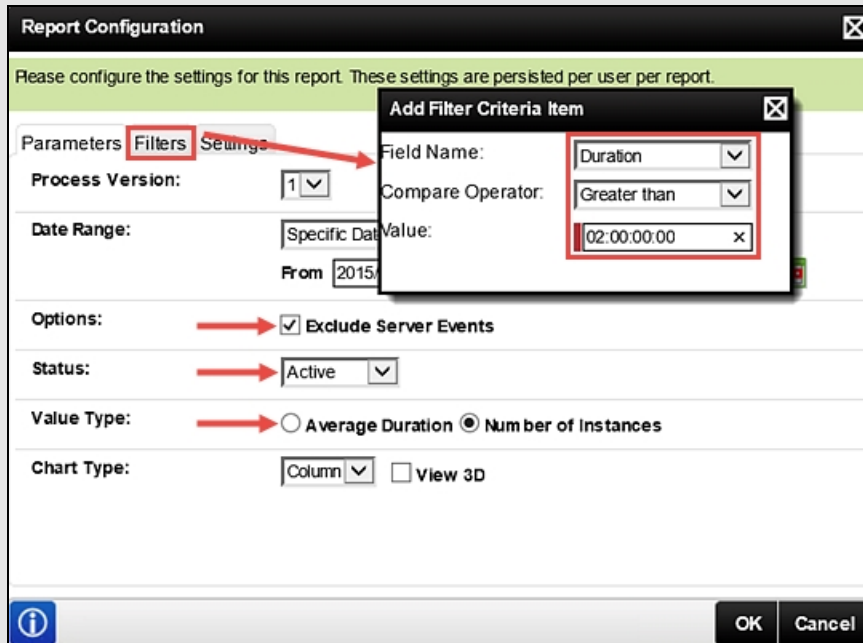
Status: Completed

Value Type: Average Duration Number of Instances

Chart Type: Column View 3D

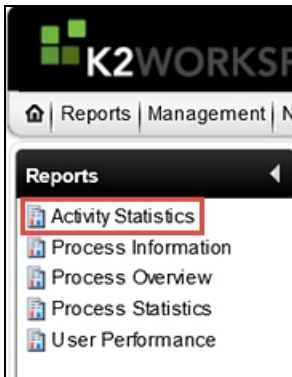
2. Run a second Activity Statistics Report to determine if there are any tasks still running after two

days. This time, exclude the server events.



Step 2 Walkthrough

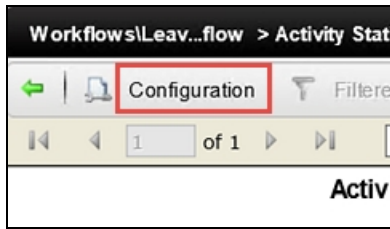
- a. From the K2 Workspace landing page, click the **Activity Statistics** Report link in the Reports pane. (It may take a minute or so for the report to open.)



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 pre-defined 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.

Parameters Filters Settings

Process Version: 1

Date Range: Specific Date Range

From: 2015/04/04 **To:** 2015/05/06

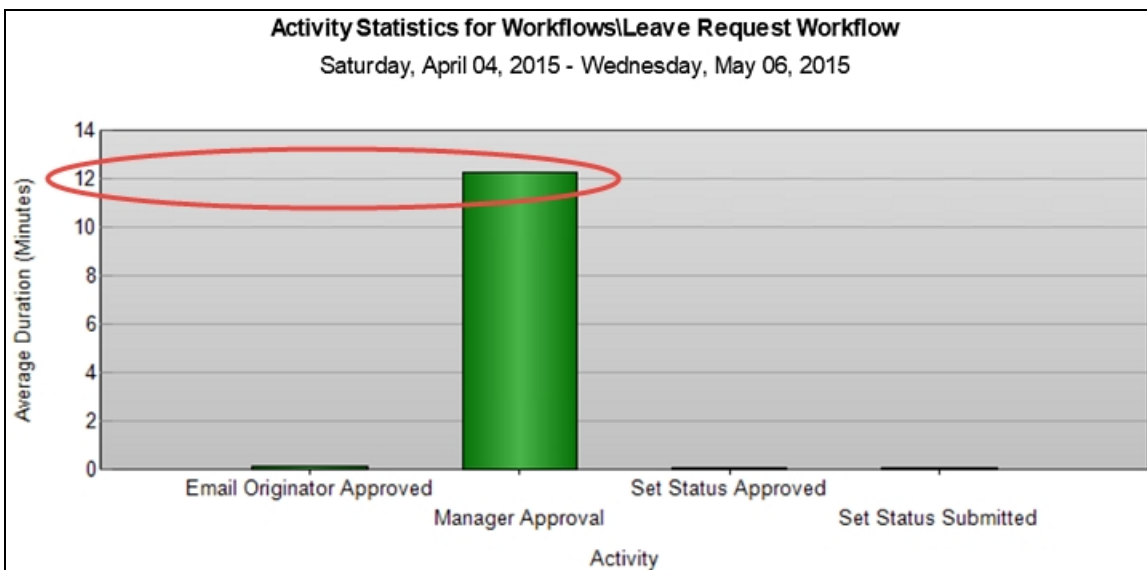
Options: Exclude Server Events

Status: Completed

Value Type: Average Duration Number of Instances

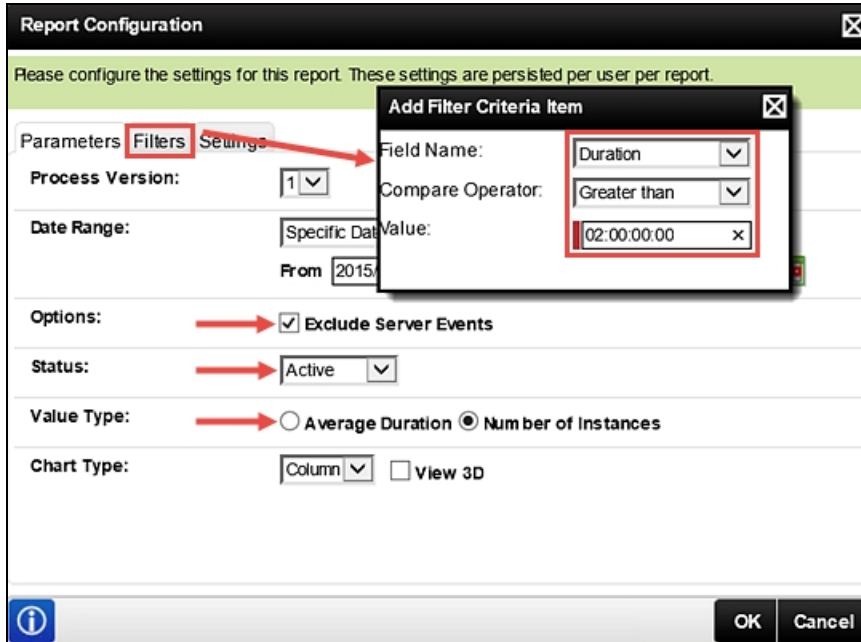
Chart Type: Column View 3D

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.

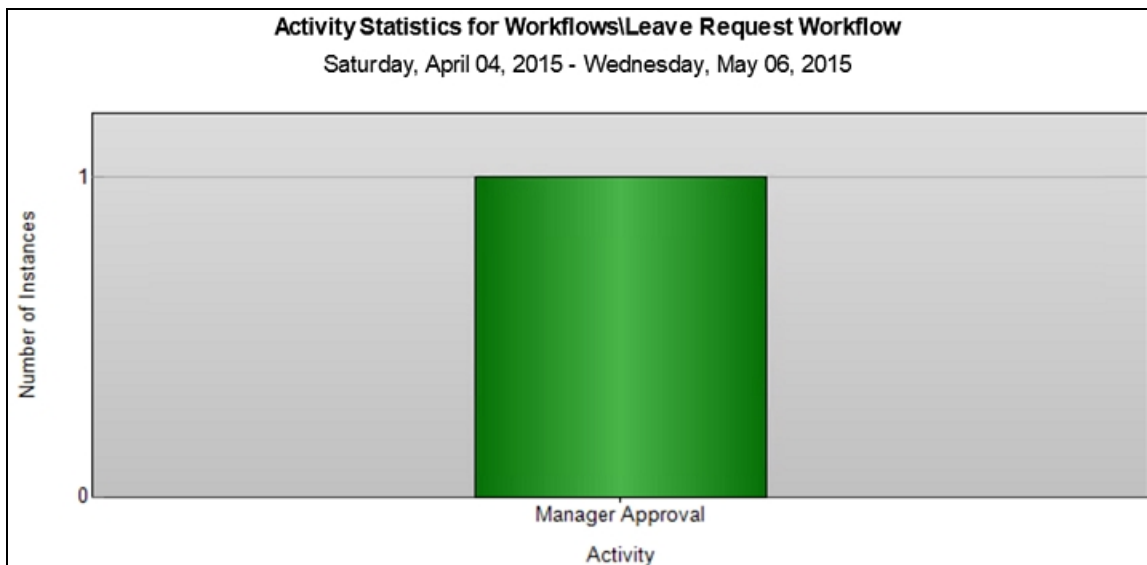


d. Now we want to see if there are any Process Instances still running after two days. Open the **Report Configuration** and make the following changes to the Parameter and Filter options:

- **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)



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

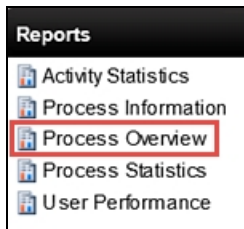
1. Open the **Process Overview** Report, then the **Leave Request Workflow** Process.
2. Click on any **Process Instance** that shows a **Completed** Status. Was this Leave Request approved or not approved? How can you tell this?
3. Access the **Audit** trail for the Manager Approval step. Check to see 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.

Process Name	Destination	Status	Priority	Start Date	Finish Date	Duration
Manager Approval	K2: DENALLIX\ADMINISTRATOR	Completed	Medium	5/6/2015 11:40:11 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 Name	Folder	Instances	Average Duration
Leave Request Workflow	Workflow s	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: B flows\Leave Request Wc C							
Process Folio A	Originator	Status	Priority	Start Date C	Finish Date	Duration	D
K2 smartforms Training View Flow icon	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 - Appl	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.

Data Slot Data XML Data Slot XML Data Data Audit Audit							
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.

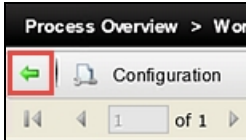
f. Click on the **Audit** link. Now we are looking at the fine details for the Manager Approval step. If there were any deviations from the normal path, they would show up here. And indeed, we can see that the Manager Approval task was redirected from Bob (Anthony's manager) to Denallix Administrator.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

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

Audit Description	User Name	Date
World 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.

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.

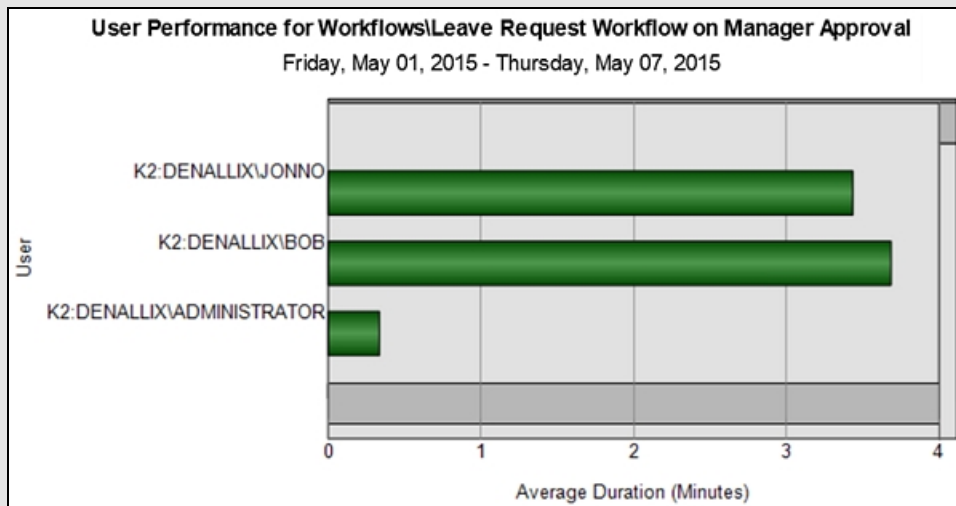
Step 4: The User Performance Report

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

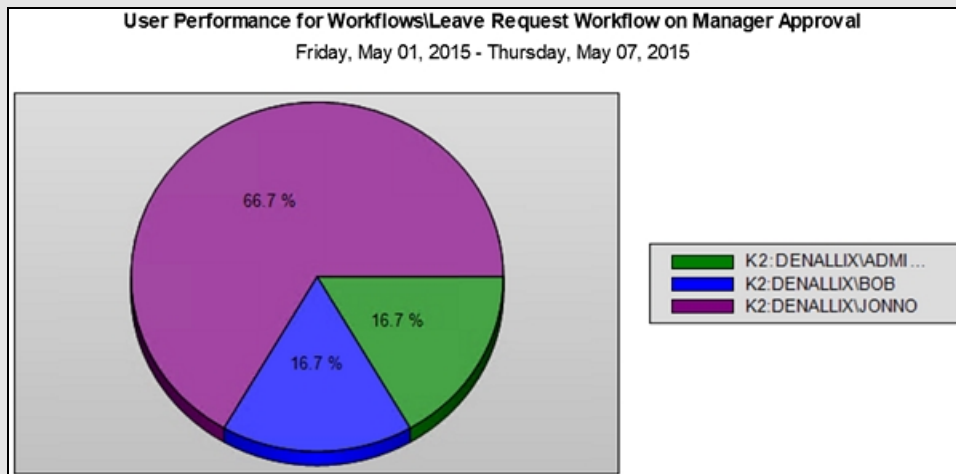
In this step, we want to generate two versions of the **User Performance** Report.

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.

- 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.

Parameters Filters Settings

Activity: Manager Approval

Date Range: Last 7 Days

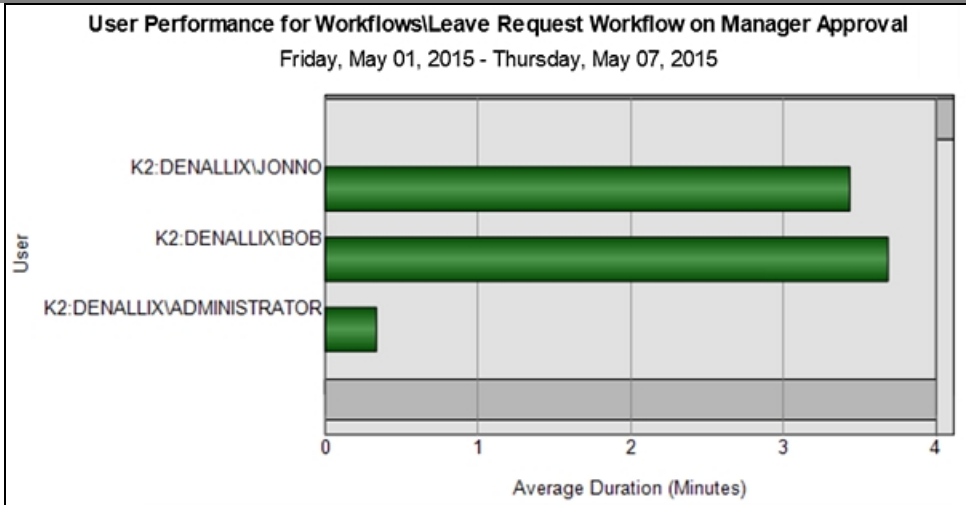
From 2015/05/07 To 2015/05/07

Value Type: Average Duration Number of Instances

Chart Type: 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

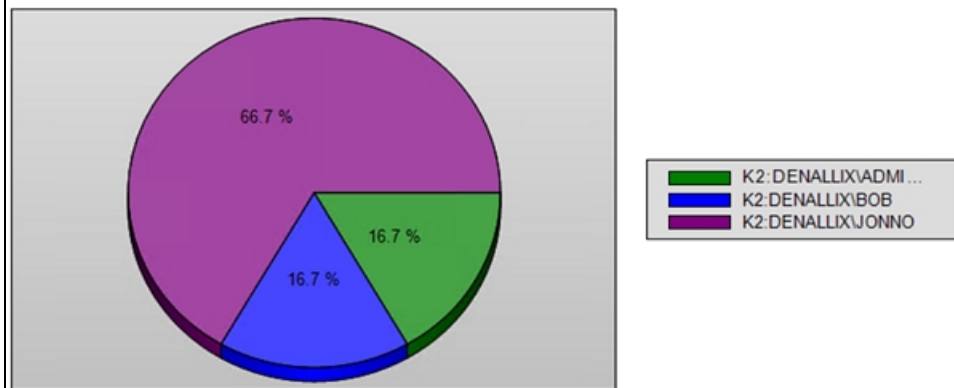
Chart Type: Pie View 3D

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?

User Performance for Workflows\Leave Request Workflow on Manager Approval

Friday, May 01, 2015 - Thursday, May 07, 2015



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.

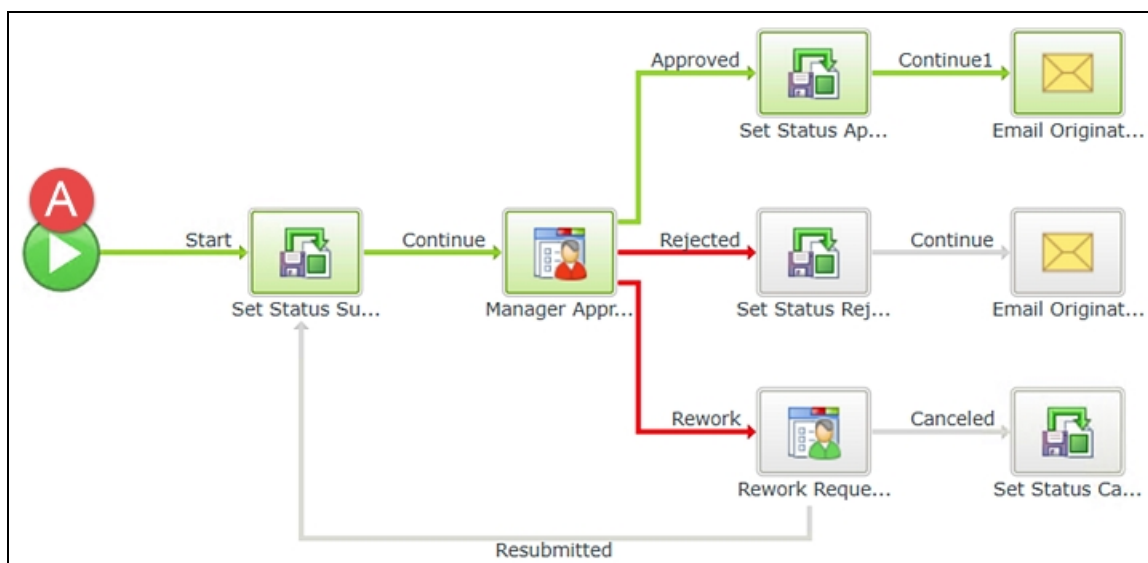
Step 5 Walkthrough

- 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.DENALLX\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.DENALLX\ADMINISTRATOR	Completed	Medium	5/4/2015 9:59:20 AM	5/4/2015 10:02:40 AM	00:00:03:21
On-line training - Appli	K2.DENALLX\BLAKE	Active	Medium	5/4/2015 10:04:46 AM		03:04:33:02
Travel Time for Corporate Meeting	K2.DENALLX\CODI	Completed	Medium	5/4/2015 10:05:46 AM	5/4/2015 10:52:10 AM	00:00:46:24
June Vacation	K2.DENALLX\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.DENALLX\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.DENALLX\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

Overview | Process Data

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)

Overview | Participants | Process Data

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)				
Overview	Participants	Process Data		
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.

Data Slot Data XML Data Slot XML Data Data Audit Audit						
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
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.

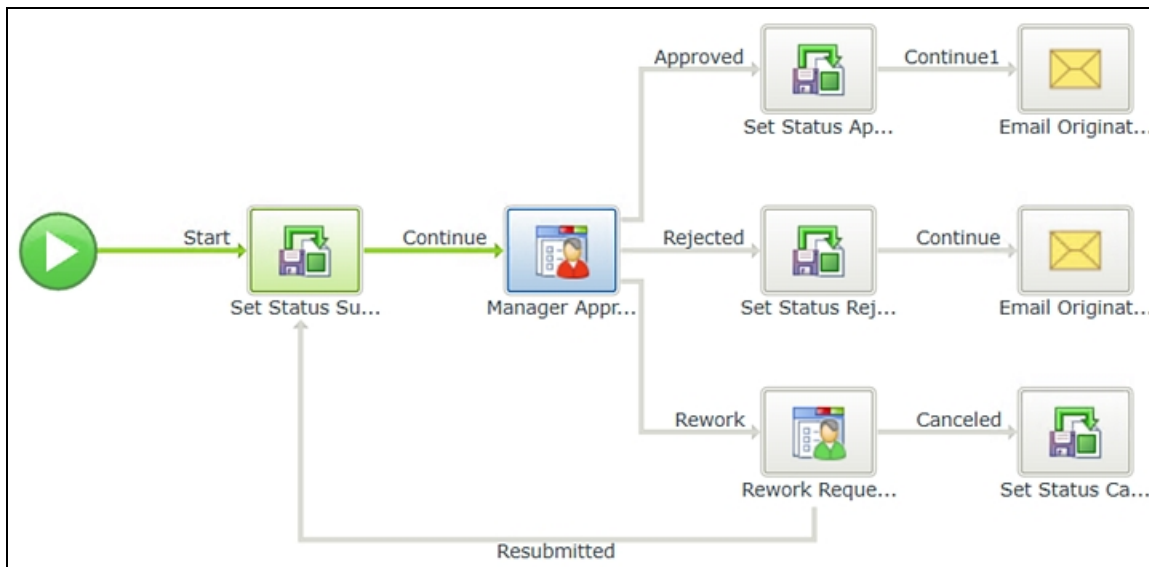
Step 6 Walkthrough

- 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

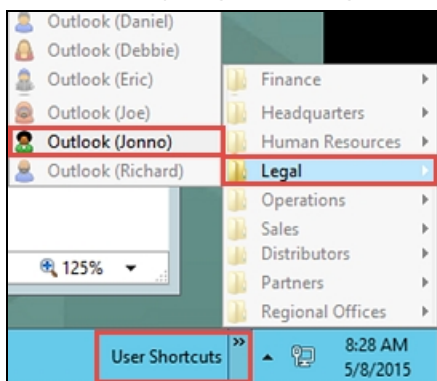
Instances that are active, you will need to submit a new Leave Request Form to create a new, active Process Instance.)

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

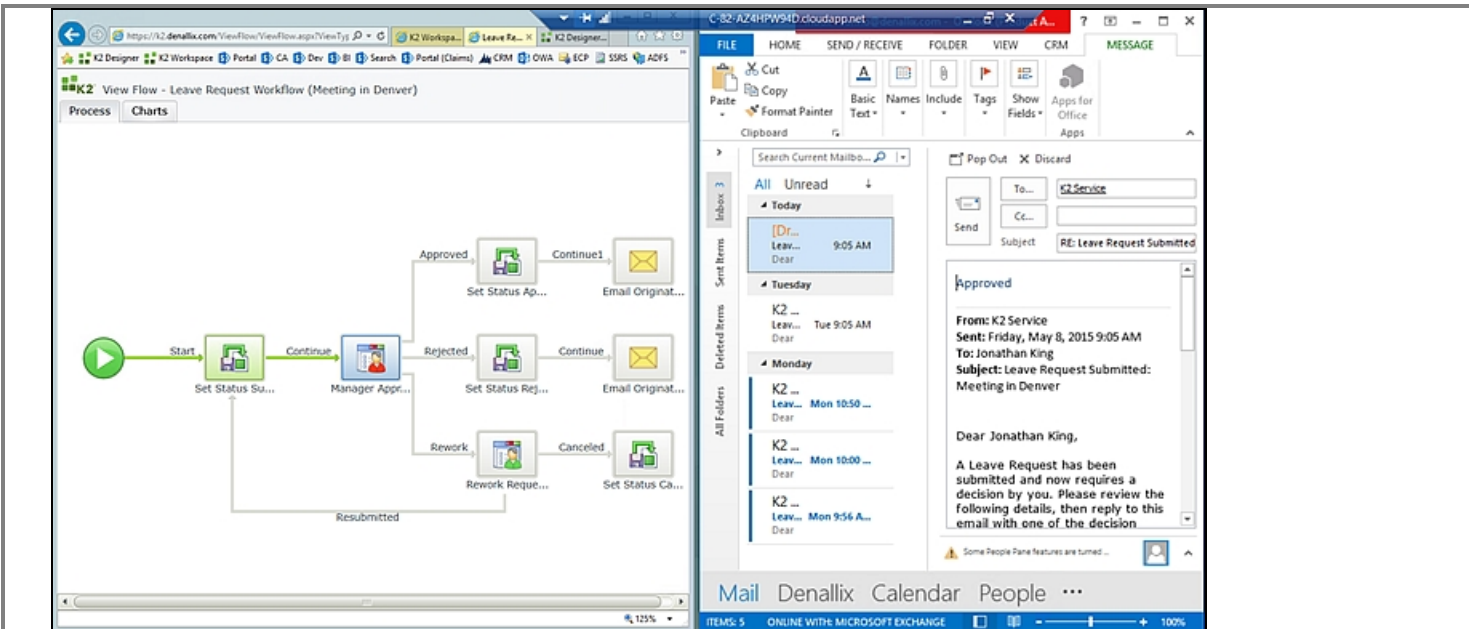
Your View Flow Report should look similar to the image below. Notice the green line flows through the system task (Set Status), then stops at the Manager Approval task. This is telling you that the workflow is currently waiting for the Manager Approval task to be actioned. In the next step, we are going 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.)

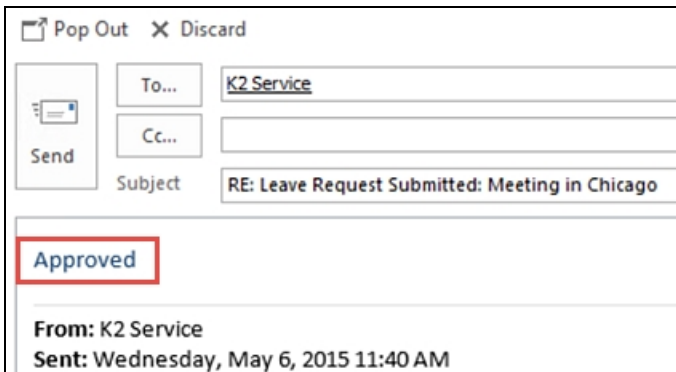


Before you action the request, adjust your View Flow Report screen and Outlook screen so that they are side-by-side. The View Flow Report can update very quickly (depending on your environment) after we action the request in Outlook. With the screens side-by-side, you will be able to see the View Flow update.

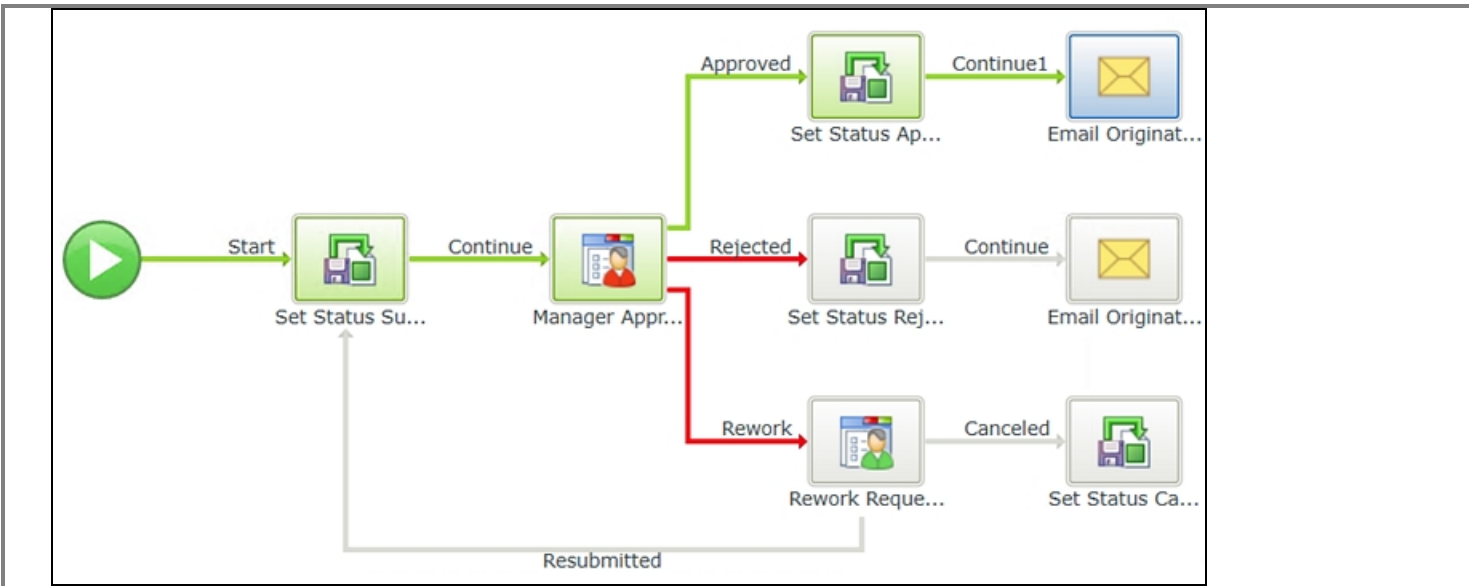


We are going to action the Leave Request using a technology called SmartActions. SmartActions allow the user to action a task by simply replying to the email, then entering one of the action options as the message body. K2 will 'read' the action and move the workflow along accordingly.

- 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.)



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



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					
		1 of 1		Select a format	Export
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 DENALLIXBOB	K2 DENALLIXBlake
Manager Approval	5/4/2015 10:04:47 AM	Set Status Submitted Event	Completed	K2Server	K2 DENALLIXBlake
July Vacation					
Manager Approval	5/11/2015 9:47:51 AM	Manager Approval	Active	K2 DENALLIXJONNO	K2 DENALLIXAdministrator
Manager Approval	5/11/2015 9:47:51 AM	Set Status Submitted Event	Completed	K2Server	K2 DENALLIXAdministrator
K2 smartforms Training					
Manager Approval	5/11/2015 9:48:26 AM	Manager Approval	Active	K2 DENALLIXBOB	K2 DENALLIXAnthony
Manager Approval	5/11/2015 9:48:26 AM	Set Status Submitted Event	Completed	K2Server	K2 DENALLIXAnthony

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.

The screenshot shows the SmartObject Explorer interface. The 'Process Overview' SmartObject is expanded, showing its 'Properties' section. A red arrow points from the 'Process Overview SmartObject' label to the 'Process Name' property. Below the properties, a table displays the results of the SmartObject execution. The table has four columns: Process Name, Folder, Instances, and Average Duration. The data rows are: Leave Request Workflow (9 instances, 00:11:00:19), Sales Orders Process (1 instance, 06:20:41:32), and Schedule Appointment (1 instance, 00:00:00:22). A red arrow also points from the 'Process Instance Overview' association to the 'Process Name' property.

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 Instance SmartObject

Process Folio	Originator	Status	Priority	Start Date	Finish Date	Duration
K2 smartforms Training	K2.DENALLK\ADMINISTRATO R	Completed	Medium	5/4/2015 9:55:16 AM	5/4/2015 10:02:41 AM	00:00:07:25
May Vacation Request	K2.DENALLK\ADMINISTRATO R	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.

Execute SmartObject: 'Process Overview' Method: 'List'

SmartObject Name: Process Overview

Method to Execute: List

Execute

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

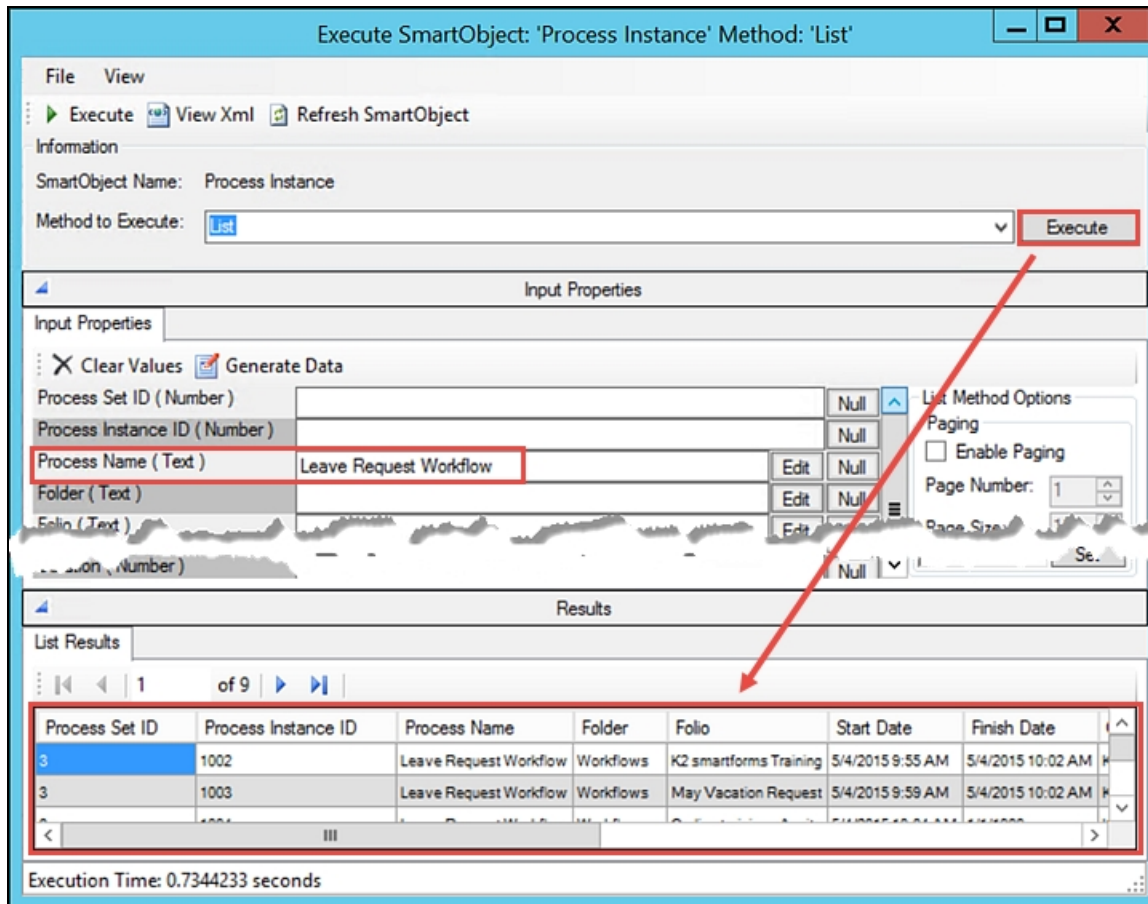
- e. Right-click the **Process Instance** SmartObject and select **Execute SmartObject**. Enter the following parameter for the **Process Name**:

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.



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 way.

Name:

Description:

Report Type

Tabular Reports

Tabular reports are the simplest and fastest way to list your data.

Summary Reports

Summary reports list your data with subtotals and other summary information.

Matrix Reports

Matrix reports list summaries of your data in a grid against both horizontal and vertical criteria.

Navigation: Back Next Finish Cancel

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.

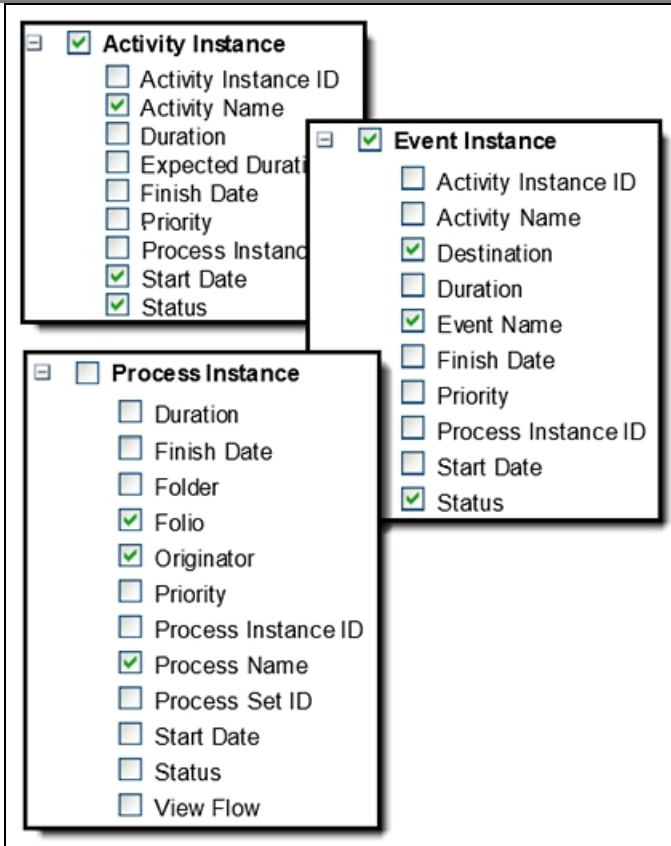
- d. Scroll to the bottom of the data source options and expand the **Workflow Reports** category, then **Workflow General**. Click the **Activity Instance** SmartObject to add it to the Related Data pane.

Primary Data Source	Related Data
<ul style="list-style-type: none"> Workflow Reports Workflow General <ul style="list-style-type: none"> Activity Data <li style="border: 1px solid red; padding: 2px;">Activity Instance Activity Instance Destination Activity Slot Data Activity Slot XML 	<ul style="list-style-type: none"> <input type="checkbox"/> Activity Data <input checked="" type="checkbox"/> Activity Instance <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Activity Instance ID <input checked="" type="checkbox"/> Activity Name <input checked="" type="checkbox"/> Duration <input checked="" type="checkbox"/> Expected Duration <input checked="" type="checkbox"/> Finish Date <input checked="" type="checkbox"/> Priority <input checked="" type="checkbox"/> Process Instance ID

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.

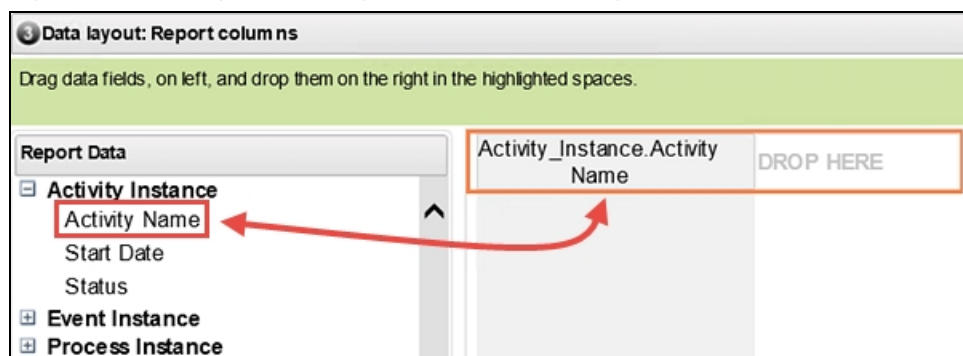
- e. For the **Activity Instance** SmartObject, CHECK (only) the following properties:
- Activity Name, Start Date, Status
- f. Locate the **Event Instance** SmartObject and CHECK the following properties:
- Destination, Event Name, Status
- g. Locate the **Process Instance** SmartObject and CHECK the following properties:
- Folio, Originator, Process Name

Your Related Data selections should reflect the image below. Click **Next** to continue.



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.



- 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.

Activity Name	Date Started	Event Instance	Event_Instance.Status	Event_Instance
---------------	--------------	----------------	-----------------------	----------------

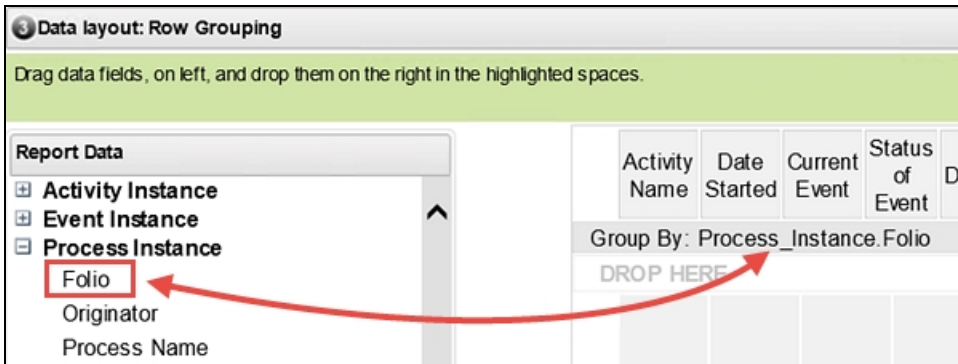
- k. Repeat this process for each of the column titles until you have the following column titles for your report:

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.

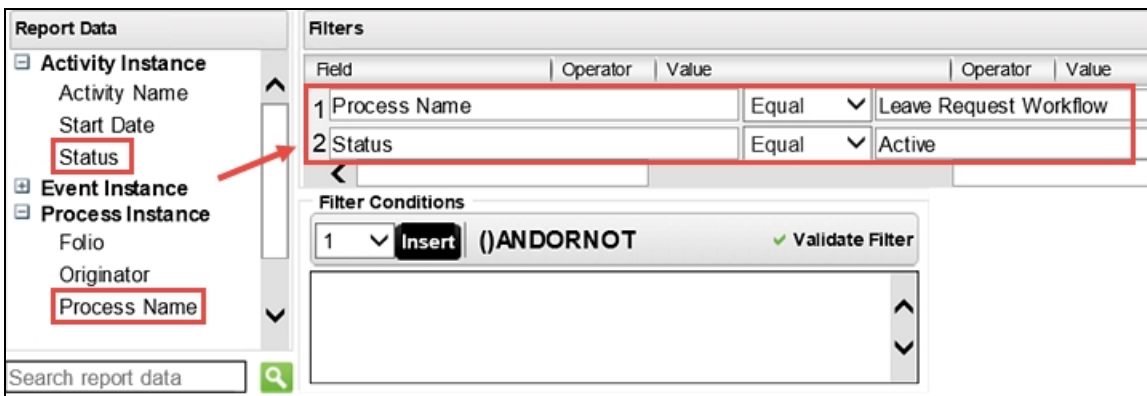
- l. On the **Data layout: Row Grouping** screen, drag the **Process Instance > Folio** property into the **DROP HERE** box on the design canvas. Click **Next** to continue.



- 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.



- 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.

Arial 11 B I U A

Activity Name	Date Started	Current Event	Status of Event	Destination	Originator
Group By: Process_Instance.Folio					
xxxxxxxxxxxx	1999/01/01	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx

- r. We are now ready to view our Custom Report. On the **Finish** screen, click the **View my report** link to run the report.

6 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

1 of 1 Select a format Export Refresh report

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.DENALLIXBOB	K2.DENALLIXBlake
Manager Approval	5/4/2015 10:04:47 AM	Set Status Submitted Event	Completed	K2Server	K2.DENALLIXBlake
July Vacation					
Manager Approval	5/11/2015 9:47:51 AM	Manager Approval	Active	K2.DENALLIXJONNO	K2.DENALLIXAdministrator
Manager Approval	5/11/2015 9:47:51 AM	Set Status Submitted Event	Completed	K2Server	K2.DENALLIXAdministrator
K2 smartforms Training					
Manager Approval	5/11/2015 9:48:26 AM	Manager Approval	Active	K2.DENALLIXBOB	K2.DENALLIXAnthony
Manager Approval	5/11/2015 9:48:26 AM	Set Status Submitted Event	Completed	K2Server	K2.DENALLIXAnthony

- 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**.

Report Viewer >> Open Leave Requests

Report Options

1 of 1

Select a format
Excel
Acrobat (PDF) file

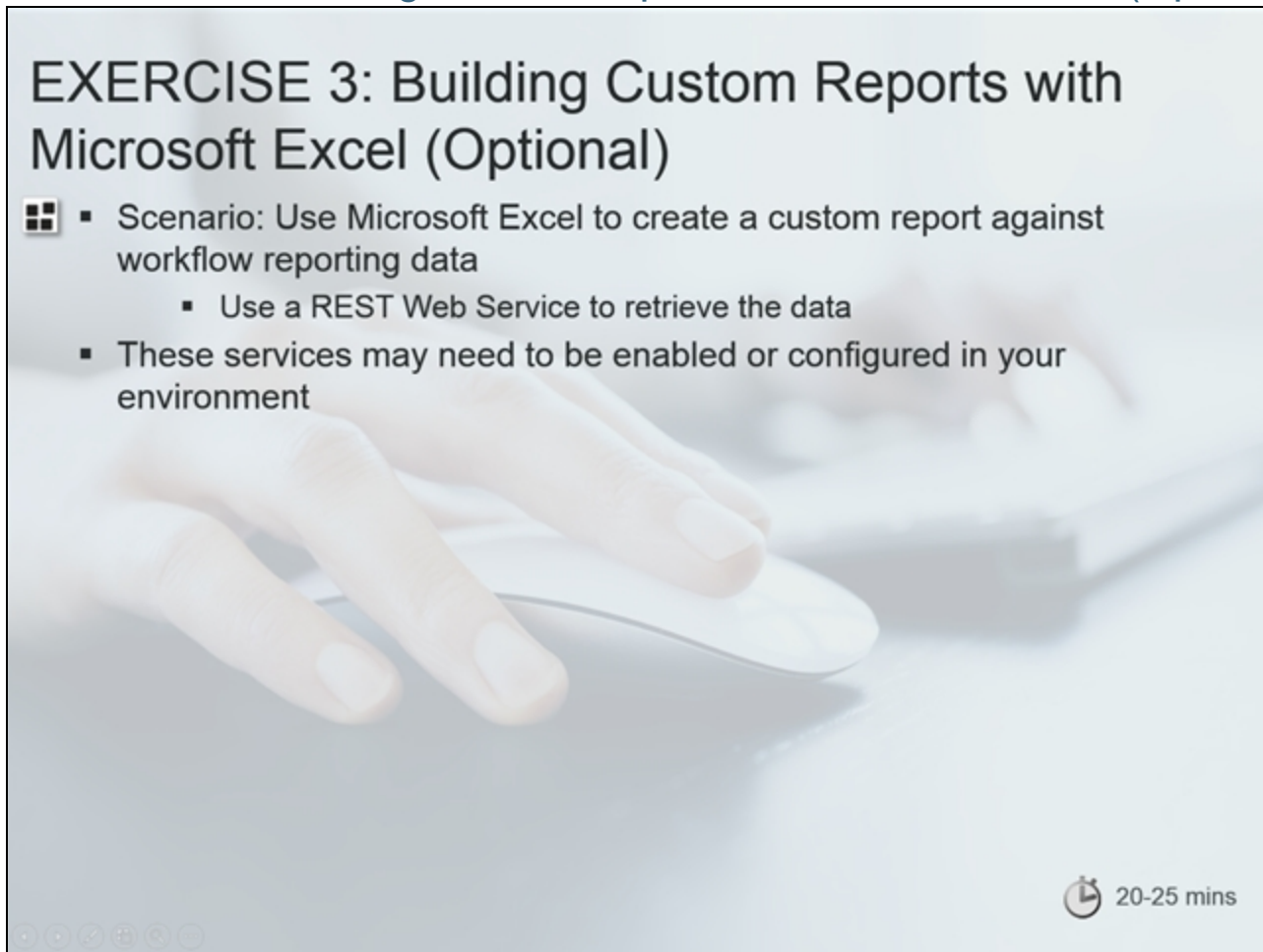
Export

Activity Name	Date Started	Current Event	Status of Event	Destination	Originator
On-line training - Appit					
Mana					ALLIXBlake
Mana					ALLIXBlake
July					
Mana	5/4/2015 10:04:47 AM	Manager Approval	Active	K2.DENALLIXBOB	K2.DENALLIXBlake
Mana	5/4/2015 10:04:47 AM	Set Status Submitted Event	Completed	K2Server	K2.DENALLIXBlake
K2 s					
July Vacation					
Mana	5/11/2015 09:47:51 AM	Manager Approval	Active	K2.DENALLIXJONNO	K2.DENALLIXAdministrator
Mana	5/11/2015 09:47:51 AM	Set Status Submitted Event	Completed	K2Server	K2.DENALLIXAdministrator
K2 smartforms Training					
Mana	5/11/2015 09:48:26 AM	Manager Approval	Active	K2.DENALLIXBOB	K2.DENALLIXAnthony
Mana	5/11/2015 09:48:26 AM	Set Status Submitted Event	Completed	K2Server	K2.DENALLIXAnthony

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

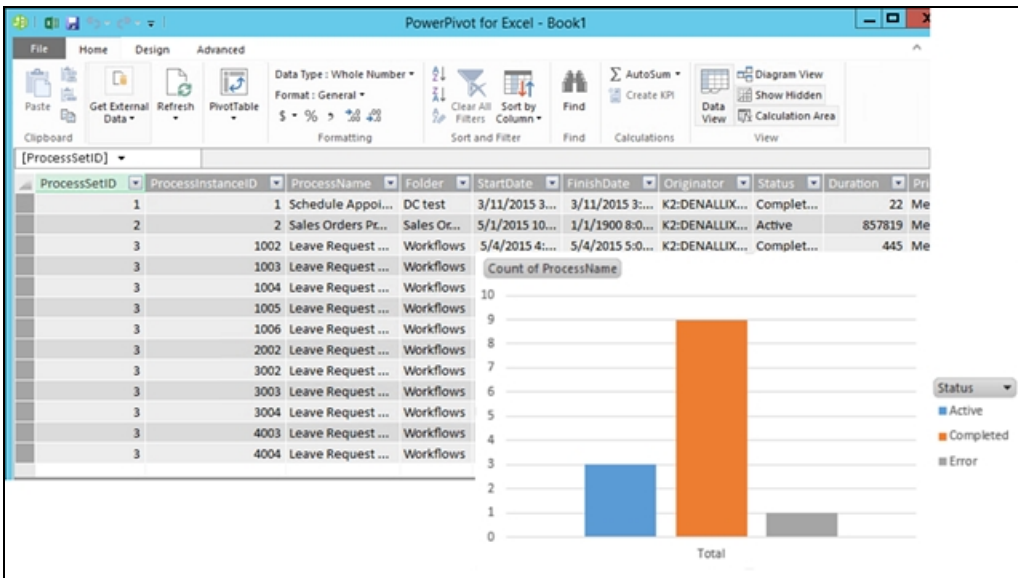
20-25 mins



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.



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}&\${top}={top}&\${skip}={skip}&filterXml={filterXml}&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](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](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	Priority
1	1	Schedule Appoi...	DC test	3/11/2015 3:...	3/11/2015 3:...	K2:DENALLIX...	Comple...	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...	Comple...	445	Me
3	1003	Leave Request ...	Workflows	5/4/2015 4:...	5/4/2015 5:0...	K2:DENALLIX...	Comple...	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...	Comple...	2784	Me
3	1006	Leave Request ...	Workflows	5/4/2015 5:...	5/4/2015 5:0...	K2:DENALLIX...	Comple...	224	Me
3	2002	Leave Request ...	Workflows	5/5/2015 4:...	5/5/2015 4:0...	K2:DENALLIX...	Comple...	95	Me
3	3002	Leave Request ...	Workflows	5/6/2015 6:...	5/6/2015 6:4...	K2:DENALLIX...	Comple...	129	Me
3	3003	Leave Request ...	Workflows	5/8/2015 4:...	5/8/2015 4:2...	K2:DENALLIX...	Comple...	1170	Me
3	3004	Leave Request ...	Workflows	5/8/2015 4:...	5/8/2015 4:2...	K2:DENALLIX...	Comple...	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](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_Leaming/Leave_Request/SmartObjects</endpoint>

  <endpoint>https://k2.denallix.com:8888/SmartObjectServices/wcf/K2_Leaming/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}](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 **managedEndpoints** 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>
```

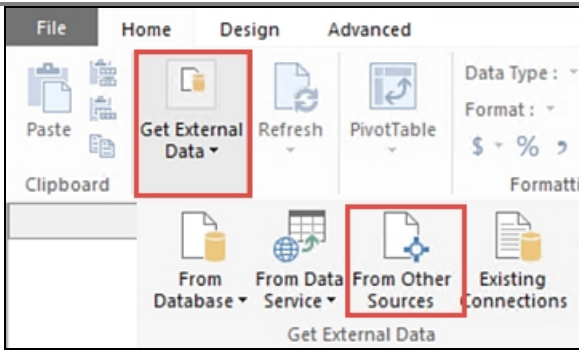
- **Restart the K2 blackpearl Service.** (**Start > Administration Panel > Services**)

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.

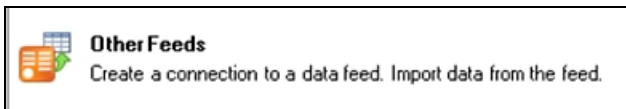
- Edit the URL so that it matches the example below.
[https://k2.denallix.com:8888/SmartObjectServices/rest/Workflow Reports/Workflow General/Process Instance/List?\\$format=ATOM](https://k2.denallix.com:8888/SmartObjectServices/rest/Workflow Reports/Workflow General/Process Instance/List?$format=ATOM) (and remove the remaining string).
- Launch **Excel**. Create a new Workbook. Click the **PowerPivot** tab, then click **Manage**.



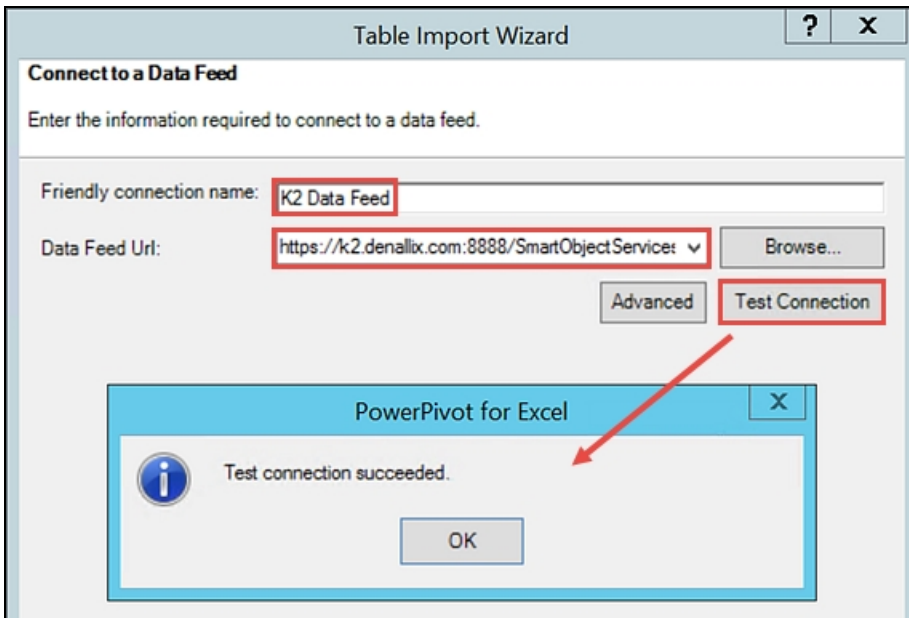
- When the Manage PowerPivot window opens, select **Get External Data**, then **From Other Sources**.



f. The **Connect to a Data Source** Import Wizard opens. Select **Other Feeds** then click **Next**.

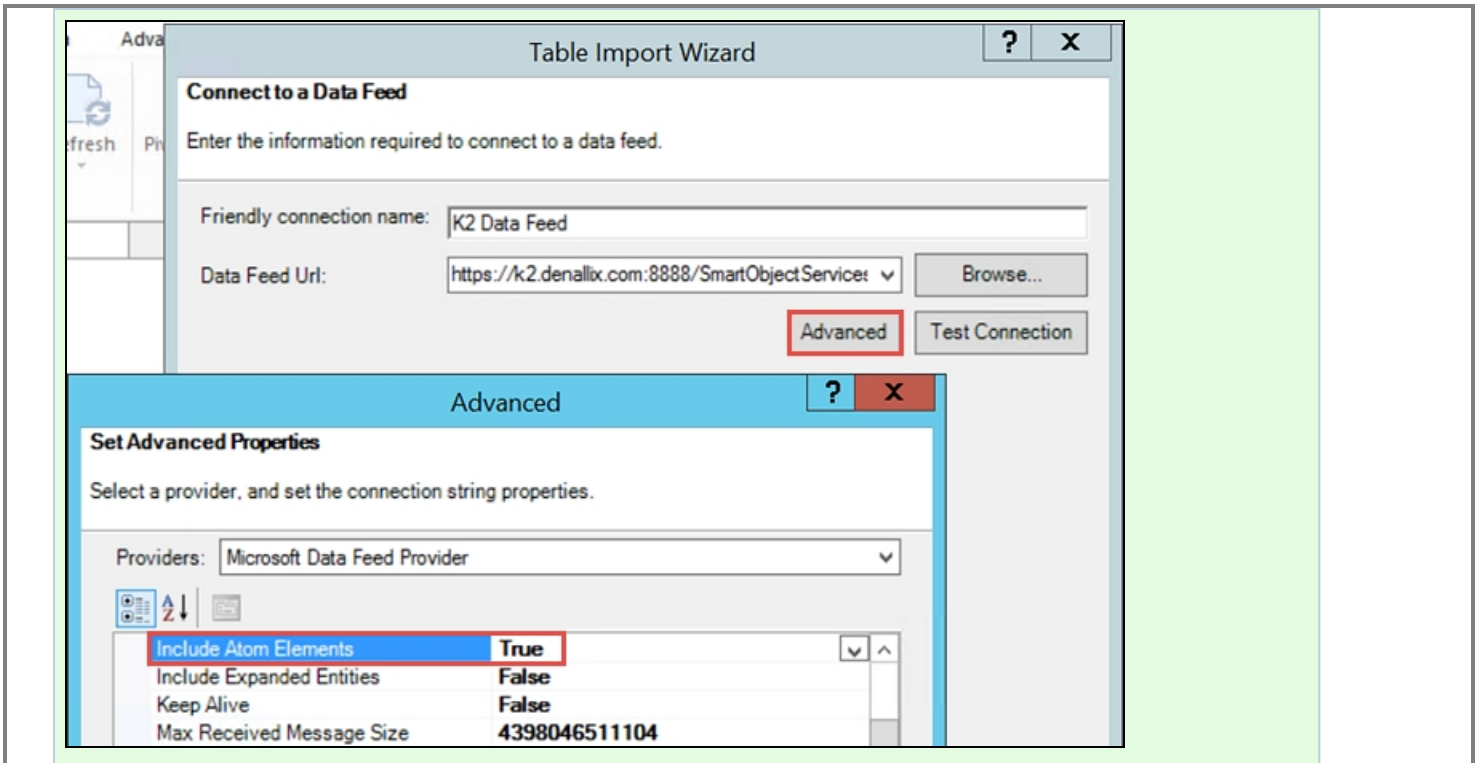


g. For the **Friendly connection name**, enter *K2 Data Feed* then **copy the URL** we configured in the previous steps into the **Data Feed Url** field. **Test** the connection.



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.



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
1	1	Schedule Appoi...	DC test	3/11/2015 3:...	3/11/2015 3:...	K2:DENALLIX...	Comple...	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...	Comple...	445 Me
3	1003	Leave Request ...	Workflows	5/4/2015 4:...	5/4/2015 5:0...	K2:DENALLIX...	Comple...	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...	Comple...	2784 Me
3	1006	Leave Request ...	Workflows	5/4/2015 5:...	5/4/2015 5:0...	K2:DENALLIX...	Comple...	224 Me
3	2002	Leave Request ...	Workflows	5/5/2015 4:...	5/5/2015 4:0...	K2:DENALLIX...	Comple...	95 Me
3	3002	Leave Request ...	Workflows	5/6/2015 6:...	5/6/2015 6:4...	K2:DENALLIX...	Comple...	129 Me
3	3003	Leave Request ...	Workflows	5/8/2015 4:...	5/8/2015 4:2...	K2:DENALLIX...	Comple...	1170 Me
3	3004	Leave Request ...	Workflows	5/8/2015 4:...	5/8/2015 4:2...	K2:DENALLIX...	Comple...	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.

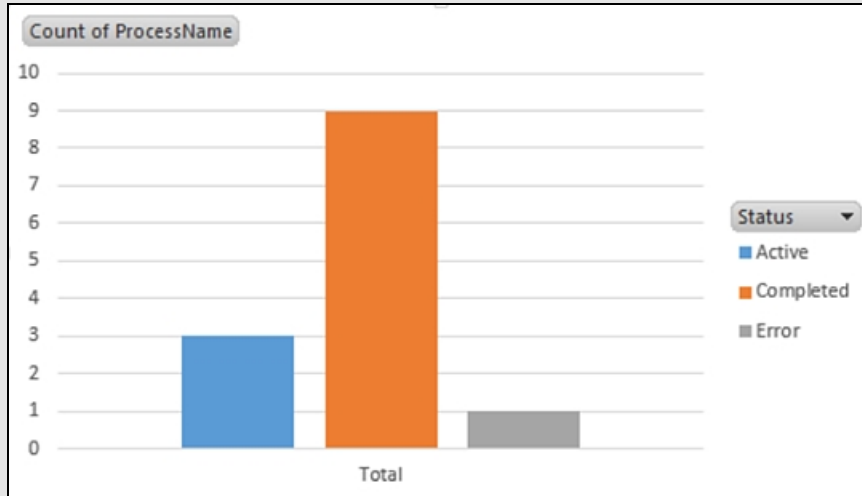
Step 2: Create a PivotChart

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

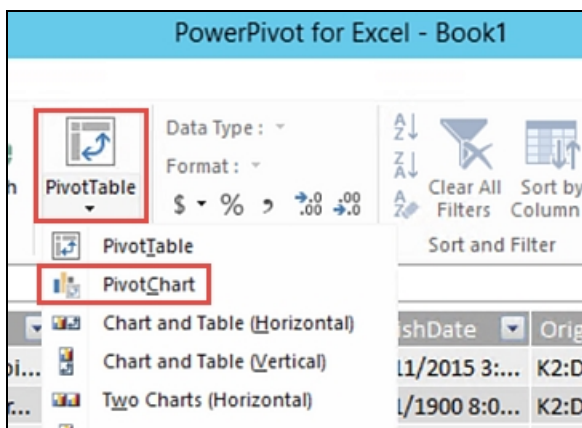
1. Create a **PivotChart** from the PowerPivot table created in the previous step.
2. Configure the **Status** property to be the chart **Legend** and the **ProcessName** to be the chart **Sum**. Make a note of the current Active and Completed values.



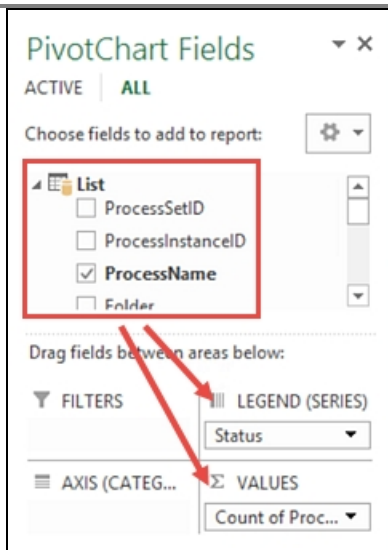
3. In K2 Workspace, action one of the **(Active) Leave Request Workflow** Process Instances. You can either redirect a Worklist Item to yourself and action it directly from K2 Workspace, or you can open Outlook for the Destination user and action the request using SmartActions.
4. Return to the **PivotChart** 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**.

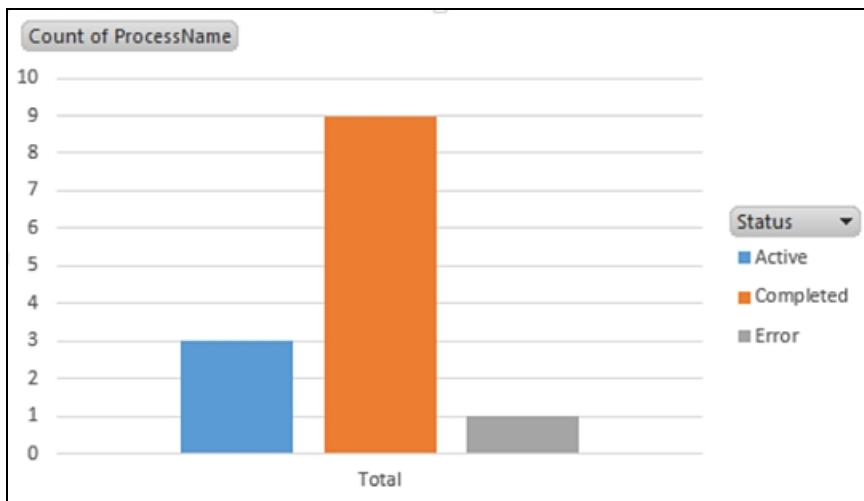


- b. Select **New Worksheet**, then click **OK**.
The PivotChart we are going to configure will show the status values for all Processes. We want to see how many Processes have Completed, how many are Active and how many are in an Error state.
- c. In the PivotChart Fields pane, expand the List options. 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

Delegate | Redirect | Release

Selected Filter: (None)

Use the following fields to further refine the selected filter

Process Full Name

Activity Name

Event Name

Folio

Destination

Worklist Date

Search

- f. A list of active Leave Request Worklist Items is displayed. CHECK the box to the left of one of the **Active** Worklist 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.

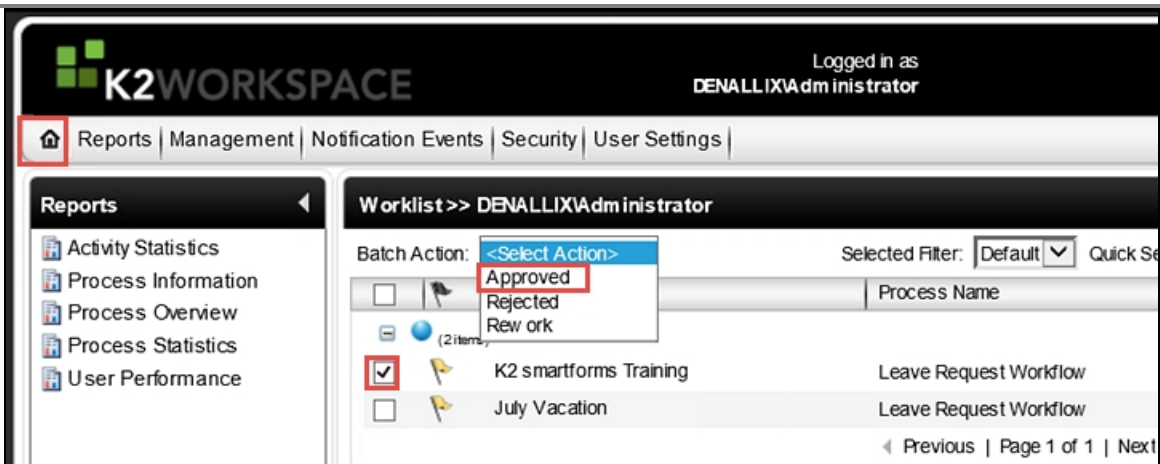
k2.denallix.com:5555 > Workflow Server > Worklists

Delegate | **Redirect** | Release

Selected Filter: (None)

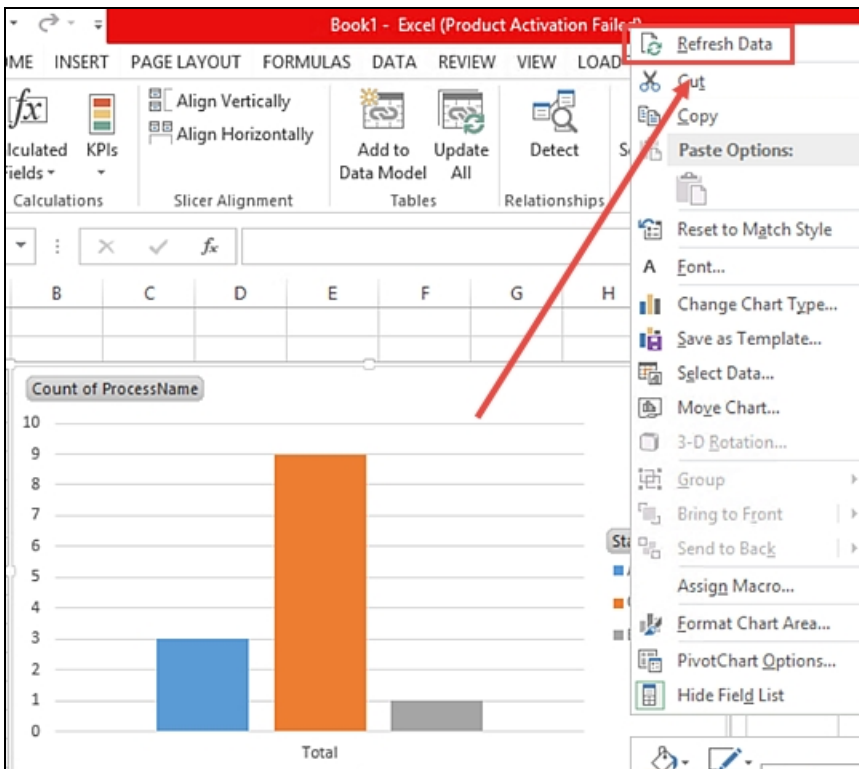
	Process Full Name ▲	Activity Name	Event Name	Folio	Destination	Worklist Date	Status
<input type="checkbox"/>	Workflows\Leave Request Workflow	Manager Approval	Manager Approval	On-line training - Appit	K2:DENALLIX\BOB	5/4/2015 10:04:48 AM	Available
<input type="checkbox"/>	Workflows\Leave Request Workflow	Manager Approval	Manager Approval	July Vacation	K2:DENALLIX\Administrator	5/11/2015 9:47:53 AM	Available
<input checked="" type="checkbox"/>	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.

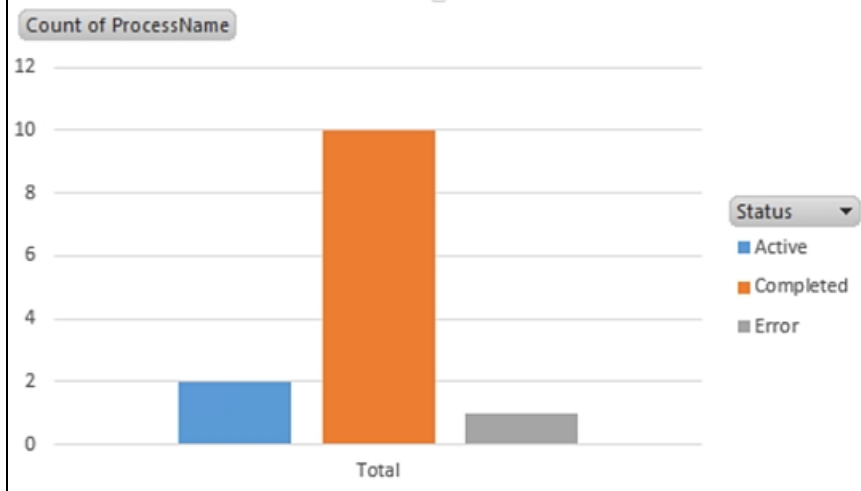


Now that we have actioned the request, the Process Instance will have completed. We will return to our PivotChart and refresh it. We should see the Count for Completed instances go up by one, and the Count for Active instances, go down by one. This will demonstrate the live data feed configuration.

- 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.

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

© 2008-2016 SOURCECODE TECHNOLOGY HOLDINGS, INC. ALL RIGHTS RESERVED. SOURCECODE SOFTWARE PRODUCTS ARE PROTECTED BY ONE OR MORE U.S. PATENTS. OTHER PATENTS PENDING. SOURCECODE, K2, K2 BLACKPEARL, K2 SMARTFORMS AND APPIT ARE REGISTERED TRADEMARKS OR TRADEMARKS OF SOURCECODE TECHNOLOGY HOLDINGS, INC. IN THE UNITED STATES AND/OR OTHER COUNTRIES.

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