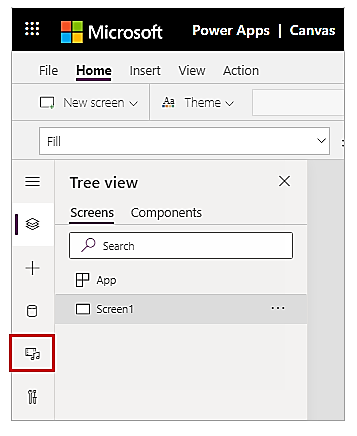
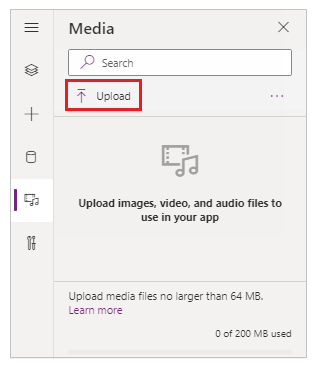
**Using Multimedia Files in Canvas Apps**

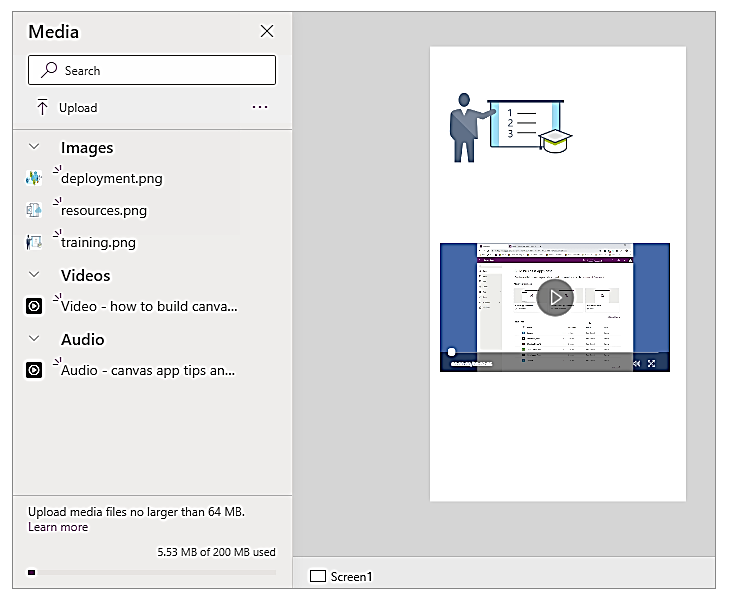
You can add images, audio and video files to a canvas app. Add video from streaming services such as Microsoft Stream, Azure Media Services, or 3rd party streaming services, such as YouTube. Or use input controls such as Pen Input to collect signatures.

**Add images, audio, or video using the media pane**

To use the Media pane to add, remove or use media files in your app:

1. Select Media from the left pane.



**Add images from the cloud to your app**

In this scenario, you save images in a cloud storage account, OneDrive for Business. You use an Excel table to contain the path to the images, and you display the images in a gallery control in your app.

1. Download CreateFirstApp.zip, and extract the Assets folder to your cloud storage account.
2. Rename the Assets folder to Assets\_images.
3. In an Excel spreadsheet, create a one-column table, and fill it with the following data.

A screenshot of a computer

Description automatically generated

To copy the file path from OneDrive for Business, select a file, and then select Path (Copy direct link) from the details pane on the right side of the screen.

1. Name the table Jackets, and name the Excel file Assets.xlsx.
2. In your app, add the Jackets table as a data source.
3. Optionally, update your app orientation to Landscape.
4. Select Insert > Gallery, and then select Horizontal.
5. Optionally, select text field and then the heading field under the first image, and delete them to keep only images on the screen.

**Upload pen drawings to the cloud**

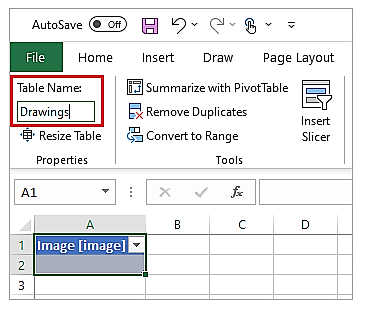
In this scenario, you learn how to upload pen drawings to your data source, OneDrive for Business, and examine how the drawings are stored there.

1. In Excel, add Image [image] to cell A1.
2. Create a table using the following steps:
   1. Select cell A1.
   2. On the Insert ribbon, select Table.
   3. In the dialog box, select My table has headers, and then select OK.

A screenshot of a computer

Description automatically generated

* 1. Name the table Drawings:



1. Save the Excel file to OneDrive for Business as SavePen.xlsx.
2. In Power Apps, create a blank app with Tablet layout.
3. In your app, add the OneDrive for Business account as a data source:
   1. Select View menu, and then select Data sources.
   2. Select Add data source, and then select OneDrive for Business.
   3. Select SavePen.xlsx.
   4. Select the Drawings table, and then select Connect.
4. Select Insert > Input, and then select Pen Input.
5. Rename the new control MyPen:

A screenshot of a computer program

Description automatically generated

1. On the Insert tab, add a Button control, and set its OnSelect property to this formula:

Patch(Drawings, Defaults(Drawings), {Image:MyPen.Image})

1. Add a Horizontal gallery control (Insert tab > Gallery).
2. Optionally, select text field and then the heading field under the first image, and delete them to keep only images on the screen.
3. Set gallery Items property to Drawings. The Image property of the gallery control is automatically set to ThisItem.Image.
4. Arrange the controls so that your screen resembles as shown below:

A screenshot of a computer

Description automatically generated

**Known limitations**

* To enable better performance while loading an app, following size restrictions apply:
* The total size of all media files uploaded to an app can't exceed 200 MB.
* Maximum size of an individual media file in an app can't exceed 64 MB.
* Supported media file types: .jpg, .jpeg, .gif, .png, .bmp, .tif, .tiff, .svg, .wav, .mp3, .mp4.
* Cloud-storage known limitations apply when connecting your app with cloud-based storage.

**Create an App with Relational Data**

By following the steps in this series of topics, you can discover concepts about relational data as implemented in a sample database in Microsoft Dataverse. You can also explore sample business apps, both canvas and model-driven, for managing that data and earn practical experience by creating such an app. This first topic explains how to install the Northwind Traders database in your own environment and gain access to the sample apps, which you can open for editing to reveal how they were built.

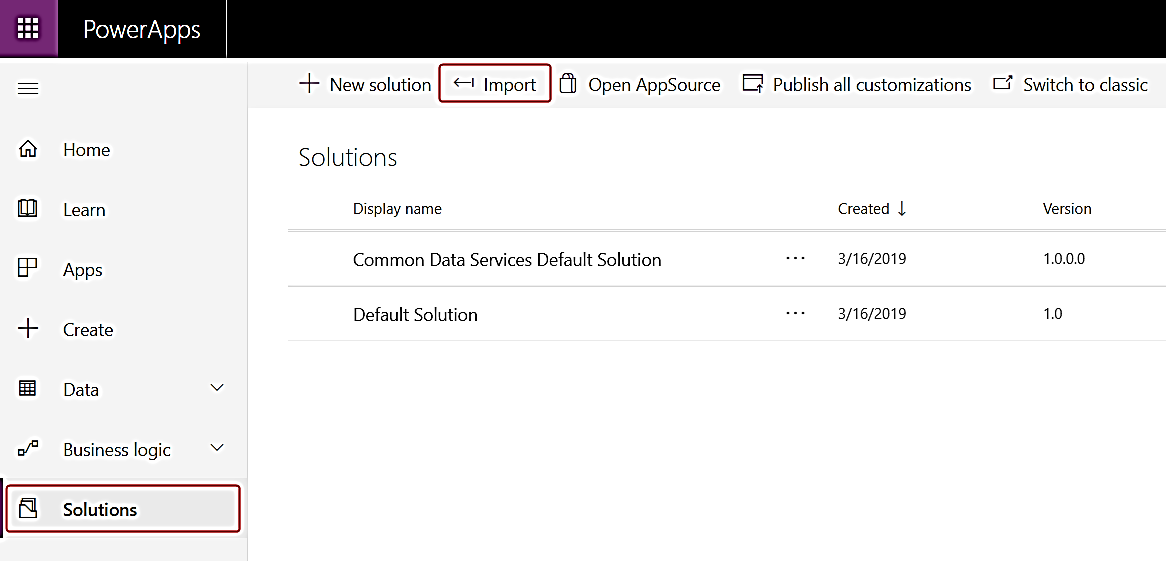
**Install Northwind Traders database and apps**

1. Download the solution

<https://download.microsoft.com/download/f/8/d/f8ddbb69-5499-4776-bd41-00f3bae050a6/NorthwindTraders_final.zip>

This solution file (.zip) contains the definitions of tables, choices, and business processes; the canvas and model-driven apps; and any other pieces that are used together.

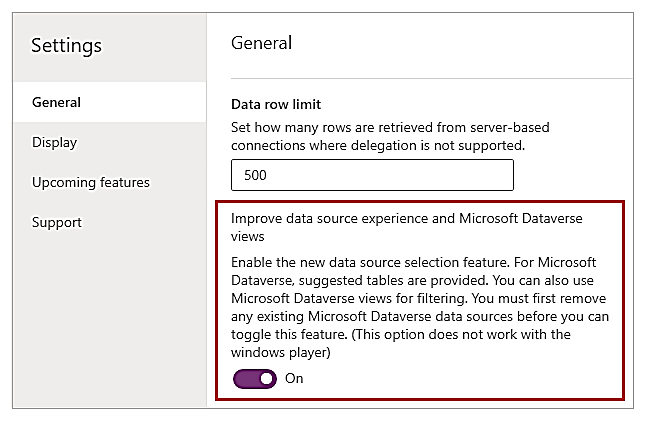
1. Install the solution
   1. Sign in to Power Apps, and then ensure that you're working in an environment that contains a Dataverse database.
   2. In the left navigation pane, select Solutions. If the item isn’t in the left navigation pane, select …More and then select the item you want. Then select Import in the toolbar near the top of the screen:



* 1. In the Select Solution Package page, select Browse.
  2. Find the file that you downloaded, and then select Open.
  3. In the Import Options page, select Import to confirm SDK message handling, which the sample requires and follow thru the wizard.

**Load the sample data**

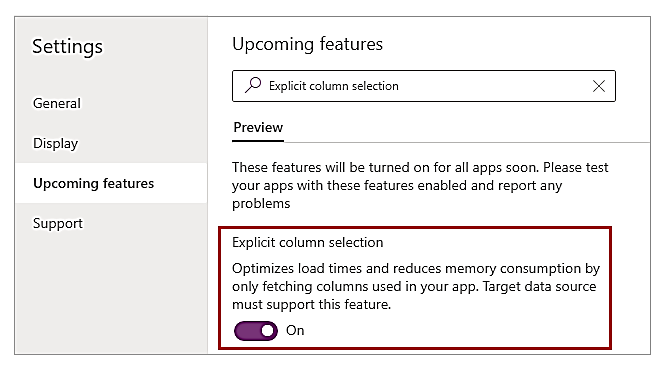
1. Select Apps, and then select **Northwind Sample Data**.
2. When the app asks for permission to interact with Dataverse, select Allow.
3. If you created a canvas app with either the Dataverse or the Dynamics 365 connector before November 2019, then you might not have the benefits of using the current native connection experience for the Dataverse.



1. Improve data source experience and Microsoft Dataverse views is On:

Either you already converted your canvas app to use this feature, or you started an app with default setting of On for this feature. No further actions required.

You may also want to enable the Explicit Column Selection feature:



**Note**

* Improve data source experience and Microsoft Dataverse views is not supported on Power Apps for Windows. You must turn this feature Off when using Power Apps for Windows.
* When this feature is Off, you'll see the following message. You can ignore this message when using Power Apps mobile app for Windows platform.

***This app is using a connector for the Microsoft Dataverse or Dynamics 365 that will not be supported past Oct 1, 2020.***

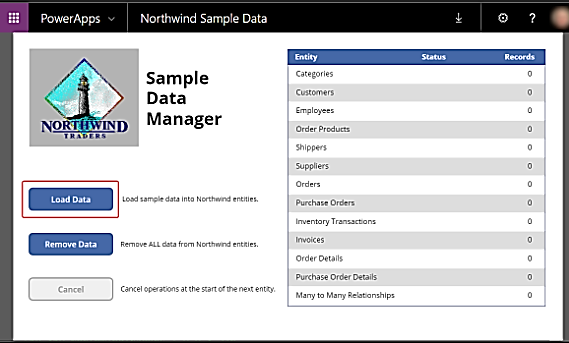
1. Relational data, option sets and other new features for Microsoft Dataverse is Off:

Check Retired section under Upcoming features. If set to Off, continue with the following instructions as a first step in the conversion.

**Important**

* **If you don't see Relational data, option sets and other new features for Microsoft Dataverse in General settings, or if it’s already On, skip the following steps and continue to the next section.**

1. After the app loads and shows that the sample tables contain no records, select Load Data to populate the tables.



As the app loads the data, dots march across the top of the app, and the number of records increases.

Tables are loaded in a specific order so that relationships can be established between records. For example, the Order Details table has a many-to-one relationship with the Orders and Order Products tables, which are loaded first.

You can cancel the process at any time by selecting Cancel, and you can remove the data at any time by selecting Remove Data:

A screenshot of a computer

Description automatically generated

1. When the data finishes loading, the last row (Many to Many Relationships) shows Done, and the Load Data and Remove Data buttons are enabled again:

A screenshot of a data manager

Description automatically generated

1. Sample apps

The Northwind solution includes these apps for interacting with this data:

* Northwind Orders (Canvas)
* Northwind Orders (Model-driven)

You open these apps the same way that you opened the app in the previous procedure.

1. Canvas

This single-screen app offers a simple master-detail view of the Orders table, where you can view and edit a summary of the order and each line item for an order. A list of orders appears near the left edge, and you can select an arrow in that list to show a summary and the details of that order.

A screenshot of a computer

Description automatically generated

1. Model-driven

This app operates on the same data (in the Orders table) as the canvas app. In the list of orders, show more information about an order by selecting its number:

A screenshot of a computer screen

Description automatically generated

**(Part 1) Create an order gallery in a canvas app**

A screenshot of a computer

Description automatically generated

1. Create a blank app
   1. Create a blank canvas app, and give it a name such as "My Northwind Orders (Canvas app)".
2. Add the data
   1. On the left-pane, select Data > Add data > search for, and select Orders table.
   2. Select View, Data sources, Add data source.

The Orders table contains many fields of various types:

A screenshot of a computer

Description automatically generated

Each field has a Display name and a Name, which is sometimes called the logical name.

Both names refer to the same thing. In general, you'll use the display name when you build an app, but some cases require the more cryptic Name, as noted in a procedure.

* 1. As we will be working with screens and controls next, in Power Apps Studio switch back to the Tree View on the left hand side by pressing the three stacked squares icon. You can return to the Data Sources at any time by pressing the cylinder icon.

1. Create the order gallery
   1. On the Insert tab, select Gallery > Blank vertical to add a Gallery control, which will show the orders.

A screenshot of a computer

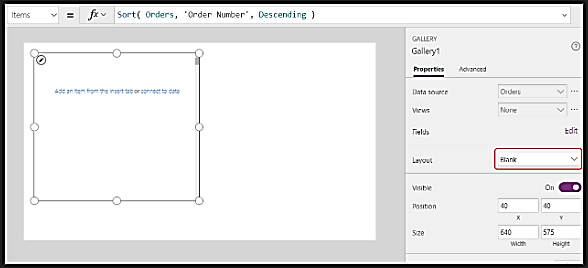
Description automatically generated

The control will be placed on the canvas and a fly out dialog will appear asking which data source to connect to.

* 1. We could connect it directly to Orders here, but instead we'd like to control the sort order of the gallery. Ignore the fly out dialog and in the formula bar set the gallery's Items property to this formula:

Sort( Orders, 'Order Number', Descending )

* 1. After a few moments the Result View will appear below the formula bar. Pull down on the arrow at the left to see the result of our formula. Scroll to the right to see the Order Number column and ensure it is sorted the way want (highest to lowest).
  2. In the Properties tab near the right edge, open the Layout list:



* 1. In the list of options, select Title and subtitle:
  2. Two Label controls are added in the gallery's template. By default, these controls show two columns of the Orders table, which you'll change next. The gallery's template is replicated vertically for each record in the table.
  3. Select Edit (next to Fields) in the Properties tab near the right edge.
  4. In the Data pane, select Title1 (or select the upper label in the gallery's template).

In the formula bar, set the label's Text property to this expression:

"Order " & ThisItem.'Order Number'

The order number appears at the top of each gallery item. In the gallery template, ThisItem grants access to all fields in the Order table.

* 1. In the Data pane, select Subtitle1 (or select the lower label in the gallery's template):

In the formula bar, set the label's Text property to this expression:

ThisItem.Customer.Company

After you enter this formula, it may show a red squiggly error for a moment. The error should clear if you select anything outside the formula bar and then return the cursor to the formula bar. If the error persists or you don't see a value, select the View tab, select Data sources, and then refresh the Orders table by selecting the ellipsis (...) to the right of the data-source name.

When you specify ThisItem.Customer, you're leveraging a many-to-one relationship between the Orders and Customers tables and retrieving the customer record that's associated with each order. From the customer record, you're extracting data in the Company column for display.

You can show all the relationships from the Orders table to other tables, including the Customer table:

A screenshot of a computer

Description automatically generated

* 1. Close the Data pane by selecting the close icon (x) in its upper-right corner.

1. Show each order's status

In this procedure, you'll add space in the gallery for a label and configure it to show each order's status in a different color based on the data.

* 1. In the gallery's template, reduce the width of the first label, Title1:

A screenshot of a computer

Description automatically generated

* 1. Repeat the previous step with the second label, Subtitle1:
  2. With the gallery template (or a control in the template) selected, select Label on the Insert tab, move the new label to the right of the Title1 label:
  3. Set the Text property of the new label to this expression:

ThisItem.'Order Status'

A screenshot of a computer

Description automatically generated

In the Orders table, the Order Status field holds a value from the Orders Status choice. A choice is similar to an enumeration in other programming tools. Each set of options is defined in the database, so users can specify only those options that are in the set. The Orders Status choice is also global, not local, so you can use it in other tables:

A screenshot of a computer

Description automatically generated

Each option in a set has a name that appears if you show it in a label. These names can be localized, and the app recognizes the same option whether an English user selects Apple, a French user selects Pomme, or a Spanish user selects Manzana. For this reason, you can't create a formula that relies on a hard-coded string for an option, as this topic demonstrates later.

In formulas, you must surround Order Status with single quotation marks because it contains a space. However, that name functions the same way as any other name in Power Apps, such as Customer or Company, does.

* 1. On the Home tab, increase the font size of the status label to 20 points, and right align the text:

A screenshot of a computer

Description automatically generated

* 1. In the formula bar, set the Color property of the status label to this formula:

Switch( ThisItem.'Order Status',

'Orders Status'.Closed, Green,

'Orders Status'.New, Black,

'Orders Status'.Invoiced, Blue,

'Orders Status'.Shipped, Purple

)

Power Apps prevents you from creating a formula that relies on a hard-coded string for each option in a set because such formulas could produce inappropriate results if the option names are localized. Instead, the Switch function determines the color based on whatever string appears in the label based on the user's settings.

With this formula in place, different status values appear in different colors, as the previous graphic shows.

1. Display each order's total
   1. Select the first item in the gallery, which is the gallery's template
   2. On the Insert tab, select Label to add another label
   3. Move the new label so that it appears under the status label
   4. In the formula bar, set the new label's Text property to this formula

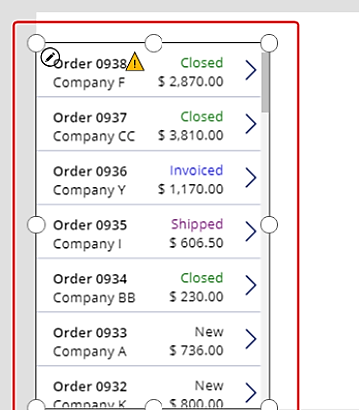
Text( Sum( ThisItem.'Order Details', Quantity \* 'Unit Price' ), "[$-en-US]$ #,###.00" )

In this formula, the Sum function adds up the records in the Order Details table that are associated with each record in the Order table through a one-to-many relationship. These line items make up each order, and you'll use the same one-to-many relationship to show and edit the line items in the lower-right area of the screen.

This formula shows a blue underline and a delegation warning because Dataverse doesn't support delegation of complex aggregate functions (for example, the sum of a multiplication). You can ignore this information because no order in this example will contain more than 500 line items. If necessary for a different app, you can increase that limit in App settings.

The Text function in this formula adds a currency symbol and formats the result with thousands and decimal separators. As written, the formula includes the language tag for U.S. English ([$-en-US]) and a dollar symbol ($). If you remove the language tag, it will be replaced with one based on your language settings, and the label will show the appropriate formats for that tag. If you leave the dollar symbol, the label will show the appropriate currency symbol based on the user's settings. However, you can force a different symbol to appear by replacing the dollar symbol with the one that you prefer.

* 1. On the Home tab, change the font size of the newest label to 20 points, and right align its text:
  2. Move the gallery to the left edge of the screen, and decrease the gallery's width to close up some space.
  3. Increase the gallery's height so that it's almost as tall as the screen, but leave a little room at the top for a title bar, which you'll add at the start of the next topic:



**(Part 2) Create a summary form in a canvas app**



1. Add a title bar
   1. In the Tree view pane, select Screen1 to ensure that you don't accidentally add a control to the order gallery
   2. On the Insert tab, select Label to insert a Label control:

A screenshot of a computer

Description automatically generated

The new label should appear only once, above the gallery. If it appears in each item of the gallery, delete the first instance of the label, ensure that the screen is selected (as the previous step describes), and then insert the label again.

* 1. Move and resize the new label to span the top of the screen:

A screenshot of a white box

Description automatically generated

* 1. Double-click the text of the label, and then type Northwind Orders.
  2. On the Home tab, format the label:
* Increase the font size to 24 points.
* Make the text bold.
* Make the text white.
* Center the text.
* Add a dark-blue fill to the background.

A blue and white sign with white text

Description automatically generated

1. Add an Edit form control
   1. On the Insert tab, insert an Edit form control:
   2. Move and resize the form to cover the upper-right corner of the screen under the title bar:

A screenshot of a computer

Description automatically generated

* 1. In the Properties pane, select the Data source drop down, select the Orders data source.

1. Add and arrange fields
   1. In the Properties tab near the right edge, select Edit fields to open the Fields pane.
   2. If the Fields pane is not empty, remove the fields that have already been inserted.
   3. After the fields list is empty, select Add field, and then select the check boxes for the Customer and Employee fields.
   4. Add the **Customer** and **Employee** fields to the Edit form control.
   5. Scroll down until these fields appear, and then select their check boxes:

* Notes
* Order Date
* Order Number
* Order Status
* Paid Date
  1. At the bottom of the Fields pane, select Add, and then close the Fields pane.
  2. The form shows seven fields, which may be in a different order:

A screenshot of a computer

Description automatically generated

Note

* If any field shows a red error icon, a problem might have occurred when data was pulled from the source. To resolve the error, refresh the data:
* On the View tab, select Data sources.
* In the Data pane, select Data sources.
* Next to Orders, select the ellipsis (...), select Refresh, and then close the Data pane.
* If the combo box for the customer or employee name still shows an error, check the Primary text and SearchField of each box by selecting it and then opening the Data pane. For the customer box, both fields should be set to nwind\_company. For the employee box, both fields should be set to nwind\_lastname.
  1. With the form selected, change the number of columns in the form from 3 to 12 in the Properties tab near the right edge.

This step adds flexibility as you arrange the fields:

Change then number of columns in the Edit form control.

Many UI designs rely on 12-column layouts because they can evenly accommodate rows of 1, 2, 3, 4, 6, and 12 controls. In this topic, you'll create rows that contain 1, 2, or 4 controls.

* 1. Move and resize the fields by dragging their handles, just as you would any other control, so that each row contains these data cards in the specified order:

First row: Order Number, Order Status, Order Date, and Paid Date

Second row: Customer and Employee

Third row: Notes

**Note**

You might find it easier to widen the Notes, Customer, and Employee data cards before you arrange them.

A screenshot of a computer

Description automatically generated

1. Hide time controls

In this example, you don't need the time portions of the date fields because that level of granularity can distract the user. If you delete them, you might cause problems in formulas that rely on those controls to update date values or determine the position of another control in the data card. Instead, you'll hide the time controls by setting their Visible property.

* 1. In the Tree view pane, select the Order Date data card.
  2. While holding down the Shift key, select the hour, minute, and colon-separator controls in the Order Date data card.

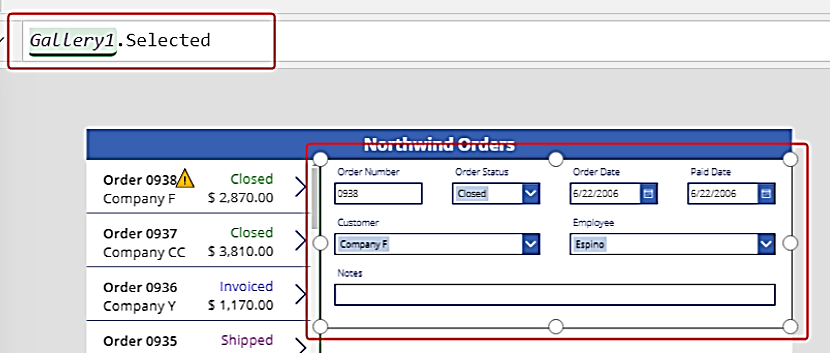
A screenshot of a computer

Description automatically generated

* 1. Set the controls' Visible property to false.
  2. Resize the Date picker control to show the complete date:
  3. Next, you'll repeat the last few steps for the Paid Date field.

1. Connect the order gallery
   1. In the Tree view pane, collapse the form to more easily find the name of the order gallery, and then, if necessary, rename it to Gallery1.
   2. Set the summary form's Item property to this expression:

Gallery1.Selected



The form shows a summary of whatever order the app user selects in the list.

1. Replace a data card

Order number is an identifier that Dataverse assigns automatically when you create a record. This field has a Text input control by default, but you'll replace it with a label so that the user can't edit this field.

* 1. Select the form, select Edit fields in the Properties tab near the right edge, and then select the Order number field
  2. Open the Control type list

A screenshot of a computer

Description automatically generated

* 1. Select the View text data card:
  2. Close the Fields pane.
  3. The user can no longer change the order number
  4. On the Home tab, change the order number's font size to 20 points so that the field is easier to find:

1. Use a many-to-one relationship

The Orders table has a many-to-one relationship with the Employees table: each employee can create many orders, but each order can be assigned to only one employee.

When the user selects an employee in the Combo box control, its Selected property provides that employee's entire record from the Employees table. As a result, you can configure an Image control to show the picture of whatever employee the user selects in the combo box.

* 1. Select the Employee data card

A screenshot of a computer

Description automatically generated

* 1. In the Advanced tab near the right edge, unlock the data card so that you can edit formulas that were previously read-only:

A screenshot of a computer

Description automatically generated

* 1. In the data card, reduce the width of the combo box to make room for the employee picture:
  2. On the Insert tab, select Media > Image
  3. An image appears in the data card, which expands to accommodate it:
  4. Resize the image, and move it to the right of the combo box:
  5. Move and resize the image control.
  6. Set the Image property of the image to this formula, replacing the number at the end of DataCardValue if necessary:

DataCardValue7.Selected.Picture

A screenshot of a computer

Description automatically generated

The picture of the selected employee appears.

* 1. While holding down the Alt key, select a different employee in the combo box to confirm that the picture also changes.

1. Add a Save icon
   1. In the Tree view pane, select Screen1, and then select Insert > Icons > Check
   2. The Check icon appears in the upper-left corner by default, where other controls might make the icon difficult to find
   3. On the Home tab, change the Color property of the icon to white, resize the icon, and move it near the right edge of the title bar
   4. In the Tree view pane, confirm that the form's name is Form1, and then set the icon's OnSelect property to this formula:

SubmitForm( Form1 )

A screenshot of a computer

Description automatically generated

When the user selects the icon, the SubmitForm function gathers any changed values in the form and submits them to the data source. Dots march across the top of the screen as the data is submitted, and the order gallery reflects the changes after the process finishes.

* 1. Set the icon's DisplayMode property to this formula:

If( Form1.Unsaved, DisplayMode.Edit, DisplayMode.Disabled )

If all changes in the form have been saved, the icon is disabled and appears in the DisabledColor, which you'll set next.

* 1. Set the icon's DisabledColor property to this value:

Gray

The user can save changes to an order by selecting the Check icon, which is then disabled and dimmed until the user makes another change

1. Add a Cancel icon
   1. On the Insert tab, select Icons > Cancel
   2. The icon appears in the upper-left corner by default, where other controls might make the icon difficult to find
   3. On the Home tab, change the icon's Color property to white, resize the icon, and move it to the left of the Check icon
   4. Set the Cancel icon's OnSelect property to this formula:

ResetForm( Form1 )

The ResetForm function discards all changes in the form, which returns it to its original state.

* 1. Set the Cancel icon's DisplayMode property to this formula:

If( Form1.Unsaved Or Form1.Mode = FormMode.New, DisplayMode.Edit, DisplayMode.Disabled )

This formula differs slightly from the one for the Check icon. The Cancel icon is disabled if all changes have been saved or the form is in New mode, which you'll enable next. In that case, ResetForm discards the new record.

* 1. Set the Cancel icon's DisabledColor property to this value:

Gray

The user can cancel changes to an order, and the Check and Cancel icons are disabled and dimmed if all changes have been saved

1. Add an Add icon
   1. On the Insert tab, select Icons > Add.
   2. The Add icon appears in the upper-left corner by default, where other controls might make it difficult to find
   3. On the Home tab, set the Color property of the Add icon to white, resize the icon, and move it to the left of the Cancel icon
   4. Set the Add icon's OnSelect property to this formula:

NewForm( Form1 )

The NewForm function shows a blank record in the form.

* 1. Set the Add icon's DisplayMode property to this formula:

If( Form1.Unsaved Or Form1.Mode = FormMode.New, DisplayMode.Disabled, DisplayMode.Edit )

The formula disables the Add icon under these conditions:

* The user makes changes but doesn't save or cancel them, which is the opposite behavior from the Check and Cancel icons.
* The user selects the Add icon but makes no changes.
  1. Set the Add icon's DisabledColor property to this value:

Gray

The user can create an order if they make no changes or they save or cancel any changes they've made. (If the user selects this icon, they can't select it again until they make one or more changes and then save or cancel those changes):

Note

* If you create and save an order, you might need to scroll down in the order gallery to show your new order. It won't have a total price because you haven't added any order details yet.

1. Add a Trash icon
   1. On the Insert tab, select Icons > Trash.
   2. The Trash icon appears in the upper-left corner by default, where other controls might make it difficult to find
   3. On the Home tab, change the Trash icon's Color property to white, resize the icon, and move it to the left of the Add icon
   4. Set the Trash icon's OnSelect property to this formula:

Remove( Orders, Gallery1.Selected )

The Remove function removes a record from a data source. In this formula, the function removes the record that's selected in the order gallery. The Trash icon appears near the summary form (not the order gallery) because the form shows more details about the record, so the user can more easily identify the record that the formula will delete.

* 1. Set the Trash icon's DisplayMode property to this formula:

If( Form1.Mode = FormMode.New, DisplayMode.Disabled, DisplayMode.Edit )

This formula disables the Trash icon if the user is creating a record. Until the user saves the record, the Remove function has no record to delete.

* 1. Set the Trash icon's DisabledColor property to this value:

Gray

The user can delete an order.

A screenshot of a computer

Description automatically generated

**(Part 3) Create a detail gallery in a canvas app**

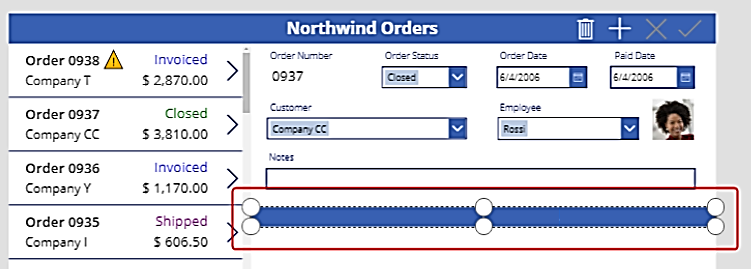


1. Create another title bar
   1. At the top of the screen, select the Label control that acts as a title bar, copy it by pressing Ctrl-C, and then paste it by pressing Ctrl-V:

A screenshot of a computer

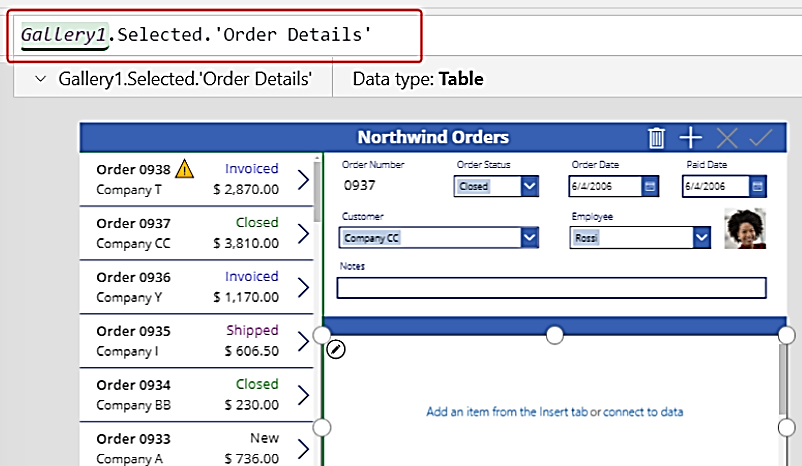
Description automatically generated

* 1. Resize and move the copy so that it appears just under the summary form.
  2. Remove the text from the copy



1. Add a gallery
   1. Insert a Gallery control with a Blank vertical layout:
   2. Close the fly out data source dialog, and then resize and move the detail gallery to the lower-right corner, below the new title bar:
   3. Set the Items property of the detail gallery to this formula:

Gallery1.Selected.'Order Details'



If an error appears, confirm that the order gallery is named Gallery1 (in the Tree view pane near the left edge). If that gallery has a different name, rename it Gallery1.

You've just linked the two galleries. When the user selects an order in the order gallery, that selection identifies a record in the Orders table. If that order contains one or more line items, the record in the Orders table is linked to one or more records in the Order details table, and data from those records appears in the detail gallery. This behavior reflects the one-to-many relationship that was created for you between the Orders and Order Details tables. The formula that you specified "walks" that relationship by using dot notation.

1. Show product names
   1. In the detail gallery, select Add an item from the Insert tab to select the **gallery template**

**Note:**

* + - **Ensure that you've selected the gallery template instead of the gallery itself. The bounding box should be slightly inside the gallery's boundary and probably shorter than the gallery's height. As you insert controls into this template, they are repeated for each item in the gallery.**
  1. On the Insert tab, insert a label into the detail gallery.

The label should appear within the gallery; if it doesn't, try again, but make sure to select the gallery's template before you insert the label.

* 1. Set the new label's Text property to this formula:

ThisItem.Product.'Product Name'

If no text appears, select the arrow for Order 0901 near the bottom of the order gallery.

* 1. Resize the label so that the full text appears:

A screenshot of a computer

Description automatically generated

This expression walks from a record in the Order Details table. The record is held in ThisItem over to the Order Products table through a many-to-one relationship.

The Product Name column (and other columns that you're about to use) are extracted.

1. Show product images
   1. On the Insert tab, insert an Image control into the detail gallery:
   2. Resize and move the image and the label to be side by side.

**Tip**

* **For fine-grained control over the size and the position of a control, start to resize or move it without pressing the Alt key, and then continue to resize or move the control while you hold down the Alt key:**
  1. Set the image's Image property to this formula:

ThisItem.Product.Picture

Again, the expression references a product that's associated with this order detail and extracting the Picture field to display.

A screenshot of a computer

Description automatically generated

* 1. Reduce the height of the gallery's template so that more than one Order Detail record appears at a time:

A screenshot of a computer

Description automatically generated

1. Show product quantity and cost
   1. On the Insert tab, insert another label into the detail gallery, and then resize and move the new label to the right of the product information.
   2. Set the new label's Text property to this expression:

ThisItem.Quantity

This formula pulls information directly from the Order Details table (no relationship required).

A screenshot of a computer

Description automatically generated

* 1. On the Home tab, change the alignment of this control to Right
  2. On the Insert tab, insert another label into the detail gallery, and then resize and move the label to the right of the quantity label.
  3. Set the new label's Text property to this formula:

Text( ThisItem.'Unit Price', "[$-en-US]$ #,###.00" )

If you don't include the language tag ([$-en-US]), it will be added for you based on your language and region. If you use a different language tag, you'll want to remove the $ just after the close square bracket (]), and then add your own currency symbol in that position.

A screenshot of a computer

Description automatically generated

* 1. On the Home tab, change the alignment of this control to Right:
  2. On the Insert tab, insert another label control into the detail gallery, and then resize and move the new label to the right of the unit price.
  3. Set the new label's Text property to this formula:

Text( ThisItem.Quantity \* ThisItem.'Unit Price', "[$-en-US]$ #,###.00" )

Again, if you don't include the language tag ([$-en-US]), it will be added for you based on your language and region. If the tag is different, you'll want to use your own currency symbol instead of the $ just after the close square bracket (]).

A screenshot of a computer

Description automatically generated

* 1. On the Home tab, change the alignment of this control to Right
  2. You're done adding controls to the detail gallery for now.
  3. In the Tree view pane, select Screen1 to ensure that the detail gallery is no longer selected.

1. Add text to the new title bar
   1. On the Insert tab, insert another label on to the screen:
   2. Resize and move the new label above the pictures of the products in the second title bar, and then change the text's color to white on the Home tab.
   3. Double-click the label's text, and then type Product:

A screenshot of a computer

Description automatically generated

* 1. Copy and paste the product label, and then resize and move the copy above the quantity column.
  2. Double-click the new label's text, and then type Quantity:
  3. Copy and paste the quantity label, and then resize and move the copy above the unit-price column.
  4. Double-click the new label's text, and then type Unit Price:
  5. Copy and paste the unit-price label, and then resize and move the copy above the extended-price column.
  6. Double-click the text of the new label, and then type Extended:
  7. Change label text to Extended.

A screenshot of a computer

Description automatically generated

1. Display order totals
   1. Reduce the height of the detail gallery to make room for the order totals at the bottom of the screen:
   2. Copy and paste the title bar in the middle of the screen, and then move the copy to the bottom of the screen
   3. Copy and paste the product label from the middle title bar, and then move the copy to the bottom title bar, just to the left of the Quantity column.
   4. Double-click the new label's text, and then type this text:

Order Totals:

A screenshot of a computer

Description automatically generated

* 1. Set the new label's Text property to this formula:

Sum( Gallery1.Selected.'Order Details', Quantity )

This formula shows a delegation warning, but you can ignore it because no single order will contain more than 500 products.

* 1. On the Home tab, set the new label's text alignment to Right:

A screenshot of a computer screen

Description automatically generated

* 1. Copy and paste this label control, and then resize and move the copy under the Extended column.
  2. Set the copy's Text property to this formula:

Text( Sum( Gallery1.Selected.'Order Details', Quantity \* 'Unit Price' ), "[$-en-US]$ #,###.00" )

This formula shows a delegation warning, but you can ignore it because no single order will contain more than 500 products.

A screenshot of a computer

Description automatically generated

1. Add space for new details

In any gallery, you can show data but you can't update it or add records. Under the detail gallery, you'll add an area where the user can configure a record in the Order Details table and insert that record into an order.

* 1. Reduce the height of the detail gallery enough to make room for a single-item editing space under that gallery.

A screenshot of a computer

Description automatically generated

* 1. On the Insert tab, insert a label, and then resize and move it under the detail gallery.
  2. Double-click the text of the new label, and then press Delete.
  3. On the Home tab, set the new label's Fill color to LightBlue:

A screenshot of a computer

Description automatically generated

1. Select a product
   1. On the Insert tab, select Controls > Combo box
   2. In the fly out dialog, select the Order Products data source.
   3. In the Properties tab for the combo box, select Edit (next to Fields) to open the Data pane. Ensure that the Primary text and SearchField are set to nwind\_productname.

You specify the logical name because the Data pane doesn't support display names in this case yet:

A screenshot of a computer

Description automatically generated

* 1. Close the Data pane.
  2. In the Properties tab near the right edge, scroll down, turn off Allow multiple selection, and ensure that Allow searching is turned on:

A screenshot of a computer

Description automatically generated

* 1. Resize and move the combo box to the light-blue area, just under the product-name column in the detail gallery:

A screenshot of a computer

Description automatically generated

In this combo box, the user will specify a record in the Product table for the Order Details record that the app will create.

* 1. While holding down the Alt key, select the combo box's down arrow.

**Tip**

* **By holding down the Alt key, you can interact with controls in Power Apps Studio without opening Preview mode.**
  1. In the list of products that appears, select a product:

A screenshot of a computer

Description automatically generated

1. Add a product image
   1. On the Insert tab, select Media > Image
   2. Resize and move the image to the light-blue area under the other product images and next to the combo box.
   3. Set the Image property of the image to:

ComboBox1.Selected.Picture

A screenshot of a computer screen

Description automatically generated

You're using the same trick as you used to show the employee picture in the summary form. The Selected property of the combo box returns the entire record of whatever product the user selects, including the Picture field.

1. Add a quantity box
   1. On the Insert tab, select Text > Text input:
   2. Resize and move the text-input box to the right of the combo box, under the quantity column in the detail gallery:

A screenshot of a computer screen

Description automatically generated

By using this text-input box, the user will specify the Quantity field of the Order Details record.

* 1. Set the Default property of this control to "".
  2. On the Home tab, set the text alignment of this control to Right.

1. Show the unit and extended prices
   1. On the Insert tab, insert a Label control.
   2. Resize and move the label to the right of the text-input control, and set the label's Text property to this formula:

Text( ComboBox1.Selected.'List Price', "[$-en-US]$ #,###.00" )

A screenshot of a computer

Description automatically generated

This control shows the List Price from the Order Products table. This value will determine the Unit Price field in the Order Details record.

**Note**

* **For this scenario, the value is read-only, but other scenarios might call for the app user to modify it. In that case, use a Text input control, and set its Default property to List Price.**
  1. On the Home tab, set the text alignment of the list-price label to Right:
  2. Copy and paste the list-price label, and then resize and move the copy to the right of the list-price label.
  3. Set the new label's Text property to this formula:

Text( Value(TextInput1.Text) \* ComboBox1.Selected.'List Price', "[$-en-US]$ #,###.00" )

A screenshot of a computer

Description automatically generated

This control shows the extended price based on the quantity that the app user specified and the list price of the product that the app user selected. It's purely informational for the app user.

* 1. Double-click the text-input control for quantity, and then type a number.

The Extended price label recalculates to show the new value:

A screenshot of a computer

Description automatically generated

1. Add an Add icon
   1. On the Insert tab, select Icons > Add
   2. Resize and move this icon to the right edge of the light-blue area, and then set the icon's OnSelect property to this formula:

Patch( 'Order Details',

Defaults('Order Details'),

{

Order: Gallery1.Selected,

Product: ComboBox1.Selected,

Quantity: Value(TextInput1.Text),

'Unit Price': ComboBox1.Selected.'List Price'

}

);

Refresh( Orders );

Reset( ComboBox1 );

Reset( TextInput1 )

A screenshot of a computer

Description automatically generated

In general, the **Patch** function updates and creates records, and the specific arguments in this formula determine the exact changes that the function will make.

* + - The first argument specifies the data source (in this case, the Order Details table) in which the function will update or create a record.
    - The second argument specifies that the function will create a record with the default values for the Order Details table unless otherwise specified in the third argument.
    - The third argument specifies that four columns in the new record will contain values from the user.
      * The Order column will contain the number of the order that the user selected in order gallery.
      * The Product column will contain the name of the product that the user selected in the combo box that shows products.
      * The Quantity column will contain the value that the user specified in the text-input box.
      * The Unit Price column will contain the list price of the product that the user selected for this order detail.

**Note**

**You can build formulas that use data from any column (in the Order Products table) for whatever product the app user selects in the combo box that shows products. When the user selects a record in the Order Products table, not only does the product’s name appear in that combo box but also the product’s unit price appears in a label. Each lookup value in a canvas app references an entire record, not just a primary key.**

The Refresh function ensures that the Orders table reflects the record that you’ve just added to the Order Details table. The Reset function clears the product, quantity, and unit-price data so that the user can more easily create another order detail for the same order.

* 1. Press F5, and then select the Add icon. The order reflects the information that you specified.
  2. (optional) Add another item to the order.
  3. Press Esc to close Preview mode.

1. Remove an order detail
   1. In the center of the screen, select the template of the detail gallery:
   2. On the Insert tab, select Icons > Trash:
   3. Resize and move the Trash icon to the right side of the detail gallery's template, and set the icon's OnSelect property to this formula:

Remove( 'Order Details', ThisItem ); Refresh( Orders )

A screenshot of a computer

Description automatically generated

As of this writing, you can't remove a record directly from a relationship, so the Remove function removes a record directly from the related table. ThisItem specifies the record to remove, taken from the same record in the detail gallery where the Trash icon appears.

Again, the operation uses cached data, so the Refresh function informs the Orders table that the app has changed one of its related tables.