



# HCL Digital Solutions Factory Tour

Your session will begin shortly...

Powered by

 HCL Domino Volt

 HCL SOFTWARE

# Domino Volt Tutorial



**Martin Lechleider**

Product Management – Domino Volt  
[martin.lechleider@hcl.com](mailto:martin.lechleider@hcl.com)

Project Steps	Pages	Resource	
1. Creating catalog app from spreadsheet	5	<a href="#">MobilePhoneCatalog.xls</a>	cumulative build of the app 
2. Add order form and table for selection	6-8	<a href="#">MobilePhoneOrders1.volt</a>	
3. Getting details for the selected item and presenting it nicely	10-13	<a href="#">MobilePhoneOrders2.volt</a>	
4. Getting information on the requester and workflow approver(s)	15-19	<a href="#">MobilePhoneOrders3.volt</a>	
5. Creating the workflow stages and rules	21		
6. Dynamic role assignments and email notifications	22-26	<a href="#">MobilePhoneOrders4.volt</a>	
7. Rules for a dynamic user experience	28-33	<a href="#">MobilePhoneOrders5.volt</a>	
8. Add a workflow approval log	35-37	<a href="#">MobilePhoneOrders6.volt</a>	

# Tutorial assumptions

---

- Domino Volt 1.0.1 or later
- Domino directory has display names, internet email and manager name populated in person documents
- Users should have some experience with Volt. If you are new to Volt, the [Introductory step by step tutorial](#) is better place to start

# 1. Creating catalog app from spreadsheet

[MobilePhoneCatalog.xls](#)

### Create a new app

**Choose your Excel spreadsheet**

Choose fields, names, and types

Enter new application details

**Choose your Excel spreadsheet**

Spreadsheet (.xls or .xlsx):

MobilePhoneCatalog.xls

**How to structure your spreadsheet**

- Data needs to be in table format.**
- Use the first row as a header.**
- Each column will become a field in the application.**
- Use multiple sheets for more than one table.**

## 2. Add order form and table for selection

### ► Create the order form and selection table

2) Add table with:

- Brand
- Model
- Platform
- Storage
- UID

3) Add UID field on main form page

The screenshot shows a software development tool interface. The main workspace displays a form titled "Mobile Phone Order System". Inside the form, there is a section labeled "Select a Phone" which contains a table. Below the table is a text input field labeled "UID". To the right of the main workspace is an "Outline" panel. The "Outline" panel shows a tree structure with "Forms" expanded to show "Order", "Page 1", and "Phone Catalog".

Brand	Model	Storage	Platform
value	value	value	value
value	value	value	value

1) Create new form called Order

Phone Catalog created from spreadsheet

Do not display UID in table

## 2. Add order form and table for selection

### ► Hide table buttons

Hide the table buttons

Select a Phone ⚙️ 🔍 ✕

[Add hint](#)

Table

Brand	Model	Storage	Platform
value	value	value	value
value	value	value	value

### Edit Table Properties

Basic **Advanced** Events

**ID:** ⓘ  
F\_Table1

**Table Style:**

Buttons below the table ▾

- Buttons below the table
- Buttons on the right
- No Buttons**

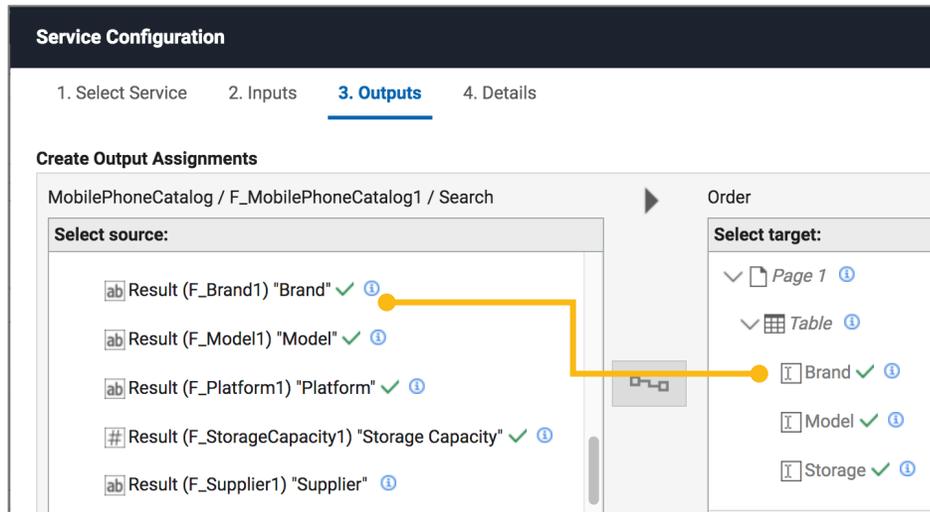
## 2. Add order form and table for selection

### ► Populate the table and enable a selection

1) Add Search Catalog Service on Table onShow event

2) Add JavaScript on Table onClick that passes the UID from the selected row to the UID field on main form page

(adjust JS based on you IDs)



Map items from the catalog to the table – including the unique record identifier

```
//get the selection
var selected = item.getSelection();

//do nothing if a row wasn't selected
if(selected === null)
    return;

// Set the UID on the main for to what is selected
BO.F_UID.setValue(selected.F_UIDTbl.getValue());
```

## 2. Add order form and table for selection

### ▶ How things should look - selection table with service

[MobilePhoneOrders1.volt](#)

### Mobile Phone Order System

**Select a Phone**

Brand	Model	Storage	Platform
Apple	iPhone 11	64	IOS
Apple	iPhone 11	128	IOS
Apple	iPhone 11 pro	64	IOS
Apple	iPhone 11 pro	256	IOS
Samsung	Galaxy Fold 5G	512	Android

**UID**

### 3. Getting details for the selected item and presenting it nicely

#### ► Create a section for selected phone details

The screenshot shows a form editor interface with a form containing several input fields. The fields are arranged in a grid-like structure. The fields are: UID, Brand, Platform, Supplier, Price, Model, and Storage. The Price field is highlighted with a blue border and a double-headed arrow, indicating it should be a number or currency field. The Storage field is highlighted with a blue border and a double-headed arrow, indicating it should be an image field. The text "Click 'Edit Properties' to add an image file" is positioned next to the Storage field. The form is enclosed in a blue border with a title bar at the top containing a gear icon, a refresh icon, and a close icon. The form is divided into sections by dashed lines. The annotations are as follows:

- 1) Create new section: Points to the right side of the form editor.
- 2) Move UID into section: Points to the UID field.
- 3) Add fields as shown: Points to the Brand, Platform, and Supplier fields.
- 4) Make sure price is a number or currency field: Points to the Price field.
- 5) Add image item. Set height to 200: Points to the Storage field.

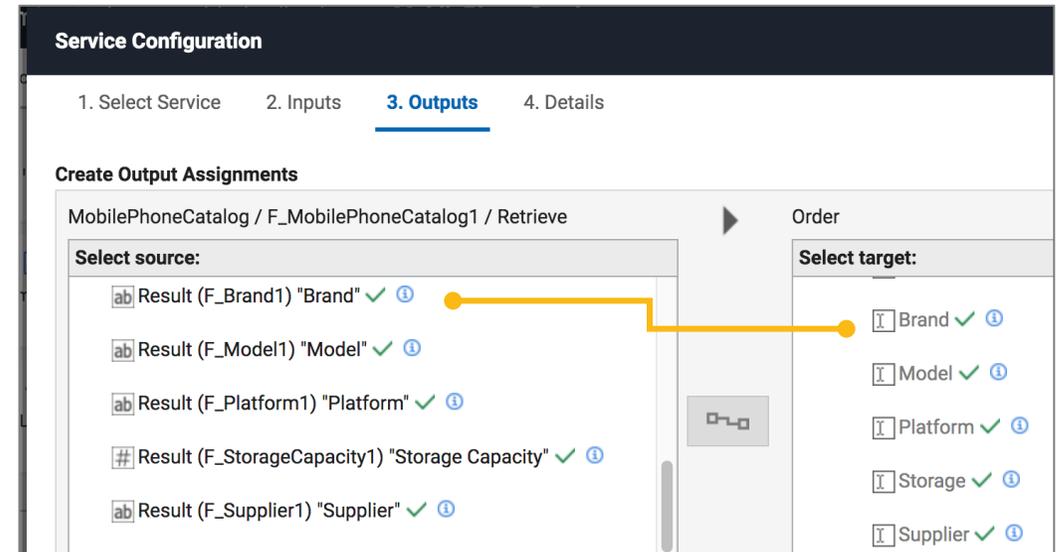
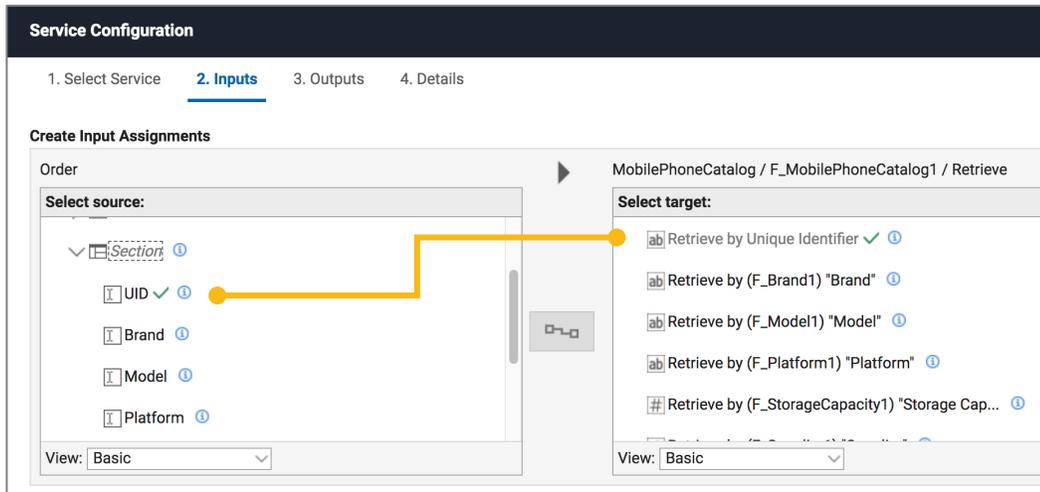
### 3. Getting details for the selected item and presenting it nicely

► Get details on selected phone

1) Add Retrieve Catalog Service on the UID onChange event

2) On inputs - map the UID to Retrieve by Unique Identifier

3) On outputs – map items from catalog to fields in section



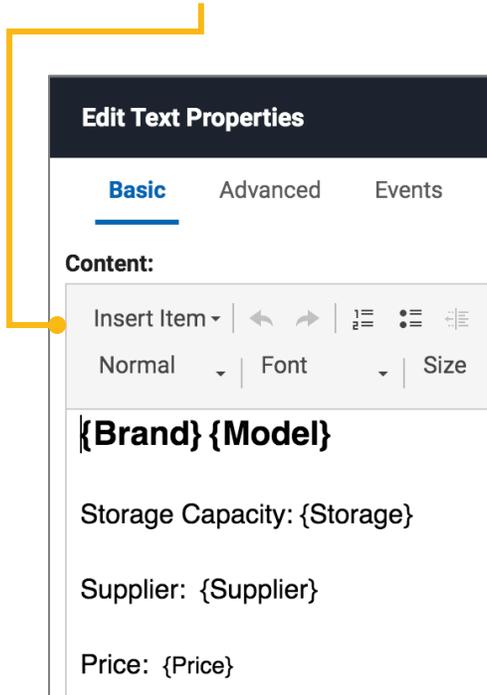
Map picture url to image item (not shown in screenshot)



### 3. Getting details for the selected item and presenting it nicely

#### ► Improve presentation of phone details with Text item

Use Insert Item drop down to add “echo text”

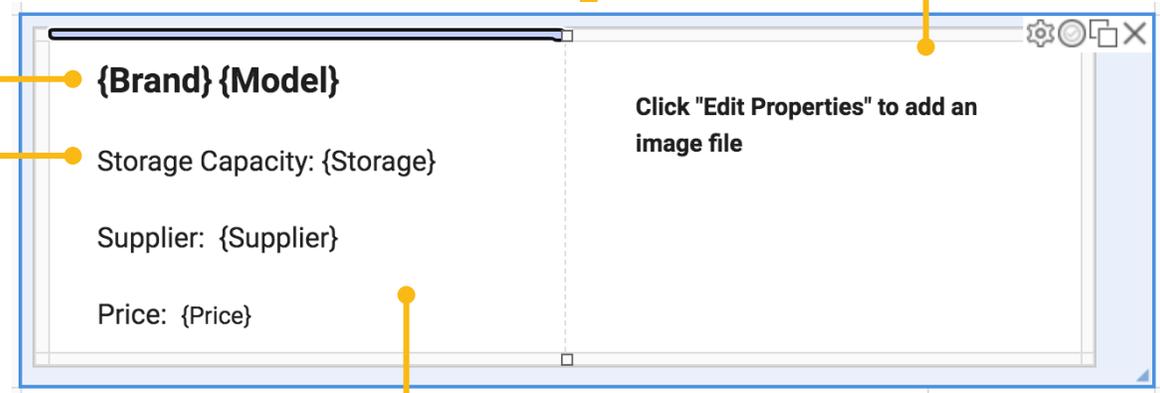


1) Add new section

2) Move the image item into the section

20px and bold

16px



3) Add text item with “echo text”

### 3. Getting details for the selected item and presenting it nicely

- ▶ Hide the phone detail data section and show presentation section

On the phone details section onShow event add the following line of JavaScript

```
item.setVisible(false);
```

<input type="checkbox"/> UID	<input type="text"/>
<input checked="" type="checkbox"/> Brand	<input type="text"/>
<input checked="" type="checkbox"/> Model	<input type="text"/>
<input type="checkbox"/> Platform	<input type="text"/>
<input type="checkbox"/> Storage	<input type="text"/>
<input checked="" type="checkbox"/> Supplier	<input type="text"/>
<input type="checkbox"/> Price	<input type="text"/>

Set a rule on the section so it shows when UID has a value

Rules	
Show rules related to: ⓘ	Rule Name: Rule 4 ✓
Section - F_Section2	
▪ Rule 4 ✕	Perform this action: ⓘ
	Section - F_Section2 Show
	When the following condition is true: ⓘ
UID	Has value

### 3. Getting details for the selected item and presenting it nicely

- ▶ How things should look - detail presentation of selected phone

[MobilePhoneOrders2.volt](#)

#### Mobile Phone Order System

Select a Phone

Brand	Model	Storage	Platform
Apple	iPhone 11	64	IOS
Apple	iPhone 11	128	IOS
Apple	iPhone 11 pro	64	IOS
Apple	iPhone 11 pro	256	IOS
Samsung	Galaxy Fold 5G	512	Android

#### Apple iPhone 11

Storage Capacity: 128

Supplier: Apple Store

Price: \$749.00



# 4. Getting information on the requester and workflow approver(s)

- ▶ Create sections for requester and workflow approver information

Create a new section and add all these fields

Fields will be populated by Domino Directory service calls.

The screenshot shows a form editor interface with a grid of fields. On the left side, there are three fields: 'Requester', 'Level 1 Approver', and 'Level 2 Approver'. On the right side, there are three fields: 'Requester Email', 'Level 1 Email', and 'Level 2 Email'. A yellow callout box labeled 'Display Names' is positioned over the 'Level 1 Approver' field. A yellow line connects this callout to the 'Level 1 Name' field in the second screenshot.

Use email items, not single line since we will use these for notifications

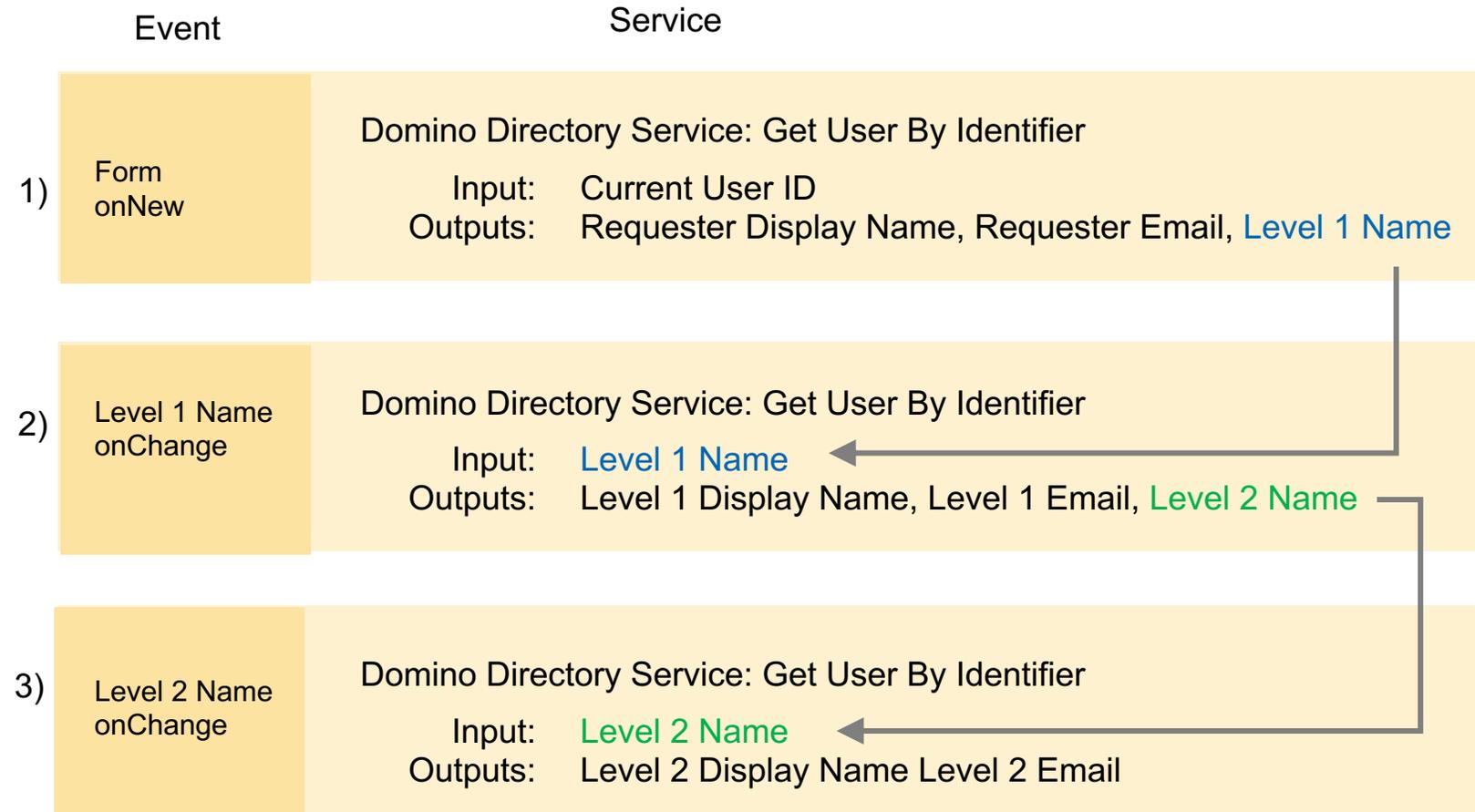
The screenshot shows a form editor interface with two fields: 'Level 1 Name' and 'Level 2 Name'. A yellow callout box labeled 'Names' is positioned over the 'Level 1 Name' field. A yellow line connects this callout to the 'Level 1 Name' field in the first screenshot.

Add to hidden section

These fields will be used to avoid displaying canonical names

## 4. Getting information on the requester and workflow approver(s)

### ▶ Sequence of service calls to get requester and approver information



**Domino Directory structure**

Name	Display Name	Email	Manager Name
-----	-----	-----	-----
-----	-----	-----	-----
-----	-----	-----	-----

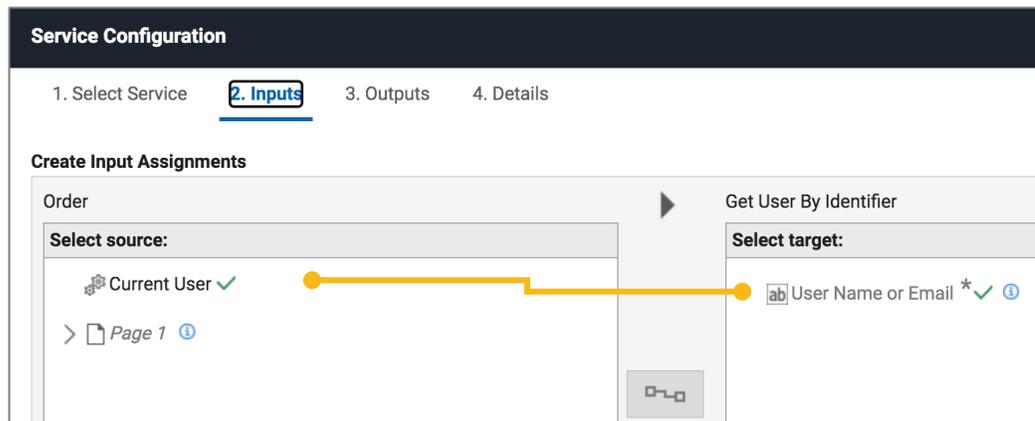
## 4. Getting information on the requester and workflow approver(s)

### ► Get workflow requester display name, email and manager name

#### On the Form onNew event

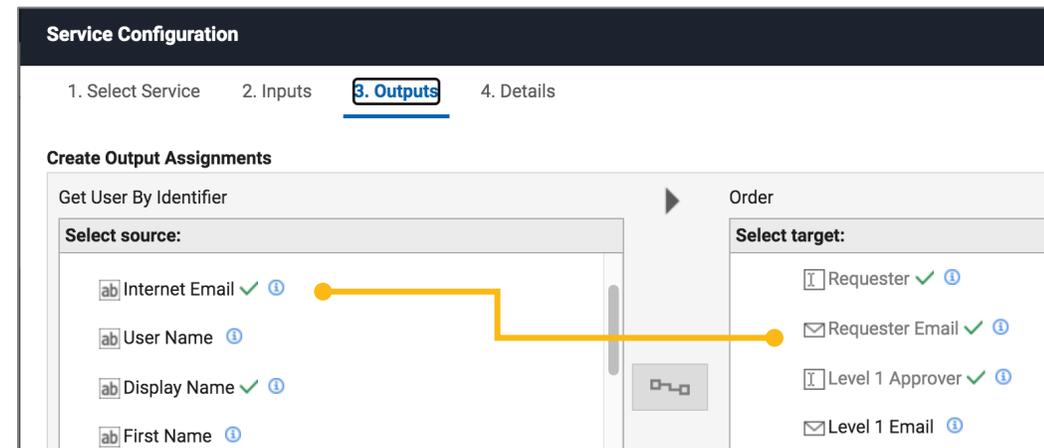
call the Domino Directory service Get User By Identifier

#### 1) Map Current User to input



The screenshot shows the 'Service Configuration' interface with the 'Inputs' tab selected. Under 'Create Input Assignments', the 'Get User By Identifier' service is configured. The 'Select source' dropdown is set to 'Current User' and the 'Select target' dropdown is set to 'User Name or Email \*'. A yellow line connects the 'Current User' source to the 'User Name or Email \*' target.

#### 2) On outputs – map Display Name and Internet Email to Requester and Requester Email, Manager to Level 1 Name



The screenshot shows the 'Service Configuration' interface with the 'Outputs' tab selected. Under 'Create Output Assignments', the 'Get User By Identifier' service is configured. The 'Select source' dropdown is set to 'Internet Email' and the 'Select target' dropdown is set to 'Requester'. A yellow line connects the 'Internet Email' source to the 'Requester' target. Other targets visible include 'Requester Email', 'Level 1 Approver', and 'Level 1 Email'.

#### 3) Map Manager (not seen in screenshot) to Level 1 Name

4) Conditionally execute the service so it only happens for the Start stage

```
if (BO.getCurrentStage() === 'ST_Start')
{
form.getServiceConfiguration("SC_FirstDirectoryCall").callService();
}
```

## 4. Getting information on the requester and workflow approver(s)

- ▶ Add call to get level 1 email, level 1 display name and level 2 approver name

### On the Level 1 Name onChange event

call the Domino Directory service Get User By Identifier

1) Map Level 1 Name to input

The screenshot shows the 'Service Configuration' interface with the 'Inputs' tab selected. Under 'Create Input Assignments', the 'Order' component is connected to the 'Get User By Identifier' service. The 'Select source' dropdown for the service is set to 'Level 1 Name'. The 'Select target' dropdown is set to 'User Name or Email \*'. A yellow line indicates the mapping from the source to the target.

2) On outputs map Internet Email to Level 1 Email and Manager to Level 2 Name

The screenshot shows the 'Service Configuration' interface with the 'Outputs' tab selected. Under 'Create Output Assignments', the 'Get User By Identifier' service is connected to the 'Order' component. The 'Select source' dropdown for the service is set to 'Internet Email'. The 'Select target' dropdown is set to 'Level 1 Approver'. A yellow line indicates the mapping from the source to the target. Below this, another mapping is shown where the source is 'Manager' and the target is 'Level 2 Name'.

## 4. Getting information on the requester and workflow approver(s)

### ► Add call to get level 2 email and display name

#### On the Level 2 Name onChange event

call the Domino Directory service Get User By Identifier

1) map Level 2 Name to to input

**Service Configuration**

1. Select Service   **2. Inputs**   3. Outputs   4. Details

**Create Input Assignments**

Order

**Select source:**

- Level 1 Approver ⓘ
- Level 1 Email ⓘ
- Level 2 Approver ✓ ⓘ

**Select target:**

- User Name or Email \* ✓ ⓘ

The screenshot shows the 'Inputs' tab of the 'Service Configuration' interface. It displays the 'Create Input Assignments' section for the 'Get User By Identifier' service. The 'Select source' list includes 'Level 1 Approver', 'Level 1 Email', and 'Level 2 Approver'. The 'Select target' list includes 'User Name or Email'. A yellow line connects 'Level 2 Approver' to 'User Name or Email'.

2) On outputs map Internet Email to Level 2 Email and Display Name to Level 2 Approver

**Service Configuration**

1. Select Service   2. Inputs   **3. Outputs**   4. Details

**Create Output Assignments**

Get User By Identifier

**Select source:**

- Internet Email ✓ ⓘ
- User Name ⓘ
- Display Name ✓ ⓘ
- First Name ⓘ

**Select target:**

- Level 1 Approver ⓘ
- Level 1 Email ⓘ
- Level 2 Approver ✓ ⓘ
- Level 2 Email ✓ ⓘ

The screenshot shows the 'Outputs' tab of the 'Service Configuration' interface. It displays the 'Create Output Assignments' section for the 'Get User By Identifier' service. The 'Select source' list includes 'Internet Email', 'User Name', 'Display Name', and 'First Name'. The 'Select target' list includes 'Level 1 Approver', 'Level 1 Email', 'Level 2 Approver', and 'Level 2 Email'. Yellow lines connect 'Internet Email' to 'Level 1 Email' and 'Display Name' to 'Level 2 Approver'.

# 4. Getting information on the requester and workflow approver(s)

## ▶ How things should look - workflow approver section

[MobilePhoneOrders3.volt](#)

Apple	iPhone 11 pro	64	IOS
Apple	iPhone 11 pro	256	IOS
Samsung	Galaxy Fold 5G	512	Android

< back

[This phone is \\$900 or greater and requires 2 levels of approval](#)

**Apple iPhone 11 pro**



Storage Capacity: 64

Supplier: Apple Store

Price: \$999.00

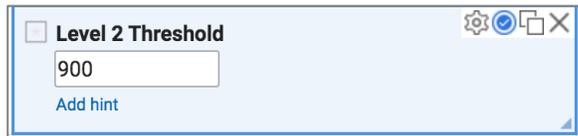
<b>Requester</b>	<b>Requester Email</b>
<input type="text" value="Admin"/>	<input type="text" value="martin.lechleider@hcl.com"/>
<b>Level 1 Approver</b>	<b>Level 1 Email</b>
<input type="text" value="Andrew Manby"/>	<input type="text" value="andrew.manby@pnp-hcl.com"/>
<b>Level 2 Approver</b>	<b>Level 2 Email</b>
<input type="text" value="Rajesh Iyer"/>	<input type="text" value="rajesh_i@hcl.com"/>

# 5. Creating the workflow stages and rules

## ▶ Define stages and conditional approval path

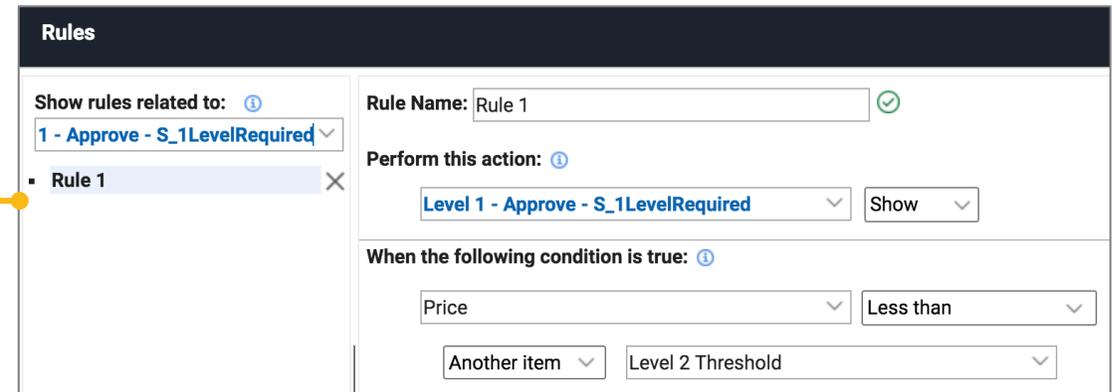
*2 Levels are required if the phone is \$900 or greater than*

1) Add number field to the hidden section and set default value to 900

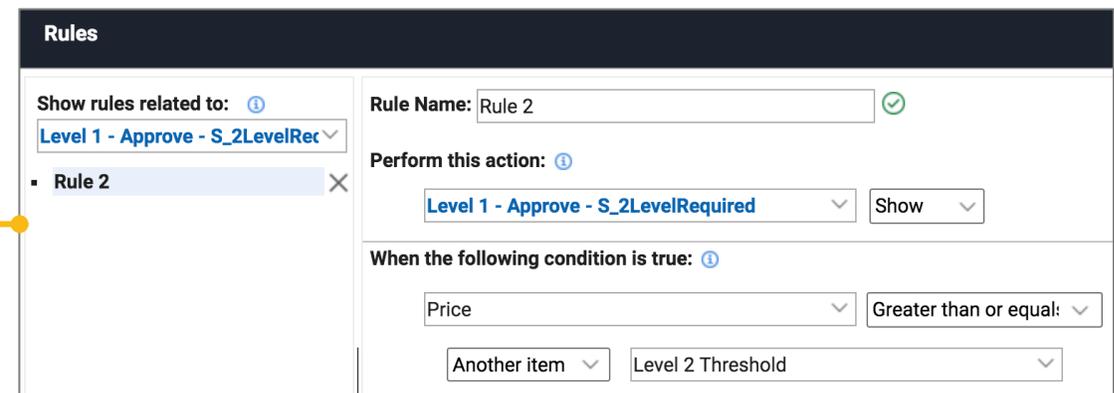


A screenshot of a form field titled "Level 2 Threshold". It includes a checkbox, a text input field containing the number "900", and an "Add hint" link.

3) Set rule on approve actions base on less than or equal than or greater than threshold filed

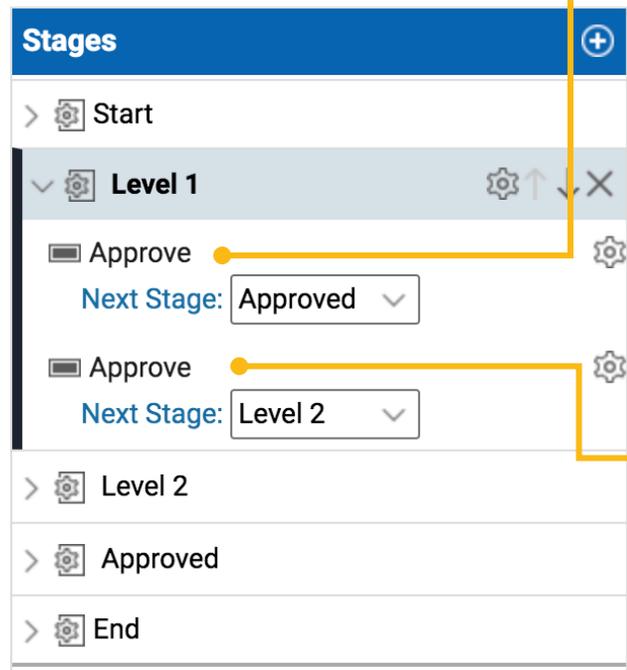


A screenshot of the "Rules" configuration interface for "Rule 1". It shows the rule name, the action "Level 1 - Approve - S\_1LevelRequired", and the condition "Price Less than Another item Level 2 Threshold".



A screenshot of the "Rules" configuration interface for "Rule 2". It shows the rule name, the action "Level 1 - Approve - S\_2LevelRec", and the condition "Price Greater than or equal: Another item Level 2 Threshold".

1) Add Stage for Level1, Level2, and Approved



A screenshot of the "Stages" configuration interface. It shows a sequence of stages: Start, Level 1, Level 2, Approved, and End. The "Level 1" stage is expanded to show two "Approve" actions. The first "Approve" action has "Next Stage" set to "Approved", and the second "Approve" action has "Next Stage" set to "Level 2".

# 6. Dynamic role assignments and email notifications

## ► Define open role for approvers

### Define Roles

Define a list of roles for this application

Add and Manage Roles

1.  ✕  
 Open  Closed
2.  ✕  
 Open  Closed
3.  ✕  
 Open  Closed
4.  ✕ +  
 Open  Closed

1) Define the role as open

### Order - Level 1

Roles	Create	Read	Update	Delete
Administrator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Initiator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Record Owner	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Approver	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

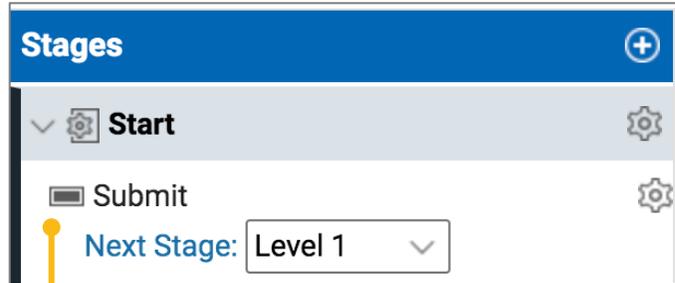
2) Specify update privileges for Level 1 and Level 2 stages

### Order - Level 2

Roles	Create	Read	Update	Delete
Administrator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Initiator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Record Owner	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Approver	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

# 6. Dynamic role assignments and email notifications

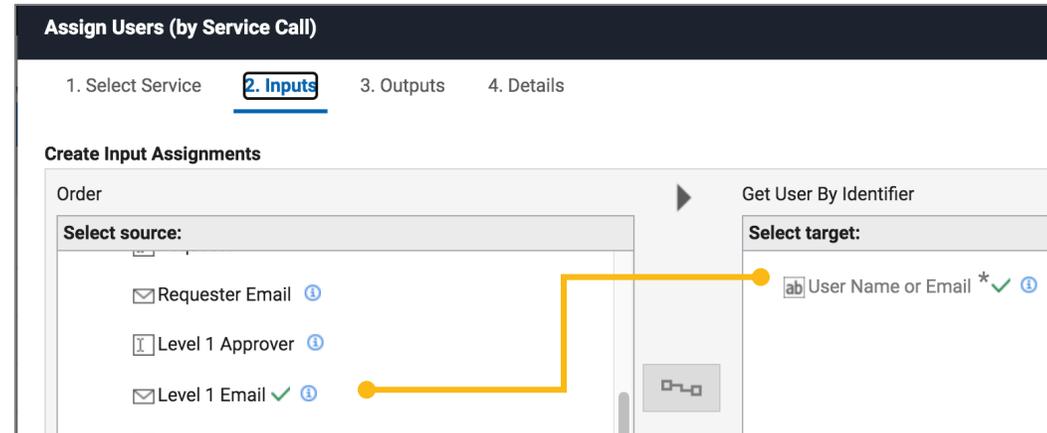
## ▶ Level 1 - Dynamic Role Assignment



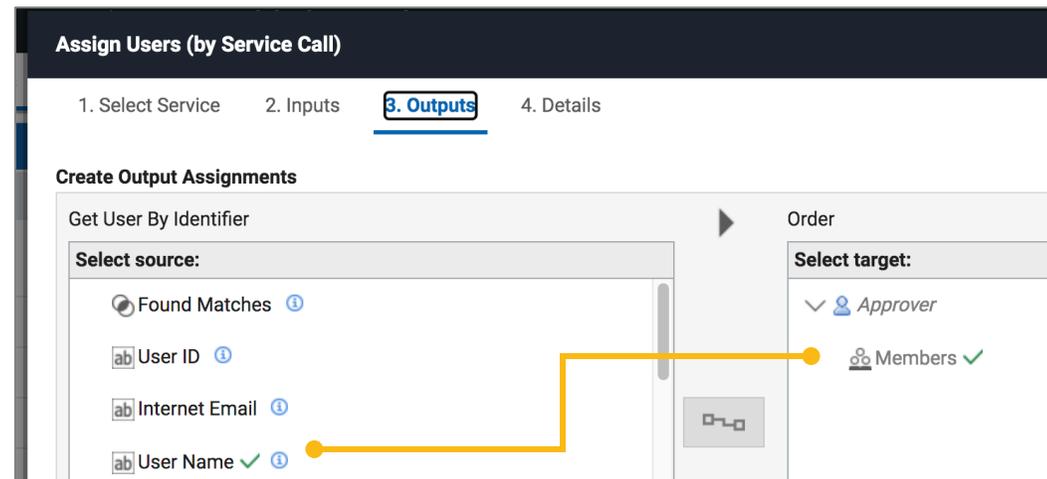
### On Submit Action

Assign User by Service  
Domino Directory / Get  
User By Identifier

on inputs – map Level 1 Email to User Name or Email



on outputs – map User Name to Approver role



## 6. Dynamic role assignments and email notifications

### ▶ Level 1 - Notification

**Stages**

- Start
- Submit  
Next Stage: Level 1

**On Submit Action**  
Send Email

**Send To**  
{Level 1 Email} field value

**Edit Properties**

To:

A value typed here:

Insert Item ▾

{Level 1 Email}

Use the Insert Item drop  
down to select the field

# 6. Dynamic role assignments and email notifications

## ▶ Level 2 - Dynamic Role Assignment (do the same thing)

**Stages**

- Start
- Level 1
  - Approve (Next Stage: Approved)
  - Approve (Next Stage: Level 2)
- Level 2
- Approved

**On Submit Action**  
Assign User by Service  
Domino Directory / Get  
User By Identifier

on inputs – map Level 1 Email to User Name or Email

**Assign Users (by Service Call)**

1. Select Service | **2. Inputs** | 3. Outputs | 4. Details

Create Input Assignments

Order: Get User By Identifier

Select source:

- Requester Email
- Level 1 Approver
- Level 1 Email
- Level 2 Approver
- Level 2 Email

Select target:

- User Name or Email \*

on outputs – map User Name to Approver role

**Assign Users (by Service Call)**

1. Select Service | 2. Inputs | **3. Outputs** | 4. Details

Create Output Assignments

Order: Get User By Identifier

Select source:

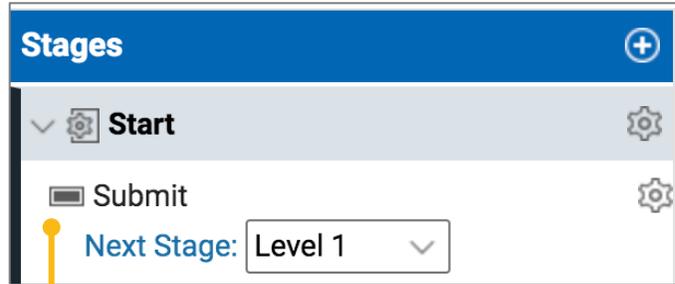
- Found Matches
- User ID
- Internet Email
- User Name

Select target:

- Approver
- Members

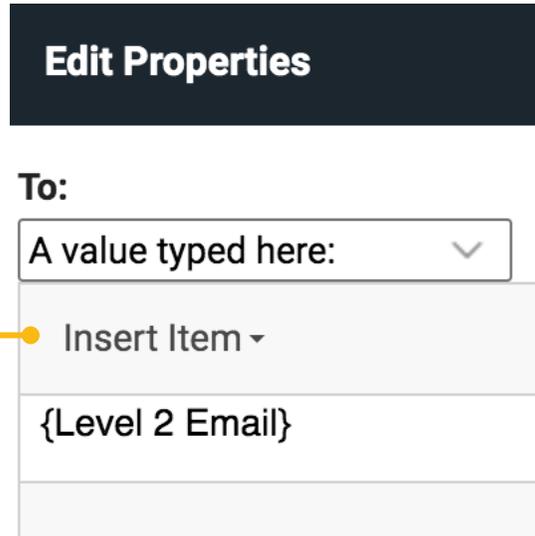
## 6. Dynamic role assignments and email notifications

### ► Level 2 – Notification (do the same thing)



**On Submit Action**  
Send Email

**Send To**  
{Level 2 Email} field value



Use the Insert Item drop  
down to select the field

# 6. Dynamic role assignments and email notifications

## ▶ How things should look – workflow stages

[MobilePhoneOrders4.volt](#)

The screenshot displays the HCL Domino Volt interface for editing a workflow. The main window is titled 'Order - Level 1 » Page 1' and shows a form titled 'Mobile Phone Order System'. The interface includes a top navigation bar with 'Forms', 'Access', 'Stages', 'Style', 'Events', 'Settings', and 'Validation'. The 'Stages' tab is active, showing a workflow diagram with stages: Start, Level 1 (containing two 'Approve' stages), Level 2, Approved, and End. The 'Approve' stages have dropdown menus for 'Next Stage' (Approved and Level 2). The main form area contains a table with columns 'Brand', 'Model', 'Storage', and 'Platform', and a section for product details with fields for '{Brand} {Model}', 'Storage Capacity: {Storage}', 'Supplier: {Supplier}', and 'Price: {Price}'. There are also 'Approve' buttons and an 'Add' button at the bottom of the form.

## 7. Rules for a dynamic user experience

### ► Conditionally show the '2 levels required' indicator

Add text item to top of section to inform when 2 levels are required

This phone is \${Level 2 Threshold} or greater and requires 2 levels of approval

**{Brand} {Model}**

Storage Capacity: {Storage}

Supplier: {Supplier}

Price: {Price}

Click "Edit Properties" to add an image file

Set a rule on the **text item** so it shows when price is equal to or greater than the level 2 threshold

**Rules**

Show rules related to: **F\_Text2**

- Rule 3

Rule Name: Rule 3

Perform this action: **F\_Text2** Show

When the following condition is true: Price Greater than or equal: Another item Level 2 Threshold

# 7. Rules for a dynamic user experience

## ► Add additional field and navigation

5) Add text item

6) Add order button

The screenshot shows a mobile application form with the following elements and annotations:

- Header:** A navigation bar with a back arrow and the text "< back". An orange line connects this to the annotation "5) Add text item".
- Text:** A blue text line: "This phone is \${Level 2 Threshold} or greater and requires 2 levels of approval".
- Fields:** Fields for "{Brand} {Model}", "Storage Capacity: {Storage}", "Supplier: {Supplier}", and "Price: {Price}" are present.
- Text:** A black text instruction: "Click 'Edit Properties' to add an image file".
- Form Elements:** A "Business Justification" section with a checkbox and a text input field. An orange line connects this to the annotation "1) Add multiline field business justification".
- Buttons:** A blue "Order" button. An orange line connects this to the annotation "6) Add order button".
- Form Elements:** "Level 1 Manager Comments" and "Level 2 Manager Comments" sections, each with a checkbox and a text input field. Orange lines connect these to the annotations "2) Add multiline Level 1 Comments" and "3) Add multiline Level 2 Comments" respectively.

1) Add multiline field business justification

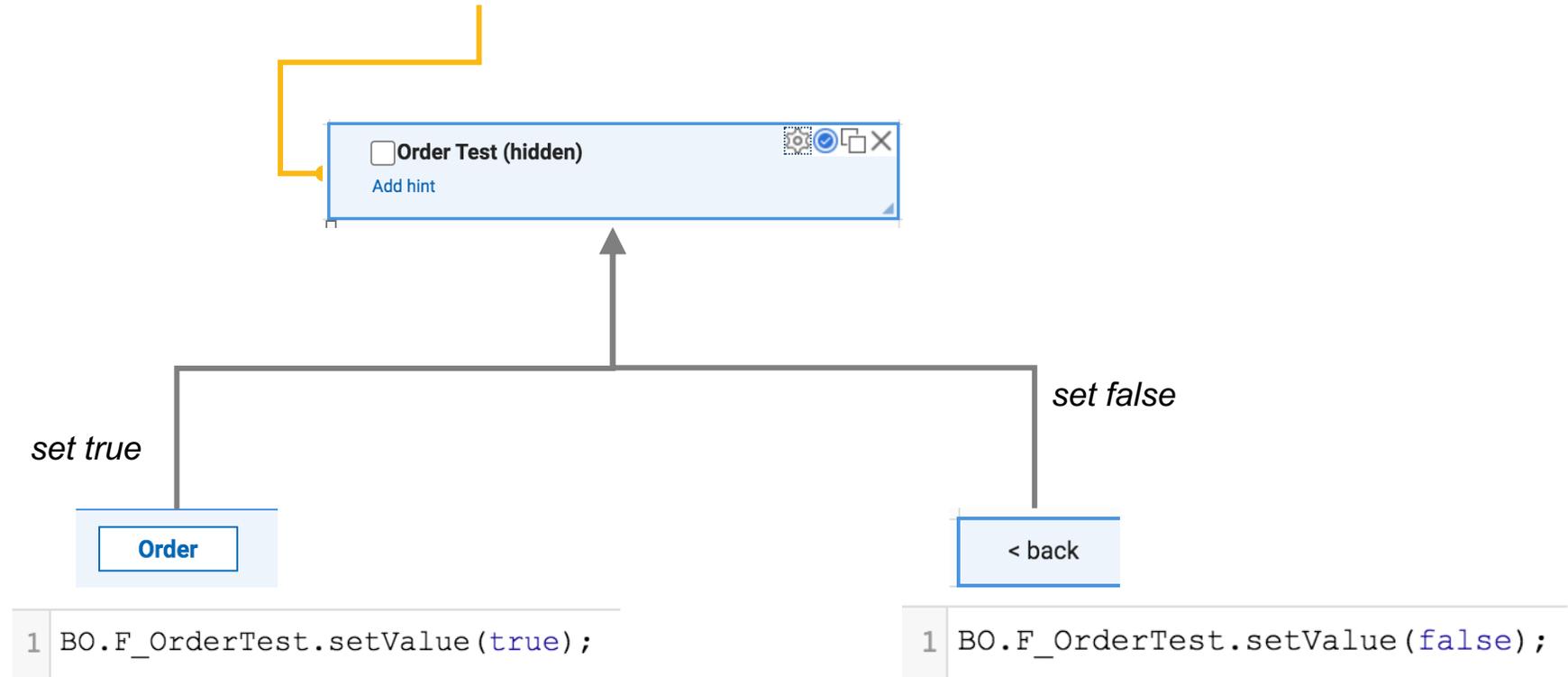
2) Add multiline Level 1 Comments

3) Add multiline Level 2 Comments

## 7. Rules for a dynamic user experience

### ► Determine if order button or back text has been clicked

1) Add Checkbox in the hidden section to record if order button has been clicked



2) Add JavaScript to the button onClick to set checkbox 'true'

3) Add JavaScript to the '< back' text item onClick to set checkbox 'false'

# 7. Rules for a dynamic user experience

## ► Making the form dynamic based on checkbox value

The screenshot displays the 'Rules' configuration window. On the left, a sidebar shows 'Show rules related to: F\_BackText' and 'Rule 6'. The main area contains a table of rules for 'F\_BackText':

Item	Action	Control
Start - Submit	Show	+
Workflow Approver Section	Show	+
Details Presentation Section	Show	+
F_BackText	Show	+
Select a Phone	Hide	-
Order	Hide	-
Business Justification	Show	+

Below the table, a condition is defined: 'When the following condition is true: Order Test (hidden) Equals A fixed value'. The 'Select item' dropdown is set to 'Checked'.

Yellow callout lines point to the following elements:

- Show submit action button
- Show workflow approver section
- Show detail presentation section
- Show '<back>' text item
- Hide phone selection table
- Hide order button
- Show business justification
- When Order Test is checked (true)

# 7. Rules for a dynamic user experience

- ▶ Hide the approver comments in start

The screenshot displays a user interface for an 'Order - Start' process. On the left, a 'Stages' sidebar shows a sequence of steps: Start, Submit, Level 1, Level 2, Approved, and End. The 'Start' stage is currently active. The main content area shows a form with the following elements: 'Supplier: {Supplier}', 'Price: {Price}', a 'Business Justification' text area, an 'Order' button, and two comment sections: 'Level 1 Manager Comments' and 'Level 2 Manager Comments'. Annotations with yellow lines point to the 'Level 1 Manager Comments' and 'Level 2 Manager Comments' fields, indicating they should be hidden.

Hide Level 1  
Comments in Start

Hide Level 2  
Comments in Start

# 7. Rules for a dynamic user experience

- ▶ Set visibility and read only rule in Level 1

Do the same in Level 2

The screenshot shows a software interface with a 'Stages' sidebar on the left and a main form area on the right. The sidebar has a 'Level 1' stage selected. The main form area contains several fields: 'Brand', 'Model', 'Storage Capacity', 'Supplier', 'Price', 'Business Justification', 'Level 1 Manager Comments', and 'Level 2 Manager Comments'. A blue 'Order' button is located below the 'Business Justification' field. Three yellow lines connect text labels on the right to specific fields in the form: 'Lock the Business Justification field in Level 1' points to the 'Business Justification' field, 'Show Level 1 Comments in Level 1' points to the 'Level 1 Manager Comments' field, and 'Hide Level 2 Comments in Level 1' points to the 'Level 2 Manager Comments' field.

When Order Test is checked (true)

# 7. Rules for a dynamic user experience

## ▶ How things should work – rules and navigation

[MobilePhoneOrders5.volt](#)

### Mobile Phone Order System

Select a Phone

Brand	Model	Storage	Platform
Apple	iPhone 11	64	IOS
Apple	iPhone 11	128	IOS
Apple	iPhone 11 pro	64	IOS
Apple	iPhone 11 pro	256	IOS
Samsung	Galaxy Fold 5G	512	Android

[This phone is \\$900 or greater and requires 2 levels of approval](#)

#### Apple iPhone 11 pro

Storage Capacity: 64

Supplier: Apple Store

Price: \$999.00



### Mobile Phone Order System

[< back](#)

[This phone is \\$900 or greater and requires 2 levels of approval](#)

#### Apple iPhone 11 pro

Storage Capacity: 64

Supplier: Apple Store

Price: \$999.00



**Business Justification**

**Requester**

**Requester Email**

**Level 1 Approver**

**Level 1 Email**

**Level 2 Approver**

**Level 2 Email**

## 8. Add a workflow approval log

### ▶ Adding fields and text to show approval status

Add the following fields to hidden section

**Order Created**  
 Date  Time

**When L1 Approved**  
 Date  Time

**When L2 Approved**  
 Date  Time

The values of these will be set via JavaScript on the Form onValidButtonPressed event

Add text item – with echo text as shown – to the approver section

<input type="checkbox"/> <b>Requester</b> <input type="text"/>	<input type="checkbox"/> <b>Requester Email</b> <input type="text"/>
<input type="checkbox"/> <b>Level 1 Approver</b> <input type="text"/>	<input type="checkbox"/> <b>Level 1 Email</b> <input type="text"/>
<input type="checkbox"/> <b>Level 2 Approver</b> <input type="text"/>	<input type="checkbox"/> <b>Level 2 Email</b> <input type="text"/>

- ✔ Order submitted by {Requester} on {Order Created}
- ✔ Level 1 was approved by {Level 1 Approver} on {When L1 Approved}
- ✔ Level 2 was approved by {Level 2 Approver} on {When L2 Approved}

Set the first text item so that it does not show in Start stage

## 8. Add a workflow approval log

### ▶ Adding function to record approval changes

Add JavaScript to Form validButtonPressed event.  
(adjust code to your IDs)

```
//record order creation time
if(pActionId === 'S_Submit')
{
BO.F_OrderCreated.setValue(new Date());
}

//record L1 approval time
if(pActionId === 'S_1LevelRequired' || pActionId
=== 'S_2LevelRequired')
{
BO.F_L1ApproverTime.setValue(new Date())
}

//record L2 approval time
if(pActionId === 'S_Level2Approval')
{
BO.F_L2ApproverTime.setValue(new Date());
}
```

Records when order submitted

Records when Level 1 approved

Records when Level 2 approved

## 8. Add a workflow approval log

### ▶ Adding rules to display approval status

Created}

⚙️ ✓ 📄 ✕

✓ Level 1 was approved  
by {Level 1 Approver} on {When  
L1 Approved}

⚙️ ✓ 📄 ✕

✓ Level 2 was approved  
by {Level 2 Approver} on {When  
L2 Approved}

Set rules on the Level 1 and Level 2 text item so that are shown when the timestamp fields have a value

**Rules**

Show rules related to: ⓘ

F\_Text4

- Rule 11 ✕

Rule Name: Rule 11 ✓

Perform this action: ⓘ

F\_Text4 Show

When the following condition is true: ⓘ

When L1 Approved Has value

### Mobile Phone Order System

Select a Phone

Brand	Model	Storage	Platform
Apple	iPhone 11	64	IOS
Apple	iPhone 11	128	IOS
Apple	iPhone 11 pro	64	IOS
Apple	iPhone 11 pro	256	IOS
Samsung	Galaxy Fold 5G	512	Android

[This phone is \\$900 or greater and requires 2 levels of approval](#)

#### Apple iPhone 11 pro

Storage Capacity: 64

Supplier: Apple Store

Price: \$999.00



[Order](#)

### Mobile Phone Order System

[< back](#)

[This phone is \\$900 or greater and requires 2 levels of approval](#)

#### Apple iPhone 11 pro



Storage Capacity: 64

Supplier: Apple Store

Price: \$999.00

**Business Justification**

---

<b>Requester</b>	<b>Requester Email</b>
<input type="text" value="Admin"/>	<input type="text" value="martin.lechleider@hcl.com"/>
<b>Level 1 Approver</b>	<b>Level 1 Email</b>
<input type="text" value="Andrew Manby"/>	<input type="text" value="andrew.manby@pnp-hcl.com"/>
<b>Level 2 Approver</b>	<b>Level 2 Email</b>
<input type="text" value="Rajesh Iyer"/>	<input type="text" value="rajesh_i@hcl.com"/>

[Submit](#)

# Finished app – approval steps

## Level 1 approval

Mobile Phone Order System

This phone is \$900 or greater and requires 2 levels of approval

**Apple iPhone 11 pro**

Storage Capacity: 64

Supplier: Apple Store

Price: \$999.00

**Business Justification**

**Level 1 Manager Comments**

---

<b>Requester</b>	<b>Requester Email</b>	
<input type="text" value="Admin"/>	<input type="text" value="martin.lechleider@hcl.com"/>	✔ Order submitted by Admin on 10/16/2020, 2:26 PM
<b>Level 1 Approver</b>	<b>Level 1 Email</b>	
<input type="text" value="Andrew Manby"/>	<input type="text" value="andrew.manby@pnp-hcl.com"/>	
<b>Level 2 Approver</b>	<b>Level 2 Email</b>	
<input type="text" value="Rajesh Iyer"/>	<input type="text" value="rajesh_i@hcl.com"/>	

Approve

## Level 2 approval

Mobile Phone Order System

This phone is \$900 or greater and requires 2 levels of approval

**Apple iPhone 11 pro**

Storage Capacity: 64

Supplier: Apple Store

Price: \$999.00

**Business Justification**

**Level 1 Manager Comments**

**Level 2 Manager Comments**

---

<b>Requester</b>	<b>Requester Email</b>	
<input type="text" value="Admin"/>	<input type="text" value="martin.lechleider@hcl.com"/>	✔ Order submitted by Admin on 10/16/2020, 2:26 PM
<b>Level 1 Approver</b>	<b>Level 1 Email</b>	✔ Level 1 was approved by Andrew Manby on 10/16/2020, 2:28 PM
<input type="text" value="Andrew Manby"/>	<input type="text" value="andrew.manby@pnp-hcl.com"/>	
<b>Level 2 Approver</b>	<b>Level 2 Email</b>	
<input type="text" value="Rajesh Iyer"/>	<input type="text" value="rajesh_i@hcl.com"/>	

Submit

## Approval Complete

Mobile Phone Order System

This phone is \$900 or greater and requires 2 levels of approval

**Apple iPhone 11 pro**

Storage Capacity: 64

Supplier: Apple Store

Price: \$999.00

**Business Justification**

**Level 1 Manager Comments**

**Level 2 Manager Comments**

---

<b>Requester</b>	<b>Requester Email</b>	
<input type="text" value="Admin"/>	<input type="text" value="martin.lechleider@hcl.com"/>	✔ Order submitted by Admin on 10/16/2020, 2:26 PM
<b>Level 1 Approver</b>	<b>Level 1 Email</b>	✔ Level 1 was approved by Andrew Manby on 10/16/2020, 2:28 PM
<input type="text" value="Andrew Manby"/>	<input type="text" value="andrew.manby@pnp-hcl.com"/>	
<b>Level 2 Approver</b>	<b>Level 2 Email</b>	✔ Level 2 was approved by Rajesh Iyer on 10/16/2020, 2:29 PM
<input type="text" value="Rajesh Iyer"/>	<input type="text" value="rajesh_i@hcl.com"/>	

Send to End

**HCL**

*Relationship*<sup>TM</sup>  
BEYOND THE CONTRACT

\$8.4 BILLION ENTERPRISE | 132,000 IDEAPRENEURS | 44 COUNTRIES



WATCH THE FILM