



Build an App in 15 Minutes or Less

Serena Business Manager (SBM) version 10.x

Introduction and Objective

This document will provide a quick tutorial on how to design and deploy a simple business processes with SBM for the “Build an App in 15 Minutes or Less” workshop.

It will use a simple, intuitive process common to everyone-- Vacation Request.

We will create an application consisting of the typical elements: workflow, fields, and forms. Then deploy it to our environment for use.

Assumption

You have installed all components of SBM. To actually deploy the application you will also need the appropriate rights. If you do not have the ability to deploy the application you can still create it.

Procedures

For complete step by step instructions for designing business application or processes with SBM Composer, please refer to the SBM Composer Guide.

The steps below will provide guidance and some details on how to build the Vacation Request application.

Step 1: Create New Process Application

Open Composer and create a new process application. Give your Application a meaningful name, such as “Vacation.”

The actual names for your Application, states, and transitions are not critical nor do they have to match the suggested examples.

Configure Process App ? X

Press F1 for help with these terms

Process app (Application Process App)

Process app name: Vacation

Category: [Dropdown]

Category provides a means of sorting in your local cache or remote repositories. Select one or type in your own.

Application (Application)

Application name: Vacation

Internal name: VACATION

Tab name: Vacation

The Tab name is what users see when running the process app.

Table (in Application)

Table name: Vacation

Singular item name: Vacation

Database table name: VACATION

Prefix: USR

Workflow (in Application)

Workflow name: Vacation

OK Cancel

Step 2: Workflow Design

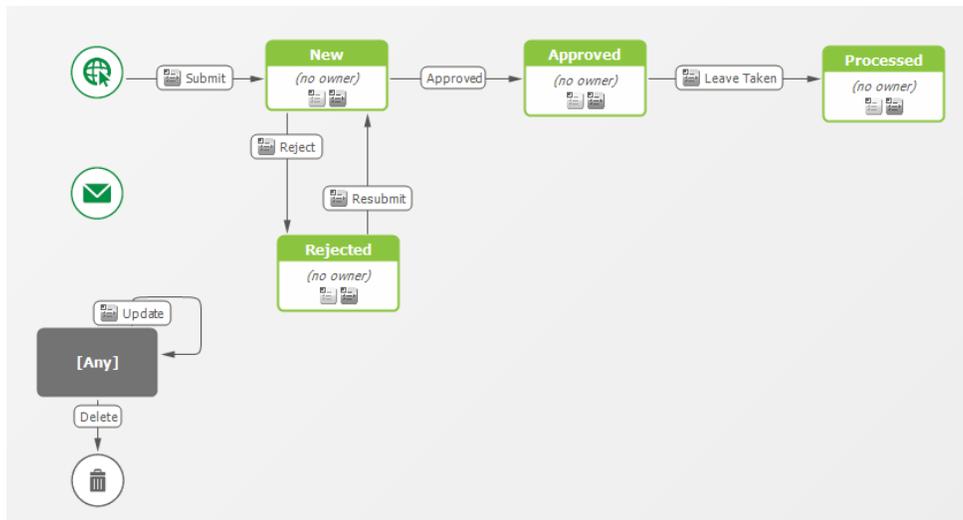
Under the Workflow palette develop the Vacation Request process with the following states:

- Approved
- Processed
- Rejected

Using the Workflow palette, create the following regular transition:

- Approve (from New to Approved)
- Leave Taken (from Approved to Processed)
- Reject (from New to Rejected)
- Resubmit (from Rejected to New)

When complete, your workflow should look something like Figure below.



At this point we have completed the drawing of the process workflow. Next, we need to understand the information or data we need to gather in each step of our Vacation Request business process.

Step 3: Data Design

Go to the “Data Design” tab in the launch pane. The next step is to define the information you want to capture about your Vacation Requests. You will do this by selecting additional fields from the table palette and dragging them (similar to what we

have done for states and transitions) onto the middle window that contains the fields currently in the *Vacation* table.

For your Vacation Request process, the following information is needed:

Database Field Name	Type	Section	Comments
START_DATE	Date/Time	Standard	Option: Date Only (mm/dd/yyyy) Required: Yes
END_DATE	Date/Time	Standard	Option: Date Only (mm/dd/yyyy) Required: Yes
PROPOSED_HOURS	Numeric	Standard	Option: Integer Required: Yes
ACTUAL_HOURS	Numeric	Standard	Option: Integer Required: No
REJECT_REASON	Text	Standard	Option: Memo
MANAGER	User	Manager	Options: Associate to Administrator Role
TITLE	Text	Standard	Change to Full Name

You now have created the fields that will contain the information you need for the Vacation process. But before we continue, we need to make one change to a system-defined field, Title. This field really isn't needed for our example. But instead of deleting it, we will just use it to capture the requester's name. Select the field Title in the property editor, go to the General tab, and change the name from Title to Full Name.

When complete, your primary table should look something like Figure below.

Field name	Type	Database field name	Section	Dependent fields
Active/Inactive	Binary/Tertiary	ACTIVEINACTIVE	Manager	N/A
Actual Hours	Numeric	ACTUAL_HOURS	Standard	N/A
End Date	Date/Time	END_DATE	Standard	N/A
Item Id	Text	ISSUEID	Standard	N/A
Item Type	Single Selection	ISSUETYPE	Standard	N/A
Last Modified Date	Date/Time	LASTMODIFIEDDATE	System	N/A
Manager	User	MANAGER	Standard	N/A
Name	Text	TITLE	Standard	N/A
Owner	User	OWNER	Manager	N/A
Project	Project	PROJECTID	Manager	N/A
Proposed Hours	Numeric	PROPOSED_HOURS	Standard	N/A
Reject Reason	Text	REJECT_REASON	Standard	N/A
Start Date	Date/Time	START_DATE	Standard	N/A
State	State	STATE	Manager	N/A
Submit Date	Date/Time	SUBMITDATE	Advanced	N/A
Submitter	User	SUBMITTER	Advanced	N/A

Now you are finished with designing your data fields.

Next, we need to design the forms to present to the users of our vacation request process.

Step 4: Visual Design

Composer provides support for the creation and use of forms when items (Vacation Requests in this example) are within a particular state or in a transition between states.

Best practice is to go through your workflow and define the forms for each state and transition. You can reuse forms where it's logical and utilize actions on forms to facilitate this. Ideally, create as many as you need to guide the end-user but not too many which becomes an administrative nightmare.

For this workshop we will limit the forms to three (3) total. One for submitting the request called *Transition Form*, one for viewing the request called *State Form*, and one for rejecting the request called *Reject Form*.

Transition Form

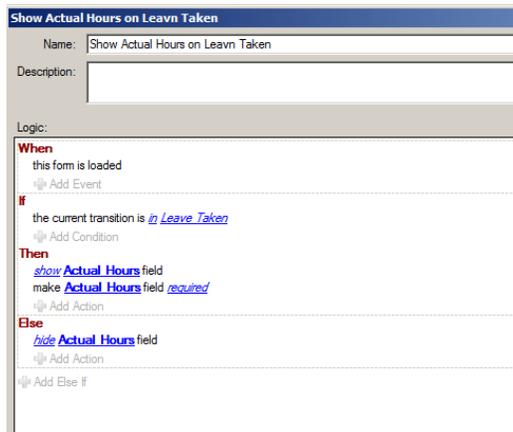
Go to the “Visual Design” tab in the launch pane. Right click on the Forms element and select Add New | Transition Form.

You have the option to create an empty form or create a new one (icons at the right side of the Property Editor). Select **Based on ‘quick’ form** from the **New** list. You will see that all fields and all sections appear in the form.

Composer automatically inserts all the data fields onto the form. At this point, you may delete the fields that the user is not required to fill-out during the submission of their vacation request. You may also re-arrange the location of the fields on your form to present it to the user in a more logical and meaningful way.

Since we can reuse the form for most of our transitions, we will put some logic on this form to only show the actual time taken when it's appropriate—on the Leave Taken transition.

Under the property editor of the form, create an **Action** like the figure below.



Arrange your Submit transition form to look something like Figure below.

The screenshot shows a software form with a title bar containing the text "{Project} - {Item ID}: {Item Title}". Below the title bar, there is a message: "The buttons to complete or cancel this transition will be displayed here." followed by an "Actions" dropdown menu and three icons (envelope, window, question mark). The form is divided into two main sections: "Standard" and "Manager".

Standard Section:

Name:	Name		
Start Date:	Start Date	End Date:	End Date
Manager:	Manager	Proposed Hours:	Propos...
Actual Hours:	Actual Hours		

Manager Section:

Active/Inactive:	Active/In...	Submit Date:	Submit Date
Owner:	Owner	State:	State

State Form

Now let's create the view form for the manager.

Go to the "Visual Design" tab in the launch pane. Right click on the Forms element and select **Add New | State Form**.

You have the option to create an empty form or create a new one just like before only this time you also have to option to create one based on another form.

Since the State Form format is very similar to the Transition Form, we will simply duplicate it. So select **Based on 'another form'**.

You can removed the fields not needed or leave them on the form. Do so whichever way you like and format it to something similar as below.

The image shows a screenshot of a web form layout for a project. The form is titled "{Project} - {Item ID}: {Item Title}" and has a "Details" button and a "Social" link. Below the title, there is a message: "The buttons to transition from this state will be displayed here." and an "Actions" dropdown menu. The form is divided into four main sections, each with a dashed border and a blue "A" label on the left and right sides:

- Standard:** Contains fields for "Name", "Item Id", "Start Date", "End Date", "Manager", and "Proposed Hours".
- Manager:** Contains fields for "Active/Inactive", "Submit Date", "Owner", and "State".
- State Change History:** Contains the text "State change history details will be shown here."
- Change History:** Contains the text "Change history details will be shown here."

Reject Form

Now let's create the reject form. For this one, you can create new or you can also create based on the Transition Form then simply remove some fields and add the Reject field.

Since we want to make the reason for rejecting the vacation request required, we will use an Action on this form.

Under the property editor of the form, create an **Action** like the figure below.

The image shows a screenshot of the "Reject Reason Required" form property editor. The form is titled "Reject Reason Required" and has a "Name" field with the value "Reject Reason Required" and an "Enabled" checkbox checked. The "Description" field is empty. The "Logic" section contains the following rules:

- When:** this form is loaded
- Then:** make **Reject Reason** field required

At the bottom of the editor, there are "OK" and "Cancel" buttons.

You can removed the fields not needed or leave them on the form. Do so whichever way you like and format it to something similar as below.

The screenshot shows a form editor window titled "{Project} - {Item ID}: {Item Title}". Below the title bar, there is a message: "The buttons to complete or cancel this transition will be displayed here." and an "Actions" dropdown menu. The form is divided into two sections:

- Standard:** Contains a "Reject Reason:" label and a text area with a "Reject Reason" placeholder.
- Manager:** Contains four fields: "Active/Inactive:" with a dropdown menu, "Submit Date:" with a date picker, "Owner:" with a dropdown menu, and "State:" with a dropdown menu.

Red hatched lines separate the two sections. Blue vertical bars with the letter 'A' are on the left and right sides of the form.

Although you may want to create custom forms for each of the States and transitions, given the time for this Workshop we have to end here.

At this point we have completed the designing of the process workflow, the data we need to capture, and we designed the forms to present to the users of our vacation request process.

But we're not done yet. We need to assign owners on the states and associate our forms.

Step 5: Final Touches on Workflow

Go back to the "Workflow Design" tab in the launch pane. You will again see the nice workflow you created. Now, you want to make sure that you have the right level of security. So let's start by setting some ownership to the different states you have created.

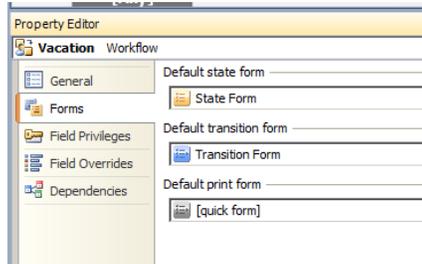
Select the "New" state and in the property editor, select Manager (this is the Manager field you created in step above) in the drop down list associated with the Owner field.

Select the "Approved" and "Rejected" states and set Submitter as the owner for these states.

Notice that Composer uses a specific color coding for each state-owned by the same person. This allows for easy grouping per owner.

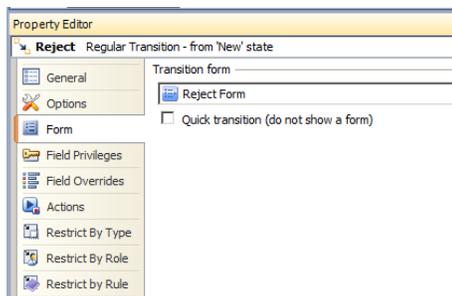
Now we need to set when the forms will be utilized in our workflow.

Click anywhere in the workflow pane. Under the property editor, click on **Forms**.
Set the Default State Forms and Default Transition Forms to State Form and Transition Form respectively.

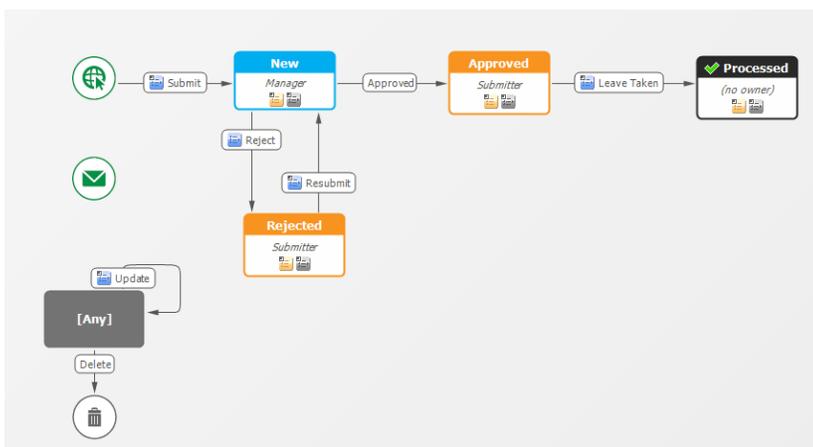


Now we need to set the Reject Form on the Reject transition.
Click on the Reject transition in the workflow.

Under the property editor, click on Forms.
Set the Transition form to Reject Form from the drop-down list.



When complete your Workflow should look something like Figure below.



That's it. You are now ready to deploy and test out your new Vacation Request application.