



K2 Learning

Leave Request (Basic)

May 2019

Leave Request (Basic Version)

In this beginner-level tutorial, you will learn how to build a simple leave request application using K2 Designer. This application will combine data (SmartObjects), Forms (SmartForms forms and views), and a workflow into the context of a process-driven business application.

The application itself is a basic Leave Request Approval application. The workflow contains a (user) Task step, where the assigned approver will either approve or reject the request. System tasks will update the request status throughout the workflow. Behind the scenes, the request data is stored in a SmartBox SmartObject, and we will use SmartForms to allow our users to interact with the application.

If you want to learn a little more about the application you are about to build and how it will behave, please review the design of the application in [Leave Request \(Basic Version\) Application Design](#).

Steps

1. Add the Application Categories
2. Build the Leave Request SmartObject
3. Build the Leave Request Item View
4. Build the Leave Request List View
5. Build the Leave Request Form
6. Create the Leave Request Workflow
7. Add a SmartObject Method Step and a Task Step
8. Add a Decision Step
9. Add Additional SmartObject Method Steps and End Step
10. Deploy the Leave Request Workflow
11. Assign Workflow Rights
12. Set the Folio Value
13. Set the Current User's Details on the Request Form
14. Set the Requester's Details on the Manager's Approval Form
15. Test the Leave Request (Basic Version) Application
16. (Optional) Clean your K2 environment by deleting application artifacts

First Step: 1. Add the Application Categories

Leave Request (Basic Version): Application Design

The Leave Request (Basic Version) application is a simple, yet complete, application that demonstrates how to combine **data (SmartObjects)**, **forms (SmartForms)**, and **workflow** into a cohesive and fully-functional application. In this tutorial, you will build a Leave Request Approval application that stores the leave request data as a SmartObject in K2 SmartBox, that uses K2 SmartForms to allow users to submit, view and approve leave requests, and a workflow that controls the leave request approval process.

Let's look at each of the components you will build in a little more detail.

Data

There are no pre-existing data sources for the Leave Request Basic application, so you will create a SmartObject to store leave request data in K2 SmartBox. You will define, or add, the properties necessary to capture the leave request details, and rely on the default methods added by K2 to create and update the data.

The Leave Request (SmartBox) SmartObject

LEAVE REQUEST SMARTOBJECT				
Add Edit Remove Remove All Move up Move down				
NAME	TYPE	KEY	REQUIRED	UNIQUE
ID	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"/>

Forms

The **Leave Request Form** contains two views: the **Leave Request Item View** and the **Leave Request List View**. The item view is the user interface for creating new or editing existing leave requests. The list view displays the current user's past requests and their status. (The list view is a read-only view). We will be using this forms to allow users to submit, view and approve leave requests.

The Leave Request Form

The screenshot shows a K2 application interface with two main sections: a 'Leave Request' form and a 'Previous Leave Requests' table. Annotations with arrows point to various elements:

- Form:** Points to the 'Leave Request' form section.
- Item View:** Points to the 'Leave Type' dropdown menu.
- Controls:** Points to the 'Request Status' text input field and the 'Create' button.
- List View:** Points to the 'Previous Leave Requests' table.

Leave Request Form Fields:

- Leave Request Title:
- Employee Name:
- Employee Email:
- Leave Start Date: (calendar icon)
- Leave End Date: (calendar icon)
- Leave Type: (dropdown arrow)
- Requester Comments:
- Request Status:
- Create button

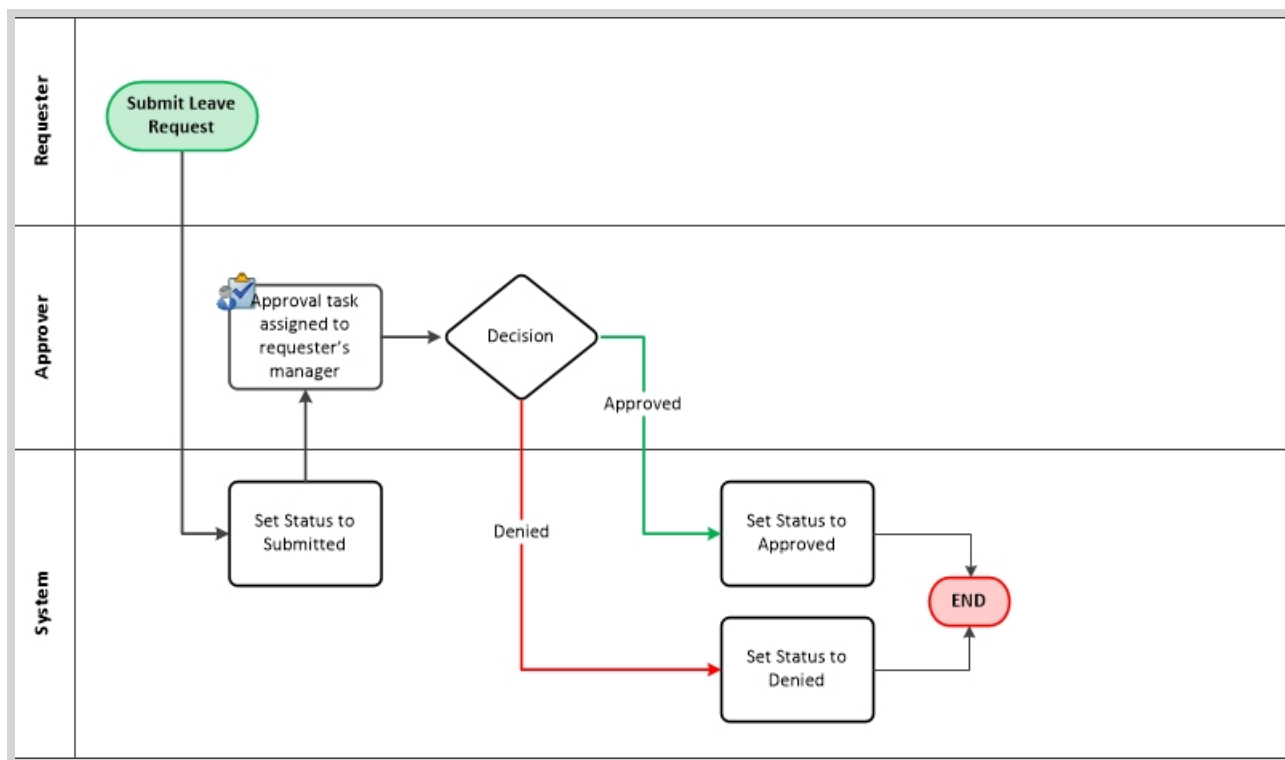
Previous Leave Requests Table:

LEAVE REQ...	EMPLOYEE ...	LEAVE STAR...	LEAVE END ...	LEAVE TYPE	REQUEST S...
Basic Test O...	Denallix Ad...	3/14/2017	3/17/2017	Study Leave	Approved
Basic Test T...	Denallix Ad...	3/20/2017	3/23/2017	Paid Time ...	Submitted
Basic Test T...	Denallix Ad...	3/13/2017	3/16/2017	Study Leave	Submitted

Workflow

Let's consider the design of the workflow component of our leave request application. In the image below, there are two swim-lanes for user tasks. The first lane represents the requester, or the person who submits the leave request. The second lane represents the person who will make the approval decision. 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.

The Leave Request Workflow represented as a swim-lane flowchart

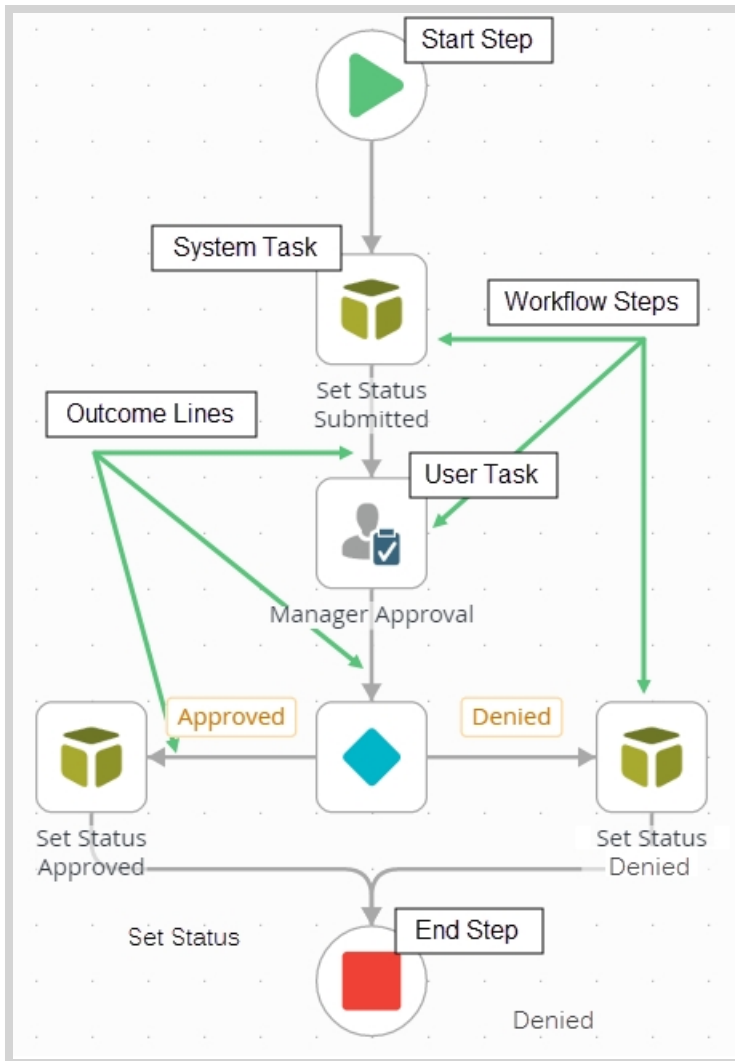


Leave Request Workflow

The Leave Request Workflow begins with a **Start** step (the green triangle). The next step is a **SmartObject Method** (system task) called **Set Status Submitted**. In this step, the K2 server updates the *Request Status* property of the Leave Request SmartObject.

The next step is a **Task** (user task) called **Manager Approval**. This is the step that requires the task recipient to decide whether to approve or deny the leave request. A **Decision** step follows and this step creates the outcomes, or lines, for each of the possible manager decisions. Each outcome has a **SmartObject Method** step that once again updates the request status property. Finally, there is a single **End** step, which provides a visual indicator that the workflow has completed.

The completed Leave Request Workflow



First Step: 1. Add the Application Categories

1. Add the Application Categories

In this step you will add categories to the K2 Designer explorer. Categories are similar to folders, and are often used to organize your K2 environment, the applications in a K2 environment, and the application elements of an application. Elements include SmartObjects, views, forms and workflows. While it is not required to create Application-specific categories and folders, they are helpful to keep your work organized. This is especially important when working in a shared environment where multiple people are building applications. In this context, we will use categories to separate your tutorials from other applications.

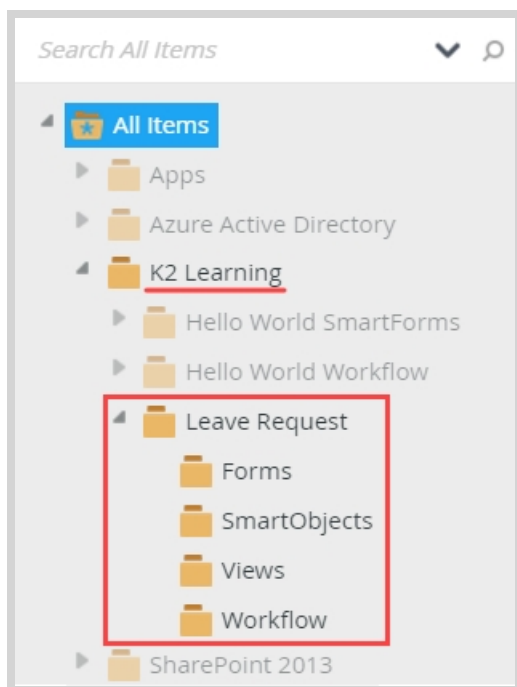
1. Add the following categories to the **All Items** node in the K2 Designer explorer.

- *K2 Learning*
 - *Leave Request*
 - *Forms*
 - *SmartObjects*
 - *Views*
 - *Workflow*

Note

If you are working in a shared environment where there are multiple builds of the same application, consider adding your initials to the application name so that you can easier determine which projects are yours. For example, *Leave Request YourInitials*, like *Leave Request KL*.

Leave Request Categories

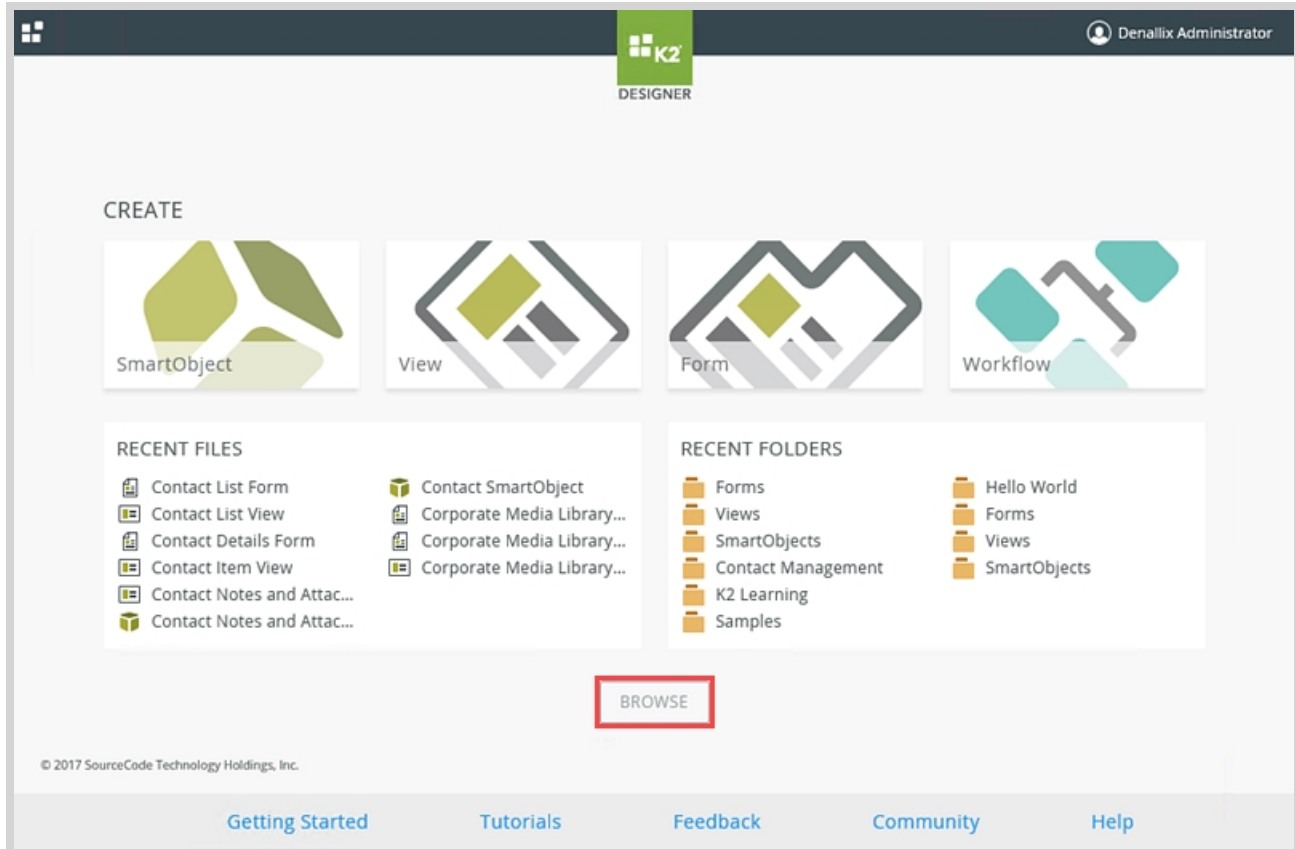


a. Launch the **K2 Designer** site.

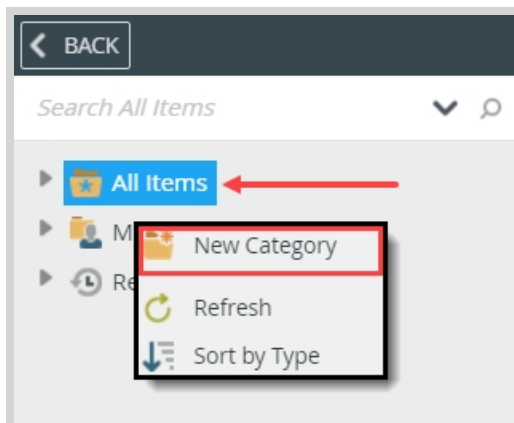
Note

If you are unsure of how to launch the **K2 Designer** site, see [Accessing K2 Sites](#).

- b. K2 Designer will open on the home page. You will be creating new categories in the explorer; to navigate to the explorer page, click the **BROWSE** button.



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



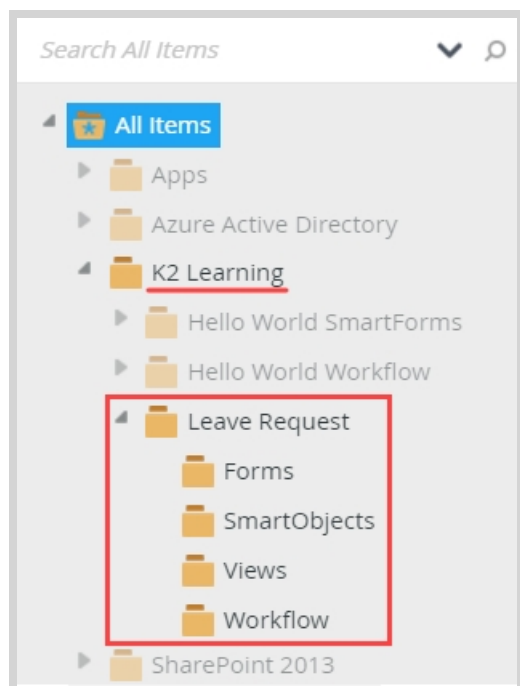
- d. For the category **Name**, enter *K2 Learning* then click **OK**. (If the K2 Learning category already exists, continue to the next step)

Note

If you are working in a shared environment where there are multiple builds of the same application, consider adding your initials to the application name so that you can easier determine which projects are yours. For example, *Leave Request YourInitials*, like *Leave Request KL*.

- e. Now we will create a category for the application itself. Locate the **K2 Learning** category, then right-click and select **New Category**. For the category name, enter *Leave Request* then click **OK**.
- f. Now we will create sub-categories to hold the various elements that will make up our application. Right-click the **Leave Request** category and select **New Category**. For the category name, enter *Forms* then click **OK**.
- g. Repeat the previous step three more times to add the following additional categories under **Leave Request**:
SmartObjects
Views
Workflow

Your explorer should look similar to the image below.



Review

In this step, you created the categories for the Leave Request (Basic Version) application. While not required, categories help keep your projects organized, especially if you are working within a shared environment. In the next step, you will create the Leave Request (SmartBox) SmartObject.

Next Step: 2. Build the Leave Request SmartObject

2. Build the Leave Request SmartObject

In this step, you will create the Leave Request (SmartBox) SmartObject in K2 Designer. This SmartObject will store the leave requests entered by our users.

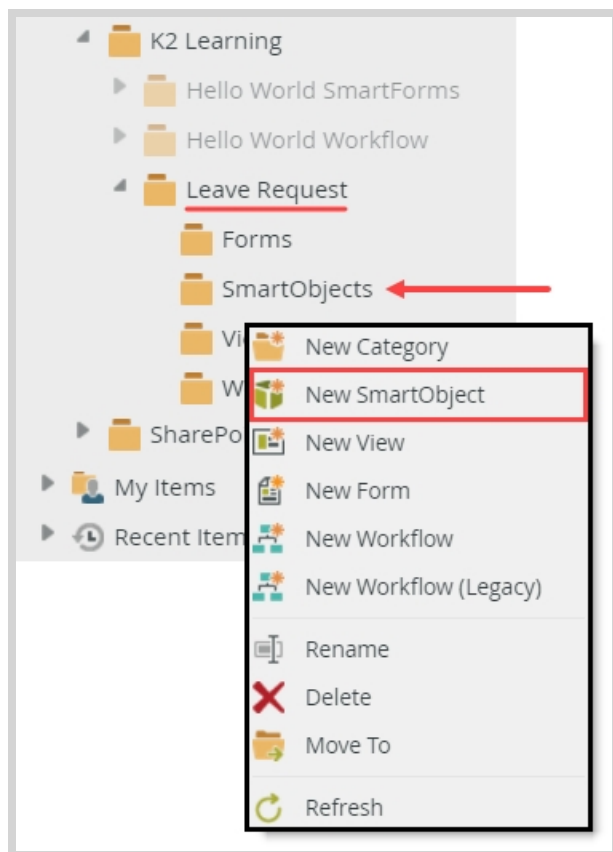
SmartBox SmartObjects are K2-provided and managed storage where K2 creates a new table on your behalf in the K2 database. You add properties (think columns in a table) and methods (think create, save, delete).

1. Begin by adding a new SmartObject to store the leave request records. Create a new SmartBox SmartObject under the **SmartObjects** category and name it *Leave Request SmartObject*, then UNCHECK the Legacy Workflow Designer option. You are using the new designer in this tutorial. (Depending on your K2 version, you may not see the legacy workflow checkbox.)

Note

If you are working in a shared environment, consider adding your initials to the SmartObject name.

- a. Create a new SmartBox SmartObject from K2 Designer to store the leave request records. Right-click the **SmartObjects** category and select **New SmartObject**.



- b. The SmartObject designer opens on the **GENERAL** settings screen. **Name** the new SmartObject *Leave Request SmartObject* then confirm the **SmartBox** option is selected. In a later step, you will learn more about the Advanced option, which connects the SmartObjects to an existing data source. UNCHECK the option to **Allow this SmartObject to be used in the Legacy Workflow Designer**, then click **CREATE**. We will not be

working with the legacy workflow designer in this tutorial.

Note

If you are working in a shared environment, consider adding your initials to the SmartObject name.

Note

Depending on your K2 installation configuration, you may not see the **Allow this SmartObject to be used in the Legacy Workflow Designer** option. If you do not see the legacy workflow designer option, it is OK to continue to the next step.

The screenshot shows the K2 SmartObject creation wizard. The 'GENERAL' tab is active. The 'Name' field contains 'Leave Request SmartObject'. The 'Description' field is empty. The 'Category' is set to 'K2 Learning\Leave Request\SmartObjects'. The 'Type' is 'SmartObject'. The 'Workflow Use' checkbox 'Allow this SmartObject to be used in the Legacy Workflow Designer' is unchecked. The 'CREATE' button is highlighted.

- Next, you will define the properties necessary for the Leave Request SmartObject. **Add** the remaining Leave Request SmartObject **properties** using the table below as a guide.

Name	Type	Notes
ID	Autonumber	K2 adds the ID automatically and it is the primary (or

Name	Type	Notes
		unique) key for each record.
<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	The workflow updates this field.

- a. On the properties screen, notice that K2 added the ID property for you, which is the primary (or unique) key for each SmartObject record. In this step, you will add the remaining properties for the Leave Request Application.

Click **Add**. Enter the property **Name** and data **Type** using the table values below the image for reference. Click the **<Add property>** link to add more properties until you have entered all eight properties, and then click **OK**.

LEAVE REQUEST SMARTOBJECT

+ Add Edit Remove Remove All Move up Move down

NAME	TYPE	KEY	REQUIRED	UNIQUE
ID	Autonumber	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Add Properties (Leave Request SmartObject)

+ Add Edit Remove Remove All

NAME	DESCRIPTION	TYPE	KEY	REQUIRED	UNIQUE
Leave Request Title		A Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

<Add property>

Name	Type	Notes
ID	Autonumber	K2 adds the ID automatically and it is the primary (or unique) key for each record.
<i>Leave Request Title</i>	Text	A unique title for each leave request.

Name	Type	Notes
<i>Employee Name</i>	Text	The person who is requesting the leave.
<i>Employee Email</i>	Text	The email of the person requesting the leave.
<i>Leave Start Date</i>	Date	What date the leave will start on.
<i>Leave End Date</i>	Date	What date the leave will end on.
<i>Leave Type</i>	Text	This will store the selected leave type, selected by a drop-down list.
<i>Requester Comments</i>	Memo	To allow the requester to enter comments.
<i>Request Status</i>	Text	The workflow updates this field with the status of the leave request as the workflow progresses.

3. **Finish** the wizard to publish the SmartObject.
The Leave Request (SmartBox) SmartObject

LEAVE REQUEST SMARTOBJECT					
Add Edit Remove Remove All Move up Move down					
NAME	TYPE	KEY	REQUIRED	UNIQUE	
ID	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"/>	

- a. Your **properties** screen should look like the image below. Now you can "finish" the SmartObject. Finishing a SmartObject publishes it to the K2 server and make it available for use in other artifacts, such

as views and forms. Click **FINISH** to publish the new SmartObject to the K2 server.

The screenshot shows the K2 SmartObject editor interface. At the top, there's a header bar with a logo, the title 'Leave Request SmartObject', and a user profile 'Denallix Administrator'. Below this is a navigation bar with tabs: 'GENERAL', 'PROPERTIES' (selected), and 'ASSOCIATIONS'. A 'FINISH' button is highlighted in red on the right. The main area displays the 'LEAVE REQUEST SMARTOBJECT' configuration. It includes a toolbar with actions: Add, Edit, Remove, Remove All, Move up, and Move down. Below the toolbar is a table with columns: NAME, TYPE, KEY, REQUIRED, and UNIQUE. The table lists the following properties:

NAME	TYPE	KEY	REQUIRED	UNIQUE
ID	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"/>

At the bottom right, there is a zoom level indicator set to 125%.

Review

In this step, you created the Leave Request (SmartBox) SmartObject. This SmartObject stores the leave request data entered by our users. In the next step, you will create the Leave Request Item View. This is the user interface for entering leave request details.

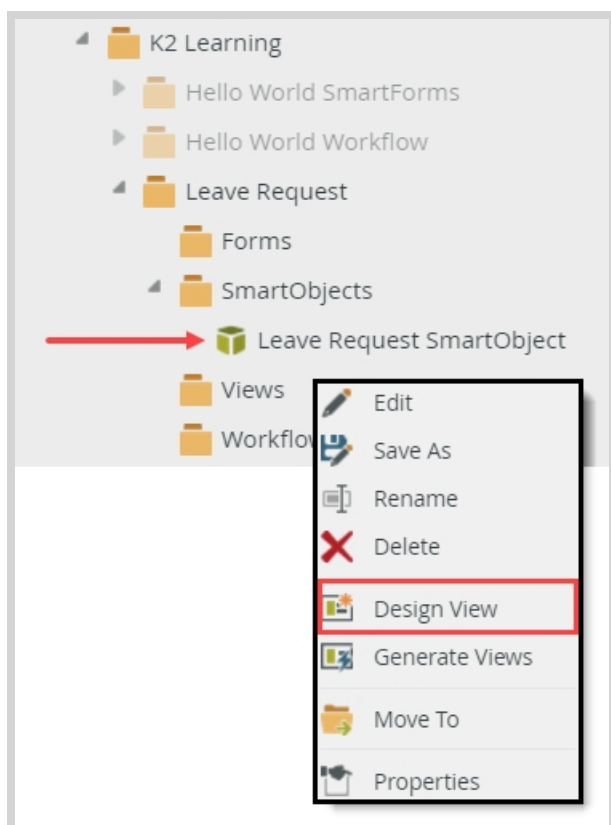
Next Step: 3. Build the Leave Request Item View

3. Build the Leave Request Item View

In this step you will design an item view from the Leave Request SmartObject. (Item views contain content from a single record, whereas list views contains a list of many records.)

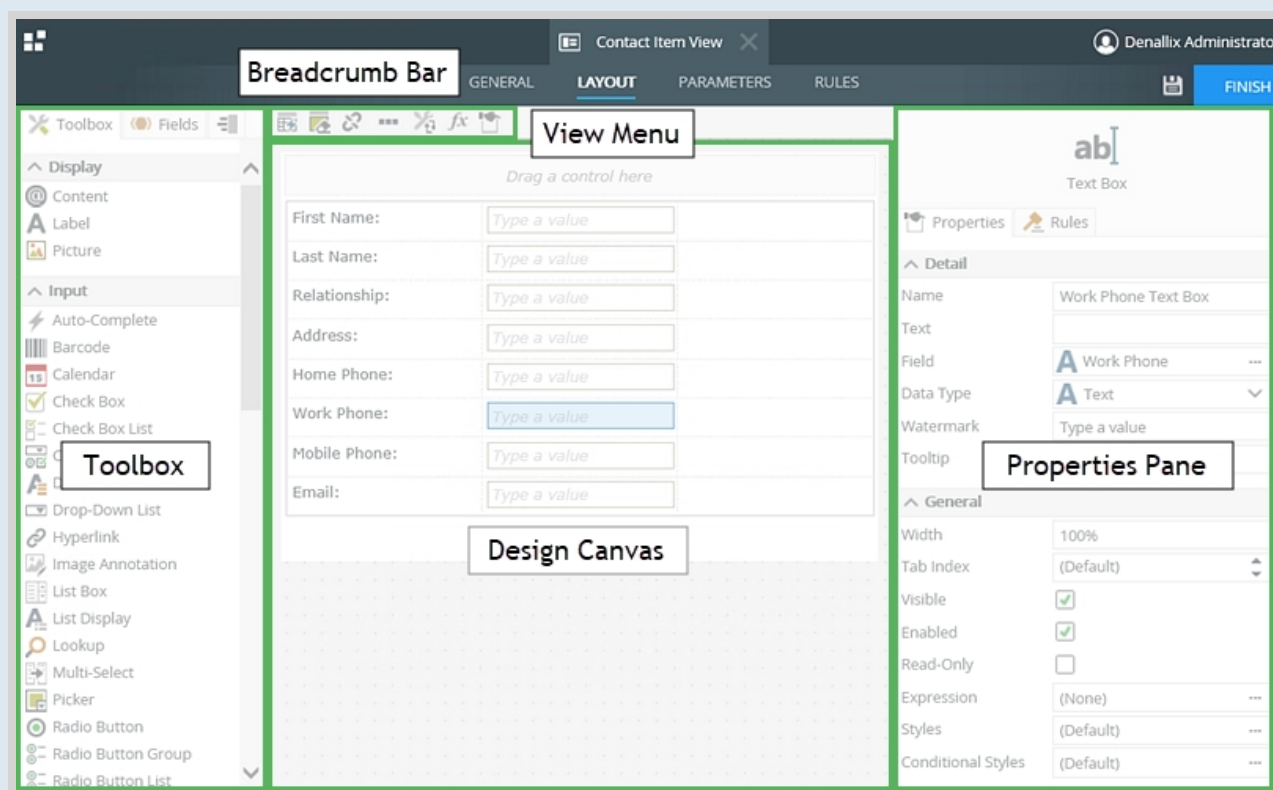
Note
Designing the view directly from the Leave Request SmartObject automatically "binds" the SmartObject to the view. In turn, K2 generates the SmartObject properties as view controls (Employee Name, Employee Email, etc.) that are already set up to "connect" with their associated SmartObject property.

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. This view is the user interface for entering leave request details.
 - a. Begin by designing a view from the Leave Request SmartObject. Right-click the **Leave Request SmartObject** (located in the SmartObjects category) and select **Design View**.



Note
K2 Designer is the tool used to build and edit SmartObjects, views, forms, and workflows. You can add controls from the Toolbox, edit the view and form layout from the View Menu, and edit control properties from the Properties pane. The Design Canvas provides a visual representation of your view or form as you go.

The K2 Designer layout and terminology



- **Breadcrumb Bar:** use the breadcrumb bar to switch designer screens
 - **General:** view details, such as name, SmartObject, view type
 - **Layout:** design canvas, view menu, toolbox, properties
 - **Parameters:** add and configure view parameters, assign default values
 - **Rules:** add new rules, edit existing rules
- **View Menu:** change control types, add functions, clear canvas
- **Toolbox:** add or change controls and methods
 - **Toolbox:** add controls directly to the design canvas
 - **Fields:** add SmartObject fields from the bound SmartObject
 - **Methods:** add methods in button format to the design canvas
- **Design Canvas:** drag and drop controls, fields and methods to create a reusable view
- **Properties Pane:** edit and configure control properties
 - **Rules** (separate tab): add, edit and configure a control's rules

b. Next, you will name the view and move it to the Views category. Because you designed the view from the Leave Request SmartObject (in the SmartObjects category), K2 created the view in the SmartObject category. To keep organized, you will move it to the Views category.

The designer opens on the **General** settings screen. Name the new view

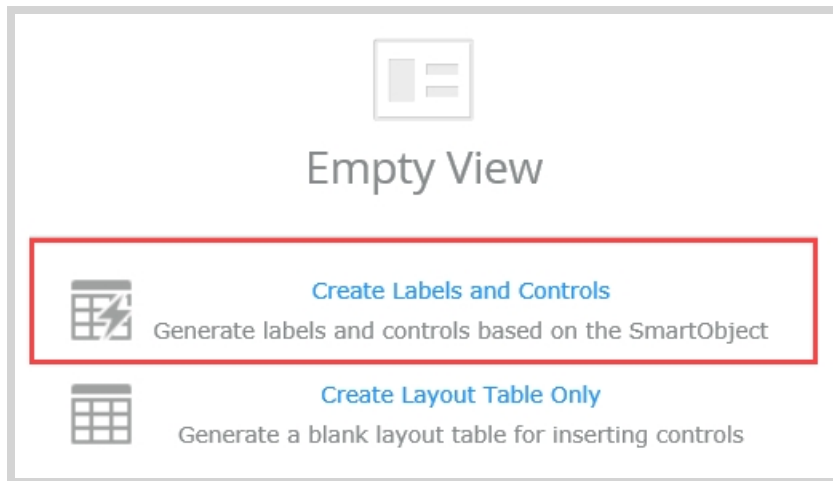
Leave Request Item View

then change the category to the **Views** category. To change the category, click the ellipses on the right side of the **Category** field, then select **Views**. Click **OK**. Confirm the **Item View** option is the view type. Click **CREATE**. (You

do not need to change the data source since you designed the item view from the SmartObject).

The screenshot shows the K2 configuration interface for a 'Leave Request Item View'. The 'GENERAL' tab is selected. The 'Name' field is 'Leave Request Item View'. The 'Description' field is empty. The 'Category' field is 'K2 Learning\Leave Request\SmartObjects'. The 'View Type' field has two options: 'Item View' (selected) and 'List View'. The 'Data Source' field is 'K2 Learning\Leave Request\SmartObjects'. A 'CREATE' button is highlighted. A 'Select a category' dialog box is open, showing a tree structure with 'Views' selected under 'Leave Request'.

2. Create the labels and controls you will use for the Leave Request user interface. Click **Create Labels and Controls** and include all the fields except the ID. Make the **Employee Email** field **Display Only**. Confirm or set two columns for the layout table. Change the **Label** position to **Left** and add the **Colon** suffix. CHECK the option for the **Create** button (only).
 - a. Next, you can choose whether to have K2 automatically generate a view for you based on the controls you select, or you can add an empty layout table and add the controls you need manually. In this application, you will let K2 generate the layout table and controls for you. On the **Layout** screen, click **Create Labels and Controls**.



- b. On the **Create Labels and Controls** screen, make the following configurations (see the screenshot below for reference)
- 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 the fields, except the **ID**.
 - This is a hidden field that will contain the 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**.
 - You do not want the user 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.
 - You are allowing K2 to perform some of the legwork. You can also add a button control to your view and configure it to create a new record in the SmartObject. In this step, K2 will generate the create button and add the necessary rules to create the record.

Click **OK** to continue.

Create Labels and Controls

Table

To hold your controls and

Columns:

Layout

Select or clear the checkboxes you want to show or hide in the Item View. A checkbox must be selected for each field.

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"/>

Labels

Choose how you want the labels to appear in the Item View.

Label Position


Colon suffix (":")

☒

Buttons

Choose how you want the buttons to appear in the Item View.

	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"/>



OK

CANCEL

Your generated view should look like the image below.

The screenshot shows the 'Leave Request Item View' in the K2 application, specifically in the 'LAYOUT' tab. The form is organized into a table-like structure with the following fields and controls:








Drag a control here	
Leave Request Title:	<input type="text" value="Type a value"/>
Employee Name:	<input type="text" value="Type a value"/>
Employee Email:	[Employee Email Data Label]
Leave Start Date:	<input type="text" value="Select a date"/>
Leave End Date:	<input type="text" value="Select a date"/>
Leave Type:	<input type="text" value="Type a value"/>
Requester Comments:	<input type="text" value="Type a value"/>
Request Status:	<input type="text" value="Type a value"/>
<input type="button" value="Create"/>	

3. To improve the layout of the view, you will make minor adjustments to the layout table and two controls. Change the width of the first column so that it is in closer alignment to the view labels. Change the **Width** of the **Leave Start Date** and **Leave End Date** controls to **50%**.
 - a. Adjust the view for better layout. First, reduce the width of the label column. Hover over the **center column line** until you see the double lines. Click+drag the column line so that it is closer to your labels.

Leave Request Title:	<input type="text" value="Type a value"/>
Employee Name:	<input type="text" value="Type a value"/>
Employee Email:	[Employee Email Data Label]
Leave Start Date:	<input type="text" value="Select a date"/>

Leave Request Title:	<input type="text" value="Type a value"/>
Employee Name:	<input type="text" value="Type a value"/>
Employee Email:	[Employee Email Data Label]
Leave Start Date:	<input type="text" value="Select a date"/>

- b. Next, adjust the width of the two calendar controls. Click to highlight the **Leave Start Date Calendar** control. In the **Properties** pane, change the **Width** to **50%** then repeat this step for the **Leave End Date Calendar** control.

Drag a control here

Leave Request Title:	<input type="text" value="Type a value"/>
Employee Name:	<input type="text" value="Type a value"/>
Employee Email:	[Employee Email Data Label]
Leave Start Date:	<input type="text" value="Select a date"/>
Leave End Date:	<input type="text" value="Select a date"/>
Leave Type:	<input type="text" value="Type a value"/>
Requester Comments:	<input type="text" value="Type a value"/>
Request Status:	<input type="text" value="Type a value"/>
<input type="button" value="Create"/>	

15

Calendar

Properties

Rules

Watermark

Select a date

Tooltip

Settings

Picker Type

Date

General

Width

50%

Tab Index

(Default)

Visible

☒

Enabled

☒

Read-Only

☐

Expression

(None)

Styles

(Custom)

4. To keep the leave type values consistent across records, change the **Leave Type Text Box** control to a **Drop-Down List** and add the following entries (for both **Value** and **Display**):

Paid Time Off

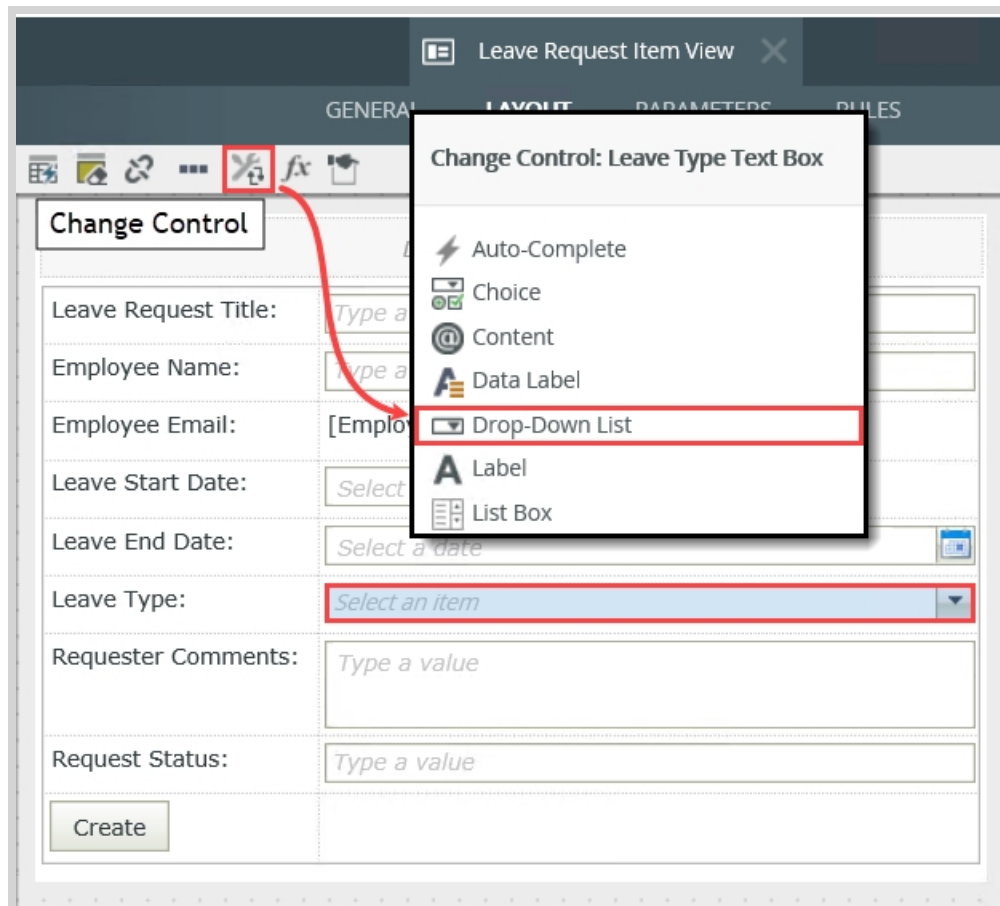
Study Leave

Family Responsibility

then change the **Width** of the **Leave Type Drop-Down List** to *50%*. Once again, changing the width makes accessing the drop-down control more convenient.

- a. To keep the leave type values consistent across records, you will change the Leave Type Text Box control to a drop-down list and add static list item values. In this way, you must select a leave type, rather than entering your own, which may or may not match your organization's recognized leave types. At some point, you may also want to run reports based on leave types. Using consistent values will help to ensure reports include all records.

Highlight the **Leave Type Text Box** control. Select the **Change Control** icon found in the View Canvas menu. Change the control to a **Drop-Down List**. Click OK.



- b. With the Leave Type Drop-Down List control still highlighted, open the **Data Source > Type** editor found in the **Properties** pane. Select the option to **Use a static list of values in the control**. Open the list editor and **Add** the following options for *both* the **Value** and the **Display**, then click OK. Click OK once again to complete the configuration.
 - *Paid Time Off*
 - *Study Leave*
 - *Family Responsibility*

Drag a control here

Leave Request Title:

Employee Name:

Employee Email:

Leave Start Date:

Leave End Date:

Leave Type:

Requester Comments:

Request Status:

Create

Drop-Down List

Properties Rules

Watermark Select an item

Tooltip

Settings

Allow Empty Selecti... ☒

Data Source

Type (None)

Configure Data Source

☒ Use a static list of values in the control

List Items:

Fixed List Configuration




+ Add Edit Delete

VALUE	DISPLAY	DEFAULT
Family	Family	<input type="checkbox"/>
Friend	Friend	<input type="checkbox"/>
Co-Worker	Co-Worker	<input type="checkbox"/>
Business Associate	Business Associate	<input type="checkbox"/>

(Add fixed list values)

- c. Change the **Width** property of the Leave Type Drop-Down List control to 50%.

5. To improve the layout of the view even more, move the **Create** button to the second cell and align it to the right.
The Leave Request Item View

Drag a control here	
Leave Request Title:	<input type="text" value="Type a value"/>
Employee Name:	<input type="text" value="Type a value"/>
Employee Email:	[Employee Email Data Label]
Leave Start Date:	<input type="text" value="Select a date"/> 
Leave End Date:	<input type="text" value="Select a date"/> 
Leave Type:	<input type="text" value="Select an item"/> 
Requester Comment..	<input type="text" value="Type a value"/>
Request Status:	<input type="text" value="Type a value"/>
	<input type="button" value="Create"/>

- a. There is one more change for the layout of the view. You will move the Create button to the right side of the layout table. The create button (or submit button) is the last control you interact with and you want its location to be easy to find. There are a number of studies that indicate on a two-column form, the best location is the lower right side of the screen.
- Click, then drag, the **Create** button into the second cell of the row. Click within the cell itself to highlight it, then click the **Right-Align** icon found in the **View Canvas** menu. If you do not see the align icons, make your screen larger to expose the icons.

Drag a control here

Right Align

Leave Request Title:

Employee Name:

Employee Email: [Employee Email Data Label]

Leave Start Date:

Leave End Date:

Leave Type:

Requester Comments:

Request Status:

Your view should look like the image below.

Drag a control here

Leave Request Title:

Employee Name:

Employee Email: [Employee Email Data Label]

Leave Start Date:

Leave End Date:

Leave Type:

Requester Comment:

Request Status:

- b. Click **FINISH** to save and exit the view. The Finish button is in the upper right corner of your screen.

Review

In this step, you created an item view from the Leave Request SmartObject. You selected which properties to use as form fields, and adjusted the layout of the view to work well for our users. You also learned some

basics about using the properties pane to configure controls.

In the next step, we will build the List view that will display a list of leave requests.

Next Step: 4. Build the Leave Request List View

4. Build the Leave Request List View

In this step you will create a read-only list view that displays the leave requestor's records. You will use this list view in two ways:

- The list view appears on the leave request form and displays the current user's previous requests.
- The list view appears on the approving manager's form and displays the leave requester's previous requests.

Later, you will learn that the two forms (requester and approving manager) are actually the same form, just configured to display the previous requests based on the form's use.

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. In a later step, you will configure a method action to call *specific* records when this view loads.
 - a. Once again, you will design this view from the Leave Request SmartObject, creating a bind between the view and the data source. In this step, you will create a list view that returns multiple records (the requester's previous leave requests). (Recall that the item view created in the previous step returns a single record.).
Right-click the **Leave Request SmartObject** and select **Design View**.
 - 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**. (By default for a list view, K2 assumes you want to display all the records from the SmartObject when the view loads. In this case, you only want the records for the leave requester, not all the records to appear. In a later step, you will configure a rule that retrieves only the leave requester's records.) Click

CREATE.

Leave Request List View

GENERAL LAYOUT PARAMETERS RULES

Name: * Leave Request List View

Description: Describe what is in this View

Category: * K2 Learning\Leave Request\Views ...

View Type:

- Item View Used to capture information.
- List View Used to display lists of information.

Data Source: * K2 Learning\Leave Request\SmartObjects

List method: * Get List

- ☒ Call this method when the form loads

CREATE DISCARD

Select a category

- All Items
- Active Directory
- eric
- K2 Learning
 - Leave Request
 - Forms
 - SmartObjects
 - Views
 - Workflow

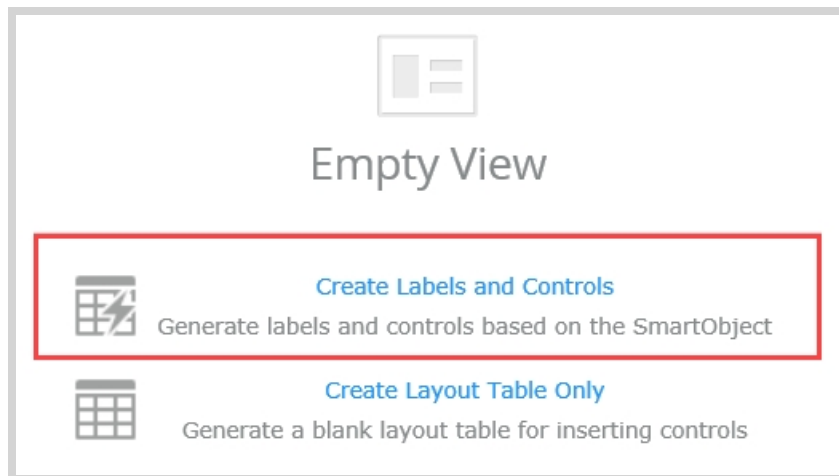
2. Create the labels and controls, then add the following fields:

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

The Leave Request List View

Drag a control here					
Leave Request Title	Employee Name	Leave Start Date	Leave End Date	Leave Type	Request Status
Leave Request Title	Employee Name	6/6/2018	6/6/2018	Leave Type	Request Status
Leave Request Title	Employee Name	6/6/2018	6/6/2018	Leave Type	Request Status
Leave Request Title	Employee Name	6/6/2018	6/6/2018	Leave Type	Request Status
Leave Request Title	Employee Name	6/6/2018	6/6/2018	Leave Type	Request Status
Leave Request Title	Employee Name	6/6/2018	6/6/2018	Leave Type	Request Status
Configure an aggregation or drag a control here					

a. Select **Create Labels and Controls**.



b. On the Create Labels and Controls screen, **Include** the following fields:

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

- c. To avoid any changes to the records, keep the **Enable list editing** option disabled. There are no other changes to the layout editor, so click **OK**.

Create Labels and Controls

Layout
Select or clear the checkboxes to show or hide in this View. All fields must be selected.

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"/>

Edit Options
Choose whether the list items can be edited. This will add menu buttons to existing rows, or delete rows.

☐ **Enable list editing**

☒ Edit all rows
☐ Edit single rows

☐ Allow the user to add new rows Create ▼ method

☐ Allow the user to edit existing rows Save ▼ method

☐ Allow the user to remove rows Delete ▼ method

☒ Enable Add new row link

Additional:
Additional options to add to the list view.

☐ Allow the user to manually refresh list

OK **CANCEL**

- d. Click **FINISH** to save and exit the view.

Review

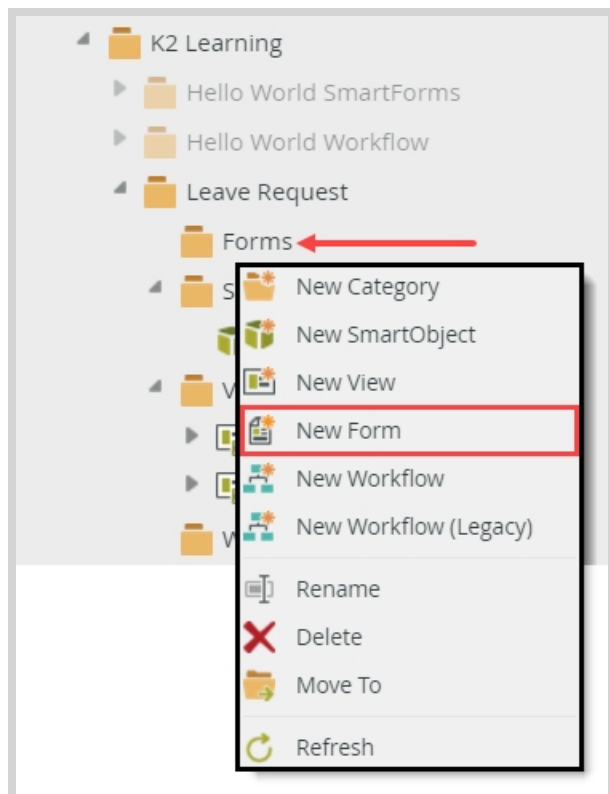
In this step, you created a list view from the Leave Request SmartObject. This view is read-only since you do not want the approver to make any changes to the request details. In the next step, you will create the Leave Request Form.

Next Step: 5. Build the Leave Request Form

5. Build the Leave Request Form

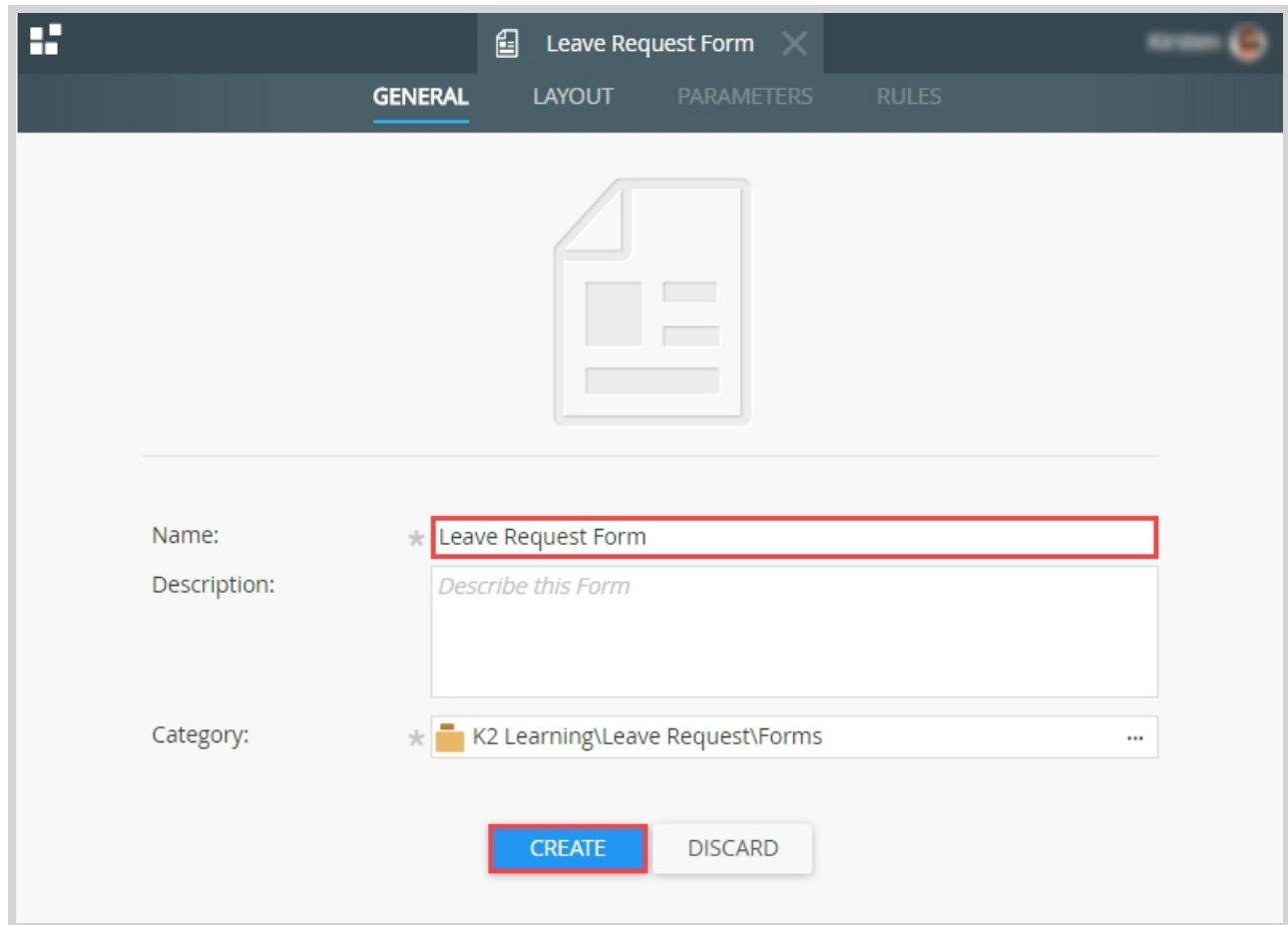
In this step, you will create the Leave Request Form, which will be the user interface for entering and approving leave requests. Users typically interact with views through forms, so you will create the form, then add the item view and the list view that you created in the previous two steps.

1. Create a **New Form** from the **Forms** category. Name the form *Leave Request Form*.
 - a. In this step, you will create the Leave Request Form from the Forms category. Because the form itself is not bound to the Leave Request SmartObject, you can create it from its category location. Right-click the **Forms** category and select **New Form**.



- b. On the **GENERAL** settings screen, **Name** the new form *Leave Request Form*

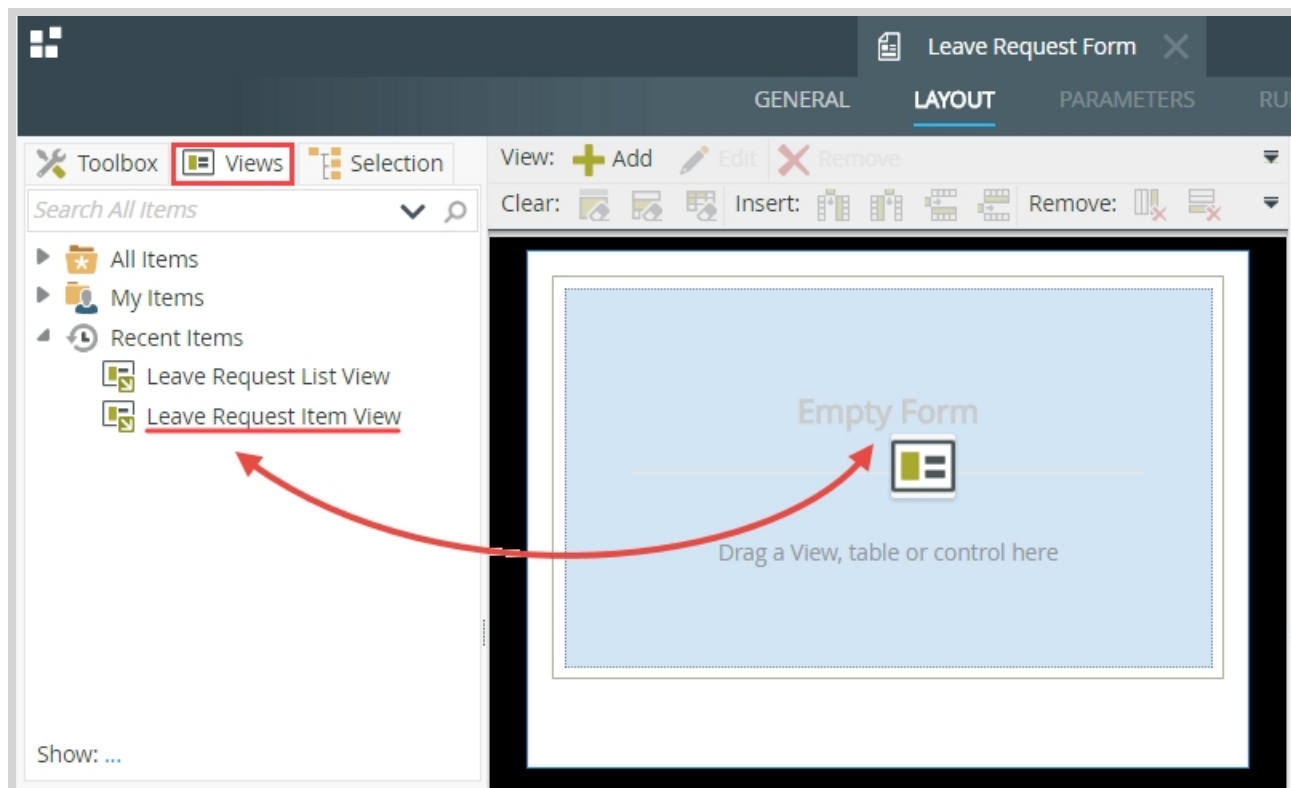
then click **CREATE**.



The screenshot shows the K2 'Leave Request Form' creation interface. The 'GENERAL' tab is selected. The 'Name' field is 'Leave Request Form', the 'Description' is 'Describe this Form', and the 'Category' is 'K2 Learning\Leave Request\Forms'. The 'CREATE' button is highlighted with a red box.

2. Add the **Leave Request Item View** and change the view title to *Leave Request*.

- a. Now that you created the form, the next steps are to add the two views. You will add the Leave Request Item View to the top of the form and the Leave Request List View below. On the left side of the screen, click the **Views** tab. Expand the **Recent Items** node if it isn't already. Drag the **Leave Request Item View** onto the form design canvas. Depending on the number of recent items in your environment, you may have to scroll to locate the item view. (You can also add the view from the **Views** category.)



- b. Click the **(Enter View Title)** view title and change it to *Leave Request* then click outside of the view to set the value. This lets you name the views within the forms to user friendly names.

3. Add the **Leave Request List View** below the item view and change the view title to *Previous Leave Requests*. The list view displays a read-only list of requests. The list view displays the leave requests from the person requesting leave.

Leave Request

Leave Request Title:

Type a value

Employee Name:

Type a value

Employee Email:

[Employee Email Data Label]

Leave Start Date:

Leave End Date:

Leave Type:

Type a value

Requester Comments:

Request Status:

Type a value

Create

Previous Leave Requests

LEAVE REQUEST TITLE

EMPLOYEE NAME

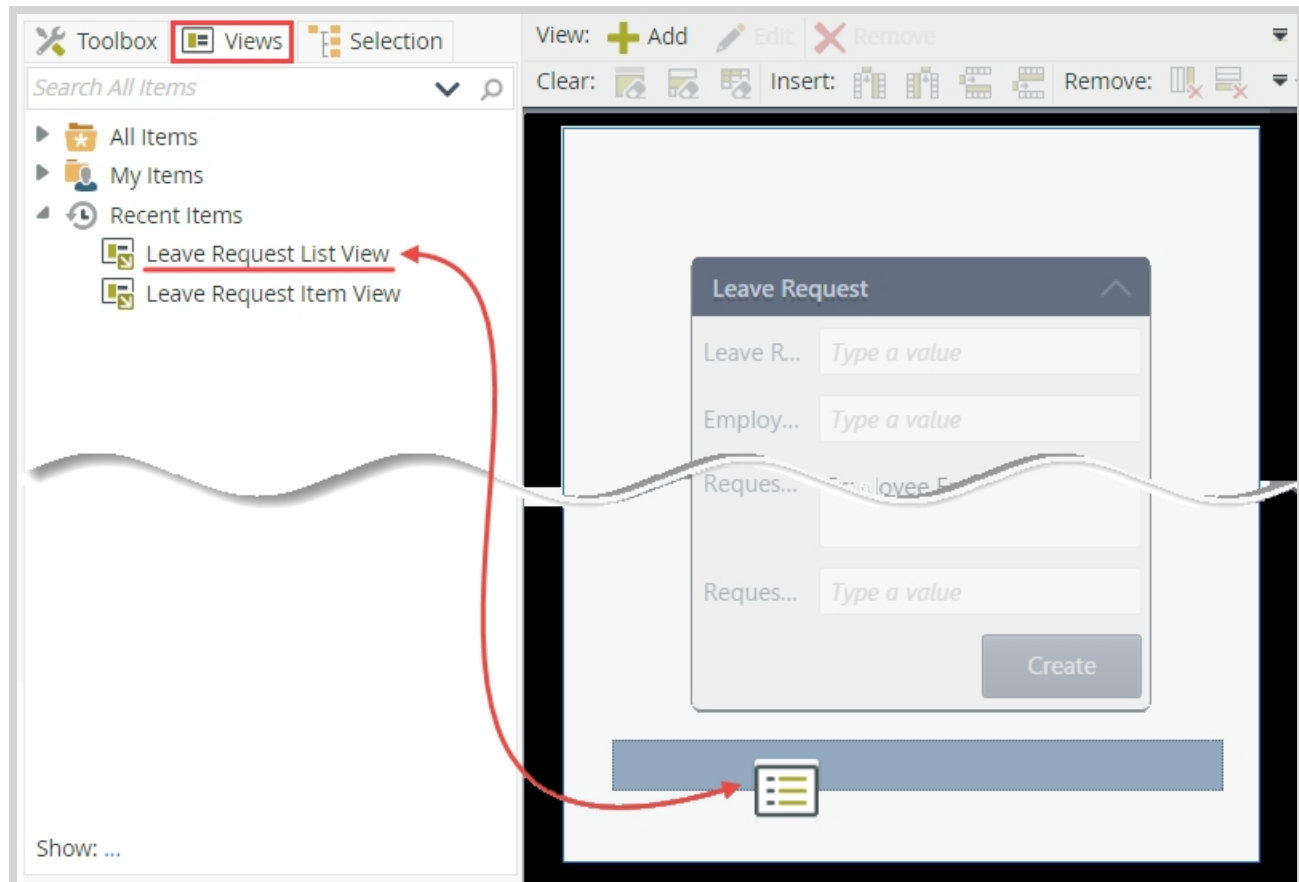
LEAVE START DATE

LEAVE END DATE

LEAVE TYPE

No items to display.

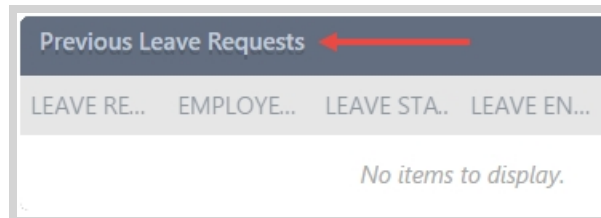
- Page 34 of 112



- b. Click the **(Enter View Title)** view title and change it to

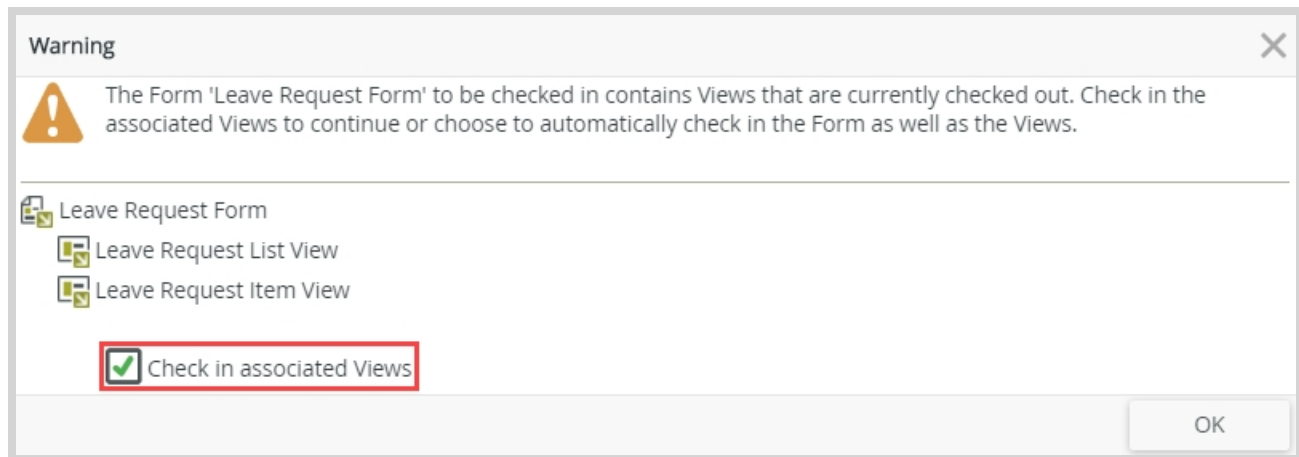
Previous Leave Requests

(Depending on who is viewing the form (the requester or the approving manager), this list will either display the current user's previous requests, or the requester's previous requests. You will configure a method action later to populate the list, depending on its use.)



- c. Click **FINISH** to save and exit the form.
- d. In the explorer, right-click the **Leave Request Form** and select **Check In**. You will see a warning about associated views. CHECK the box to **Check in associated Views**, then click **OK**.
Views and forms are not available for live use until you check them in. Checking views and forms in publishes them to the K2 server and makes them available for users. To change a view or form, you

must first check it out, make the change, then check it back in.



Review

In this step, you created the Leave Request Form, then added the Leave Request Item View and Leave Request List View to the form. You checked-in the form and view, which published them to the K2 server and made them available for users to use. In the next step, you will create the Leave Request Workflow.

Next Step: 6. Create the Leave Request Workflow

6. Create the Leave Request Workflow

In this step you will create the Leave Request Workflow. You will create a new workflow, then add steps and outcomes (lines) to automate the leave request approval process. You will add a (user) Task step and assign it to the requester's manager. The manager must decide to either approve or deny the leave request. You will add several system tasks that update the status property as the workflow moves through the steps.

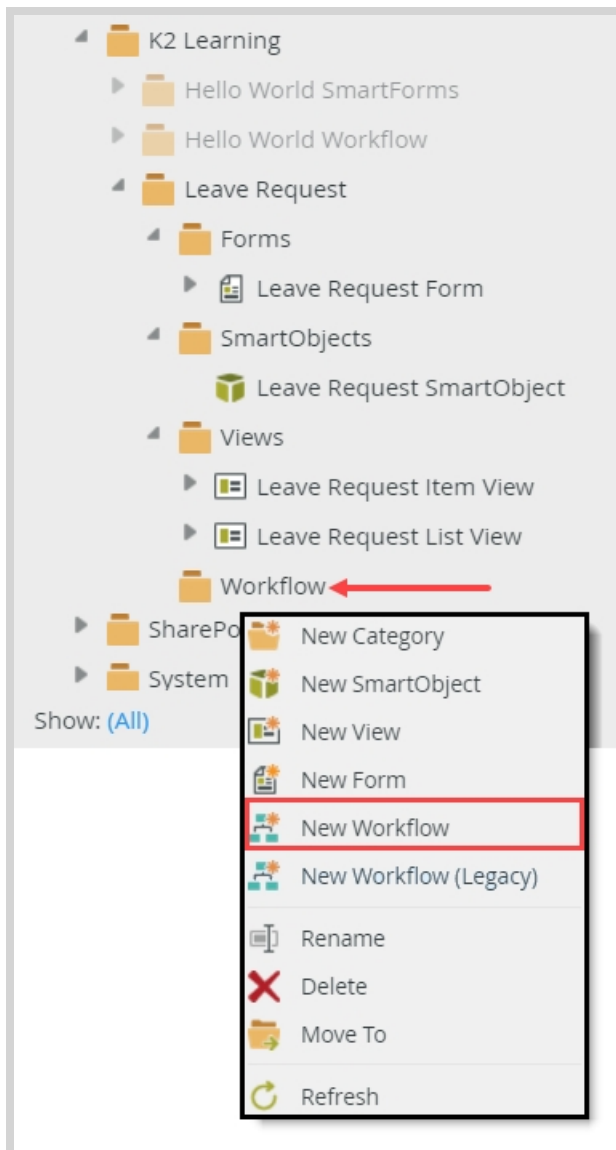
1. Create a new workflow from the **Workflow** category and **Name** it *Leave Request Workflow*.

Note

If you are working in a shared environment, consider adding your initials to the workflow name. Workflow names must be unique.

- a. Similar to creating the Leave Request Form, you will create the Leave Request Workflow directly from its category. K2 saves the workflow in the appropriate workflow category.

Right-click the **Workflow** category and select **New Workflow**. Do not select the New Workflow (Legacy) option.



- b. On the workflow designer landing page, **Name** the workflow *Leave Request Workflow* then click **CREATE**.


Note

If you are working in a shared environment, consider adding your initials to the workflow name. Workflow names must be unique.

The folder category selected is correct. You can change folders if needed by clicking the link next to the folder icon.

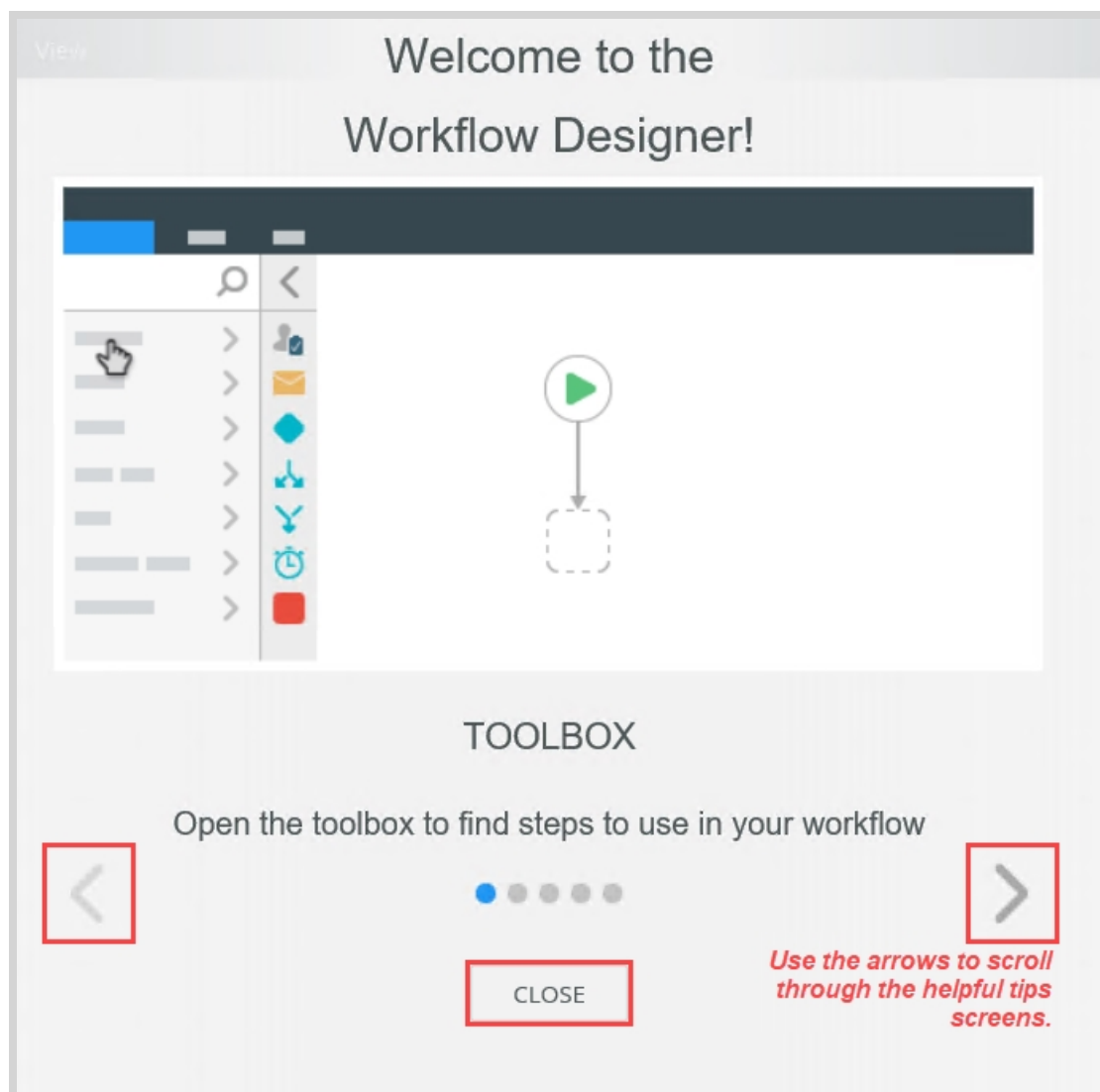
What should your workflow be called?

Leave Request Workflow

Folder  [K2 Learning/Leave Request/Workflow](#)

CREATE

2. The K2 Workflow Designer launches with a helpful tips screen. If you like, step through the helpful tips to learn more about the workflow designer. Otherwise, click CLOSE.



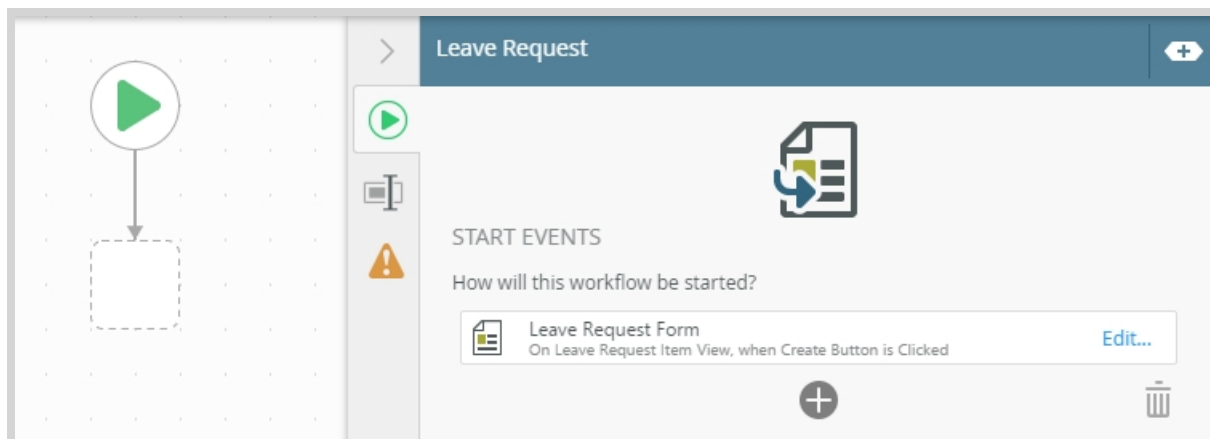
3. Begin by configuring the **Start** step. Here, you will associate the Leave Request Form as the form that starts the workflow. You will create a new State, then confirm that the start rule specifies the workflow starts when you click the Create button. Expand the **Configuration Panel** for the Start step. Add the **Leave Request Form** (SmartForm) as the form that starts the workflow. Complete the wizard configurations using the values in the table below. If a value is not specified, assume the default value.

Wizard Screen	Value
REFERENCES	Leave Request SmartObject
START RULE	Help me configure it
FORM STATE	Create a new State: Leave Request Workflow
FORM STATE	Make this state the default for the form: CHECKED

HOW DOES YOUR FORM START THE WORKFLOW?

On Leave Request Item View, when Create Button is Clicked

The Leave Request Workflow with SmartForm Association



In this step, you will configure the workflow Start button. You will associate the Leave Request Form with the workflow. Once K2 recognizes the workflow's SmartForm, you can create a new state on the form (you will learn about states later), then configure the rule on the form that tells K2 to fire the workflow when you click the Create button. Before you start the next steps, take a minute to learn about the basic layout of the K2 Workflow Designer.

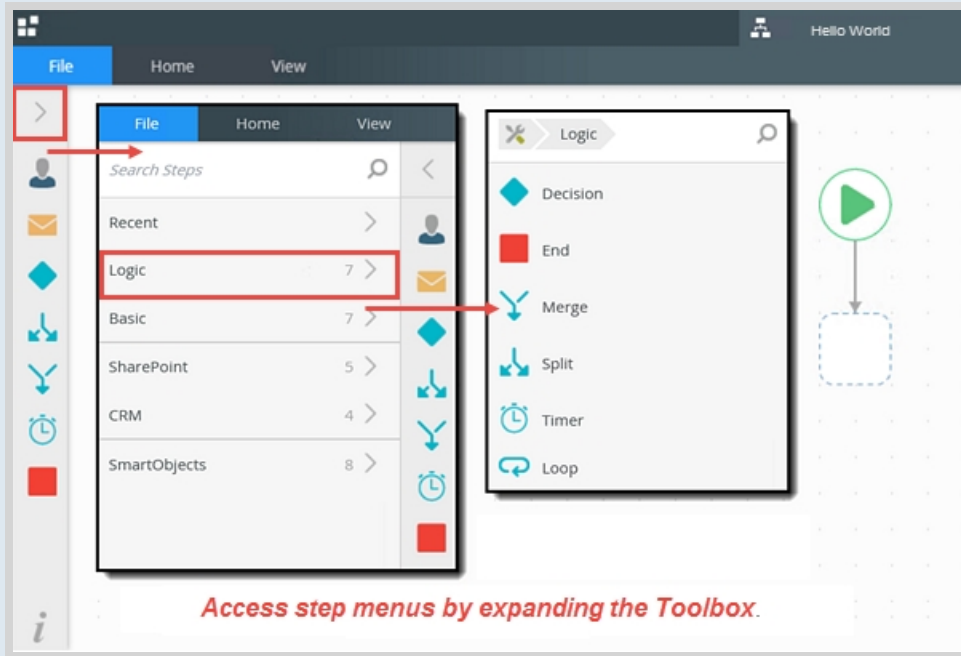
Note

The **K2 Workflow Designer** environment contains four primary sections:

- the **Toolbox** on the left-side of the page
- the **Design Canvas** in the center of the page
- the **Configuration Panel** on the right-side of the page
- the **Context Browser** located within the Configuration Panel

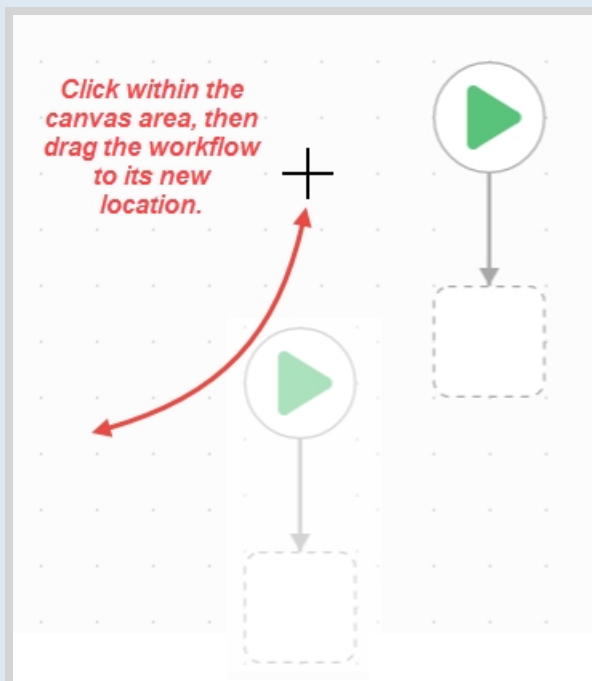
The **Toolbox** is where you will find the different types of steps that you can add to a workflow. You can drag and drop the steps you need for your workflow onto the canvas. User steps usually involve human interaction, such as a person having to make a decision for the workflow to continue. System steps, such as sending an email or updating data in some other system, are performed by the K2 server. Lines connect steps to each other, and define the "path" that the workflow will follow.

Accessing steps in the Toolbox



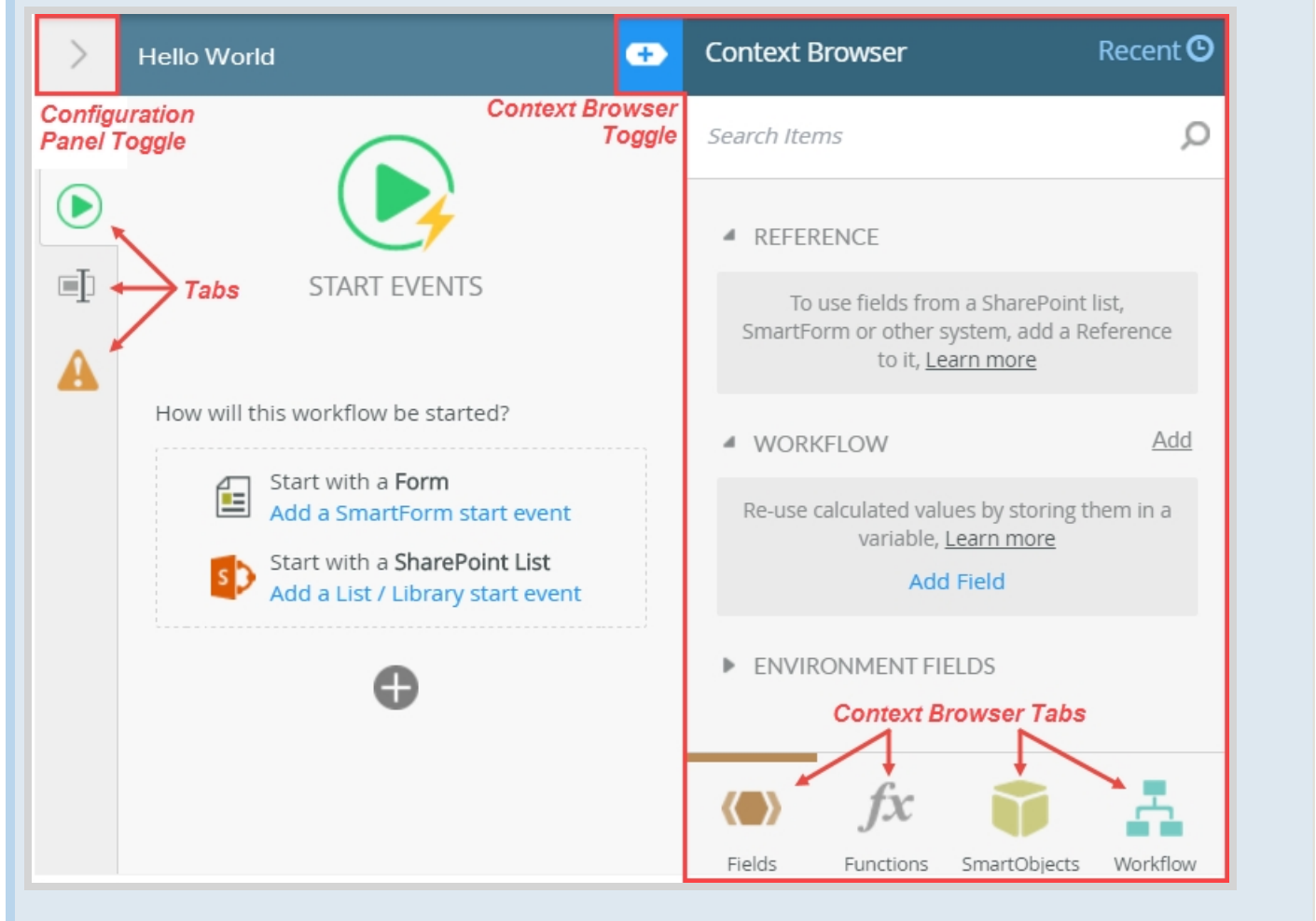
The **Design Canvas** is where you drag and drop the steps you need for constructing your workflow. You can move the workflow around by clicking within the canvas, then dragging your mouse to a new location. This is helpful if your workflow is larger than your browser screen.

Moving the workflow around the design canvas



The **Configuration Panel** contains the settings for each step. After highlighting a step on the canvas and expanding the Configuration Panel, the Configuration Panel will display a series of tabs. The tabs are specific to the step type. The Configuration Panel also contains the **Context Browser**. The **Context Browser** provides access to variables, functions, SmartObjects, and workflow variables.

Accessing step tabs in the Configuration Panel; expanding the Context Browser

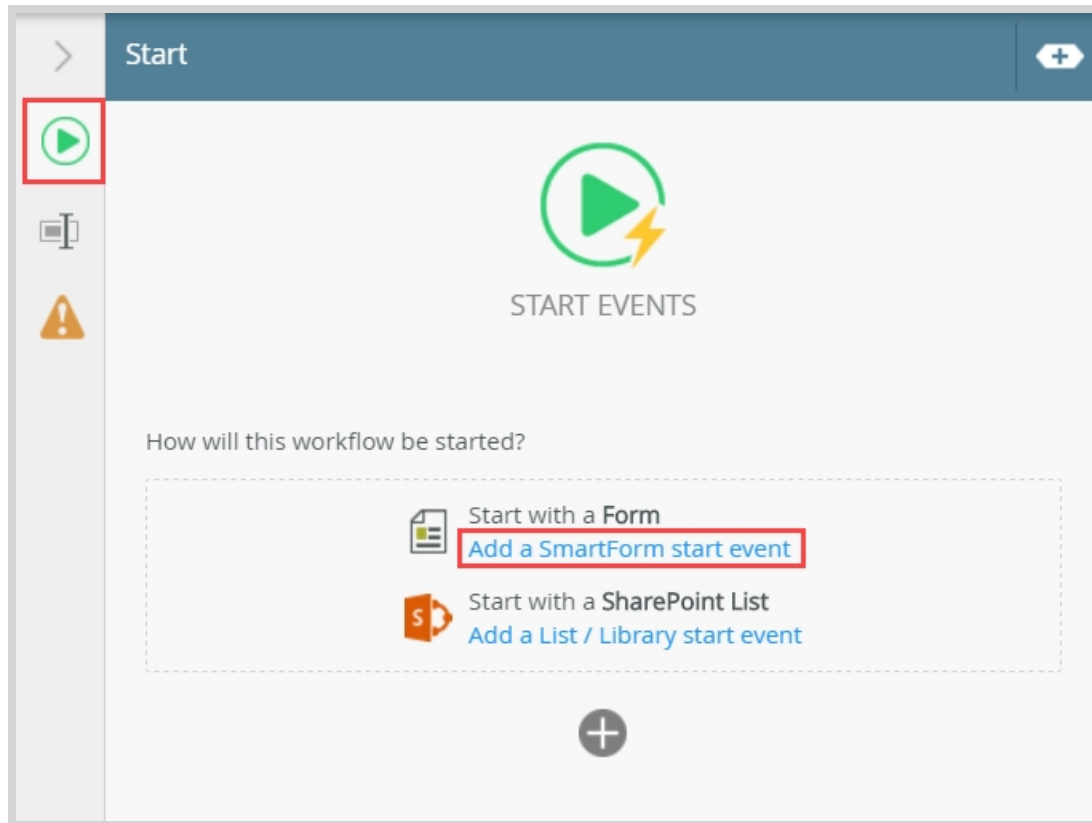


- Begin the Start step configuration. Click the green **Start** step to highlight it. When highlighted, a blue border will appear around the step. Expand the **Configuration Panel**.

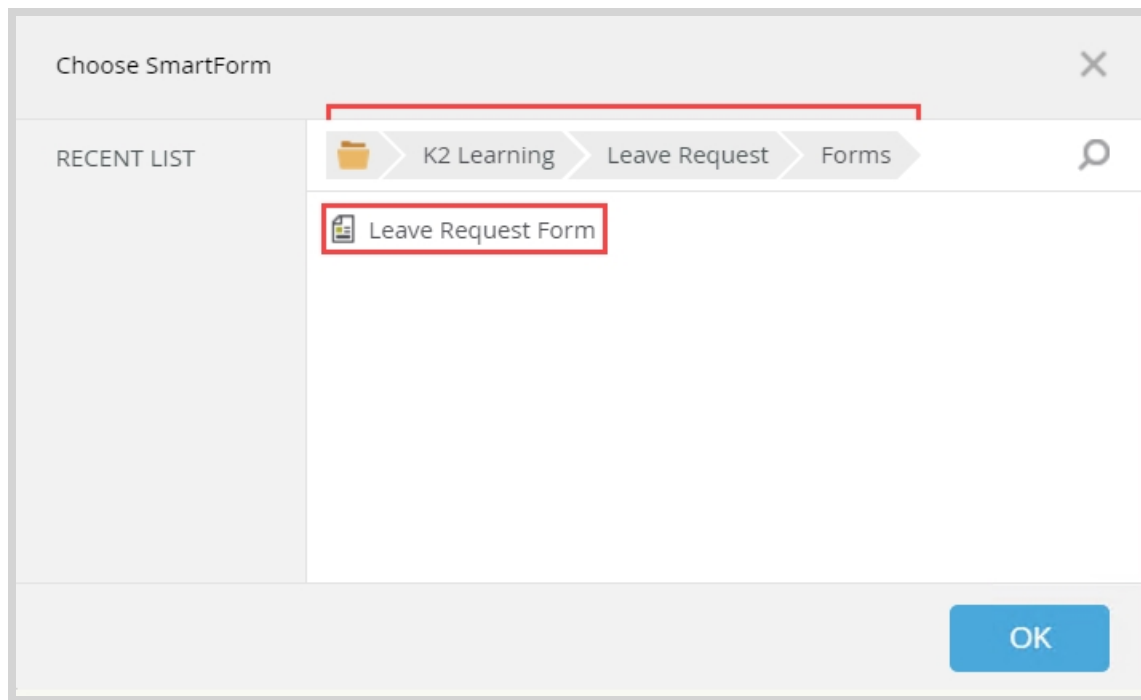


- b. Each step has associated tabs in the Configuration Panel. Clicking a tab opens the configuration settings. The Start step has three tabs: START, PROPERTIES, and ERRORS. You will begin with the START tab. The START tab is where you associate a SmartForm with a workflow. After selecting a SmartForm, the **Start Workflow from a Form** wizard launches and displays the configuration for starting the workflow. You want the workflow to start when the requester clicks the create button on the Leave Request Form. Begin your Start step configuration by clicking the **Add a SmartForm start**

event link.



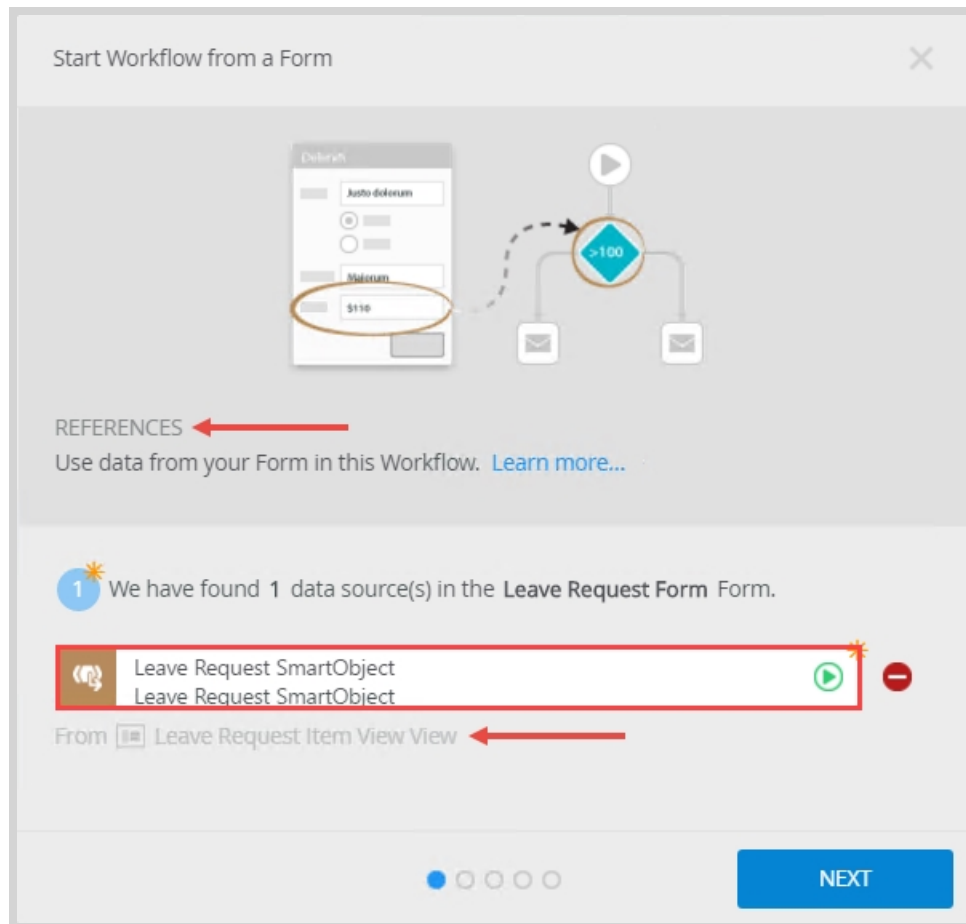
- c. Now you will select the Leave Request Form to associate the form with this workflow. On the **Choose SmartForm** screen, click **K2 Learning**, then **Leave Request**. Click **Forms**. Select the **Leave Request Form**, then click **OK**.



- d. The **Start Workflow from a Form** wizard launches. The first screen is the REFERENCES screen. K2 discovered the **Leave Request SmartObject** as the workflow data source. This is because the form contains the Leave Request Item View and the Leave Request List View. You created both views from the Leave Request SmartObject. K2 is now referencing the SmartObject in the workflow. You can now easily reference this SmartObject in other steps and rule configurations (you'll see how this works in the later steps). Click **NEXT** to continue.

Note

A *reference* is essentially an association between a workflow instance and record(s) in some data store. For example, an item added to a SharePoint list starts a workflow. K2 creates a reference between the workflow instance and the item that was created in the SharePoint List. This allows you to easily work with the list item that started the workflow.



- e. On the START RULE screen, select the option to **Help me configure it**. The wizard will walk you through the steps for configuring the start rule on the Leave Request Form. Click **NEXT** to continue.

Start Workflow from a Form

START RULE

Your **Leave Request Form** form does not seem to start this workflow yet...

☐ Do it for me!
☒ **Help me configure it**
☐ I will do it myself later

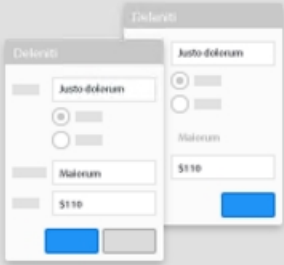
PREVIOUS NEXT


- f. On the FORM STATE screen, keep the default **Create a new State** option. Confirm the state name is **Leave Request Workflow**. Confirm the **Make this state the default for the form** option is true. (See the note box below for an explanation of states.) On this screen, you are creating the state for the form's configuration that the leave *requester* uses. Click **NEXT** to continue.

Note

A **state** represents the *configuration and behavior* of a form for a specific use. States are generally configured for a (user) Task step. For example, a request *approver's* form may have form fields disabled because as the approver, they do not need to change any form field values. A form used for *rework* will have the same fields enabled, because the workflow originator needs to update those form field values. Rather than creating two separate forms, you can use different States for each step, then use rules to control how the same form should behave in each of those two states.

Start Workflow from a Form ✕




FORM STATE  Which state will the form be in when the workflow is started?

☒ Create a new State
Recommended if this form will also be used as a Task form.

Leave Request Workflow


☐ Use an existing State

(Base State) 

[Refresh](#)

☒ Make this state the default for the form

PREVIOUS





NEXT

- g. The next screen is HOW DOES YOUR FORM START THE WORKFLOW? This is where you specify the action that starts the workflow. In this application, clicking the **Create** button starts the workflow. Confirm the FORM RULES drop-down list displays the create button clicked rule. If it does not, **change the start rule** to the create button clicked rule. Confirm the ACTIONS show that first, K2 executes the

create method, then the workflow starts. Click **NEXT** to continue.

Start Workflow from a Form



HOW DOES YOUR FORM START THE WORKFLOW? 
Choose from a list of Rules found in your form.

FORM RULES

Controls on your form start a rule, which rule should start the workflow?

On Leave Request Item View, when Create Button is Clicked

Refresh


ACTIONS

We've added the **Start Workflow** action to the rule you've chosen, move it up/down to change the order.

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

then start the Leave Request Workflow workflow

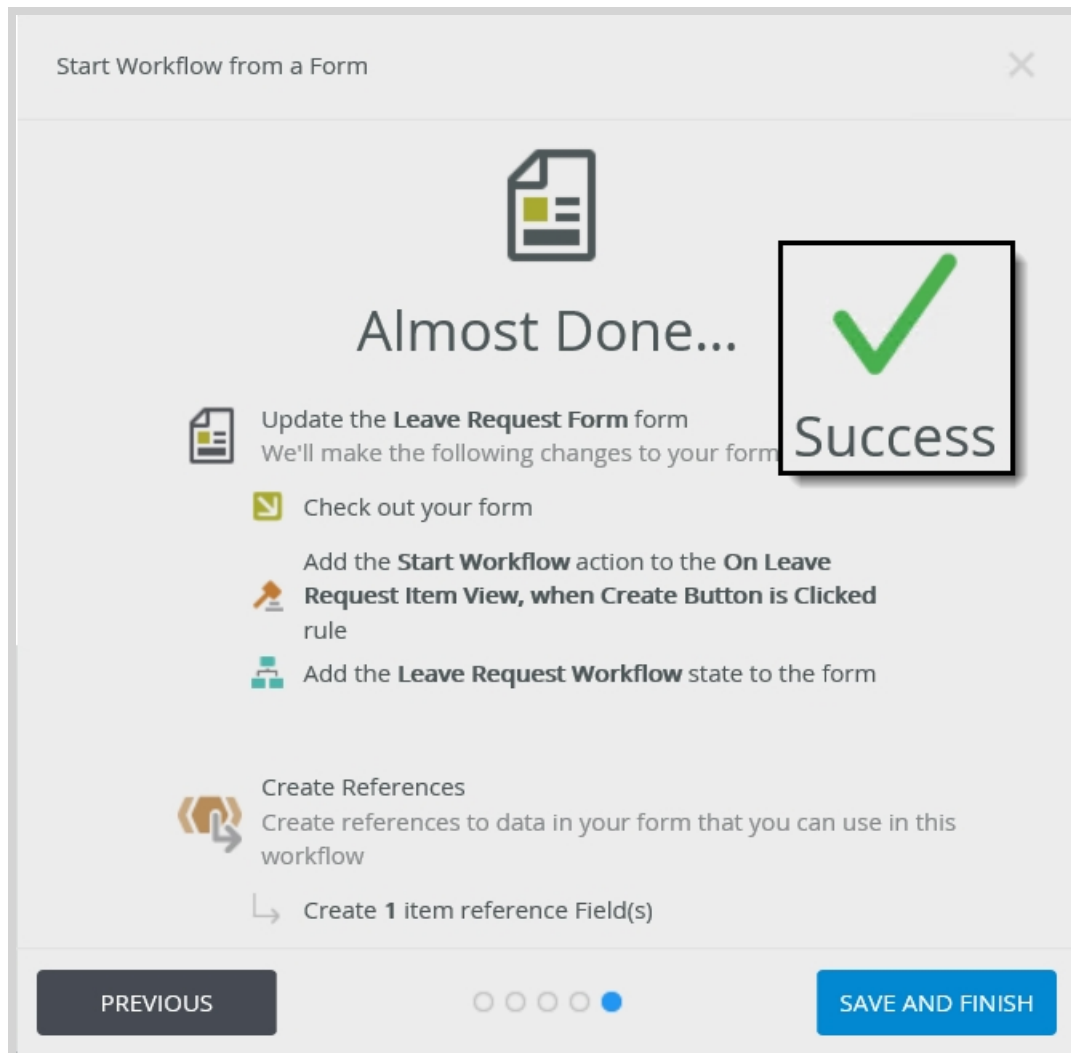
PREVIOUS



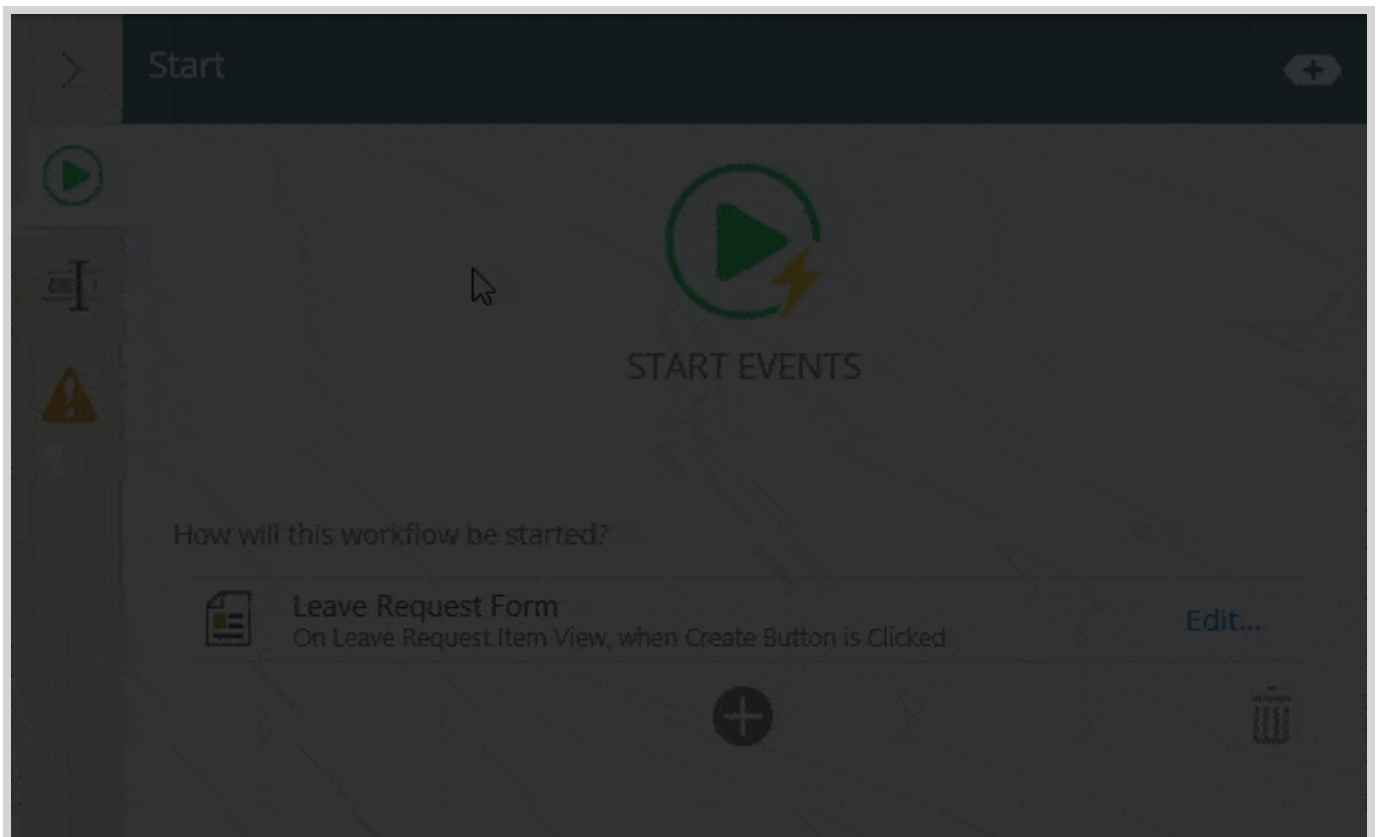
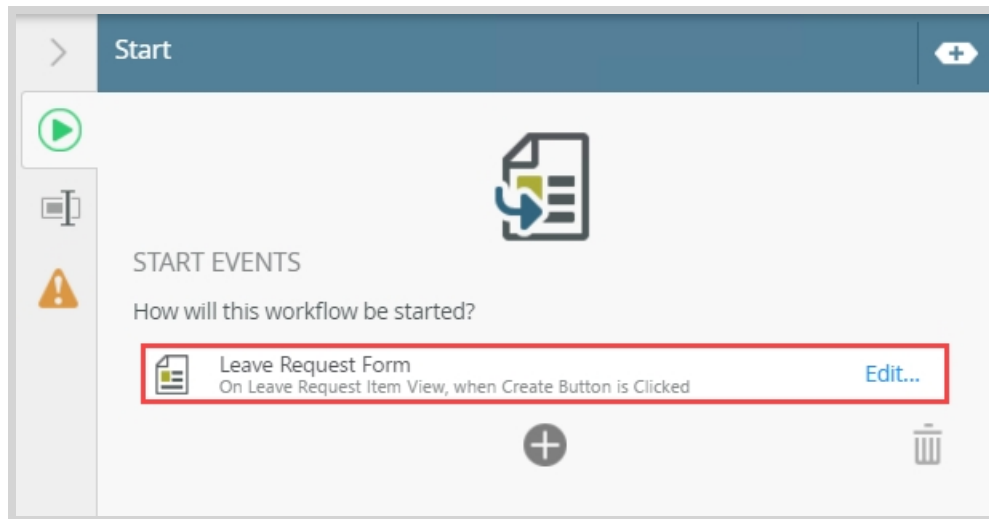
NEXT

- h. The last screen displays a summary of the actions K2 takes for configuring the START tab. This screen displays all the rule configurations that K2 performs based on the wizard settings. Review the settings, then click **SAVE AND FINISH**. When you see the **Success** check, click **OK** to close the wiz-

ard.



- i. In the Configuration Panel, notice the **Leave Request Form** starts the workflow. Collapse the **Configuration Panel**.



Review

In this step, you created the Leave Request Workflow. You selected the SmartForms option with the Leave Request Form. K2 discovered the Leave Request SmartObject and configured it as the default reference to use in the workflow. You created a new state that contains the form configuration and behavior that the *requester* sees. Last, you confirmed that clicking the create button is the action that starts the workflow. In the next step, you will add a SmartObject Method step that updates the status property. Then, you will add a Task step and configure the approval task (approve or deny) for the approving manager.

Next Step: 7. Add a SmartObject Method Step and a Task Step

7. Add a SmartObject Method Step and a Task Step

In this step you will add a SmartObject Method step to update the Request Status property in the Leave Request SmartObject. In this application, you will add several SmartObject Method steps that update the status of the leave request at different points in the workflow. When you launch the Leave Request Form, you will see this value displayed on the Previous Leave Requests list view. This lets you and the approving manager know the status of each leave request you submitted.

After the SmartObject Method step, you will add a (user) Task step. This step assigns a task (to the requester's manager) where they must review your request and either approve it or deny it.

1. Add a **SmartObject Method** step to the empty placeholder, and configure it to use the **Save** method that updates the request status property. (In K2 terms, save is the equivalent of update.) In the **Configuration Panel**, configure the step as follows:

Configuration Panel Tab	Property	Value
SmartObject Method Details	SMARTOBJECT	Leave Request SmartObject
SmartObject Method Details	METHOD	Save
SmartObject Method Details	INPUT MAPPINGS	ID = Context Browser > REFERENCES > Leave Request SmartObject > ID
SmartObject Method Details	INPUT MAPPINGS	Request Status = <i>Submitted</i>
General Properties	Name	<i>Set Status Submitted</i>

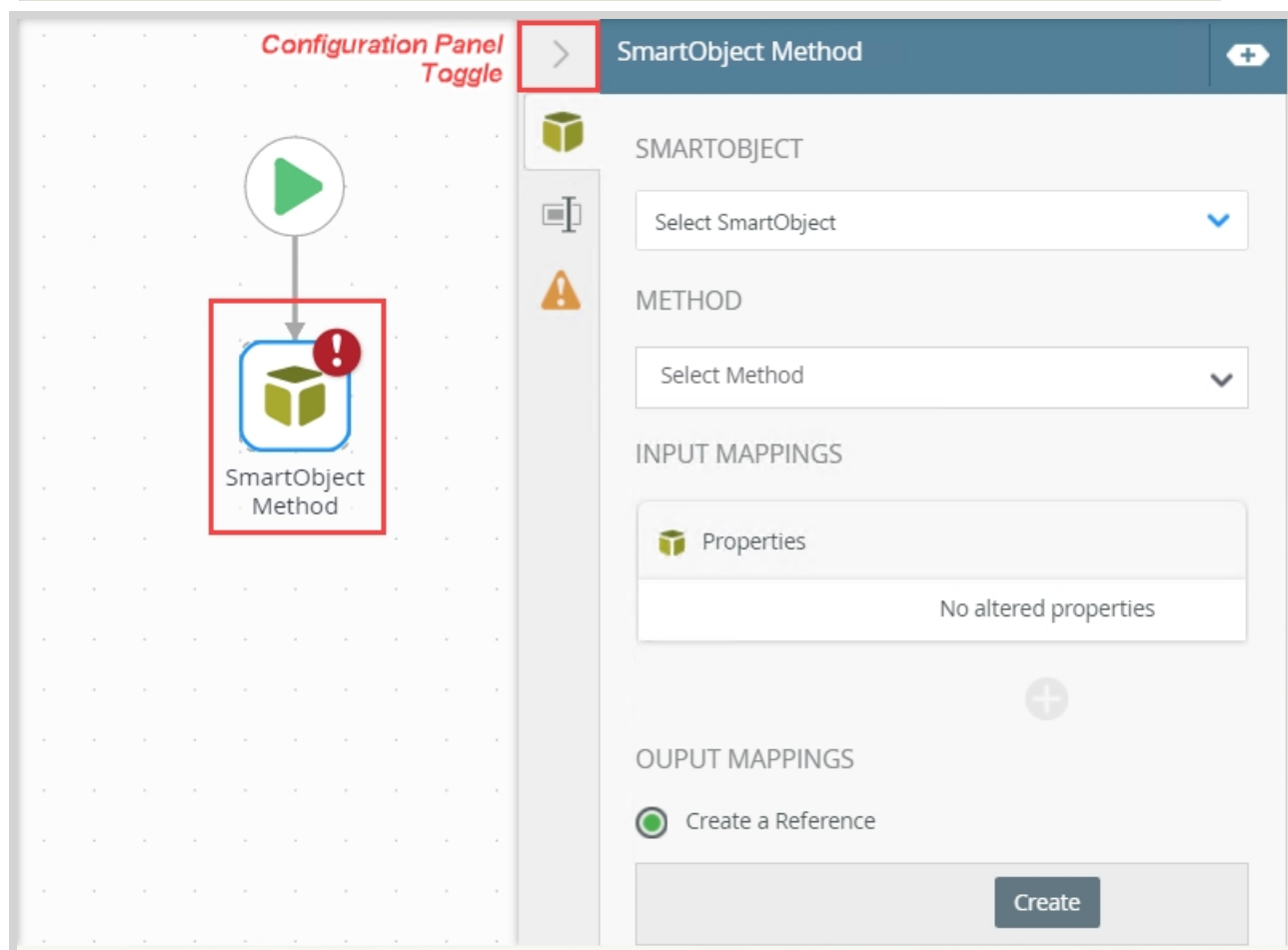
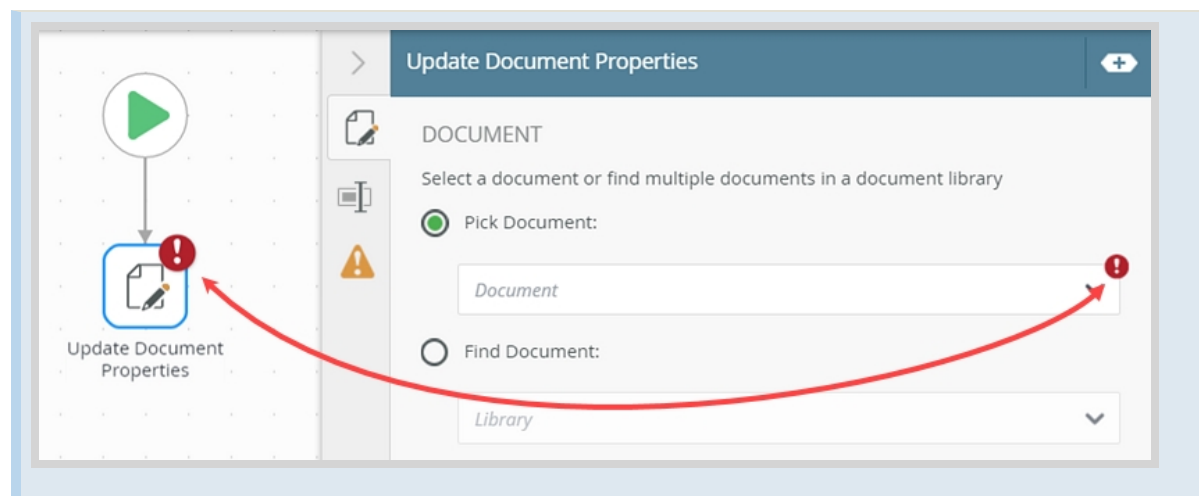
- a. Now, you will begin adding steps to the design canvas. You add, connect, and configure steps so that you eventually have a complete workflow from start to end. The first step updates the status property in the Leave Request SmartObject.
Expand the **Toolbox > Basic** node. Drag the **SmartObject Method** step into the empty placeholder on the design canvas.



- b. Next, you will configure the SmartObject Method step properties. You want the Request Status property to show that you submitted the leave request and it is now waiting for review and approval. Click the **SmartObject Method** step to highlight it, then expand the **Configuration Panel** using the Configuration Panel toggle.

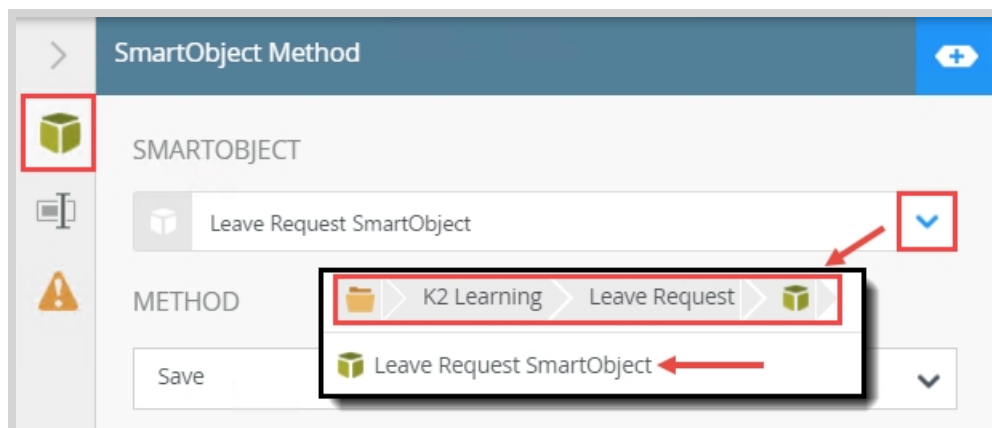
Note

When adding a step to the workflow design canvas, you may see a red warning badge. This badge simply indicates the step has required configurations. When you expand the Configuration Panel, the required configuration is also flagged.

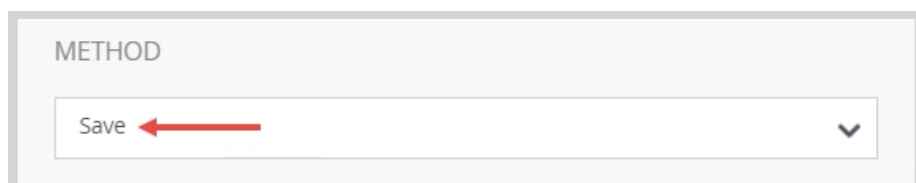


- c. First, you will select the Leave Request SmartObject so that K2 knows which SmartObject to update. Select the Leave Request SmartObject. From the **SmartObject** tab, select **Browse** from the SMARTOBJECT drop-down. Navigate to and select the **Leave Request SmartObject**. (K2 Learning

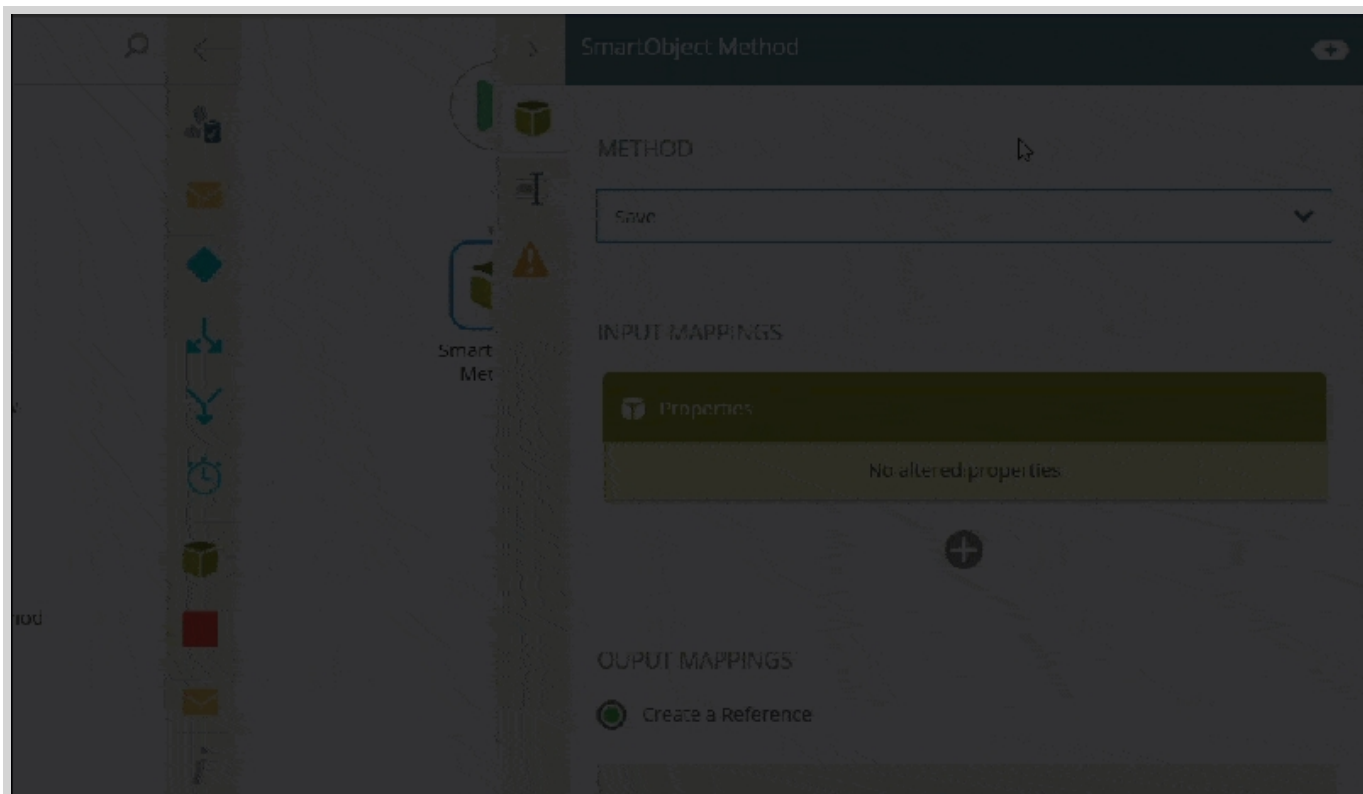
> Leave Request > SmartObjects)



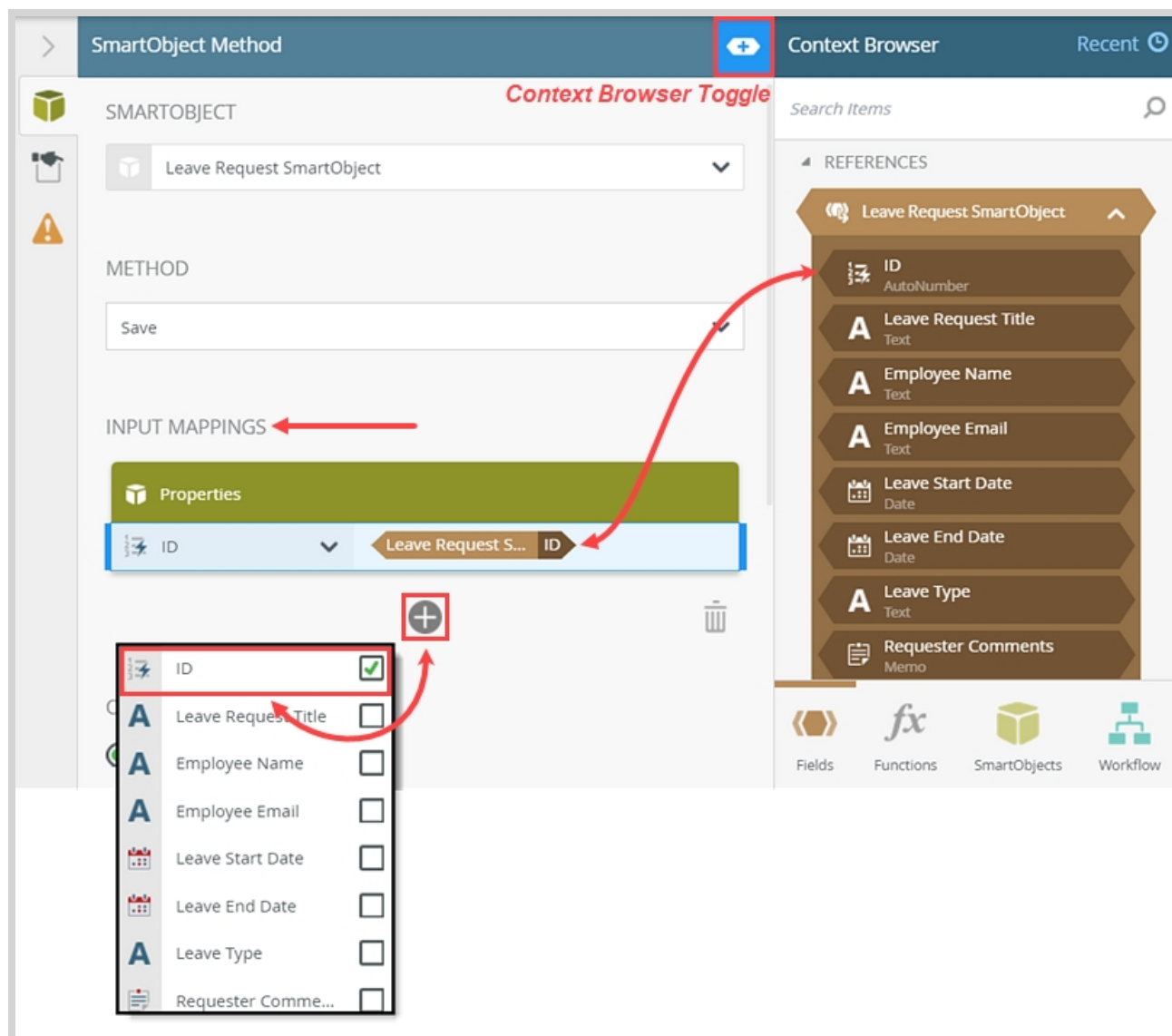
- d. Next, select the **Save** method from the METHOD drop-down. (Save is the equivalent of update in K2 terms.)



- e. Next, you will associate the current record for this workflow instance to the Leave Request SmartObject. This tells K2 *which* record to update.
Expand the **Context Browser**. Expand the REFERENCE drop-down if it isn't already. When you configured the Start Workflow wizard, K2 detected the Leave Request SmartObject and added it as a reference in the workflow.



- f. Click the **Add (+)** icon for the INPUT MAPPINGS. Select the **ID** for the current record. Drag the Leave Request SmartObject reference **ID** into the box to the right of the current record **ID**. This tells K2 to bind the current, or new record ID to the SmartObject ID property. K2 now knows *which* record to update. After you add the ID reference, collapse the **Context Browser** by clicking the toggle once again.



- g. Next, you will add the value for the request status. Click the **Add (+)** icon. Select **Request Status**. You may have to scroll down to see the status property. In the second mapping text box, type *Submitted*

then click outside the box to set the value.

The screenshot shows the K2 Configuration Panel for a SmartObject Method. The panel is divided into several sections:

- SMARTOBJECT**: A dropdown menu showing "Leave Request SmartObject".
- METHOD**: A dropdown menu showing "Save".
- INPUT MAPPINGS**: A section containing a table of input mappings.

Properties	
ID	Leave Request S... ID
A Request Status	Submitted
- OUTPUT MAPPINGS**: A section at the bottom.

A red arrow points to the "Request Status" property in the input mappings table. A red box highlights the "+" icon in the bottom right corner of the input mappings section, indicating where to click to add a new mapping.

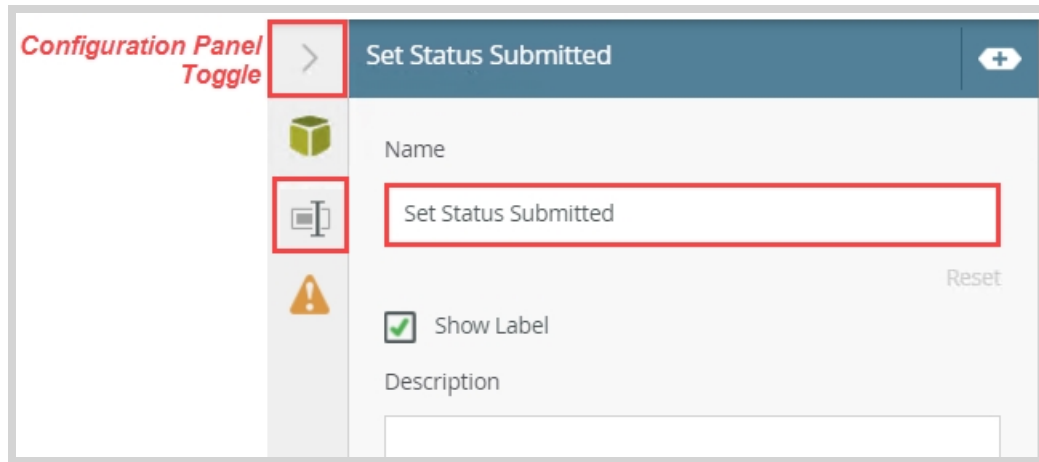
- h. To keep the design canvas organized, you can rename steps so that you know exactly what the step does.

Click the **General Properties** tab in the Configuration Panel. Change the **Name** to

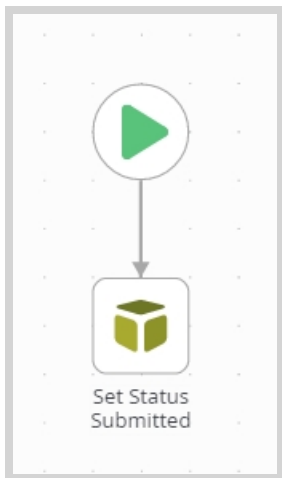
Set Status Submitted

then click outside the box to set the value. Collapse the **Configuration Panel** by clicking the Con-

figuration Panel toggle once again.



Your workflow should look like the image below.

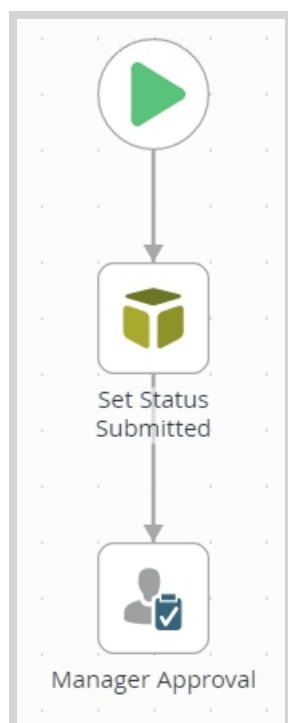


- Next, you will add and configure the Task step for the requester's manager. The manager must decide to approve or deny the leave request. Add a **Task** step and connect it to the **Set Status Submitted** step. Expand the Configuration Panel and set the following:

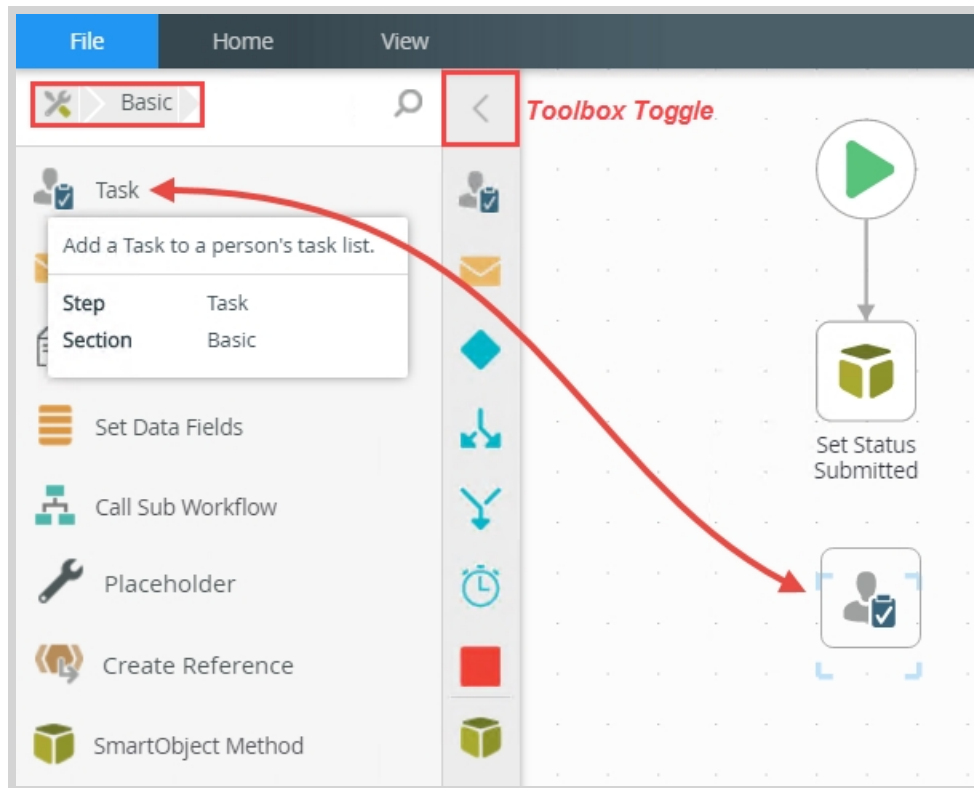
Property/Tab	Value
Instructions	<i>A leave request requires your approval. Select Approved or Denied, then click Submit to send your decision.</i>
Actions	<i>Approved</i> <i>Denied</i>
Form	Form : SmartForm > Leave Request Form Select the option to <ul style="list-style-type: none"> OPEN TASK: Help me configure it FORM STATE: Create a new State: Workflow Task (this should be the default) Keep the remaining default wizard values

Recipient	Originator Originator Manager You will add both the Originator and Originator Manager as the task recipients. You will action a task as the manager. Having tasks assigned to the originator (you) will provide content for the Reporting hands-on tutorial.
Notification	Select the default Send email task notification , do not customize.
Step Name	<i>Manager Approval</i>

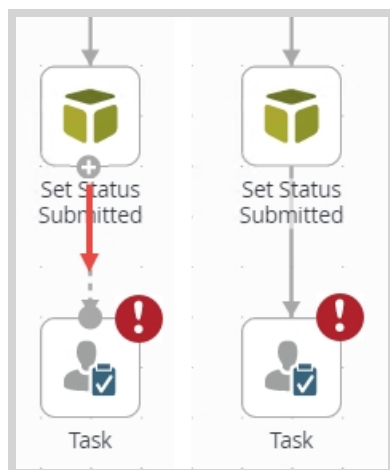
SmartObject Method Step and Task Step



- Next, you will and configure the Task step for the requester's manager. After the workflow starts, the manager receives a task notification email. They must decide to approve or deny the leave request. In this step, you will add the Task step, configure the actions (such as Approved or Denied), then assign the task to a task recipient. In this case, the recipient is the workflow originator's manager. Expand the **Toolbox > Basic** nodes. Drag a **Task** step onto the design canvas, below the Set Status Submitted step.



- b. To connect the two steps, hover your mouse over the **bottom border** of the **Set Status Submitted** step. You will see a handle appear. Click the handle and drag it into the new **Task** step. Click the canvas to set the line. (If you don't see a handle appear when you hover, click anywhere in the design canvas to refocus your mouse, then hover again.)



- c. Next, you will configure the Task step properties. Click the **Task** step to highlight it, then expand the **Configuration Panel**. Click the **Task Detail** tab (the first tab). Enter the following INSTRUCTIONS: *A leave request requires your approval. Select Approved or Denied, then click Submit to send your*

decision.

The screenshot shows the 'Task' editor interface. On the left is a vertical toolbar with icons for people, documents, users, email, and a folder. The main area is titled 'Task' and contains an 'INSTRUCTIONS' section. A red box highlights the 'INSTRUCTIONS' header icon and the text area below it. The text area contains the following text: 'A leave request requires your approval. Select Approved or Denied, then click Submit to send your decision.' At the bottom right of the text area is an 'Insert Field' button with a plus icon.

- d. **Actions** are the decision choices for the task-assigned user. In this task, there are two choices: Approved and Denied. Scroll down to the ACTIONS section, then click the **Add (+)** icon. Type over the default action and enter

Approved

then click the **Add (+)** icon once again. Type over the default action and enter

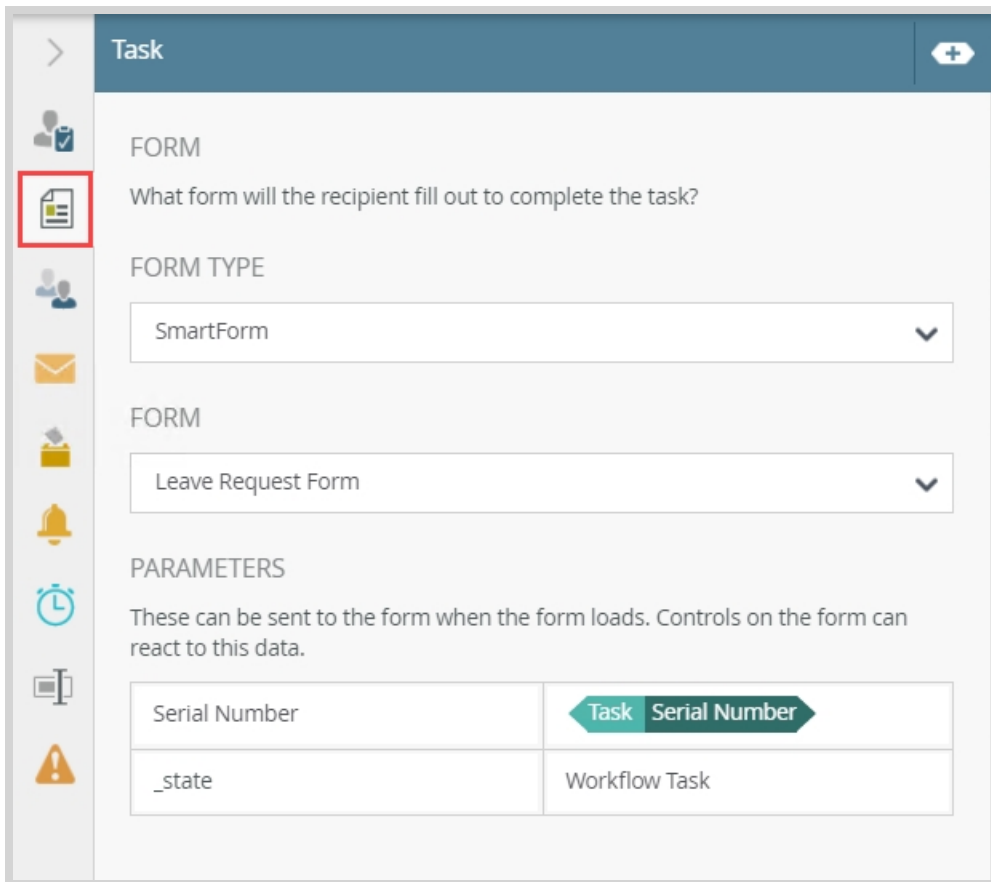
Denied

then click outside the box to set the action.

The screenshot shows the 'Task' editor interface, scrolled down to the 'ACTIONS' section. The 'INSTRUCTIONS' section is still visible at the top. The 'ACTIONS' section is titled 'What actions can you take with this task?'. It contains two action items: 'Approved' and 'Denied'. The 'Denied' item is highlighted with a blue background. At the bottom of the screen, there is a large plus icon and a trash icon.

- e. Click the **Forms** tab. Here, you will configure the form you want the task recipient to use. In this case, you will use the same Leave Request Form that the requester used, only with a different state.

The completed Forms screen.



Task

FORM
What form will the recipient fill out to complete the task?

FORM TYPE
SmartForm

FORM
Leave Request Form

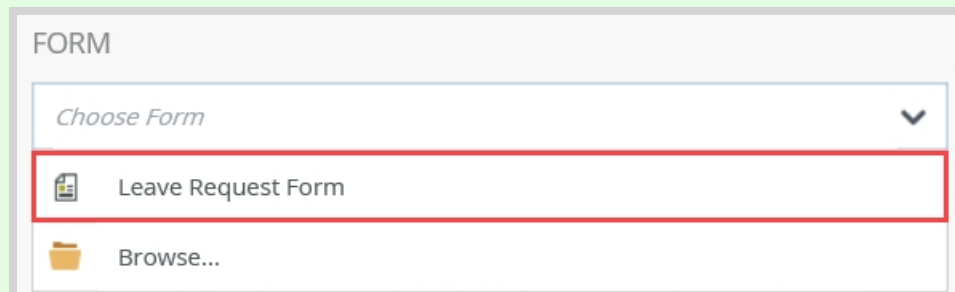
PARAMETERS
These can be sent to the form when the form loads. Controls on the form can react to this data.

Serial Number	Task Serial Number
_state	Workflow Task

- f. Set the FORM TYPE to **SmartForm**.
- g. For the FORM, click the drop-down arrow and select **Browse**. On the **Choose SmartForm** screen, click **K2 Learning**, then **Leave Request**. Click **Forms**. Select the **Leave Request Form**, then click **OK**.

Tip

In some cases, you may see **Leave Request Form** when you click the drop-down arrow for the FORM value. You can select the form if you like and skip browsing to it.

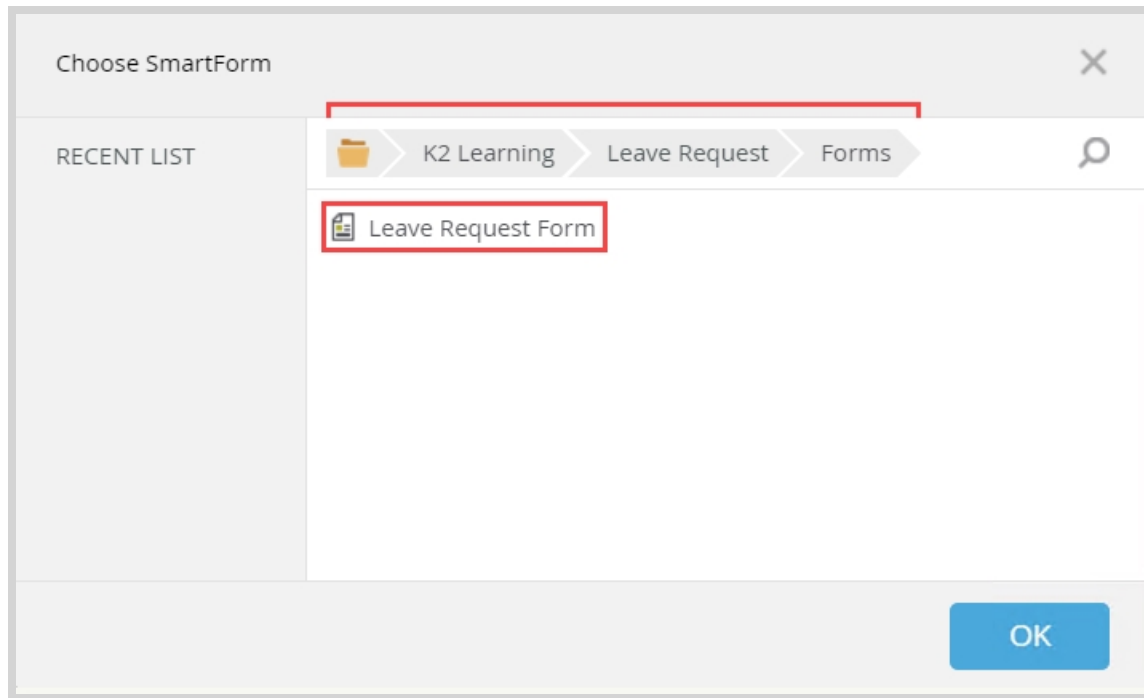


FORM

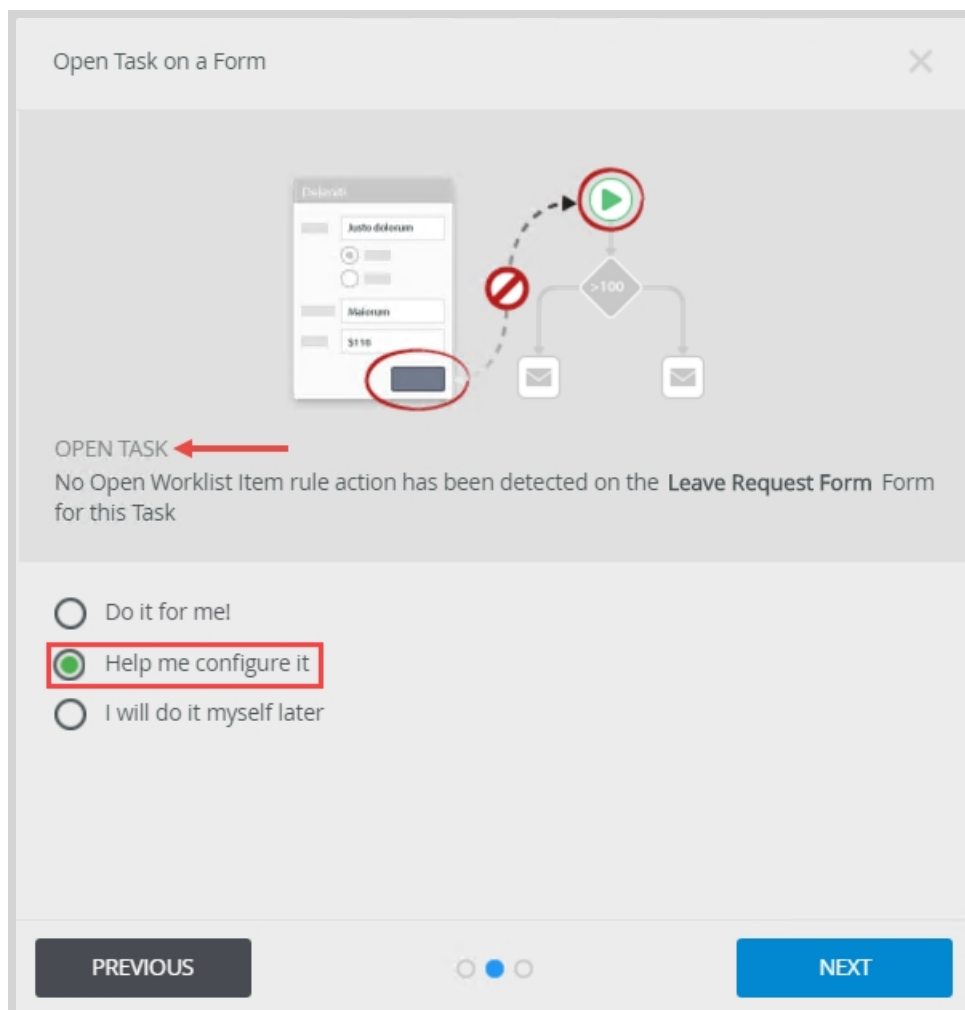
Choose Form

Leave Request Form

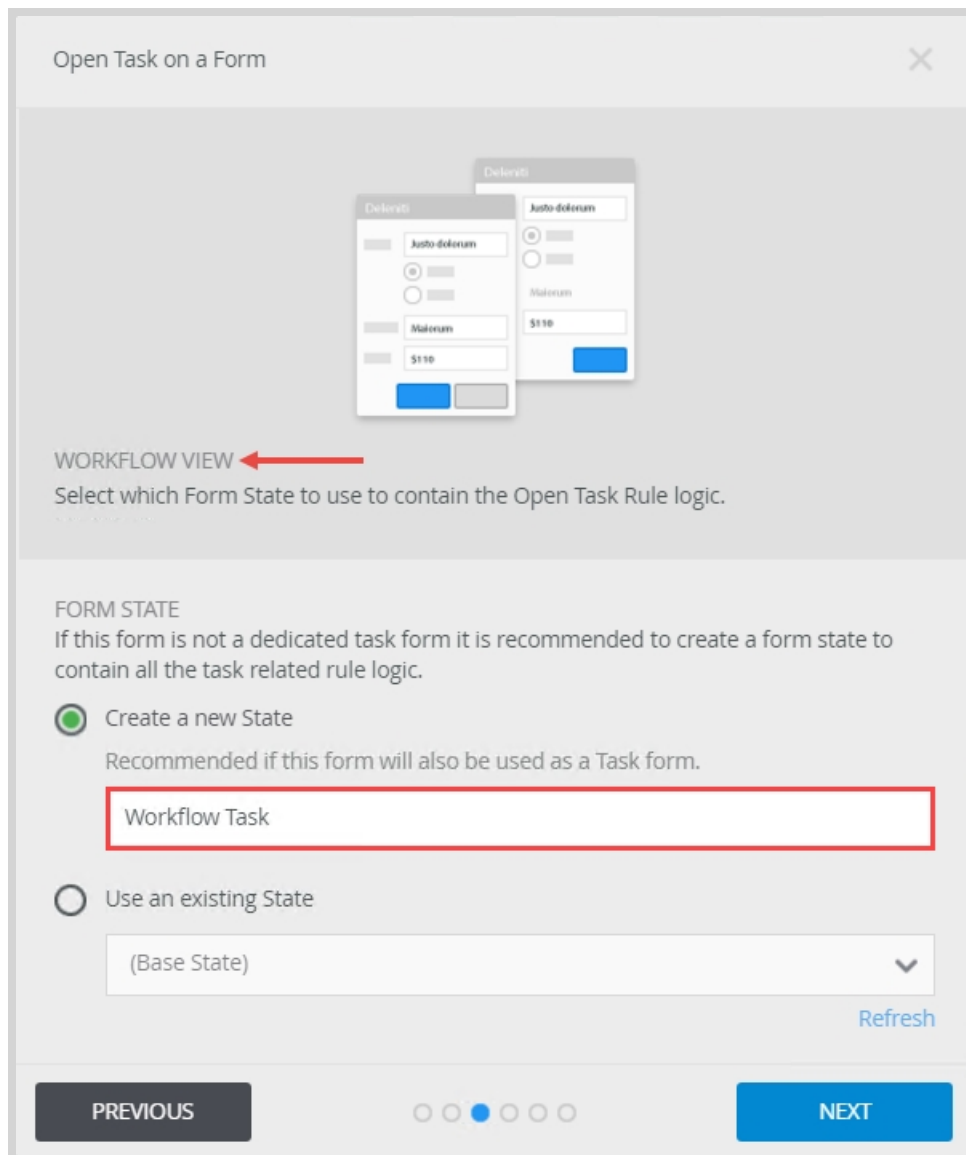
Browse...




- h. There are no changes to the REFERENCES screen, so click **NEXT**. On the OPEN TASK screen, confirm (or select if necessary) the option to **Help me configure it**, then click **Next**.



- i. You will create a new state for the approval task. This state represents the form's configuration and behavior for the requester's manager. On the WORKFLOW VIEW screen, keep the default **Workflow Task** value for the FORM STATE. Click **NEXT**.



Open Task on a Form

WORKFLOW VIEW 

Select which Form State to use to contain the Open Task Rule logic.


FORM STATE

If this form is not a dedicated task form it is recommended to create a form state to contain all the task related rule logic.

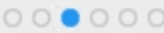
☒ Create a new State

Recommended if this form will also be used as a Task form.

☐ Use an existing State

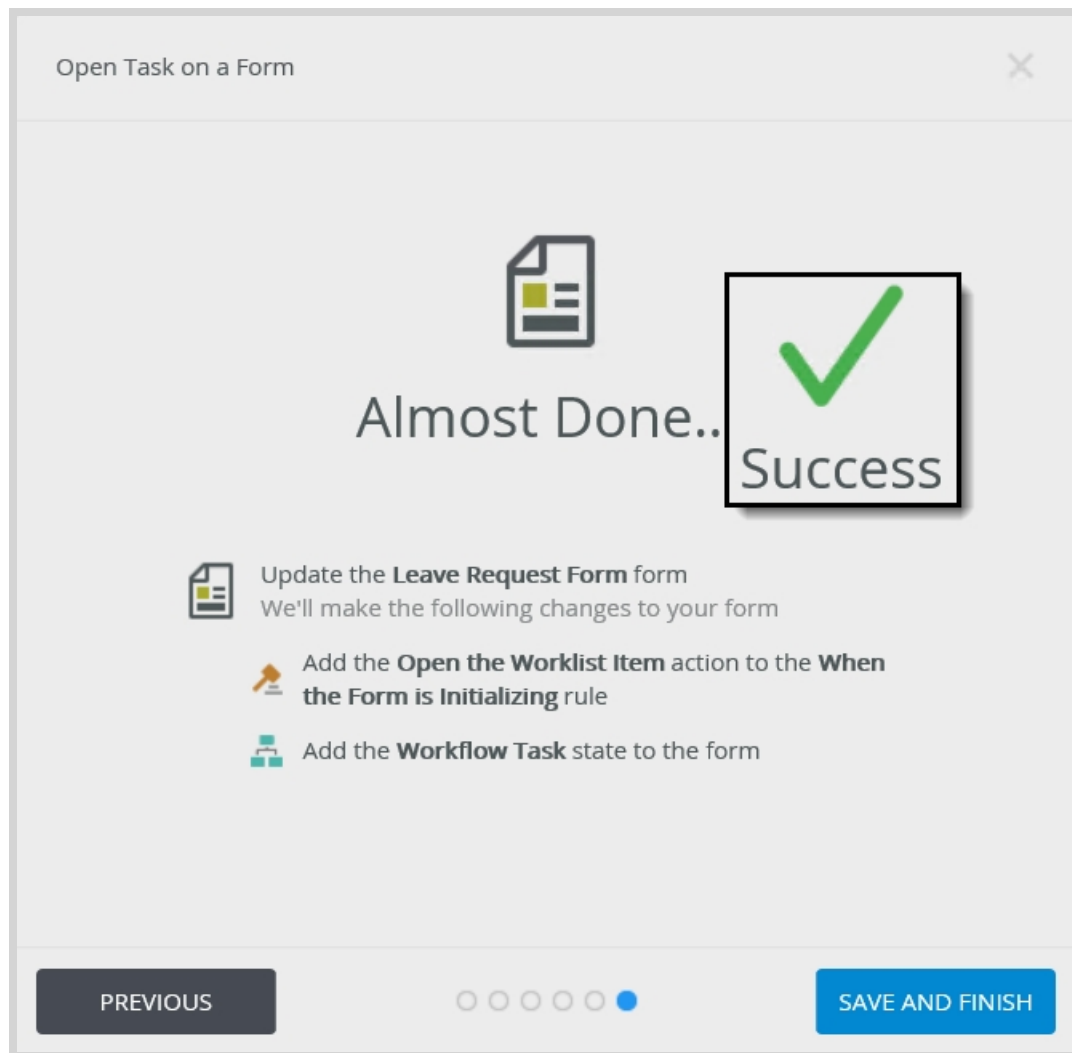
(Base State) 

[Refresh](#)

PREVIOUS  NEXT

- j. There are no changes to the OPEN TASK RULE ACTION screen, so click **NEXT**.
- k. There are no changes to the AFTER SUBMIT ACTION screen, so click **NEXT**.

- I. The last screen is the summary. Review the settings, then click **SAVE AND FINISH**. When you see the **Success** check, click **OK** to close the wizard.



- m. Next, you will assign the users who will be the task recipients. For this tutorial, you will assign the user task to both the originator and the originator's manager. Click the **Recipients** tab. **Originator** should already be the default recipient. Click the **Add(+)** icon to add another user. A new recipient box appears. Click the drop-down arrow and select **Originator Manager**.

Note

In most cases, you will only add the Originator Manager as the task recipient. However, to provide content for the Reporting tutorial, there must be several tasks assigned to you. In this case, you will action a task as the manager, while still having tasks assigned to you for reporting content.

Manager Approval

RECIPIENTS

Who should receive the task?

Originator

Originator Manager

Add Group

- n. Select the **Notification** tab. Confirm that the **Send email task notification** option is CHECKED. You will not customize the email message in this basic tutorial, so there are no edits to this screen. (In the Leave Request (Extended Version) tutorial, you learn how to customize an email using variables from the Context Browser.)

Task

TASK NOTIFICATION

Send an email task notification to the recipients when the task is created

☒ Send email task notification

☐ Customize Notification

SUBJECT

Type / Drop

BODY

- o. The last step in the task configuration is to change the step name so that is clear what the step does. Click the **General Properties** tab. Change the **Name** to *Manager Approval*

then collapse the **Configuration Browser**.

>

Manager Approval

+

Name

Manager Approval

Reset

☒

Show Label

Description

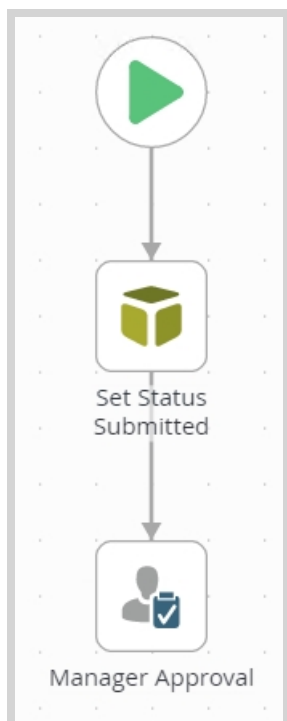
Priority

● Medium

▼

Notes

Your workflow should look like the image below.



Review

In this step, you added a SmartObject Method to the workflow. You configured the method to update the request status property. You bound the current record to the SmartObject by mapping the ID properties.

You then added a user task and assigned it to the approving manager. The manager must decide whether to approve or deny the leave request. When configuring the task form properties, you created a new state will control the look and behavior of the form that the approving manager sees. In the next step, you will add a Decision step that generates the outcomes for the Task step.

Next Step: 8. Add a Decision Step

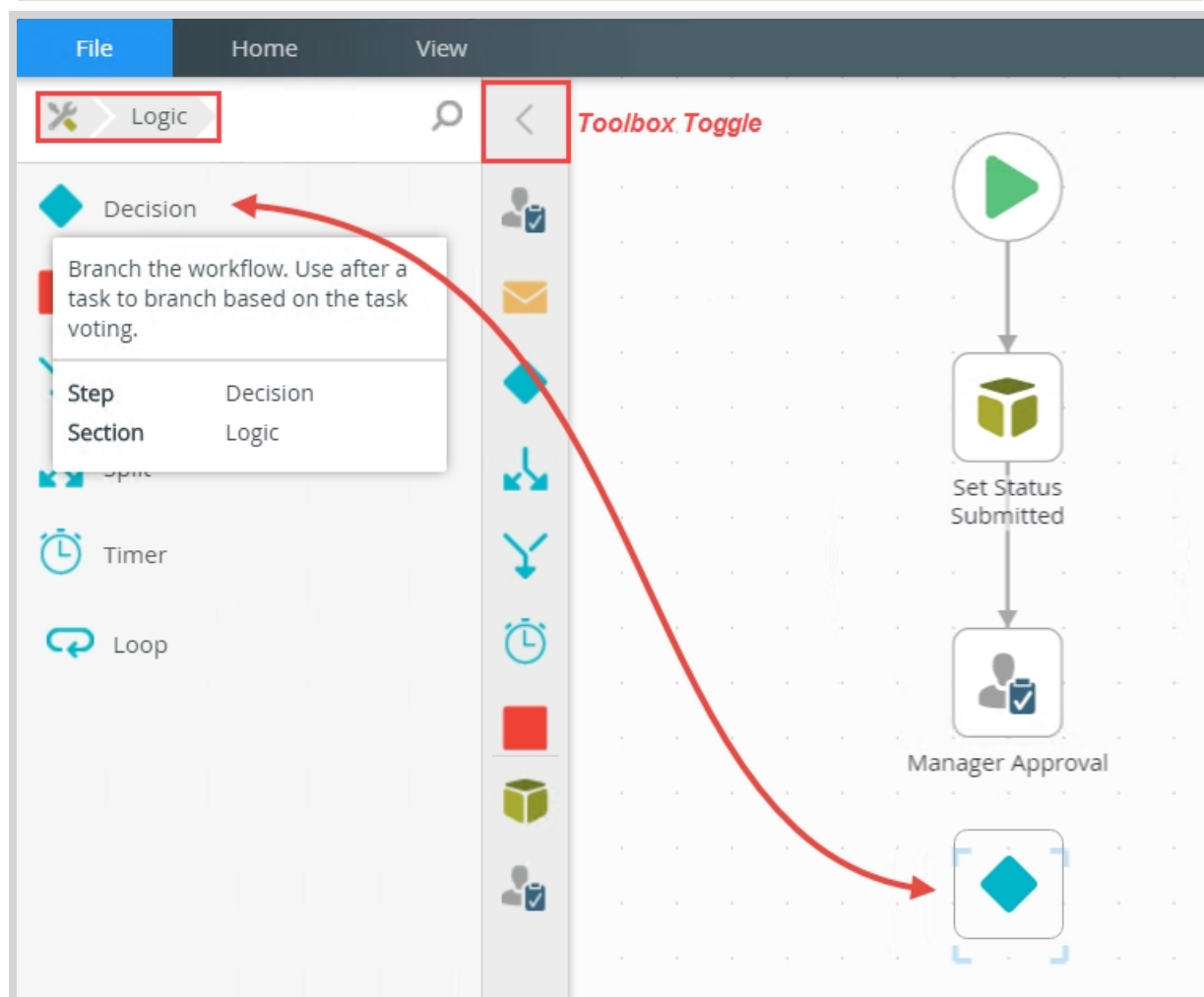
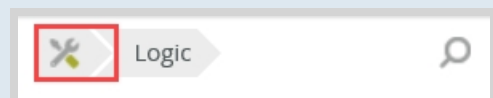
8. Add a Decision Step

In this step you will add a Decision step to the workflow. A Decision step is often used in conjunction with a user Task step. The Decision step generates lines based on the actions configured in the preceding user task, and moves the workflow along the appropriate line, depending on the decision the task recipient made. In this case, there are two possible outcomes: Approved or Denied.

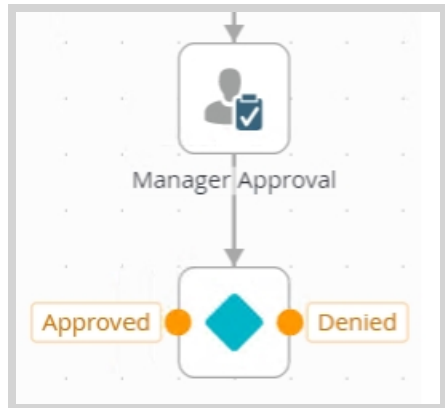
1. Add a **Decision** step to the design canvas, then connect the **Manager Approval** step to the **Decision** step.
 - a. Expand the **Toolbox > Logic** node. Drag a **Decision** step onto the design canvas, below the Manager Approval step. Collapse the **Toolbox**.

Note

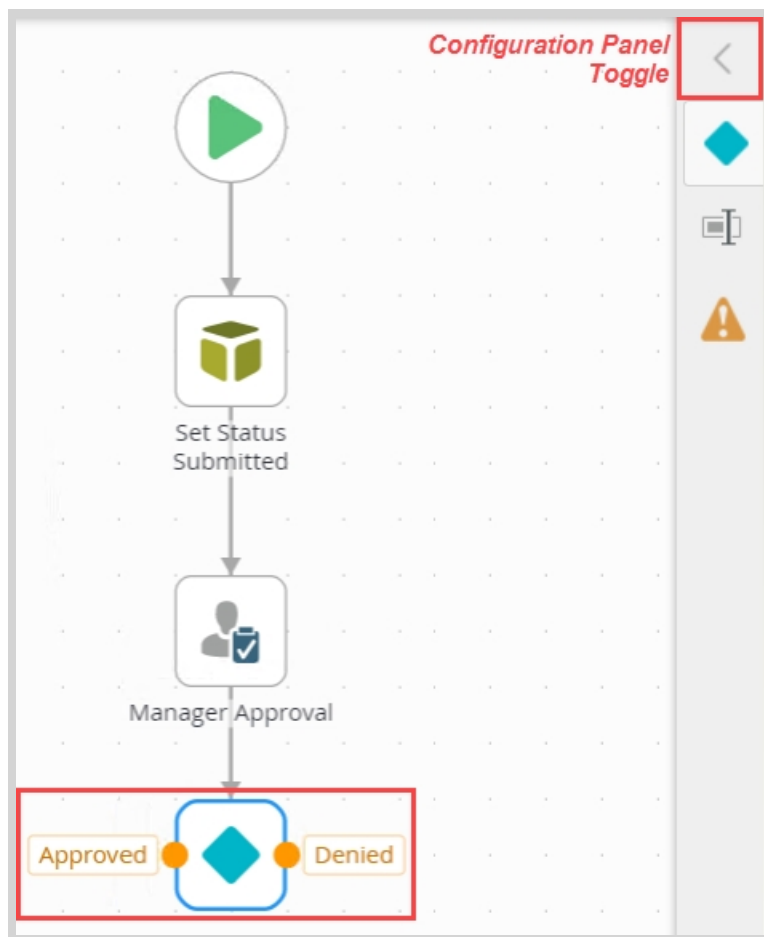
Click the **Toolbox** icon in the breadcrumb bar to return to the main menu. This is useful if you want to change categories (such as Basic or Logic).



- b. Connect the **Manager Approval** step to the **Decision** step. The Decision step generates the outcome lines for the two actions.



- c. Click to highlight the **Decision** step. Expand the **Configuration Panel**.



- d. Click the **Decision Details** tab. The DECISIONS section displays the **Approved** and **Denied** actions. You configured the actions in the Manager Approval step. The DECISIONS RULE displays the **Use Task Outcomes** by default. You also have the option to create your own rules for handling workflow

paths.

The screenshot shows the 'Decision' configuration panel in K2. The 'DECISIONS' section contains two paths: 'Approved' (linked to 'Approved' Task Result) and 'Denied' (linked to 'Denied' Task Result). The 'DECISIONS RULE' section has 'Use Task Results' selected, with a description: 'An automatic rule that follows a separate path for each task result of the linked task.' Below this, 'Automatically Add Decisions' is checked, with a description: 'We'll add decisions whenever this task step has actions or results added:'. A dropdown menu shows 'Manager Approval' selected.

2. Rename the step to *Manager Approval Decision*.

- Expand the **General Properties** tab. Change the **Name** to *Manager Approval Decision*

so that it is clear what this step does. Confirm the **Show Label** option is UNCHECKED. In this case, since the Decision step connects to the Manager Approver step, it is obvious the two steps work together, so a visual label is not needed. However, the step will show up in reports, so K2 recommends naming the step for clarity. Collapse the **Configuration Panel**.

The screenshot shows the 'Manager Approval Decision' configuration panel. The 'Name' field is set to 'Manager Approval Decision'. The 'Show Label' checkbox is unchecked. A red arrow points to the 'Name' field.

Review

In this step, you added a Decision step and connected it to the Manager Approval step. The Decision step generates the outcome (lines) based on the actions configured. The workflow will determine if any rules or conditions are valid for the outcomes, then proceed along the appropriate outcome.

In the next step, you will add SmartObject Method steps to each outcome and update the status property based on the manager's decision. Then you will add an End step, which provides a visual indicator that the workflow is complete.

Next Step: 9. Add Additional SmartObject Method Steps and End Step

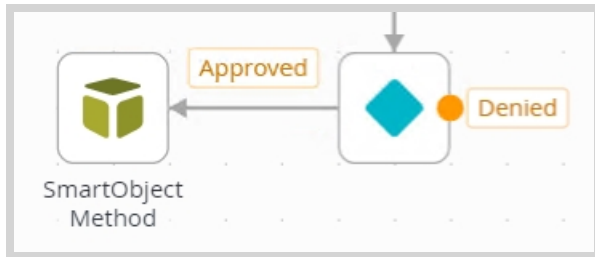
9. Add Additional SmartObject Method Steps and End Step

In this step, you will again use a SmartObject Method step to update the Leave Request SmartObject > Request Status property. You will add a SmartObject Method step to the Approved outcome and another to the Denied outcome. You will configure the save method to update the request status with either "Approved" or "Denied". You will complete the workflow by adding an End step. The End step provides a visual indicator that the workflow is complete. While not required, nor does it have any other functionality, the End step is a clear and clean method to end a workflow.

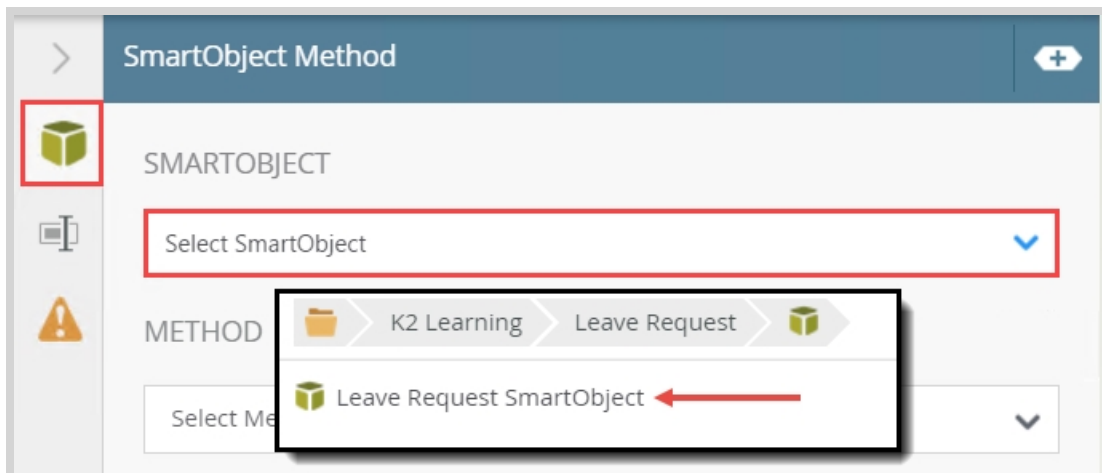
1. In this step, you will update the Request Status property so that it reflects either the approved or denied decision. Begin by adding a **SmartObject Method** step to the **Approved** outcome. Connect the two steps. Map the **ID INPUT MAPPINGS** to the **Leave Request SmartObject > ID REFERENCE**. Add the **Request Status INPUT MAPPING** and enter *Approved*. **Repeat** this step for the **Denied** outcome.

- a. This step is nearly identical to the first step you added and configured. You are going to update the Request Status property so that it reflects either the approved or denied decision. Ultimately, you will add two SmartObject Method steps to the canvas. One for the Approved outcome and one for the Denied outcome.

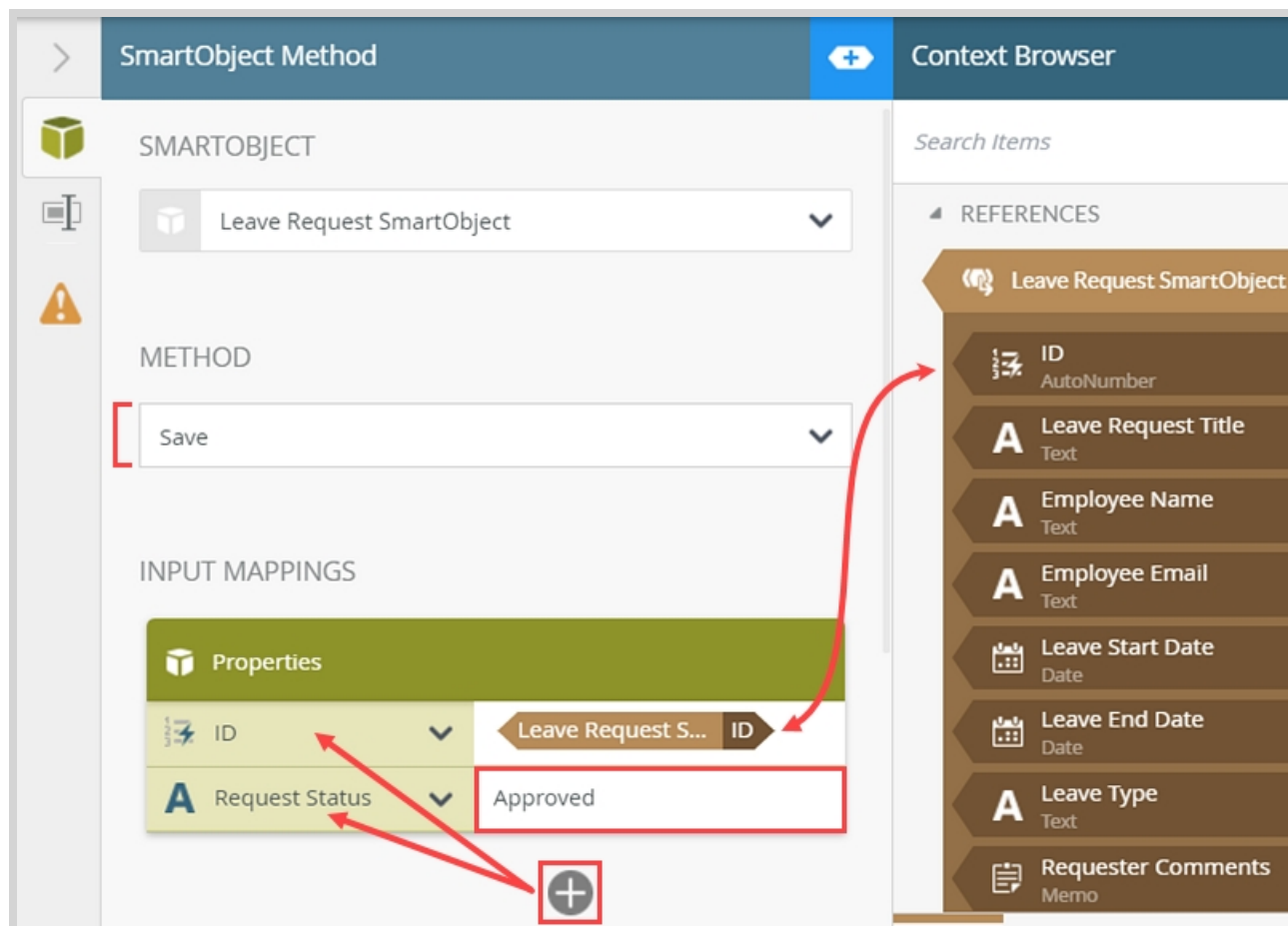
Drag a **SmartObject Method** step (**Toolbox > Basic**) onto the design canvas so that it aligns with the **Approved** outcome. Connect the **Approved** outcome to the **SmartObject Method** step.



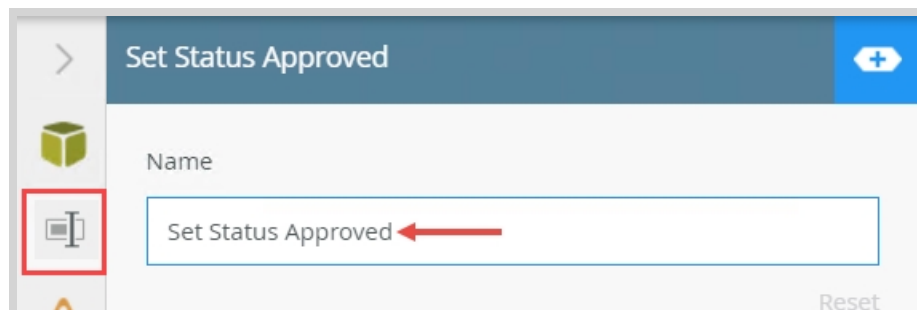
- b. Select the **SmartObject Method** step, then expand the **Configuration Panel**.
- c. Click the **SmartObject Detail** tab to confirm it is the active screen. In the SMARTOBJECT section, browse to, then select the **Leave Request SmartObject**. Click **OK**.



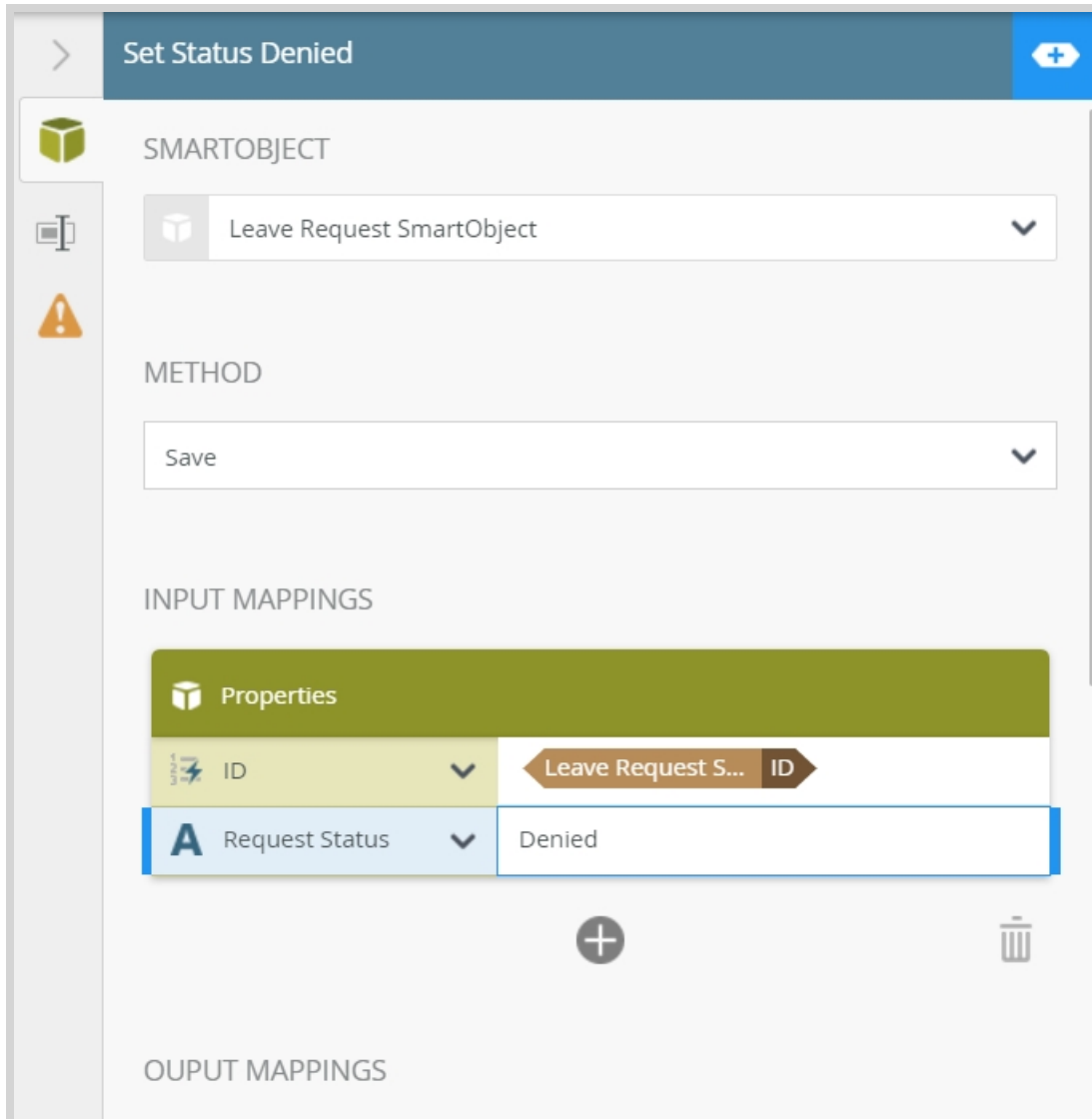
- d. Select the **Save** method from the METHOD drop-down list.
- e. Now, you will map the INPUT MAPPINGS to the Leave Request SmartObject REFERENCE so that K2 knows which record to update. In the INPUT MAPPINGS section, click the **Add (+)** icon. Select **ID** and **Request Status** from the drop-down list. If you like, you can select both properties at the same time by clicking the ID property, then clicking the Request Status property. Click outside the list menu to close it. Expand the **Context Browser**. Expand the **Leave Request SmartObject** REFERENCE. Drag the SmartObject **ID** so that it is mapped to the **ID** INPUT MAPPING. For the **Request Status** value, enter *Approved* then click outside of the box to set the value.



- f. Click the **General Properties** tab. Change the **Name** of the step to *Set Status Approved* then collapse the **Context Browser** and **Configuration Panel**.



- g. Using the same steps as above, add and configure a **SmartObject Method** for the **Denied** outcome. Be sure to rename the step to *Set Status Denied* from the **General Properties** tab.

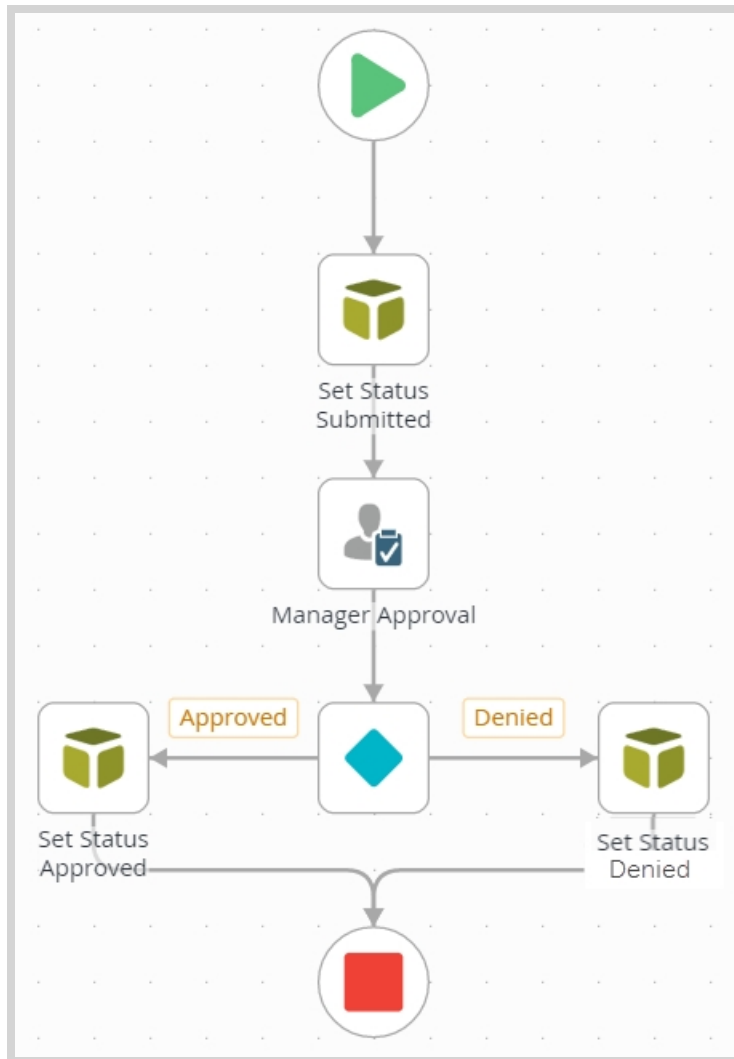


The screenshot displays the configuration interface for a K2 workflow step titled "Set Status Denied". The interface is organized into several sections:

- SMARTOBJECT**: A dropdown menu is set to "Leave Request SmartObject".
- METHOD**: A dropdown menu is set to "Save".
- INPUT MAPPINGS**: This section contains two rows of mappings:
 - The first row, under a "Properties" header, maps the "ID" property to the "ID" field of the "Leave Request S..." SmartObject.
 - The second row, marked with a blue "A" icon, maps the "Request Status" property to the value "Denied".
- OUTPUT MAPPINGS**: This section is currently empty.

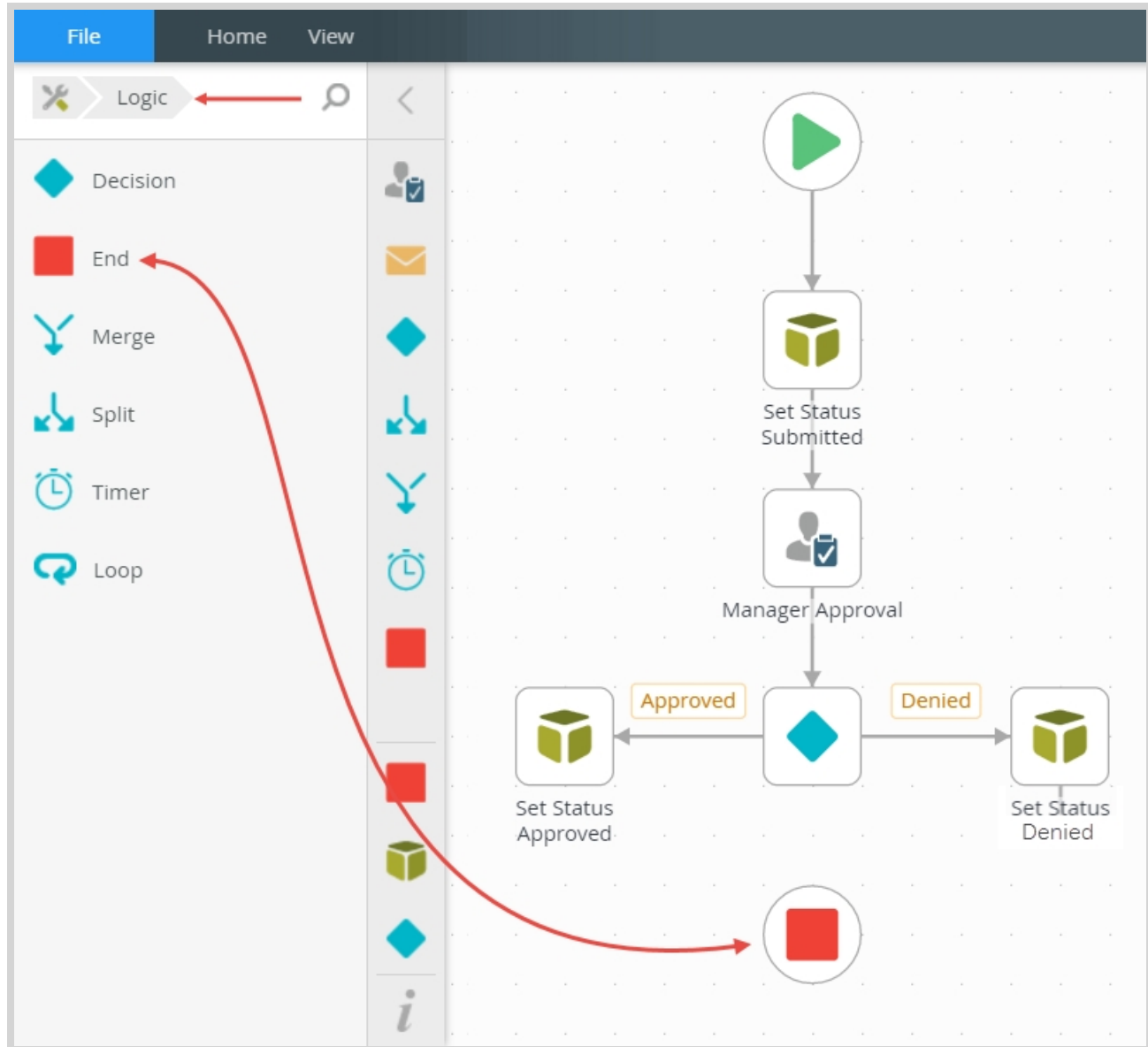
Navigation icons for adding (+) and deleting (-) steps are visible at the bottom of the configuration area.

2. To show the workflow is complete, add an **End** step. Connect the both the Approved and Denied SmartObject Method steps to the End step.

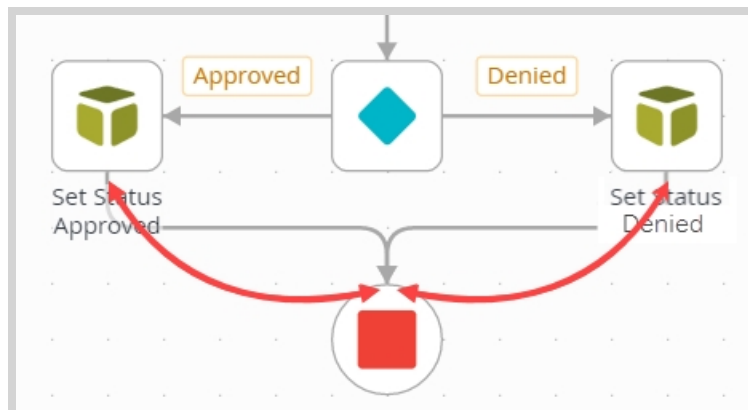
The Completed Leave Request Workflow

- a. The final task in building the workflow is to add an **End** step. (The End step is not required to finish a workflow, nor does it have any other functionality other than providing a clean indication that the workflow is complete.)

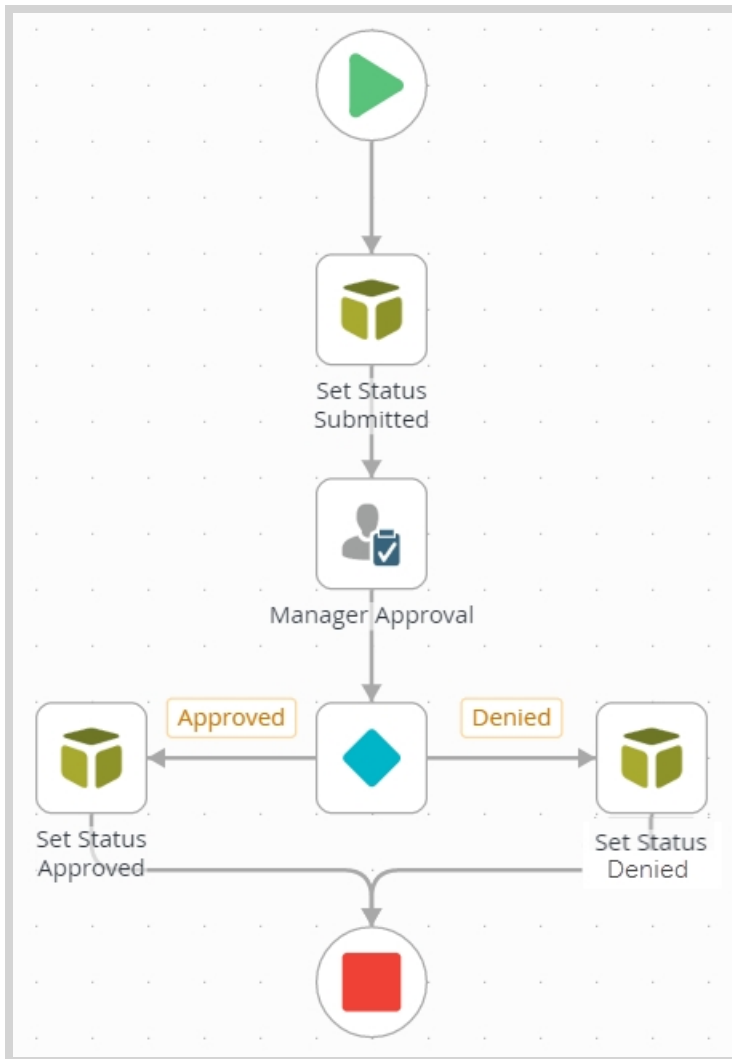
Drag an **End** step (**Toolbox > Logic**) below the **Decision** step.



- b. Connect the **Set Status Approved** step to the **End** step. Then connect the **Set Status Denied** step to the **End** step.



Your completed Leave Request Workflow should look like the image below.



Review

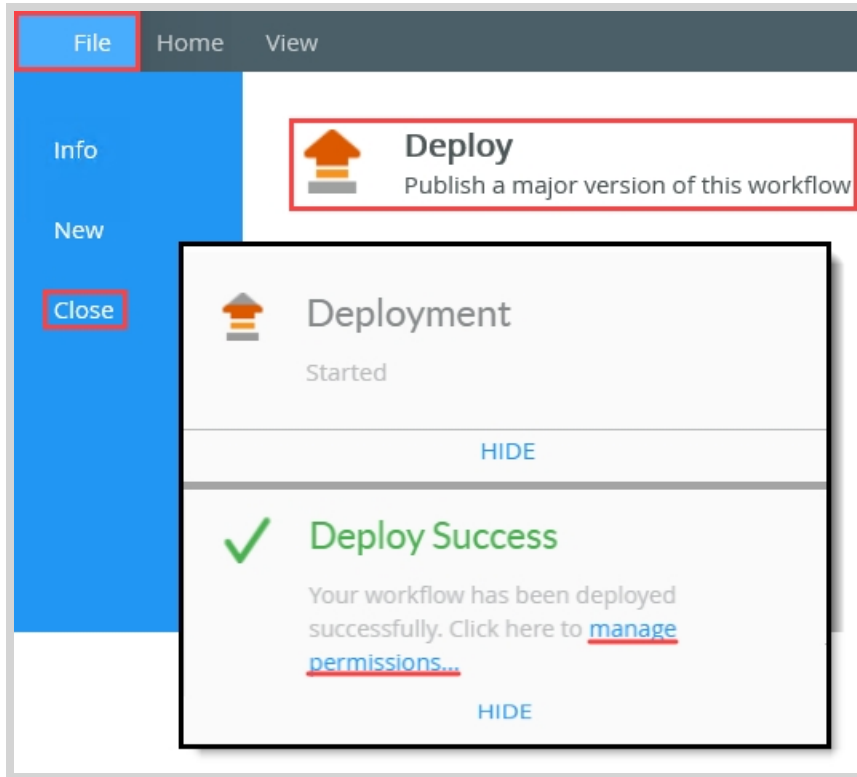
In this step, you added additional SmartObject Method steps to update the status property for the Approved and Denied outcomes. You added an End step as a visual indicator that the workflow is complete. In the next step, you will deploy the Leave Request Workflow to the K2 server.

Next Step: 10. Deploy the Leave Request Workflow

10. Deploy the Leave Request Workflow

Now you will deploy the Leave Request Workflow. Deploying a workflow publishes it to the K2 server. After deploying, the workflow is available for use. If you make any changes to the workflow, you must deploy it again so that the K2 server "sees" the changes.

1. **Deploy** the workflow. Click the **manage permissions...** link. You will set permissions in the next step.
 - a. Click the **File** tab, then **Deploy**. When you see the **Deployment Successful** bar, the deployment is complete. Next, you will grant workflow permissions. Click the **manage permissions...** link.



The K2 Management site launches. The next step contains the instructions for setting workflow permissions

Review

In this step, you deployed the workflow to the K2 server. The workflow is now available for runtime use. If you make a change to the workflow, you must redeploy it to publish the change. In the next step, you will assign workflow rights, or permissions, to domain users so that they can start an instance of the Leave Request Workflow.

Next Step: 11. Assign Workflow Rights

11. Assign Workflow Rights

In this step you will assign workflow rights, or permissions, to domain users so that they can start an instance of the Leave Request Workflow. You will grant start and view rights. This allows them to start the workflow and view reports on their process instances. You can manage workflow permissions from the K2 Management site.

1. In the **K2 Management** site, grant domain users start and view rights for the **Leave Request Workflow**.

Note

If you are in a cloud environment and cannot find "domain users", use "everyone" instead. Assign the same permissions to this group.

Leave Request Workflow Rights

Process Details

INSTANCES **RIGHTS** ROLES VERSIONS TASK LIST ERRORS REPORTS

+ Add ✕ Remove ▶ Edit ↺ Refresh

Selected Filter: Default ▼ 🔧

Quick Search: All fields ▼ 🔍

NAME	ADMIN	START	VIEW	VIEW PARTICIPATE	SERVER EVENT
Denallix Administra...	true	false	false	false	false
Domain Users	false	true	true	false	false

« < 1 > »

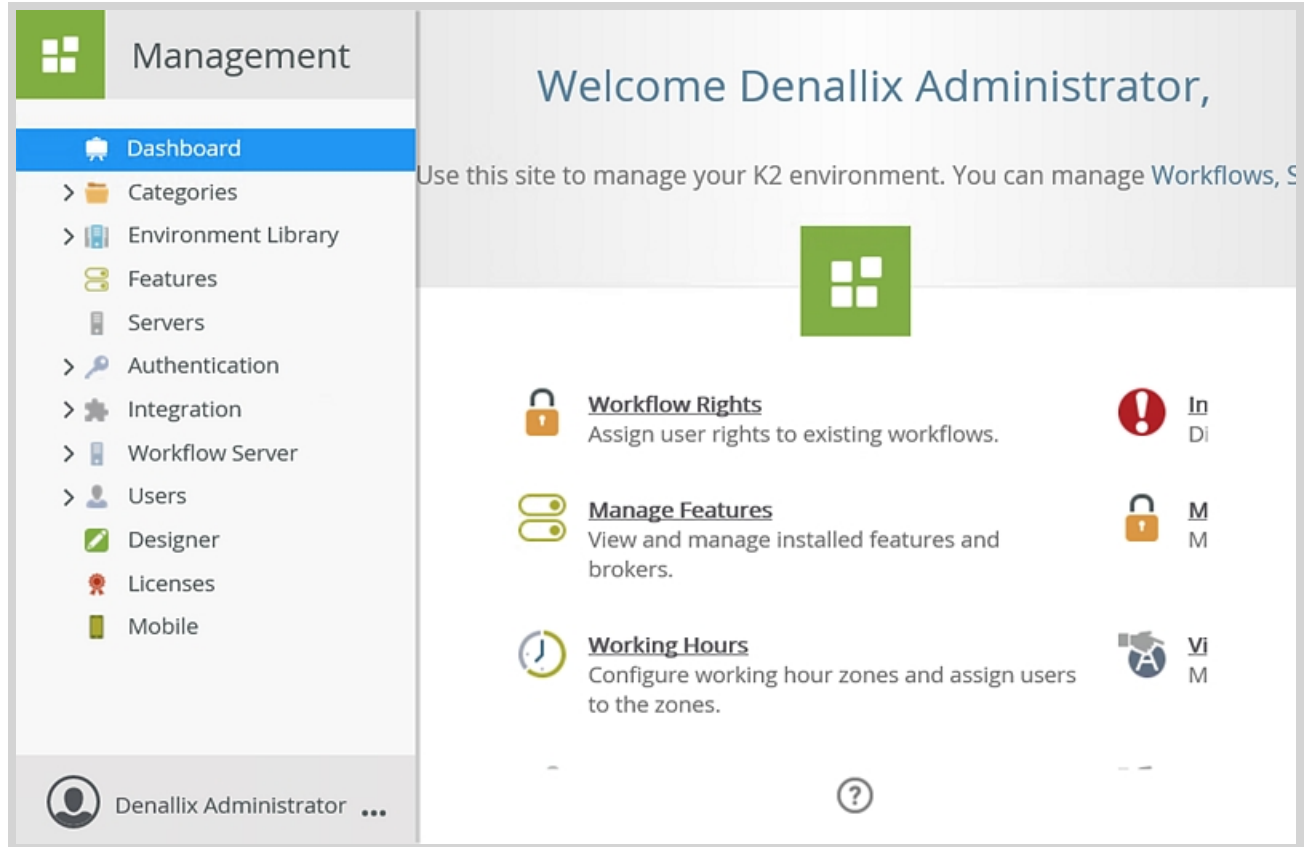
If you clicked the permissions link in the last step, skip to Step (b). The step below shows you how to open the K2 Management site if it is not already open.

- a. Launch the **K2 Management** site.

Note

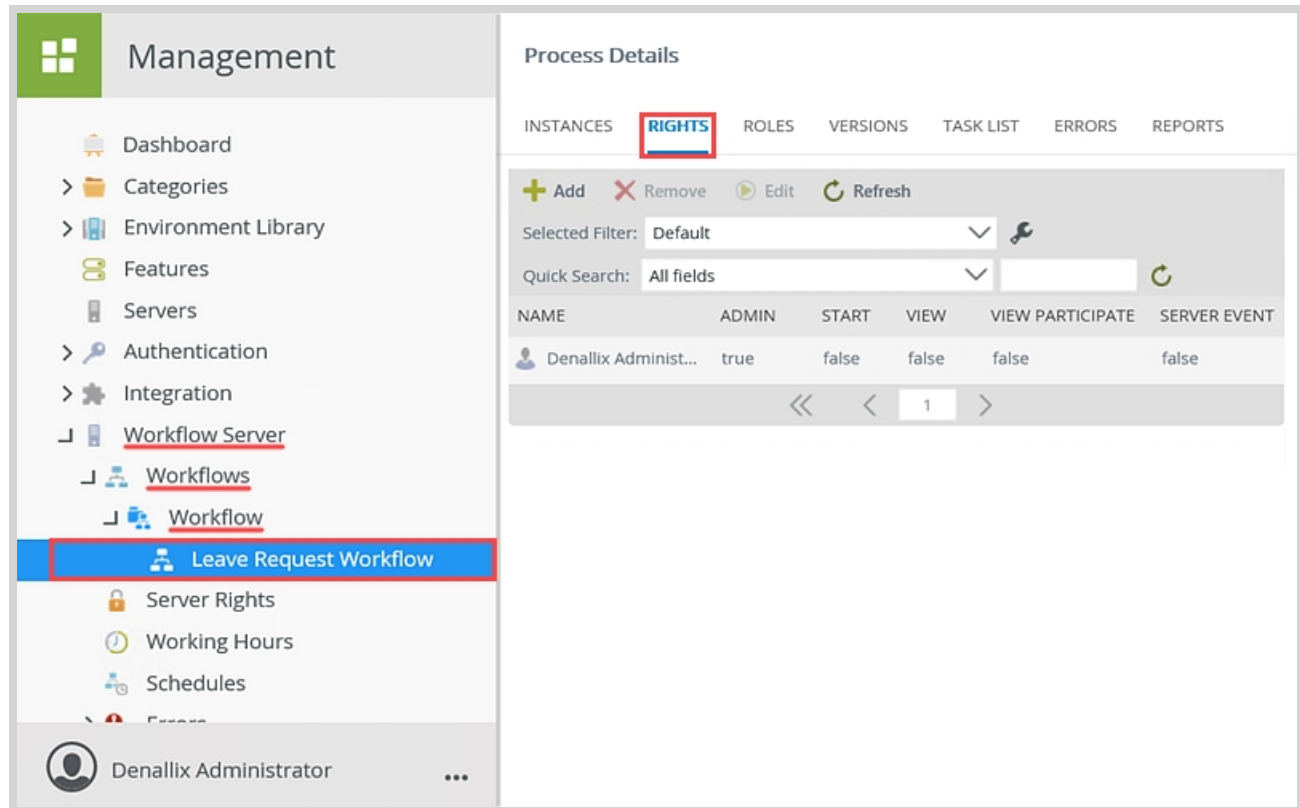
If you are unsure of how to launch the **K2 Management** site, see [Accessing K2 Sites](#).

- b. The management site opens with the dashboard. Depending on your K2 server version and permissions, your screen may be slightly different from the image below.

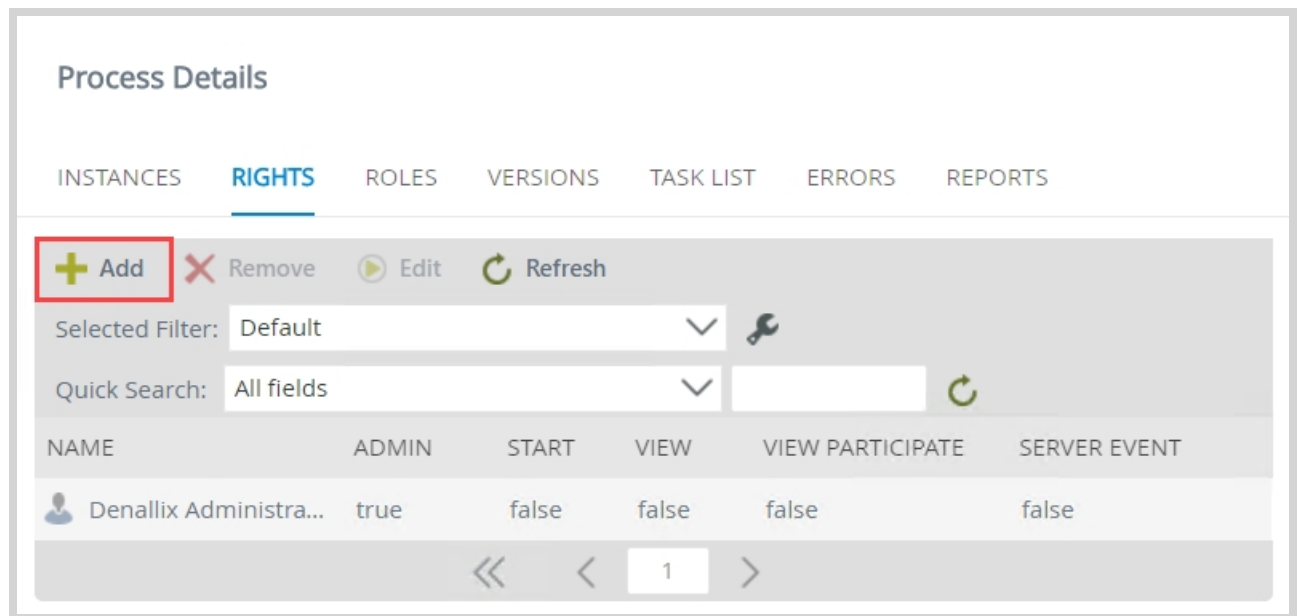


- c. Begin by selecting the Leave Request Workflow to expose the Process Details screen. Expand the **Workflow Server > Workflows > Workflow** nodes. Click to highlight **Leave Request Workflow**. The

central pane displays the Process Details screen. Click **RIGHTS**.



d. Click **Add**.



- e. On the search properties screen, change the **Search** mode to **All**. Searching "users" looks for a single user account and will return no results. Searching "groups" looks for a group of users (such as Finance Department) and again, will return no results. Searching "all" users looks for any records with those keywords. Confirm or change the **Label** to **K2**. You have the option of searching other employee management systems such as CRM. You want to search K2. In the search text box, enter *domain users* then click the **Search** button. Search results appear in the section below the search properties.

Depending on your keyword you may have several results returned. Click to highlight the **Domain Users** row, then click **Add**. Workflow rights members appear in the section below the search results. Click **Next**.

Workflow Rights - Add Users And Groups

Add Users and Groups

Search: All

domain users

Search

Label: K2

Type: Contains

+

Add

Search results are returned in this section.

NAME	ACCOUNT	TYPE	LABEL	EMAIL
Domain Users	DENALLIX\Domain Users	Group	K2	

<<

<

1

>

>>

×

Remove

Users or groups selected for workflow rights are this section.

<input type="checkbox"/>	NAME	ACCOUNT	TYPE	LABEL	EMAIL
<input type="checkbox"/>	Domain Users	DENALLIX\Domain Users	Group	K2	

<<

<

1

>

>>

Back

Next

Finish

Cancel

Note

The process of clicking *Add* twice may be confusing. The first add button (on the previous screen) directs you here (to the search properties screen). After searching, there may be many results returned, depending on the search keyword. If this is the case, you can select the appropriate user(s) or group(s), then add them to the workflow (rights) list. In this way, you can search for and add many users without leaving the search properties screen.

- f. On the workflow rights screen, CHECK the boxes for **START** and **VIEW** in the Domain Users row. You are granting domain users the right to start new workflow instances and the right to view reports on

instances they have started. Click **Finish**.

NAME	ADMIN	START	VIEW	VIEW PARTICIPA..	SERVER EVENT
Domain Users	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The **Process Details > RIGHTS** screen displays the users and groups with workflow permissions.

Process Details

INSTANCES **RIGHTS** ROLES VERSIONS TASK LIST ERRORS REPORTS

+ Add ✖ Remove ▶ Edit ↺ Refresh

Selected Filter: Default ▼ ⚙

Quick Search: All fields ▼ 🔍

NAME	ADMIN	START	VIEW	VIEW PARTICIPATE	SERVER EVENT
Denallix Administra...	true	false	false	false	false
Domain Users	false	true	true	false	false

Review

In this step, you granted workflow permissions to domain users so that they can start instances of the Leave Request Workflow. They will also be able to view reports on their process instances. In the next step, you will configure a rule to set the folio value for the workflow instance.

Next Step: 12. Set the Folio Value

12. Set the Folio Value

In this step you will edit the Leave Request Form and set the folio value. You will set the folio by editing a rule on the Leave Request Workflow (Default) state. Remember that states represent the configuration and behavior of a form at a given point in the workflow. The **Leave Request Workflow (Default)** state is the form configuration for the workflow originator, or the person requesting leave.

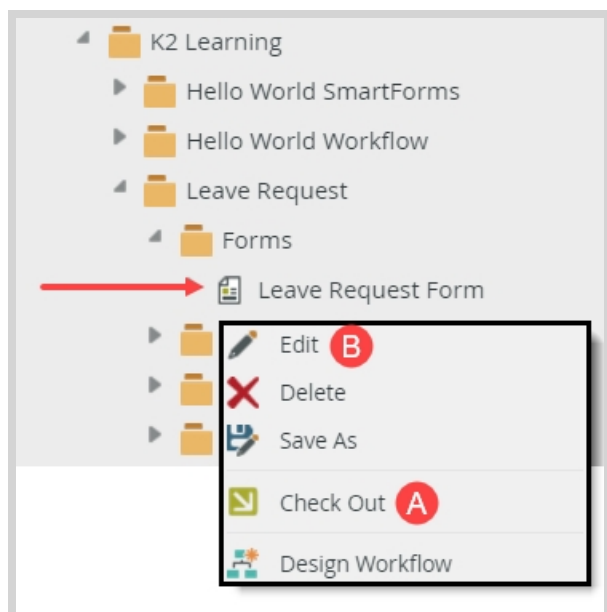
Note

A common use for the **folio** is to distinguish one workflow instance from another. The folio is often a unique value, but it does not have to be unique, nor is it required. The folio can use properties from the Context Browser as its value, or it can be set by hand, and it can be changed over the course of a workflow if you want to.

1. First, you will edit a rule associated with Leave Request Workflow (Default) state. You will use a variable from the Context Browser for the Folio value. At runtime, K2 replaces this variable with live content. Edit the **...when Create Button is Clicked** rule for the **Leave Request Workflow (Default)** state. This rule is found on the **Leave Request Form**. If you are still in the K2 Management site, switch to the **K2 Designer** browser tab to continue. Once in the designer tab, you are still in the K2 Workflow Designer, where you deployed the workflow. To return to the SmartForms designer, click **File > Close**. You should now see the categories.
 - a. Before you can edit a view or form, you must check them out. This "locks" the view or form in your name and ensures that your changes are not overwritten by another user. From the **K2 Designer** explorer, right-click and check out the **Leave Request Form**. Right-click and **Edit** the form. If the form is already checked out, just right-click and **Edit** the form.

Note

In some versions of K2, deploying the workflow checks in associated views and forms. In other versions, the views and forms remained checked out from when you created them. If your form is already checked out, right-click and **Edit** the form.



- b. Once the K2 Designer opens, switch to the rules screen. Click the **RULES** button.

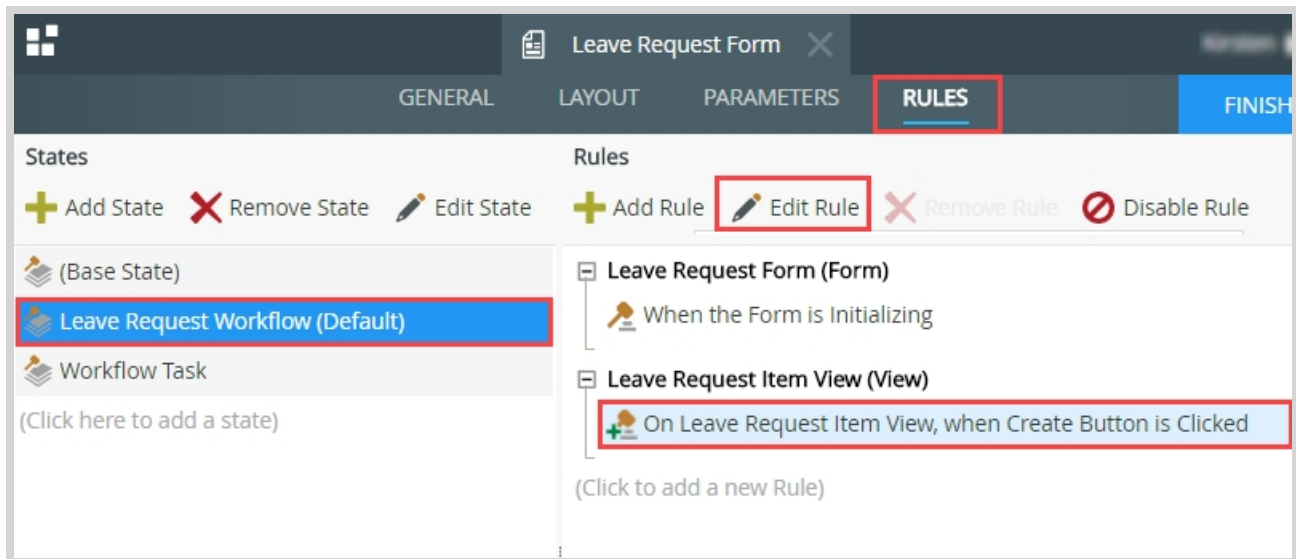
Note

Rules consist of events, conditions and actions. A valid rule contains at least one action.

- **Events** define *when* the rule should run, such as when a button is clicked, or when the form loads. (Events are optional, you can define "event-less" rules that you can call from other rules.)
- **Conditions** define *whether* the rule should run. If the criteria are true, continue the rule, and if the criteria are not met, stop the rule. For example, a condition might evaluate a form to confirm that required fields have content. (Conditions are optional - not all rules will require conditions.)
- **Actions** define *what* the rule should do. For example, show a message, start a workflow, or enable a form field. Rules can contain multiple actions that are run in sequence or in parallel.

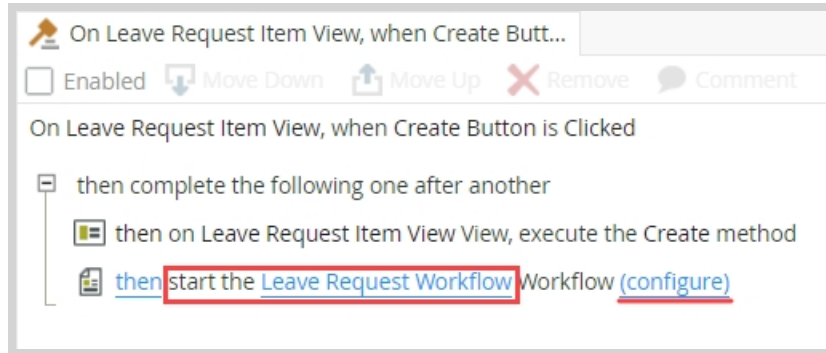


- c. Next, you will edit the rule that fires when you click the Create button. You want to edit an action in the rule to map a variable from the Context Browser to the Folio. Click to highlight the **Leave Request Workflow (Default)** state, then edit the **...when Create Button is Clicked** rule. (Remember, the Leave Request Workflow (Default) state is the form behavior that the requester sees when they want to submit a new leave request).

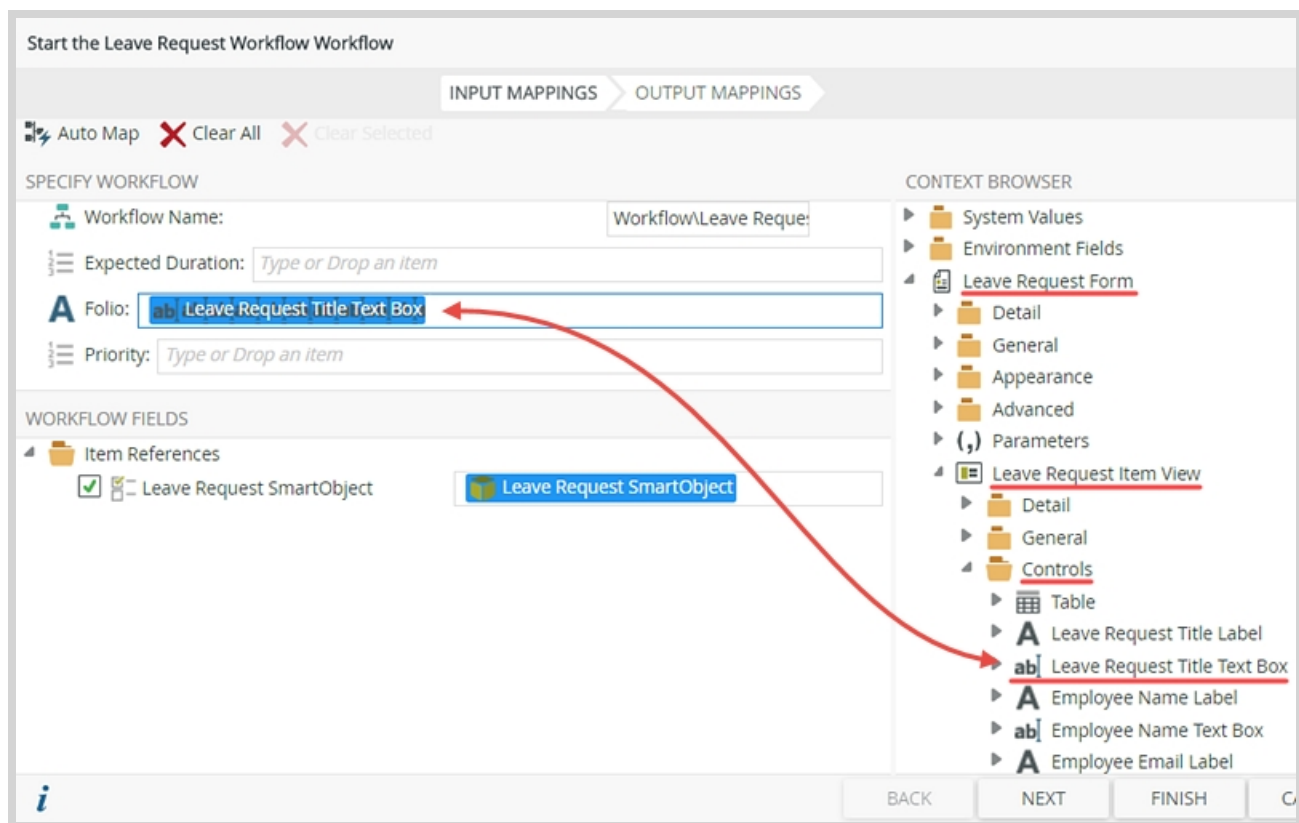


2. Locate the action that starts the workflow, then edit it by clicking the (configure) link. Map the **Context Browser > Leave Request Item View > Controls > Title Text Box** control to the **Folio**.

- a. Next, you will edit the action that starts the workflow. When the workflow starts, you want K2 to take the value from the Leave Request **Title Text Box** (on the form) and make it the **Folio** value. On the rule definition pane, locate the action **then start the Leave Request Workflow Workflow**. To edit this action, click the **(configure)** link.



- b. Expand the **Context Browser > Leave Request Form > Leave Request Item View > Controls** nodes. Drag the **Leave Request Title Text Box** into the **Folio** field. Once again, when the workflow starts, K2 will take the value from the Title Text Box and assign it as the Folio value. Click **FINISH**, then click **OK** to close the Rule Designer. Keep the form open.



Review

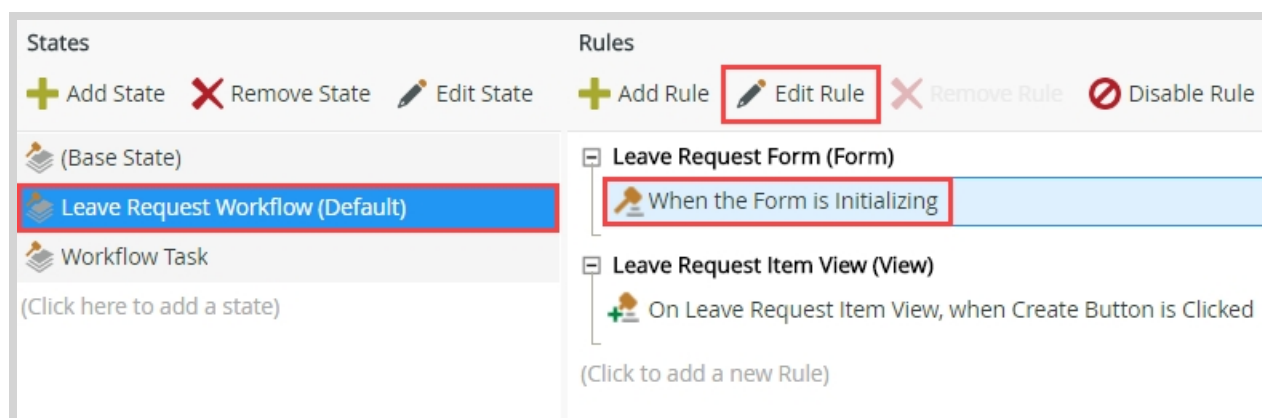
In this step, you set the folio value. While not required, the folio is a common way to distinguish instances of the same workflow. In the next step, you will configure a rule so that the form opens with the current user's name and email pre-populated. This rule will also populate the user's previous leave requests at the bottom of the form.

Next Step: 13. Set the Current User's Details on the Request Form

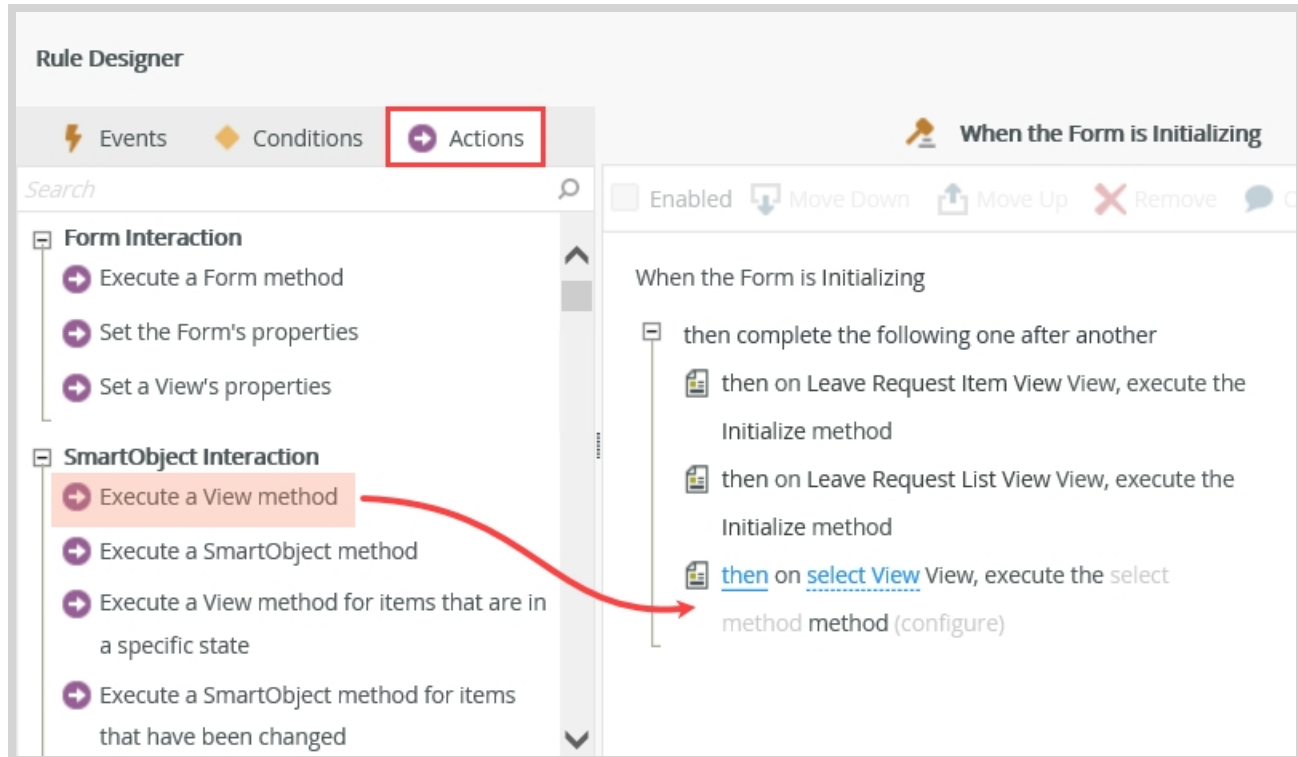
13. Set the Current User's Details on the Request Form

In this step you will edit the Leave Request Workflow (Default) state's "form initializing" rule, and add an action to load the current user's previous leave requests into the Leave Request List View. Transfer the current user's system values (Email and Display Name) to the Leave Request Item View. K2 keeps track of the current user's system details such as their display name, email, department and manager as system values. You will use two of these system values to personalize the Leave Request Form. When you launch the form, K2 will displays your name and email automatically.

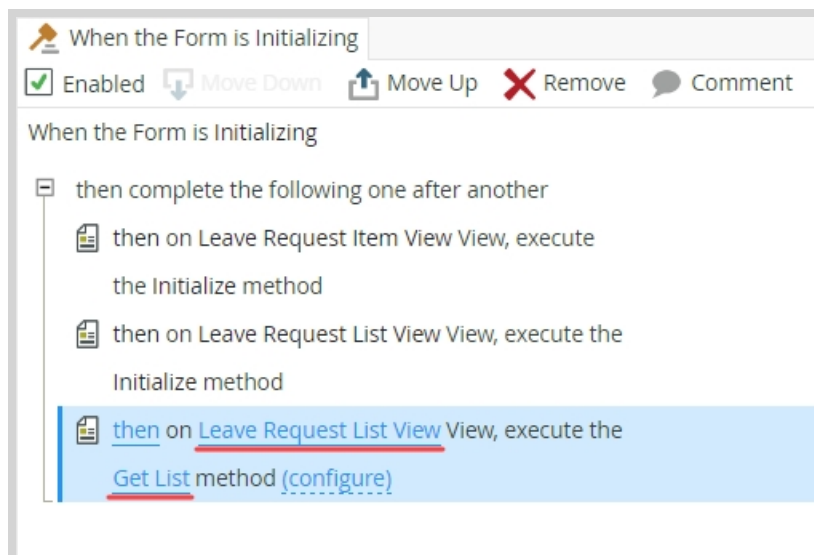
1. First, you will add an action to retrieve the current user's previous leave requests. You will populate the Leave Request List View these records as the form loads. Edit the **form initializing rule** for the **Leave Request Workflow (Default)** state on the Leave Request Form. Add an **Execute a View method** action for the **Leave Request List View, Get List** method.
 - a. Begin by editing the rule that runs when the form first loads, or initializes, for the leave requester. With the **Leave Request Workflow (Default)** state still highlighted, edit the **When the Form is Initializing** rule.



- b. Confirm the **Actions** tab is active, then click **Execute a View method** to add it to the rule definition pane.



- c. Click the **select View** link and select **Leave Request List View**. This is the view that displays the previous leave requests at the bottom of the form.
- d. Click the **select method** link and select **Get List**. The Get List method does just that, retrieves records from a SmartObject. But you do not want ALL records returned, so you will configure the action to filter the records by the employee's email.

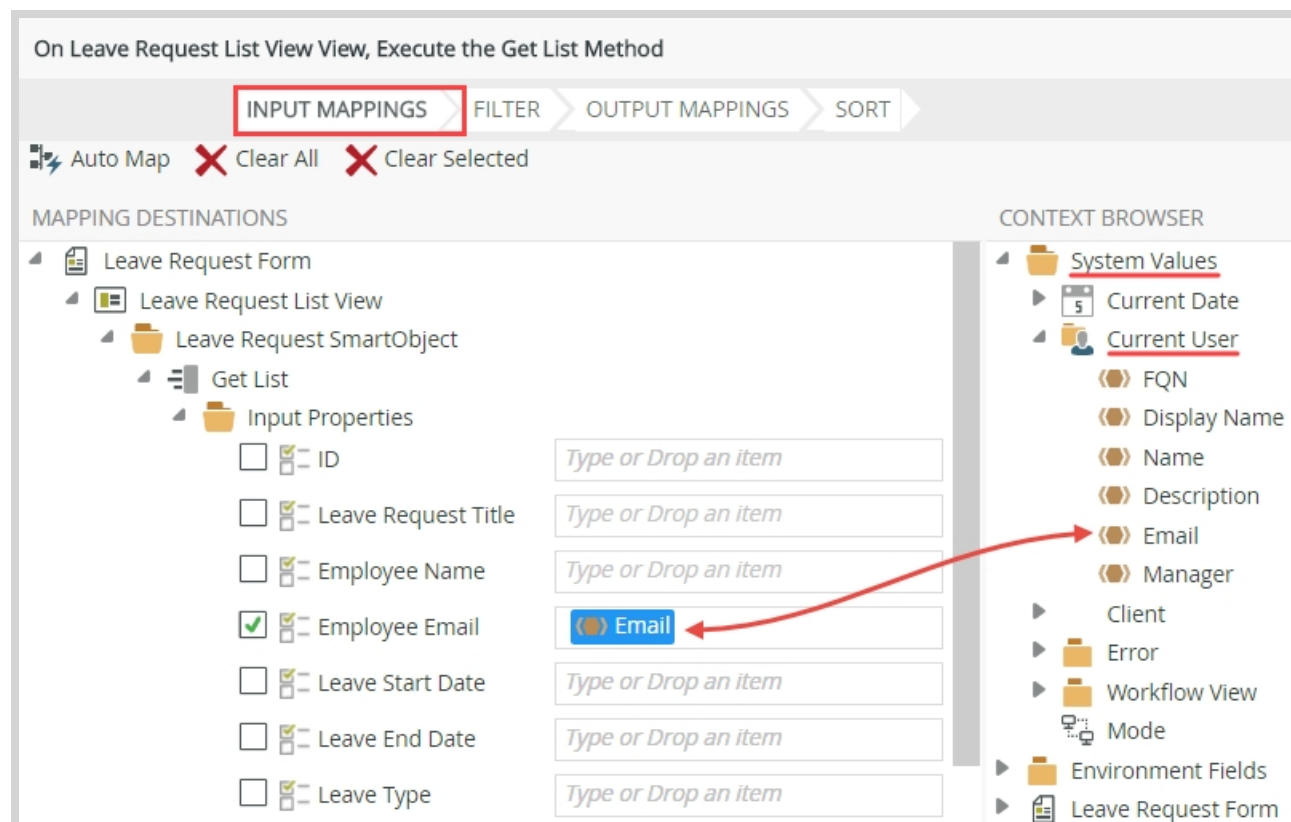


2. Map the **system value (current user) email** to the **Employee Email** input property.

- a. Now you need to tell K2 *which* SmartObject records you want returned. You want the current user's records. You will set up a filter so that K2 only returns the current user's leave requests. For this

tutorial, you will use the email as the filter, since email addresses are unique. This step also highlights why the Employee Email a data label and not an editable text box. The user cannot change a data label. The data label value is the user's system email value so it will remain consistent throughout all the records.

Click the **(configure)** link. On the **Input Mappings** screen, expand the **Context Browser > System Values > Current User** nodes, then drag the **Email** property into the **Employee Email** field. The input mapping value is the filter K2 uses when returning records.

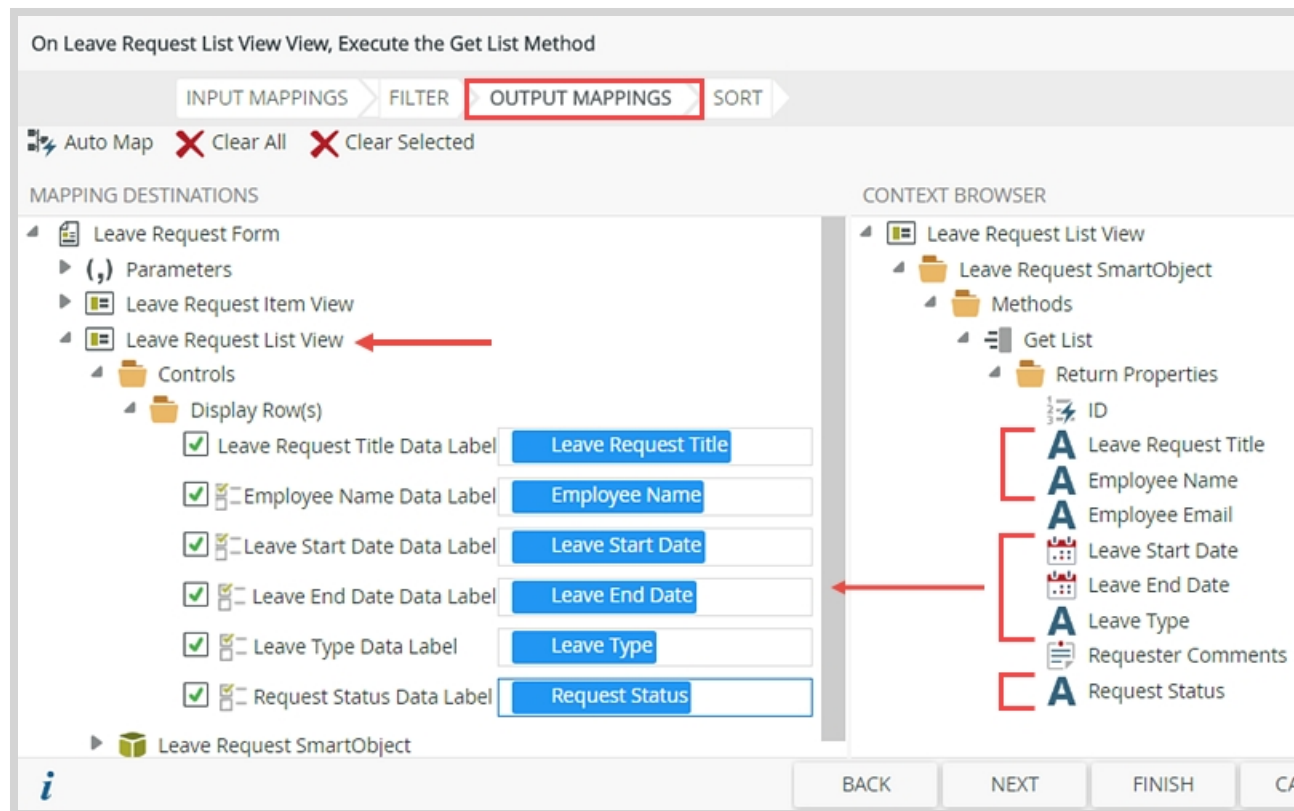


3. Map the **return properties** to the **Leave Request List View** output mappings.

- a. Now you will map the return properties K2 will take the filtered records it finds and populate the list view with the SmartObject records..

Click the **Output Mappings** button. Drag the **Return Properties** into the **Leave Request List View** fields. These are the properties that become the list view values. You may have to collapse the item view to see the list view in the Mapping Destinations pane. Click **FINISH**.

When you created the list view, you selected the columns you wanted to include. The **Display Row** fields below are those columns. You are mapping the SmartObject properties to their corresponding list view columns.



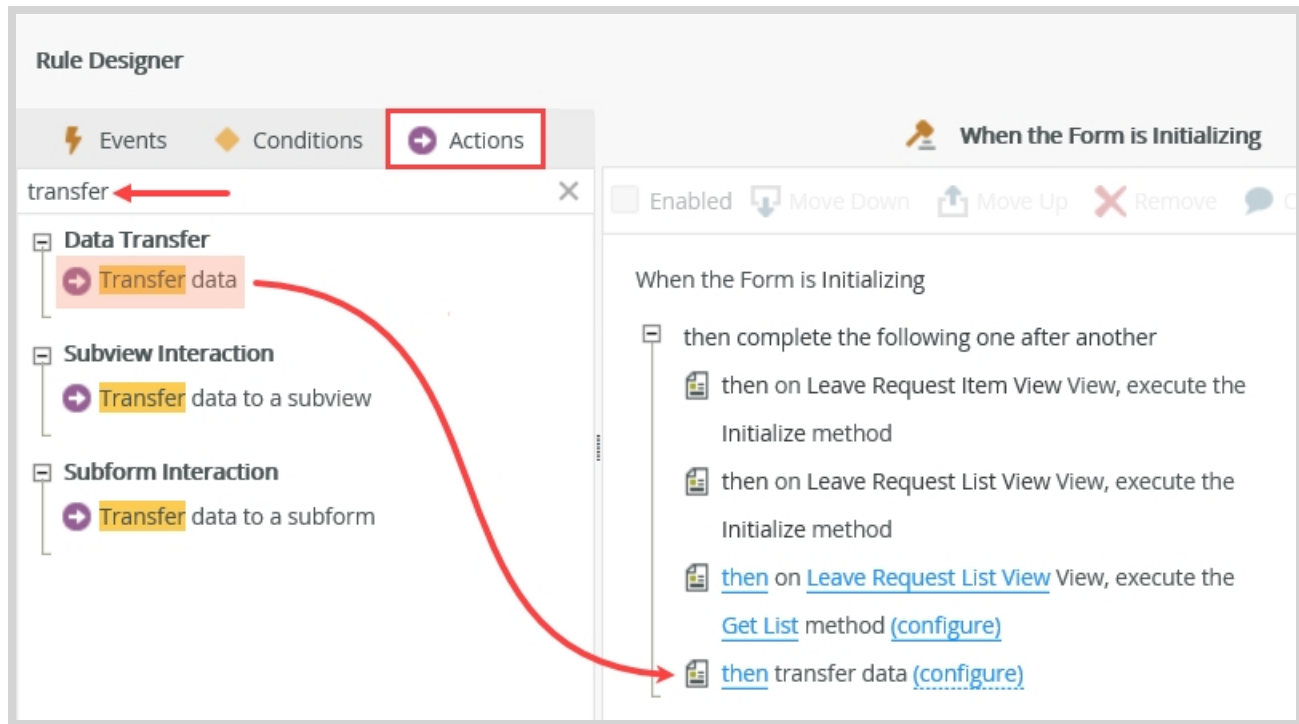
4. Add a **transfer data** action. Map the system current user **display name** and **email** to the appropriate item view controls.

- Now you will map the current user's name and email to the form so that the values load when the form loads. This is not required, but is a nice feature for your users. It also forces the leave request records to match system values. In this case, you need a consistent email address for the leave requester. As you become more familiar with rules, you can search for keywords to locate the event, condition or action you need. This eliminates having to scroll through the rule options.

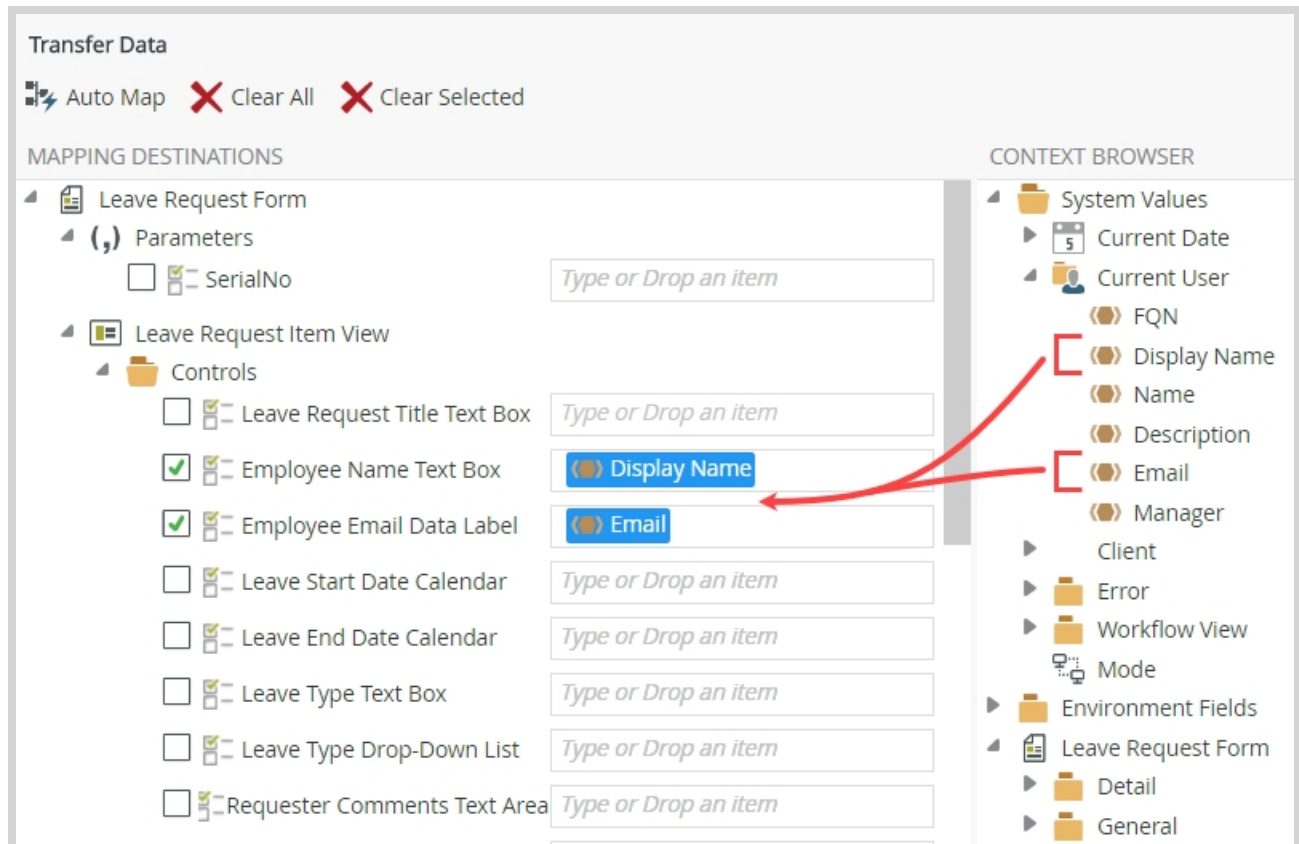
With the **Actions** tab still active, search the keyword

transfer

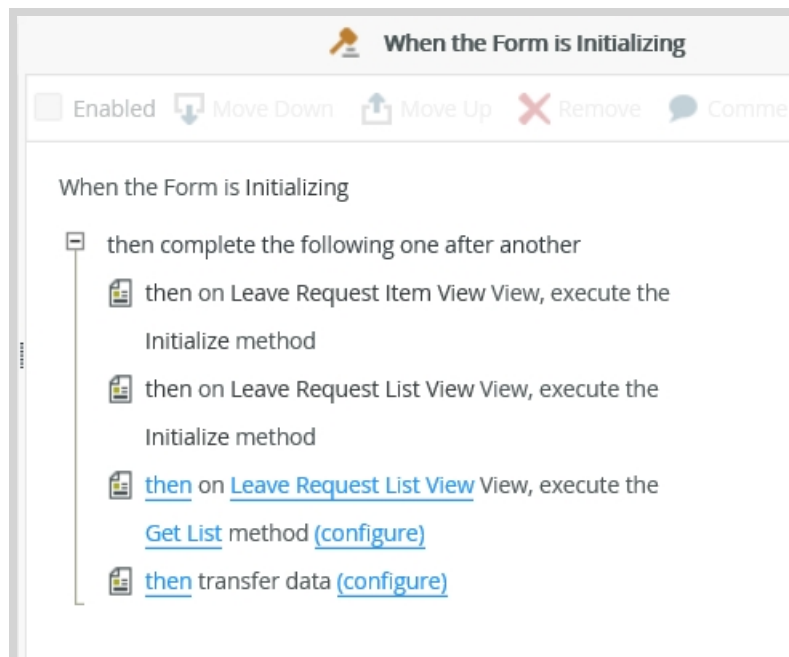
then click on **Transfer data** to add it to the rule definition pane.



- b. Click **(configure)**. Expand the **Context Browser > System Values > Current User** nodes. Drag the **Display Name** into the **Employee Name Text Box** field. Drag the **Email** into the **Employee Email Data Label** field. Click **OK**.



- c. The completed **When the Form is Initializing** rule should look like the image below. Click **OK** to exit the Rule Designer. Keep the form open, as you will work with rules in the next section.



Review

In this step, you edited the form initializing rule and added two actions. The first action retrieves the current user's leave requests, then populates the list view with the records found. You used the requester's email to filter the SmartObject records. The second action retrieves the current user's system value display name and email. When the requester launches the Leave Request Form, the name and email fields is pre-populated. In the next step, you will work with the state the approving manager sees. This time, you will populate the form with the leave requester's previous leave requests.

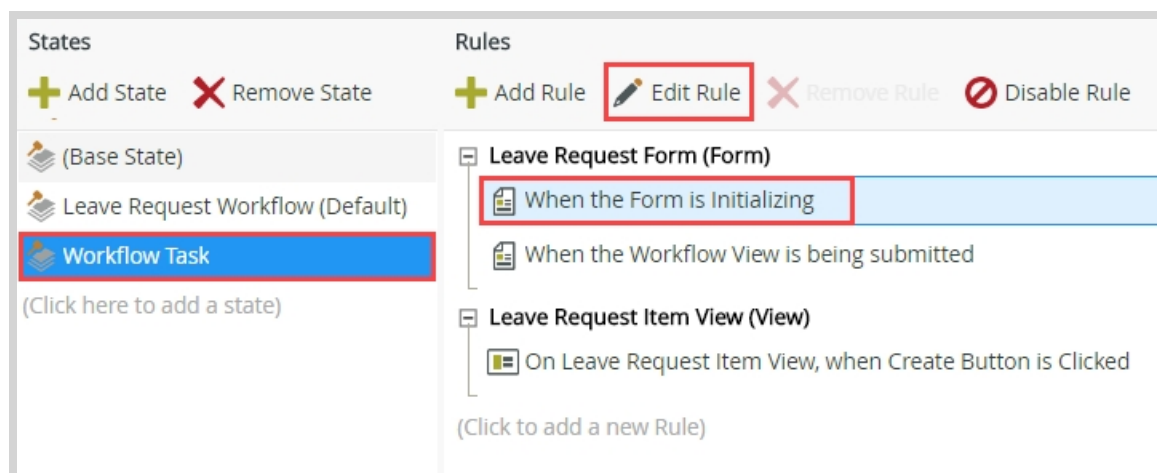
Next Step: 14. Set the Requester's Details on the Manager's Approval Form

14. Set the Requester's Details on the Manager's Approval Form

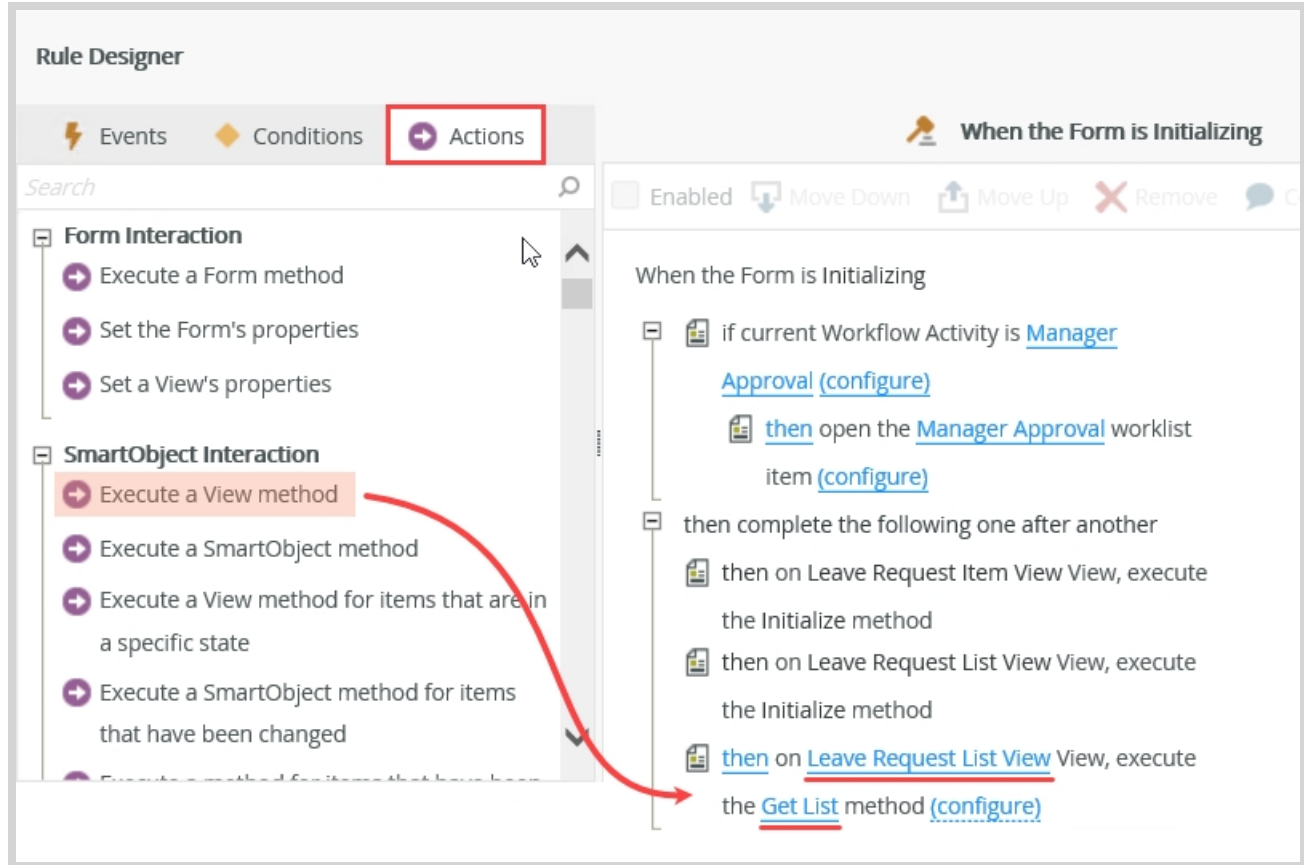
In this step, you will edit the Workflow Task state, form initializing rule, and add an action to load the leave requester's previous leave requests into the Leave Request List View. In the last step, you configured the form initializing rule for the *default* state, or the form configuration the *requester* sees. You configured the rule to use the current user's system email value as the SmartObject record filter. The Workflow Task state is the configuration the approving *manager* sees. You don't want the *current user's* records for the approval step though, because the current user is the manager! You will configure the rule to use the *requester's* email as the filter and not the current user's system email value.

You will also hide the **Create** button for the approving manager as the manager will not add any new records.

1. Like the previous step, you will first add an action to retrieve the requester's leave requests. Edit the **form initializing** rule on the **Workflow Task** state. Execute a **get list** method for the **Leave Request List View**.
 - a. Begin by editing the rule that runs when the form first loads, or initializes, for the approving manager. Highlight the **Workflow Task** state, then edit the **When the Form is Initializing** rule. This state represents the form and rule configuration for the approving manager.



- b. Confirm the **Actions** tab is active, then add an **Execute a View method** action to the rule definition pane. For the view, select **Leave Request List View**. For the method, select **Get List**.



2. Next, you will map the Employee Email property from the SmartObject instead of the System Value email. Map the **Employee Email** property from the Leave Request Item View > **Leave Request SmartObject** to the **Employee Email** input mapping.

- a. Recall in the last section, you configured the system email value as the filter for the Previous Leave Requests list view. The system value email represents the current user's email value. You do not want the current user's previous leave requests because the current user for this state is the approving manager! Therefore, you need to map the email address from the SmartObject record itself that is the email for the leave requester. You want to retrieve the leave requester's previous requests for the manager to review.

Click **(configure)**. Expand the **Context Browser > Leave Request Form > Leave Request Item View > Leave Request SmartObject** nodes. Drag the **Employee Email** into the **Employee Email** input

property.

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

INPUT MAPPINGS FILTER OUTPUT MAPPINGS SORT

Auto Map ✗ Clear All ✗ Clear Selected

MAPPING DESTINATIONS

- Leave Request Form
 - Leave Request List View
 - Leave Request SmartObject
 - Get List
 - Input Properties
 - ID ☐
 - Leave Request Title ☐
 - Employee Name ☐
 - Employee Email ☒
 - Leave Start Date ☐
 - Leave End Date ☐
 - Leave Type ☐
 - Requester Comments ☐

CONTEXT BROWSER

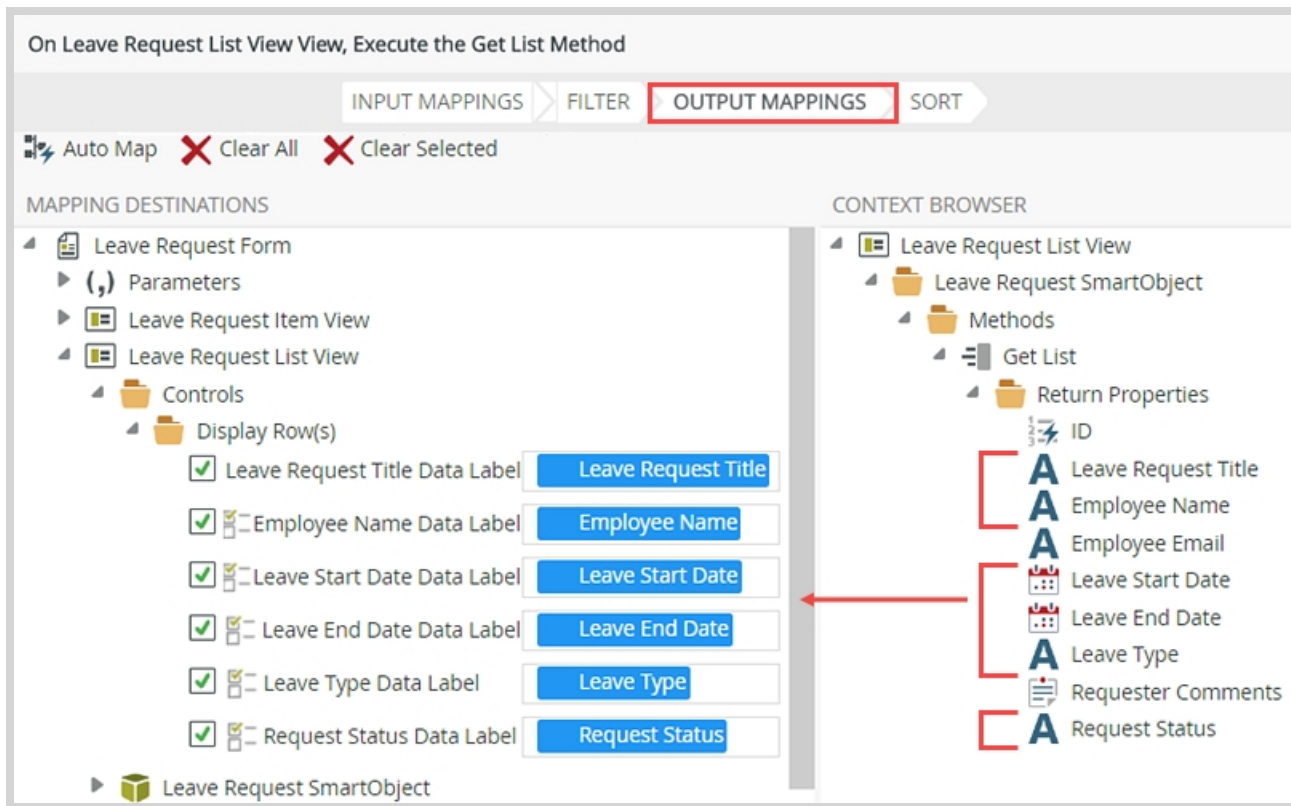
- System Values
- Environment Fields
- Leave Request Form
 - Detail
 - General
 - Appearance
 - Advanced
 - Parameters
 - Leave Request Item View
 - Detail
 - General
 - Controls
 - Leave Request SmartObject
 - ID
 - Leave Request Title
 - Employee Name
 - Employee Email
 - Leave Start Date
 - Leave End Date
 - Leave Type
 - Requester Comments
 - Request Status
 - Leave Request List View

3. Map the **return properties** to the **Leave Request List View** output mappings.

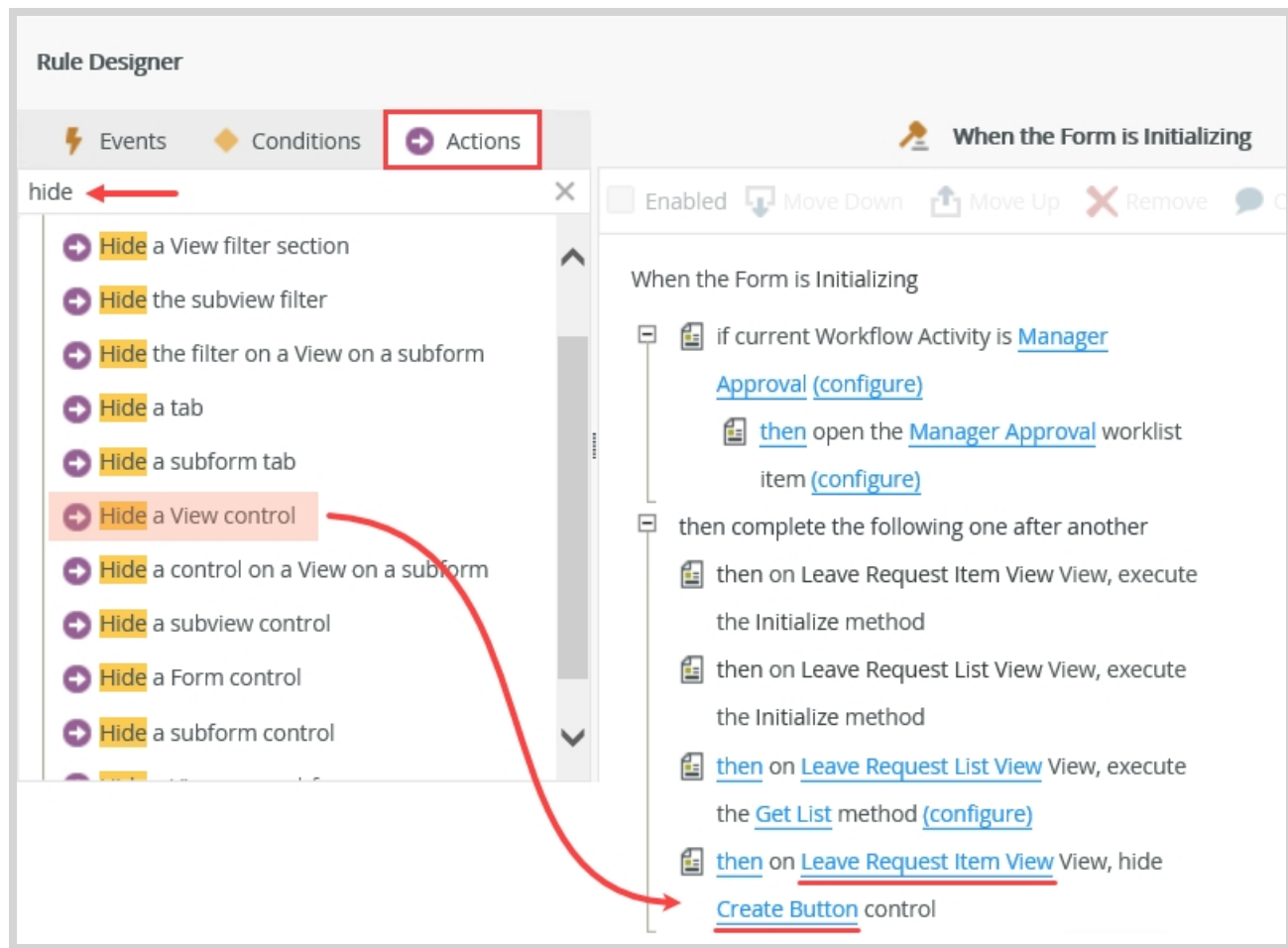
Like previous step, you will map the output mappings. These are the properties that ultimately become the list view values. Map the **Return Properties** to the **Leave Request List View > Controls > Display Row(s)** properties.

- Click **Output Mappings**. Map the **Return Properties** to the **Leave Request List View** controls. You may have to collapse the item view to see the list view in the Mapping Destinations pane. Click

FINISH.



4. The last rule configuration you will make is to **hide the create button** on the item view. The approving manager will not add any new records, so this button may be confusing. After you hide the control, **finish**, then **check in** the form.
 - a. Next, you will hide the create button. Remember that this state is the form behavior that the approving manager sees. The manager will not create any new leave requests, they are just reviewing and taking action on the requester's leave requests. Since this button may be confusing if it's displayed, you will hide it for this state. With the **Actions** tab active, search for *hide* then click **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 save and exit the form.



- b. Right-click the **Leave Request Form** and select **Check In**. If you see a message about checking in associated views, check the boxes next to the views, then click **OK**.

Review

In this step, you edited the form initialing rule for the Workflow Task state. This is the form configuration and behavior the approving manager sees. You added an action to execute a view method (Leave Request List View, Get List method) to populate the list view with the *requester's* leave requests (instead of the *current user's* records). You hid the create button from this state, because the manager will not create new records. The workflow configuration is now complete! In the next step, you will test the Leave Request application.


Next Step: 15. Test the Leave Request (Basic Version) Application

15. Test the Leave Request (Basic Version) Application


In this step you will test the Leave Request (Basic Version) application by submitting two leave requests. As the approver, you will open the workflow task form, then respond to the first request. After responding to the first request, you will open the workflow task form for the second request and confirm the list view displays correct request status. If time allows, submit several more leave requests so that you have content to administer or report against in other tutorials.


1. From the K2 Designer, **submit two leave requests** using the **Runtime URL**. The Runtime URL launches the most current version of the form published to the K2 server. Use this link on web pages and other locations to start the Leave Request Form.
 - a. Return to K2 Designer. Navigate to your **Leave Request** category. Click to highlight the **Leave Request Form**. From the central pane, click the **Runtime URL**. This will launch the most current version of the form in a web browser.


[All Items](#) > [K2 Learning](#) > [Leave Request](#) > [Forms](#) > [Leave Request Form](#)





Leave Request Form


**Edit**
Design this Form.


**Run**
Run this Form in this Window

**Run with Parameters**
Set parameters before running this Form



**Save As**
Create a copy of this Form.

**Check Out**
Check this Form out.

**Design a new Workflow**
Create a Workflow using fields in this Form.

**Delete**
Completely remove this Form.

System Name
Leave Request Form

Browse
[Testing URL](#)  [Runtime URL](#) 

Category
K2 Learning\Leave Request\Forms

Version
2.0

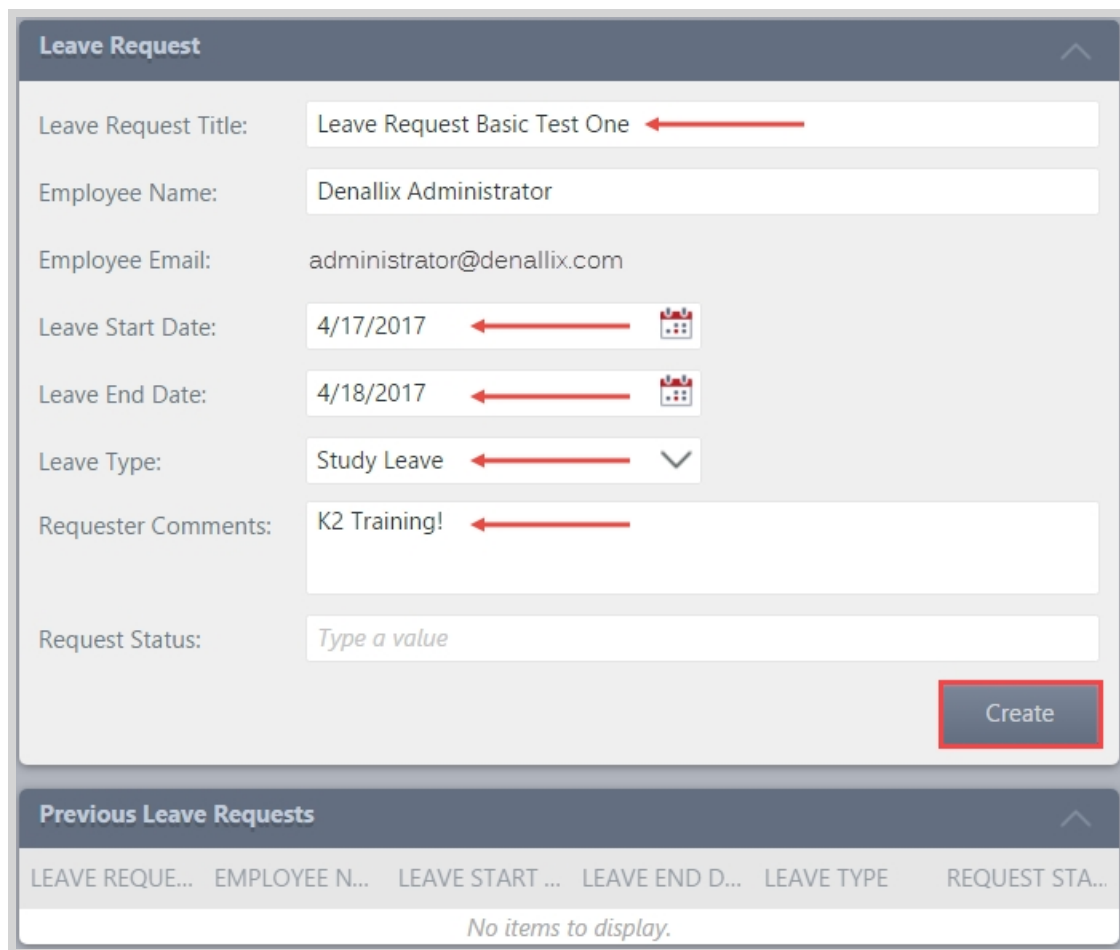
Created By
[Denallix Administrator](#) , 1/30/2018 2:45 PM

Last Modified
[Denallix Administrator](#) , 2/1/2018 9:33 AM

Status:
Checked in

- b. The form opens. The Employee Name and Employee Email fields should be pre-populated with the values of the currently logged-in user. Complete the remaining form fields, except the Request Status field. The workflow updates this value (using the SmartObject Method steps). Notice there are no pre-

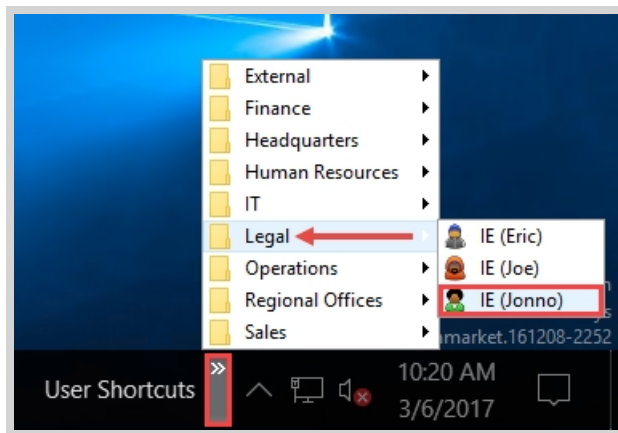
vious leave requests just yet. Click **Create**.



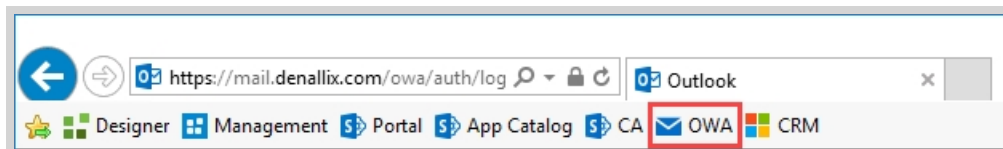
- c. When the spinner disappears, the form submitted. For now, there is no formal indicator that the form submitted. You will add those features in the extended tutorial. **Close** the browser tab.



2. As the approving manager, action the first leave request submittal. From the approving manager's mailbox, open the first task notification email, then click the **Worklist Item** link. The Worklist Item link opens the Workflow Task form. (Remember, that is the state you created for the approving manager's form.) Respond to the request with one of the actions (Approved or Denied).
Next, you will access the manager's Outlook to view the task notification emails. If you are working on a K2-provided VM, you are likely logged in as Denallix Administrator. Administrator's manager is Jonno, so you will open Jonno's Outlook. If you are working in your own environment, access the Outlook account for your manager (or other person if you configured the task to route to someone else).
 - a. If using a K2-provided VM, close all open browser windows. Click the **User Shortcuts** arrows found in the lower-right corner of the screen. Expand the **Legal** folder. Click **IE (Jonno)**.



- b. Click the **OWA** (Outlook Web Access) link in the Favorites bar.

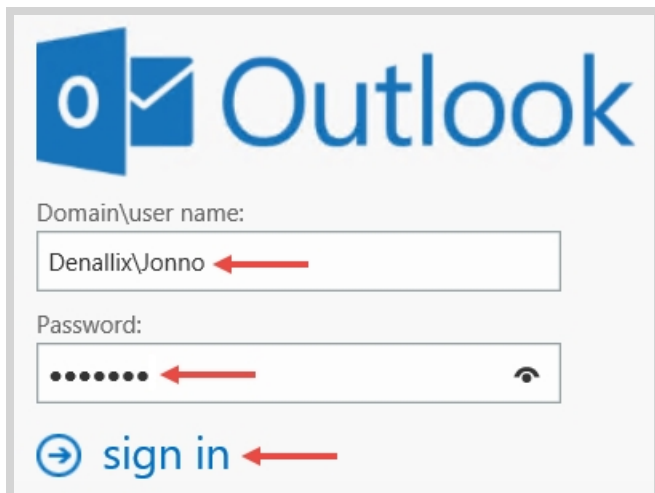


- c. When OWA launches, enter Jonno's login credentials.

Domain\user name: *Denallix\Jonno*

Password: *K2pass!*

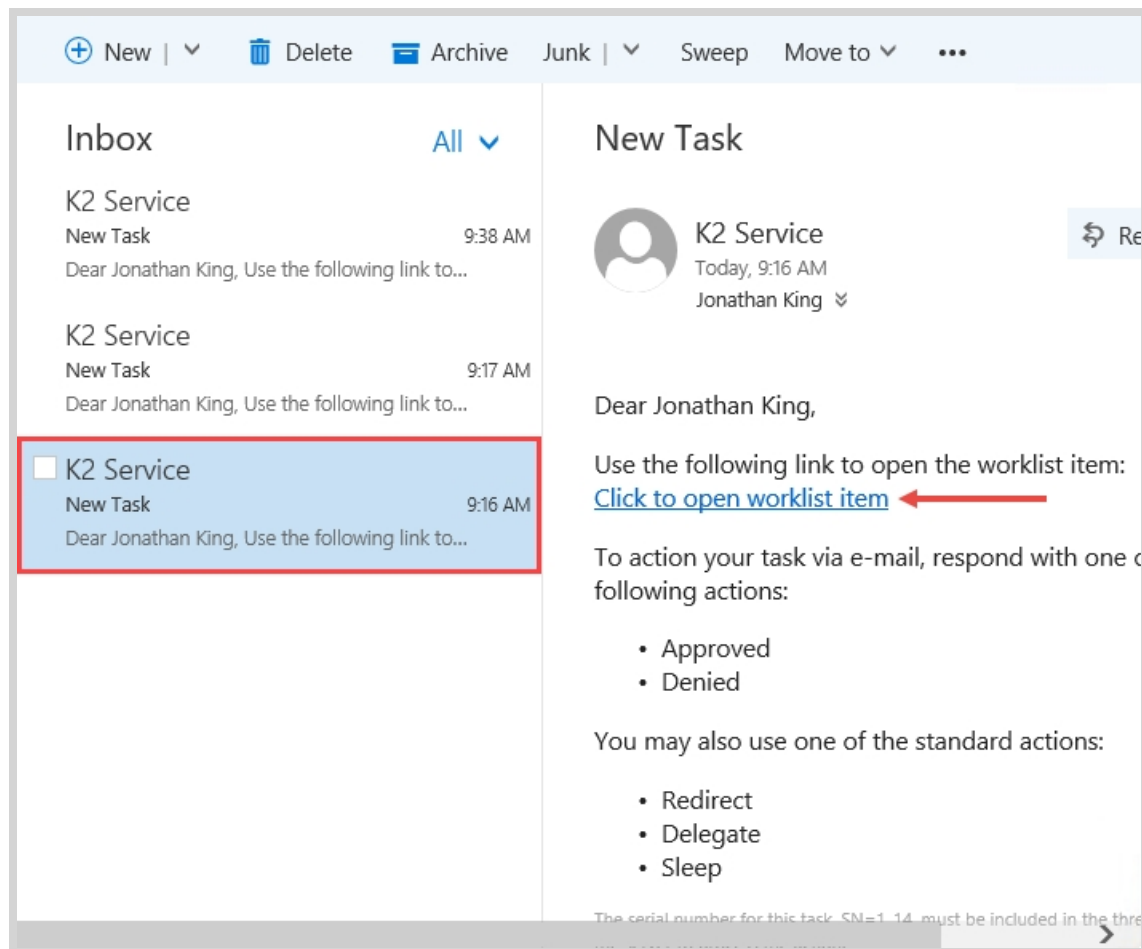
then click **sign in**.



Note

If you see Outlook settings such as Language and Time zone, select the Pacific time zone, then click **Save**.

- d. You should see task notification emails for each of the leave requests you submitted. **Open** one of the emails, then click the **worklist item** link to open the manager's approval form.



- e. Review the request details. Notice that K2 has added the **Workflow** view to the top of the form. This view contains the actions available to the approver. From the **Select Action** list, choose an action, the

click **Submit**.

Workflow

Folio:

Basic Test One

Activity Name:

Manager Approval

Instruction:

A leave request has been submitted. Open the form, then select your approval decision.

Select Action:

Approved

Submit

Leave Request

Leave Request Title:

Basic Test One

Employee Name:

Denallix Administrator

Employee Email:

administrator@denallix.com

Leave Start Date:

3/14/2017

Leave End Date:

3/17/2017

Leave Type:

Study Leave

Requester Comments:

K2 Training!

Request Status:

Submitted

Previous Leave Requests

LEAVE REQUEST TI...	EMPLOYEE NAME	LEAVE START DATE	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
Basic Test One	Denallix Administr...	3/14/2017	3/17/2017	Study Leave	Submitted
Basic Test Two	Denallix Administr...	3/20/2017	3/23/2017	Paid Time Off	Submitted
Basic Test Three	Denallix Administr...	3/13/2017	3/16/2017	Study Leave	Submitted

f. You will see a generic message the that form submitted, click **OK**. **Close** the browser tab.

3. From the manager's mailbox, open the second task notification email, then click the **Worklist Item** link. Confirm the list view at the bottom of the form displays the previous leave request. Confirm the Request Status reflects the action you took in the previous step.
 - a. Launch a worklist item from a different email. Confirm that the **Previous Leave Requests** list view reflects the decision you just made.

Previous Leave Requests					
LEAVE REQUEST TI...	EMPLOYEE NAME	LEAVE START DATE	LEAVE END DATE	LEAVE TYPE	REQUEST STATUS
Basic Test One	Denallix Administr...	3/14/2017	3/17/2017	Study Leave	Approved
Basic Test Two	Denallix Administr...	3/20/2017	3/23/2017	Paid Time Off	Submitted
Basic Test Three	Denallix Administr...	3/13/2017	3/16/2017	Study Leave	Submitted

Review

In this step, you tested your basic Leave Request application by submitting requests. You responded to one of the tasks as the manager. When submitting another request, you could see where the workflow updated the status property with the response value. In the next optional step, you learn how to clean your K2 environment by deleting the application you built in this tutorial.

Next Step: 16. (Optional) Clean your K2 environment by deleting application artifacts

16. (Optional) Clean up your K2 environment by deleting the application artifacts

Caution

If you intend to work through the **Leave Request Extended** tutorial, DO NOT complete this step! The extended version builds upon the artifacts from the basic version.

This optional step describes how you can clean your K2 environment by deleting the artifacts that you created in this tutorial. For example, if you are working in a shared environment, you may want to delete your tutorial applications to keep your environment clean and tidy.

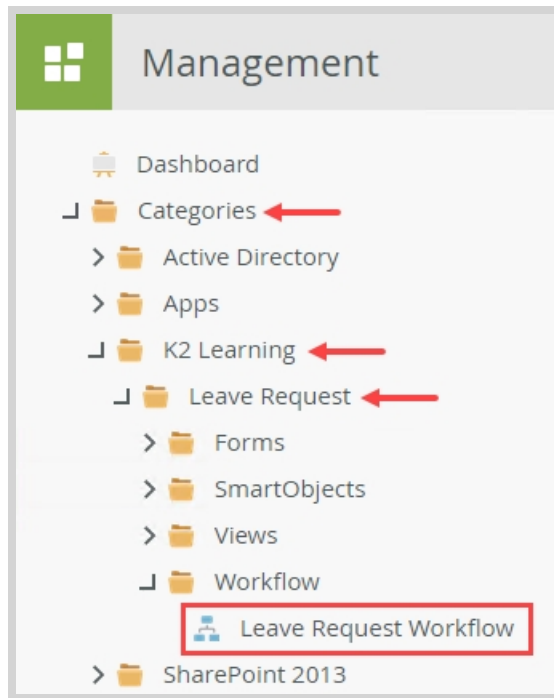
Caution

Deleting artifacts (such as views, forms, SmartObjects, workflows, service instances and so on) removes them from the K2 server permanently and you cannot retrieve or restore them easily. You should only delete artifacts once you are certain that they are no longer needed. Also, once you start deleting application artifacts, other application artifacts might stop working, because they depend on the item that you just deleted.

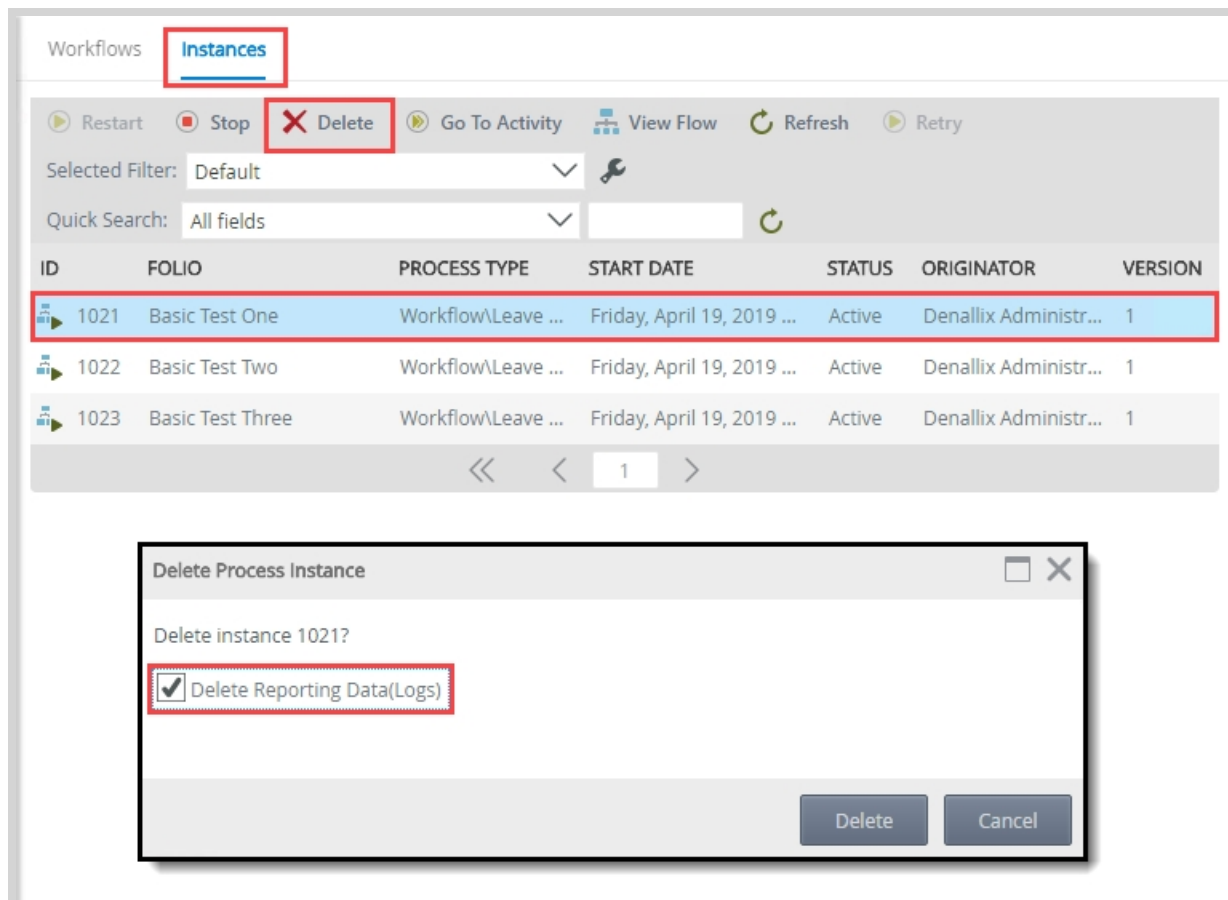
Use the K2 Management site to delete deployed workflows

Begin by deleting the deployed versions of the Leave Request Workflow from the K2 server. First, you must delete any active process instances, then delete the deployed workflow versions.

1. From the **K2 Management** site, delete any process instances associated with the Leave Request Workflow. You are deleting any process instances that are active, running, stopped, or in error. You can also choose to remove reporting data as well.
Location: **Categories > K2 Learning > Leave Request > Workflow > Leave Request Workflow > Instances** (remember that **K2 Learning** might have a different name in your environment).
 - a. From the **Categories** menu, navigate to the **K2 Learning > Leave Request** node (remember that **K2 Learning** might have a different name in your environment). Expand the **Workflow** category and select **Leave Request Workflow**.



- b. In the central pane, click the **Instances** tab. Check to see if there are any active, running, stopped, or erred process instances. If there are, select them one-by-one and click the **Delete** button (there may be more than one instance). You see a pop-up where you have the option to delete reporting logs. Check the **Delete Reporting Data (Logs)** box, then click **Delete**. This removes any reporting data generated for the process instance.



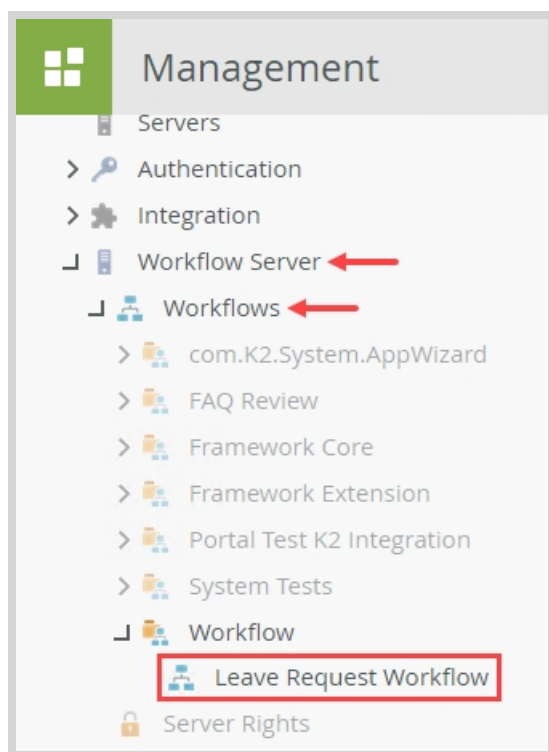
- c. Repeat the step above until you have deleted all process instances for the **Leave Request Workflow**.
2. Delete all versions of the workflow. You may see more than one version and you have the option to delete all versions or specific versions. In this case, you delete all versions along with the historical data and workflow reporting SmartObjects.

Location: **Workflow Server > Workflows > Workflow > Leave Request Workflow > Versions**.

Note

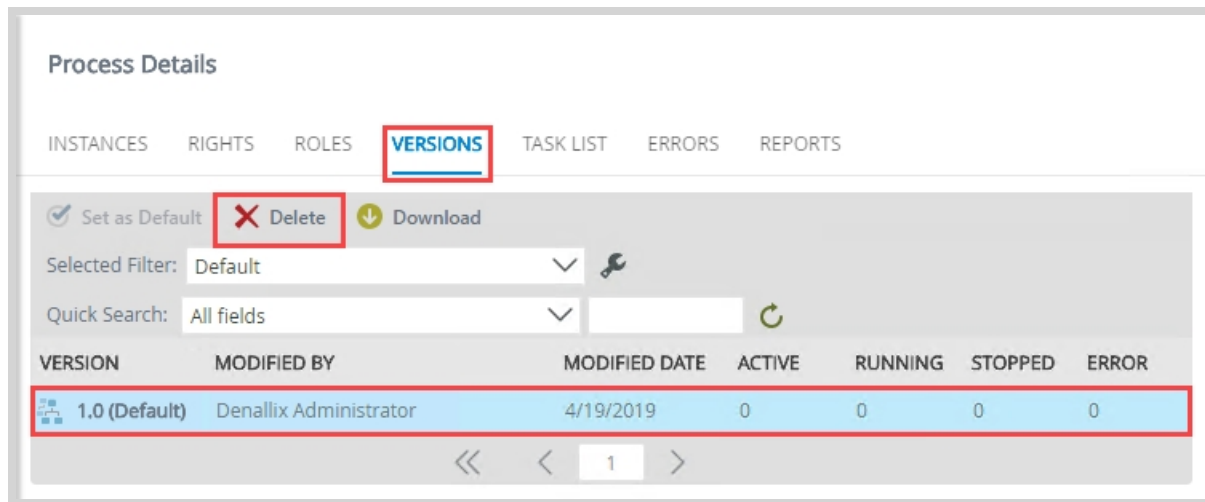
Your artifact names may not match the steps exactly if you added your initials to make them unique. When an artifact name is referenced, use the name you entered when creating it. Artifacts include views, forms, SmartObjects, workflows, service instances, roles, groups, and SharePoint lists and libraries.

- a. Expand the **Workflow Server > Workflows** node. Expand the **Workflow** category and select **Leave Request Workflow**.

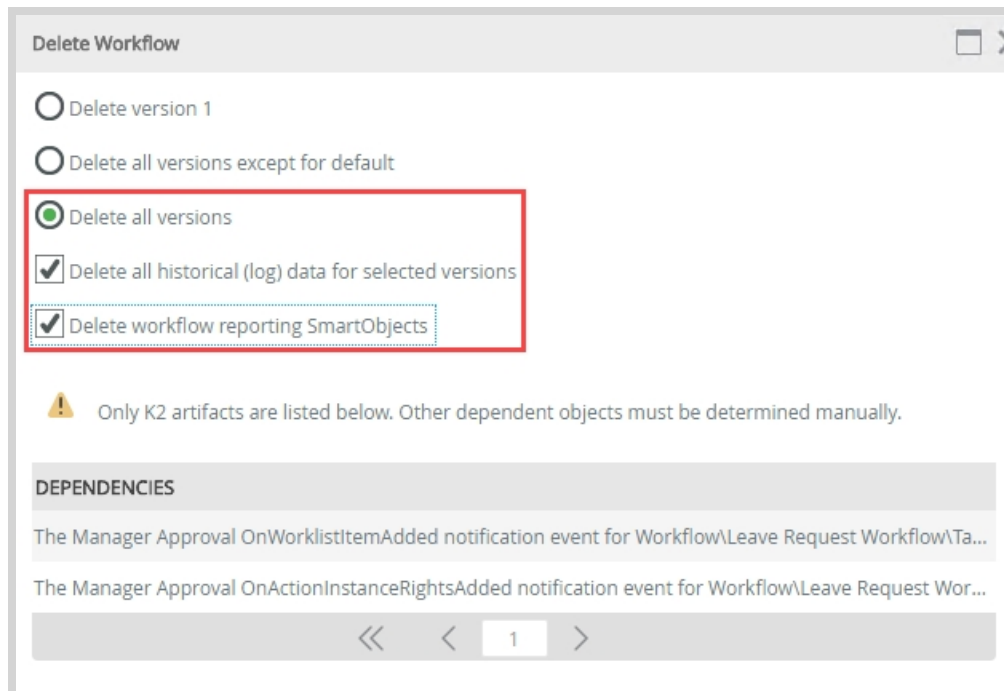


- b. In the central pane, click the **Versions** tab. In the image below, there is only one version of the Leave Request Workflow. You may have several versions, depending on how many times you deployed the workflow. Notice the columns for Active, Running, Stopped, and Error instances. You should see

zeros here, since you deleted all process instances in the prior step. Select a version and click **Delete**.



- c. On the **Delete Workflow** screen, select **Delete all versions**, then check the option to delete all **historical data** and **workflow reporting SmartObjects**. Click **OK**. Repeat this step until you have deleted all versions of the workflow.



Use K2 Designer to delete application elements

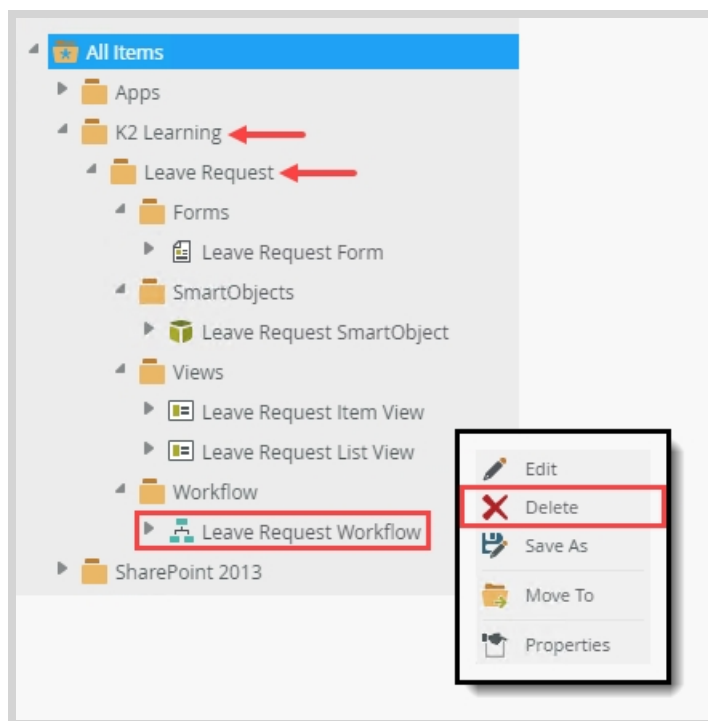
The next step is to remove application elements using K2 Designer. Here, you can remove views, forms, SmartObjects, workflow design files and categories.

- In **K2 Designer**, delete the **Leave Request Workflow** design file. (In the previous step, you removed workflows published to the K2 server. In this step, you are removing the "design" copy of your workflow.)

Location: **All Items > K2 Learning > Leave Request > Workflow**. Remember that the **K2 Learning** category might have a different name in your environment.

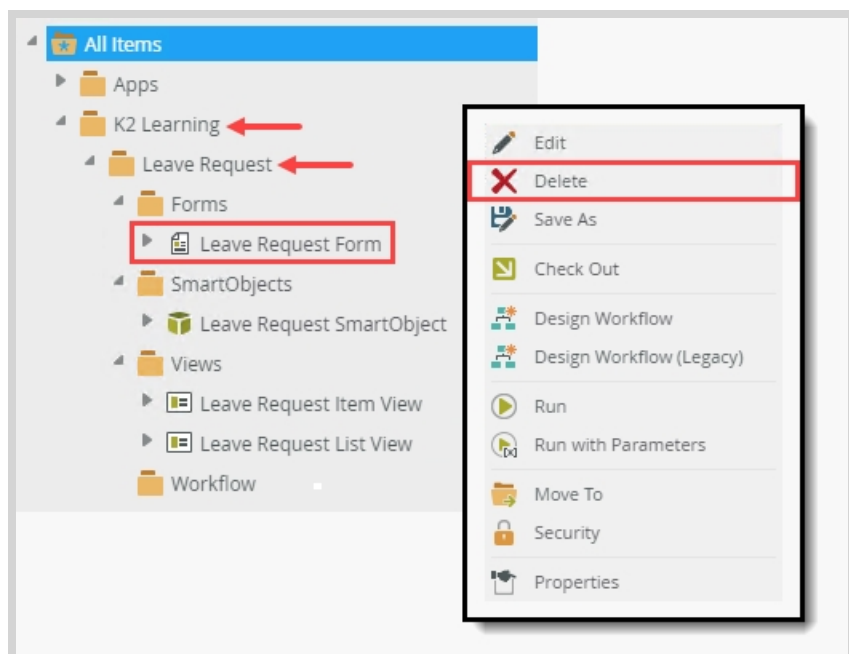
- Return to **K2 Designer**. Navigate to your **K2 Learning > Leave Request** categories - remember that

the K2 Learning category might have a different name in your environment . Expand the **Workflow** category and select **Leave Request Workflow**. Right-click and select **Delete**. Click **Yes** for the confirmation.



4. Delete the **Leave Request Form**. (You must delete forms first, followed by the views used on those forms.)
Location: **All Items > K2 Learning > Leave Request > Forms**.

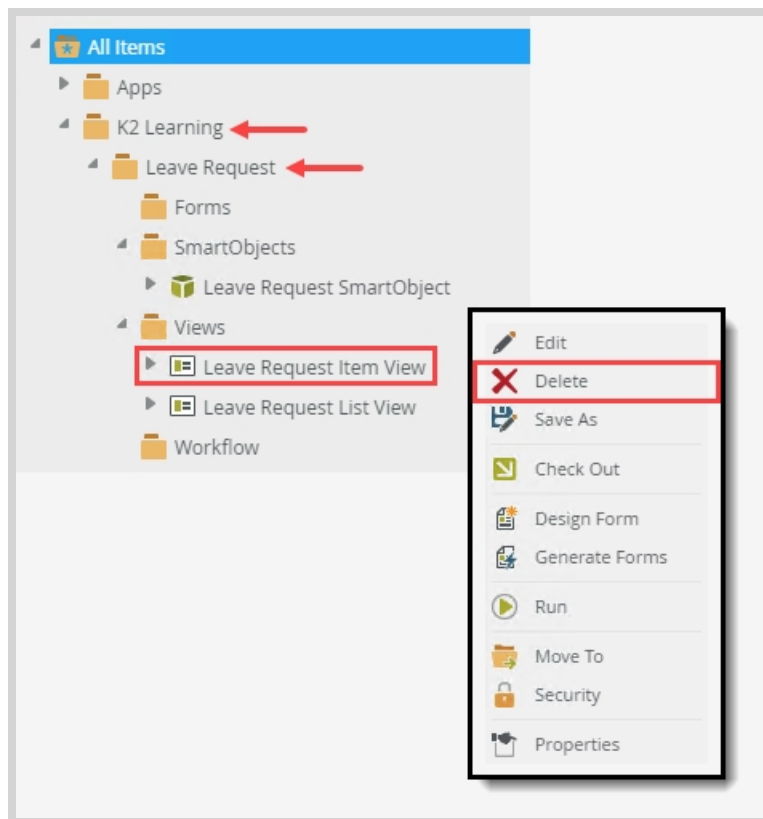
- Expand the **Forms** category.
- Right-click **Leave Request Form** and select **Delete**. Click **Yes** for the confirmation. If you see a message about associated workflows, you can ignore it and continue.



5. Delete each **view**. (There is no specific order for deleting views).

Location: **All Items > K2 Learning > Leave Request > Views**.

- a. Expand the **Views** category. Right-click each view and select **Delete**. There is no specific order you must follow for deleting views. Click **Yes** for each confirmation.

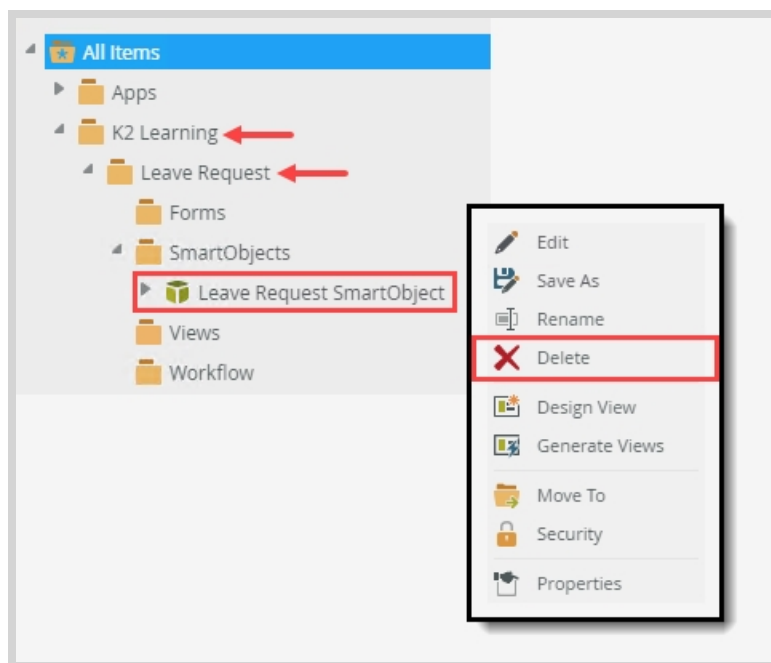


6. Delete the **Leave Request SmartObject**.

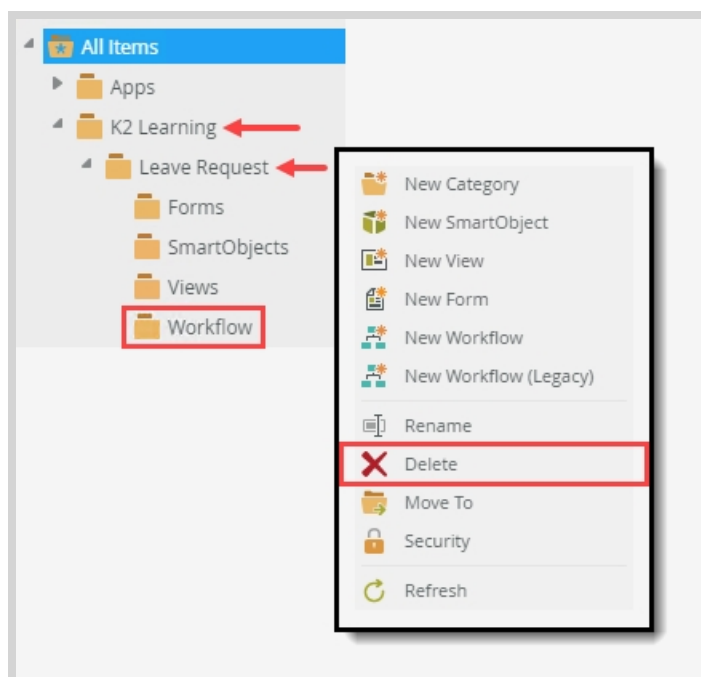
Location: **All Items > K2 Learning > Leave Request > SmartObjects**.

In the next step, you delete the Leave Request SmartObject.

- a. Expand the **SmartObjects** category. Right-click the **Leave Request SmartObject** and select **Delete**. Click **Yes** for the confirmation.



7. Delete the learning categories. You must delete the subcategories first (**Forms, Views, SmartObjects, Workflow**) and then **Leave Request**. Finally, you can delete **K2 Learning** if you do not intend building more tutorials, and remember that the **K2 Learning** category might have a different name in your environment.
 - a. Delete the application's categories. You must delete the subcategories first (**Forms, Views, SmartObjects, Workflow**) and then **Leave Request**.



- b. Finally, you can delete the **K2 Learning** category if you do not intend building more tutorials. Remember that the **K2 Learning** category might have a different name in your environment.

Summary

If everything worked as you expected: congratulations!

Having completed this tutorial, you learned about three main K2 application components: **data**, **forms** and **workflow**. At this point, you should have a basic understanding of how to build SmartObjects, views, forms and workflows, and how to combine them together in the context of a workflow-centric application.