BBSPA

Building Business Solutions with PowerApps and Flow

Mastering the Microsoft Power Platform

Complying with all applicable copyright laws is the responsibility of the user. All rights reserved. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without express written permission of Critical Path Training, LLC.

Printed and bound in Canada.

Microsoft, Windows, Power BI, PowerApps, Flow, Office 365, SharePoint and Microsoft Azure are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Course Manual Version 1.1

About This Training Course **5**
About the Hands-on Labs **7**
GitHub Repository for this course 7
Student Workstation Prerequisites 7
Copying the Student Lab Files to Your Windows PC 7
Module 01: Getting Started with the Power Platform **9**
Module 01 Lab: Getting Started with PowerApps Studio 43
Exercise 1: Create a new Microsoft 365 Trial Tenant 43
Exercise 2: Create a Trial Subscription for PowerApps Plan 2 48
Exercise 3: Create a Canvas App using the Start From Data Template 51
Exercise 4: Test the Expense Tracker Application From a Mobile Device 67
Exercise 5: Create a Canvas App using the Start from Blank Template 71
Module 02: Building Data-driven Canvas Apps **83**
Module 02 Lab: Building a Data-driven Canvas App 105
Exercise 1: Create a SharePoint List to Store Customer Data 105
Exercise 2: Create the Customer Ordering Canvas App 111
Exercise 3: Implement the Browse Customers Screen 122
Exercise 4: Implement the Add Customer Screen 135
Exercise 5: Implement the Edit Customer Screen 144
Exercise 6: Integrate a Custom Connector to Provide New Customer Data 149
Module 03: Building a Canvas App with a Shopping Cart **157**
Module 03 Lab: Building a Shopping Cart for Processing Orders 169
Exercise 1: Create The Browse Products Screen 169
Exercise 2: Implement Shopping Cart Collection for Ordering Products 176
Exercise 3: Create SharePoint Lists for Orders and Order Details 190
Exercise 4: Create the Submit Order Screen 197
Exercise 5: Create the Order Confirmation Screen (If you have time) 203
Module 04: Getting Started with Microsoft Flow **205**
Module 04 Lab: Getting Started with Microsoft Flow 247
Exercise 1: Create a New Twitter Account for Testing Purposes 247
Exercise 2: Create a Flow to Track Twitter Data in an Excel Workbook 250
Exercise 3: Create a Flow that is Manually Triggered by a Button 263
Exercise 4: Create a SharePoint List to Store Customer Data 269
Exercise 5: Create a Flow to Generate a Word Document from a SharePoint List Item 275
Module 05: Building Flows to Manage Content and Approvals **287**
Module 05 Lab: Creating Approvals in SharePoint Online using Microsoft Flow 313
Exercise 1: Create Two SharePoint Document Libraries for Uploading Photos 313
Exercise 2: Create a New Canvas App to Upload Photos to SharePoint 314
Exercise 3: Create A Flow to Automate a Photo Approval Process 327
Exercise 4: Test the Photo Approval Flow 333
Module 06: Integrating PowerApps with Power BI **337**
Module 06 Lab: Integrating Power BI with PowerApps and Flow 363
Exercise 1: Adding Power BI Content to a New App Workspace 363
Exercise 2: Embed Power BI Dashboard Tiles in PowerApps 376
Exercise 3: Extend a Power BI Report using PowerApps 379
Exercise 4: Use a Flow to Create a Real-time Dashboard in Power BI 388
Module 07: Getting Started with the Common Data Service **401**
Module 07 Lab: Working with the Common Data Service for Apps 437
Exercise 1: Create a New PowerApps Environment with a CDS Database 437
Exercise 2: Create a Canvas App to Manage Contact Entity Data 441
Exercise 3: Create a Model-driven App to Manage Contact Entity Data 449
Module 08: Integrating PowerApps with External Systems **459**
Module 08 Lab: Integrating External Systems with PowerApps and Flow 495
Exercise 1: Use the HTTP Connector to Retrieve Data from an External Web Service 495
Exercise 2: Execute a Child Flow from a Parent Flow 506
Exercise 3: Create a Custom Connector for an External Web Service 516

© Critical Path Training. 2019. All Rights Reserved

Module 09: PowerApps ALM using Managed Solutions 531

© Critical Path Training. 2019. All Rights Reserved

About This Training Course

Course Title:	Building Business Solutions with PowerApps and Flow
Course Code:	BBSPA
Audience:	Citizen Developers and Technical Professionals
Format:	In-person and Remote
Length	3 Days

Course Description

Building Business Solutions with Power Apps and Flow is an intensive 3-day training class designed for people who have already attended the Microsoft App-in-a-Day (AIID) training and are ready to move their PowerApps and Flow skills to the next level. Students will learn best practices for building canvas apps and flows to update and manage content in Azure SQL, SharePoint Online, Excel workbooks and OneDrive for Business.

After providing a foundation for building canvas apps and flows, this course teaches students how to build business solutions with the Common Data Service for Apps (CDSA) by creating custom entities and model-driven apps. The class also examines advanced techniques required in real-word PowerApps scenarios such as creating custom connectors and conducting application lifecycle management (ALM) using managed solutions and multiple environments for development, UAT and production.

Student Prerequisites

All students will require a Windows PC for lab exercises running Windows 10 or Windows 8.1. Students should already be familiar with Microsoft Excel, Office 365 and SharePoint Online. Due to the accelerated nature of this training class, it is also recommended that students attend Microsoft's App-in-a-day training before taking this class or have the equivalent hands-on experience with PowerApps and Flow.

© Critical Path Training. 2019. All Rights Reserved

About the Hands-on Labs

Building Business Solutions with PowerApps and Flow (BBSPA) includes a rich set of hands-on lab exercises designed to reinforce the key concepts and techniques that you learned in the lectures. The lab exercises for this training course were designed to run on a workstation or laptop running the Windows operating system. We recommend that you use Windows 10 but you can also Windows 8.1.

DISCLAIMER: While it is possible to complete many of the lab exercises for this course using a Mac instead of a Windows PC, the writeups for the lab exercises in this training course assume that you are using a Windows PC.

GitHub Repository for this course

Student files for this course are maintained in a GitHub fil repository at **https://github.com/CriticalPathTraining/BBSPA**.

Student Workstation Prerequisites

All lab exercises should be completed on a student workstation that has the following software.

1. Windows 10 or Windows 8.1.
2. One or more Internet browsers (Chrome, Edge, IE, etc.)
3. Microsoft Excel 2016 or Microsoft Excel 2013 - *If you don't already have it, you can optionally install it during lab 1.*

Copying the Student Lab Files to Your Windows PC

The most recent version of the electronic student files for this course are kept in the **Student** folder of the GitHub repository for this course. You can download the zip archive for this repository from the following URL.

https://github.com/CriticalPathTraining/BBSPA/archive/master.zip

It is recommended that you that you download the master zip archive and make a local copy of the **Student** folder so that you have a local copy of the files you will need on your computer workstation when going through these labs exercise. Once you download the master zip archive, open it and copy the **Student** folder to a new local folder.

Copy the **Student** folder from the master zip archive to a new local folder at **C:\Student**.

© Critical Path Training. 2019. All Rights Reserved

Module 01: Getting Started with the Power Platform

Module Description

This module introduces students to the Microsoft Power platform and explains the role of canvas apps, connectors and flows in building business solutions. The module introduces the Common Data Service for Apps (CDSA) and explains how it provides support for creating custom entities and building model-driven apps. Students will create canvas apps with PowerApps Studio and learn to write advanced expressions for screen and control properties. The module examines connectors and data binding and demonstrates using the Start with Data template. Along the way students will learn to build a canvas app for mobile devices that reads and writes customer data to a table inside an Excel workbook in OneDrive for Business.

Module Agenda

- Getting Started with the Power Platform
- Creating Canvas Apps
- Writing PowerApps Expressions
- Working with Connectors and Data Binding
- Understanding Delegation

Topics Covered

- Downloading Student Files
- Student Background Questionnaire
- What is the Power Platform?
- What Can You Build with PowerApps?
- Creating an Office 365 E5 Trial Tenant
- Microsoft 365 Admin Center
- Configuring a PowerApps Plan 2 License
- PowerApps Admin Center & Environments
- Creating a New Canvas App
- Getting Started with PowerApps Studio
- Running an App from PowerApps Studio
- Saving an App to the Cloud
- Building Apps using Screens and Controls
- Adding Controls to a Screen
- Configuring Control Properties
- PowerApps Formula Language
- Input and Output Properties
- Primitive Data Types
- Compound Data Types
- Events and State Changes
- Declarative vs Imperative Functions
- App OnStart
- Navigating Between Screens
- Understanding Connectors & Connections
- Data Binding with Galleries and Forms
- Working with the Data Pane
- Understanding Forms and Data Cards
- Changing a Field's Data Card Type
- Customizing a Data Card
- Populating a Dropdown Combobox
- Understanding Delegation
- Delegate-able Functions

Instructor Demos

- Configuring PowerApps Plan 2 Licenses
- Creating an App with the Budget Tracker App Template
- Navigating Between Screens
- Creating an App using the Start from Data Template
- Customizing Forms and Data Cards

© Critical Path Training. 2019. All Rights Reserved

© Critical Path Training. 2019. All Rights Reserved

At Critical Path Training, we try our best to stay on top of all of Microsoft's updates as they are applied to their cloud-based services such as PowerApps and Microsoft Flow. As you can imagine, this cloud cadence of monthly updates can pose a challenge to many training companies and content vendors who are faced with the problem of having to update their courseware on a monthly basis.

Critical Path Training has responded to challenge of dealing with monthly updates with a new student experience that we refer to as **Live Labs**. All the electronic student files for running demos and completing the lab exercises are published in a public GitHub repository which can be accessed with a browser at **https://github.com/CriticalPathTraining/BBSPA/archive/master.zip**.

It is recommended that you that you download the master zip archive and make a local copy of the **Student** folder so that you have a local copy of the files you will need on your computer workstation when going through these labs exercise. Once you download the master zip archive, open it and copy the **Student** folder to a new local folder. Note that each module of the course has its own folder in the **Student\Modules** folder.

© Critical Path Training. 2019. All Rights Reserved

Student Background Questionnaire

- What is your name?
- What are you doing with PowerApps and Flow?
- Which products and services have you used?
  - PowerApps and Flow
  - Microsoft Excel
  - Office 365
  - SharePoint Online
  - Power BI
  - Dynamics 365
  - Others

When you are called upon by the instructor, please provide a brief background by answering the following questions.

© Critical Path Training. 2019. All Rights Reserved

PowerApps and Microsoft Flow are to important pieces of the Microsoft initiative known as the Business Application Platform. Together with Power BI, PowerApps and Flow make up the “Power Trio” of the Business Application Platform

The Business Application Platform has been specifically designed for business users and technical specialists who do not have the background of a software developer. The ultimate goal of the Business Application Platform is to allow business users and technical specialists within an organization to build custom applications and workflow solutions in a fraction of the time compared to the traditional lifecycle of a software development project. The Business Application Platform has also been designed with a mobile-first philosophy which makes it relatively easy and very effective to build applications that target mobile devices such as iPhones and Android phones.

The Business Application Platform consist of these services

- PowerApps is a service for building and consuming web applications and mobile apps that connect to data
- Microsoft Flow is a service for automating workflow across the growing number of apps and SaaS services
- Power BI is a business analytics service which provides self-service BI and great interactive visualizations
- The Common Data Service for Apps provides the native storage of business data in the PowerApps environment
- Connectors and Gateways provide the Business Application Platform with access to external data

© Critical Path Training. 2019. All Rights Reserved

What Can You Build with PowerApps?

- Canvas Apps
  - Built using PowerApps Studio
- Connections
  - Used to connect Canvas apps to external data
- Flows
  - Used to process data and run workflows
- Common Data Service for Apps (CDS for Apps)
  - Used to create business-centric data solutions
- Model-driven Apps
  - Application platform built on top of CDS for Apps

© Critical Path Training. 2019. All Rights Reserved

In the lab exercises for this course, you will go through the steps to sign up for an Office 365 Enterprise E5 trial account. By doing this you will create a new Office 365 tenant which makes it possible to create multiple user accounts. A key point here is that you are creating a trial account for an entire Office 365 organization as opposed to creating a trial account for a single user.

When you initially create the new Office 365 tenant, you will be prompted to enter the user name and password for a new user account. This initial user account will be created with full tenant administrator capabilities. This means that this accounts with have full administrative control over user management and group management within the Office 365 tenant. This account will also be able to see and modify the organization-wide administrative settings for important cloud services such as Azure AD, SharePoint Online, Power BI and, of course, PowerApps and Microsoft Flow.

A significant benefit of creating a test environment in this fashion is that you can create additional users which makes it possible to test PowerApps and Flow scenarios such creating custom connectors and configuring an On-Premises Gateway. An Office 365 Enterprise E5 trial account allows you to add up to 25 user accounts for testing purposes. You will also be able to create new PowerApps environments so you can create a setup where you build apps and other components in a development/staging environment and then practice how to package and deploy your work to a production environment.

© Critical Path Training. 2019. All Rights Reserved

Once you have created your new Office 365 Enterprise E5 trial account, you should become familiar with the process of navigating around inside the Office 365 Admin center. For example, you need to learn how to add new user accounts and groups. In the lab exercises, you will add a secondary user account for testing purposes. For most of the class, you will log in and complete your work using your primary user account which will have tenant-level administrative permissions. The benefit of creating a second user account is that you can log in and test your apps and flows with a typical user account which does not have any administrative permissions.

The Office 365 Admin Center also allows you to purchase new subscriptions and to start trial subscriptions. This is something you will be required to do. More on that as we move to the next slide.

© Critical Path Training. 2019. All Rights Reserved

In the lab exercises for this module, there are instructions to configure your primary Office 365 user account with a PowerApps plan 2 license. The reason for this is that it provides you with the ability to view and create PowerApps environments and to work with the Common Data Service for Apps to complete design tasks such as creating a custom entity or a model-driven app.

Here are the high-level steps required to configure a PowerApps Plan 2 license.

- Navigate to the Purchase services ion of the Office 365 admin center
- Start a Trial Subscription of Microsoft PowerApps Plan 2
- Configure Your User Account with a Microsoft PowerApps Plan 2 license

Note that the upcoming lab exercises for this module will provide you with the step-by-step instructions needed to complete these steps.

© Critical Path Training. 2019. All Rights Reserved

The architecture for PowerApps and Flow is based on an important concept of “environments”. Any time you create a new app with PowerApps or a new flow, you are always doing so within the context of a specific environment.

You don’t have to think about environments when you start working with PowerApps. That’s because every new Office 365 tenant is automatically created with default environment. When you first begin to create apps and flows, you will be working within the default environment for the tenant that was created along with your new Office 365 trial account. During the first part of this training course, everything you do will be done in the default environment.

You can view and manage the environments for the current Office 365 tenant by navigating to the PowerApps Admin center at https://admin.powerapps.com. Note that you will not be able to view existing environments or create new environments until your Office 365 user account has been assigned a PowerApps Plan 2 license.

This training course provides a module on application lifecycle management (ALM) with PowerApps and Flow. At that point, the instructor will go into greater detail about creating and managing environments. As you will learn, an organization can benefit from creating multiple environments to assist with developing and staging the apps and flows they build. After building and testing apps and flows in a staging environment, an organization can design a strategy for packaging and deploying them into a production environment.

© Critical Path Training. 2019. All Rights Reserved

When you start using PowerApps, there are two important URLs that are easy to confuse with one another. The first is the URL to the PowerApps Home page which is located at https://web.powerapps.com. The second is the URL to PowerApps Studio which is https://create.powerapps.com. It's important you understand the difference between these two locations.

You should navigate to the PowerApps Home page at https://web.powerapps.com whenever you would like to create a new app or when you would like to play, edit or share an app you have already created. There is a navigation menu on the left which allows you to see and manage other PowerApps resources such as connections, custom connectors and custom entities in the Common Data Service for Apps.

Whenever you create a new app or you edit an existing app, you will be redirected to PowerApps Studio at https://create.powerapps.com. In other words, PowerApps Studio is the place where you do all your work designing, testing and debugging the apps you create with PowerApps. When you navigate to PowerApps Studio, you will notice that the browser resolves to a URL that has a locale inside it as in the case of https://us.create.powerapps.com.

© Critical Path Training. 2019. All Rights Reserved

If this is your first day using PowerApps, it will take a small bit of time to become accustomed to PowerApps Studio. In the first hands-on lab, you will get experience creating and building new apps. Once you create a new app, you must learn how to move from screen to screen and how to add new controls. You must also learn to set property values using the Properties pan and by writing formulas in the Formula bar.

PowerApps Studio provides a user interface using the following components

1. **Left navigation menu** displays app hierarchy of screens and controls.
2. **Screen designer** is the pane in the middle showing the current screen in design mode.
3. **Right-hand pane** displays Properties pane, Rules pane and Advanced pane.
4. **Property dropdown list** allows you to select property for Formula bar.
5. **Formula bar** where you write formulas in PowerApps Formula Language.
6. **Ribbon tabs** provide ribbon buttons to add and customize design elements.

PowerApps Studio is primarily available as a web-based application that is accessible through any modern browser. You should note that there is also a native version of PowerApps Studio that can be installed on the Windows operating system or on a Mac. These native versions of PowerApps Studio were more important in the early preview days of PowerApps because back then they provided functionality that was not available in the web-based version. At this point, Microsoft has improved the web-based version so it now includes all functionality that exists in the native application versions. Furthermore, Microsoft now recommends using the web-based version of PowerApps Studio over the native versions because that's where Microsoft's primary investments will be made moving forward.

© Critical Path Training. 2019. All Rights Reserved

While you are building an app, you will want to periodically run the app in order to test it and make sure the app it does what you want it to do. The way that you run an app is by clicking the 'Preview the app' button in the toolbar in the upper right corner of PowerApps Studio. Running the app makes it possible to experience what the consumers of the app will see when they launch the app.

After running and testing an app, you must learn how to quit the app and return back to design view in PowerApps Studio. You can accomplish this by clicking the button with the X icon in the top, right corner of the App Preview window. You should become comfortable running and quitting apps in PowerApps studio because you will be doing that on a regular basis as you build apps.

When you begin to work with variables and collections (covered in the next lecture), you will see that it is often required to run the app to initialize variable and collection values. In other words, you cannot initialize a variable or a collection while the app is in design mode. Instead, you must run the app and execute the actions to initialize the values for variables and collections.

© Critical Path Training. 2019. All Rights Reserved

When you create a new app from the PowerApps Home page, you are redirected from the URL of https://web.powerapps.com over to PowerApps Studio at https://create.powerapps.com. While you are building and testing an app, everything you do is simply stored in memory by the browser. If you close the browser without saving your work, then the app you are creating will be discarded. Therefore, you must save your app at periodic intervals to ensure you do not lose any work.

An app can be saved into storage in the cloud by using the Save command or the Save As command. However, before saving an app for the first time, you should navigate to the App settings page so you can enter an app title as well as configure the app color and the app icon. The App settings page also provides access to the settings for the app's screen size and orientation as well as advanced settings which makes it possible to enable new features that are still in preview.

After you save an app, you can later return to it to run the app or to open it up in edit mode to continue developing it. The user who creates an app is automatically assigned as the app owner. By default, the app is not available to other users. However, you can share an app you have created with other users so they can play it and use it as an app consumer. You can also share an app with edit permissions so another user can open the app in edit mode and continue the design work by adding new screens and controls.

© Critical Path Training. 2019. All Rights Reserved

Building Apps using Screens and Controls

- Screens provide the top-level objects in PowerApps UI
  - Your app must have one screen but can have multiple screens
  - You design screens by adding and configuring controls
  - Left tree view shows hierarchical view of screens and controls
  - You can rename screens and controls using left tree view menu

Screens represent the top-level objects in any user interface you create with PowerApps. Every app must contain at least one screen. However, an app can have multiple screens and it's common to add secondary screens when designing real-world apps. Note that only one screen can serve as the startup screen for an app. The screen which is located at the top of the left navigation menu is considered to be the startup screen. That means the top screen in the left navigation will be automatically displayed whenever the app is run. When you add secondary screens to an app you have created with PowerApps Studio, you must extend the app with extra behavior to display and navigate to these secondary screens.

You design screens by adding and configuring controls. Controls represent standard user interface elements such as buttons, labels, textboxes and combo boxes. When you want to add a new control, you must add it to a specific form. However, once you have added a control to a screen and configured its properties, you can copy that control to the clipboard and paste a copy of it into another screen within the same app.

The left navigation menu is important because it displays a hierarchical view of screens and controls that make up your app. Using the left navigation menu is often the best way to navigate to a particular screen or control. This is especially true once you have a screen with a large number of controls when it becomes hard to select the control you want in the PowerApps screen designer. Note that the left navigation menu also makes it possible to rename screens and controls to give them names that are more meaningful. While renaming screens and controls can be tedious, it is a good practice to get into because it makes apps easier to understand and to maintain.

© Critical Path Training. 2019. All Rights Reserved

PowerApps provide an extensive set of controls for building whatever user experience your app requires. To create a new control, you should first select the correct screen in the left navigation and then navigate to the Insert tab in the ribbon. Next, you can use one of the dropdown menu buttons on the Insert tab to add the type of control you want.

The Text menu provides controls that deal with text values such as Labels and Textboxes. The Controls menu provides controls such as Combo Box, Date picker and Slider that provide a more convenient user experience for entering data. You can also add controls that support data binding such as a gallery, a display form or an edit form. These controls will be explained in more details over the next few slides.

Take a look at the controls available in the Media menu. As you can see there are controls such as Camera, Microphone and Barcode that can be used to build apps that take advantage of the capabilities of a modern mobile device such as an iPhone or an Android phone. There is also an Icons menu that makes it quick and easy to add style to an app by adding familiar icons for tasks such as saving, filtering and searching through data.

© Critical Path Training. 2019. All Rights Reserved

Configuring Control Properties

- Control properties can be set two different ways
  - Property values can be set using Properties pane
  - Property values can be set using Formula bar

- Building apps with PowerApps requires shift in thinking
  - You don't write code to set property values like in VBA
  - Control properties configured using formulas
  - You develop using **declarative style** instead of **procedural style**

When it comes to configuring the property values for a control, you have two choices. First, you can configure property values using the Properties pane on the right. In the example shown on the slide above, you can see that the Properties pane displays an editable textbox for the Text property of a Label control. This textbox in the Properties pane makes it possible to view and update the Text property of the label.

As an alternative to using the Properties pane, you can edit the Text property of a control using the Formula bar which is located above the PowerApps screen designer. To edit a control property using the Formula bar, you must first select the control. Next, you must select the property you want to configure in the property selector dropdown menu to the left of the Formula bar. After that, you can configure the control property by writing a formula in the Formula bar. In the example shown on the slide above, the formula in the Formula bar sets the Text property of the Label control. While the formula used in this example is just a static string value, you can write formulas which are dynamic. The ability to write dynamic formulas for screen and control properties adds a very important aspect to the PowerApps development model.

Let's take a step back and think about how PowerApps allows you to set control properties using dynamic formulas. This leads to a development style that is quite different than times in the past when you may have written code in a programming language such as VBA, VB or C#. In these other languages, it is common to write procedural code that sets control properties. For example, you might write code behind the Reset button on a form to set the Text property of every textbox back to an empty string. This style of procedural coding is not the same style of development that you should be using when you begin working with PowerApps.

In PowerApps, you will not be writing any procedural code to set control properties. Instead, you configure control properties by writing formulas. Therefore, control properties take care of setting their own values. This leads to a development style in PowerApps that is 'declarative' instead of 'procedural'. However, in order to master this new declarative style of development, you must first learn about the syntax of the PowerApps Formula Language.

© Critical Path Training. 2019. All Rights Reserved

PowerApps defines its own Formula language for writing the formulas used to configure control properties. When Microsoft created the PowerApps Formula language, they tried to make it as similar as possible to the Excel Formula language. The reason for this is that are already 100 trillion (OK, maybe that's an exaggeration) business users that know how to write formulas in an Excel workbook. Many Excel users will feel at home as they begin to write formulas in PowerApps because it uses a syntax and functions that they are already familiar with.

The PowerApps Formula language includes a built-in set of functions. Here are a few simple examples. You can use the Text function to apply formatting to a currency value. You can use the User function to return an object with information about the current user. You can use the Filter function to filter the set of records displayed in a Gallery control.

In the next module, you will see that PowerApps Formula language provides additional syntax for compound data structures such as tables and records. You will also learn that some functions provided by the PowerApps Formula language return a value while other perform an action. If you want to take a look at Microsoft documentation for this language, you can navigate to the following URL:

https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/formula-reference

© Critical Path Training. 2019. All Rights Reserved

Input and Output Properties

- Input only
  - Value based on formula which cannot be used in other formulas
  - `Textinputbox1.Default` *(initial value)*
- Input/Output
  - Value based on formula which can be used in other formulas
  - `Textinputbox1.Fill`
- Output only
  - Value cannot be set by formula but can be used on other formulas
  - `Textinputbox1.Text` *(value always controlled by user input)*

© Critical Path Training. 2019. All Rights Reserved

Primitive Data Types

- Number: 3.141592
- Text: "Hello World"
- Boolean: True or False
- DateTime: 3/27/2018 12:00PM
- Date: 3/27/2018

- Any type can contain blank (i.e. null) values
  - Test for null value using IsBlank() function
  - Set null value using Blank() function

© Critical Path Training. 2019. All Rights Reserved

Compound Data Types

- Record
  ```
  { FirstName: "Chuck", LastName: "Sterling" }
  ```
- Table
  ```
  Table( { FirstName: "Chuck", LastName: "Sterling" },
         { FirstName: "Ted", LastName: "Pattison" } )
  ```
- Shorthand for Table with one column named value
  ```
  [ "Moe", "Curly", Larry" ]
  ```
- Records and tables can be nested
  - Table can contain records of tables of records…
  - Record can contain tables of records of tables …

© Critical Path Training. 2019. All Rights Reserved

Events and State Changes

- Formulas for event properties can contain imperative logic
  - `OnSelect`, `OnVisible`, `OnStart`, etc.
- Imperative logic is used to take action
  - Set value of global variable or context variable
  - Add item to a collection
  - Navigate between screens
  - Submit data to server
- You can chain actions together with chaining operator (;)

```
Collect(Customers, {ID: NextCustomerId, FirstName: FirstNameTextInput.Text, LastName: LastNameTextInput.Text});
Reset(FirstNameTextInput);
Reset(LastNameTextInput);
Set(NextCustomerId, NextCustomerId+1);
```

© Critical Path Training. 2019. All Rights Reserved

Declarative vs Imperative Functions

- Non-highlighted functions used to return values
  - These are declarative functions
- Highlighted functions used to perform actions
  - These are imperative functions

Abs	Collect	Day	HashTags	Max	Rand	Shuffle	TrimEnds
Acceleration	Color	Defaults	Hour	Mid	Refresh	Sin	Ungroup
Acos	ColorFade	Degrees	If	Min	Remove	Sort	Update
Acot	ColorValue	Disable	IfError	Minute	RemoveIf	SortByColumns	UpdateContext
AddColumns	Compass	Distinct	IsBlank	Mod	RenameColumns	Split	UpdateIf
And	Concat	Download	IsEmpty	Month	Replace	Sqrt	Upper
App	Concatenate	DropColumns	IsMatch	Navigate	Reset	StartsWith	User
Asin	Connection	EditForm	IsNumeric	NewForm	ResetForm	StdevP	Validate
Atan	Count	Enable	IsToday	Not	Revert	Substitute	Value
Atan2	Cos	EndsWith	Language	Notify	RGBA	SubmitForm	VarP
Average	Cot	Errors	Last	Now	Right	Sum	ViewForm
Back	CountA	EncodeUrl	LastN	Operators	Round	Switch	Weekday
Blank	CountIf	Exit	Launch	Or	RoundDown	Table	Year
Calendar	CountRows	Exp	Left	Param	RoundUp	Tan	
Char	DataSourceInfo	Filter	Len	Patch	SaveData	Text	
Choices	Date	Find	Ln	Pi	Search	Time	
Clear	DateAdd	First	LoadData	PlainText	Second	TimeValue	
ClearCollect	DateDiff	FirstN	Location	Power	Select	TimeZoneOffset	
Clock	DateTimeValue	ForAll	LookUp	Proper	Set	Today	
Coalesce	DateValue	GroupBy	Lower	Radians	ShowColumns	Trim	

© Critical Path Training. 2019. All Rights Reserved

App OnStart

- App **OnStart** property used to initialize state in app
  - Event provides support for initializing state in app
  - Commonly used to initialize app-level variables and collections
  - Right-click **App** in left navigation to run **OnStart** while in editor

© Critical Path Training. 2019. All Rights Reserved

When an app is run, PowerApps automatically displays the startup screen which is defined as the top screen in the Left Navigation menu. If you add secondary screens to an app, you must extend your app with the behavior to navigate to secondary screens. This can be accomplished by adding a Button control and configuring its OnSelect property with a formula that calls the Navigate function.

The Navigate function is an example of a function that performs an action instead of returning a value. When you call the Navigate function, the first parameter indicates the screen you want to display. The second parameters controls whether there will be a visual animation such as a fade when moving from one screen to another.

The Navigate function is complemented with a similar function named Back. The Back function provides the behavior of returning to the previous screen. For example, imagine you have an app that has a Customers screen that displays a list of customers and also a Customer screen that displays the data for a single customer at a time. On the Customers screen you can call the Navigate function to navigate to the Customer screen to display the data for a specific customer. The Customer screen could provide a Button control with an OnSelect property which calls the Back function to navigate back the Customers screen.

© Critical Path Training. 2019. All Rights Reserved

Understanding Connectors & Connections

- What is a Connector?
  - API wrapper that PowerApps uses to interact with datasource
- What is a Connection?
  - Configuration created to connect to a specific datasource
  - Each connection is created using a specific connector
  - Connection also caches login credentials and granted permissions
  - Connections can be shared across users

One of the strengths of PowerApps is that it makes it relatively easy to build an app that connects to data. Keep in mind that the data you need to access may be stored in many different types of datasources. You bring that data into your app by creating connections. Each connection is created by a specific user and may be shared across users. If you navigate to the PowerApps home page, you can see your current set of connections by navigating to Data > Connections node in the left navigation menu as shown in the screenshot in the slide above.

To become proficient at building apps with PowerApps, you must learn the key concepts involved with connectors. A connector is a wrapper around an API that PowerApps uses to interact with a specific type of datasource. Each connection you create is based on a specific connector which knows how to communicate with the datasource to which you are connecting. PowerApps provides out-of-the-box connectors for many popular services and on-premises datasources including SQL Server, SharePoint, OneDrive for Business, Azure AD, Dynamics 365, Salesforce and Twitter.

There are two important aspects of connectors and connections that relate to security. When using a connection for the first time, the user is often prompted to provide security credentials such as a user name and password to establish a secure connection to the datasource. After the user supplies security credentials the first time, the connection is able to cache those credentials and reuse them to reestablish connections to that datasource in the future without any need for user interaction.

The second security-related aspect of connections has to do with users delegating permissions to an app. A connection often allows an app to read and write to a datasource on behalf of the current user. PowerApps prompts users with an interactive dialog which allows a user to grant permissions to a connection which is required in order to read and write to a datasource on a user's behalf.

© Critical Path Training. 2019. All Rights Reserved

Data binding is one of the key features in PowerApps that dramatically simplifies the process of building an app to access data. When you create a new app using the 'Start from data' template, PowerApps Studio will use data binding when it generates the connection, the screens and the controls for the app. This means you can use data binding without really understanding how it works. However, learning how data binding really works in PowerApps will help you become a better app builder.

Data binding in PowerApps is based tabular data which is laid out in terms of tables and records. Some controls support data binding to a table of records while other controls support binding to a single record at a time. The Gallery control and the DataTable control are examples of controls that can be bound to a table of records. For example, you can create a Gallery control to display a set of expenses.

Forms are a special type of control that support data binding with a single record of data at a time. You can use a Display Form control when you want to display a single record of data in a read-only fashion. You can use an Edit Form control when you want to build a screen that allows a user to add a new record or to edit an existing record.

© Critical Path Training. 2019. All Rights Reserved

In order to customize the way data binding works, you must learn to use the Data Pane in PowerApps Studio. You use the Data Pane when you need to configure or customize the data binding for a data bound control such as the Gallery control. In order to use the Data pane, you must first select the data-bound control in the left navigation and then you can open the Data pane by navigating to the View tab in the ribbon and clicking the Data sources button. When a Gallery control is selected, the Data pane shows the details of the data binding including the connection, the data binding layout and the field mappings. It's easy to switch back and forth between different layouts and to update the field mappings to creating a better view of the underling data.

There is a common design pattern used in PowerApps where an app is created with three screens including (1) a browser screen, (2) a detail screen and (3) an edit screen. All three screens are used to data bind to the same underlying table. The browser screen contains a Gallery control with data binding to display many records at once. The detail screen uses a Display Form control to display the data for a single record in a read-only fashion. The edit screen contains an Edit Form control which allows users of the app to add new records and edit existing records.

© Critical Path Training. 2019. All Rights Reserved

In PowerApps, you use display forms and edit forms for scenarios where you need to bind to a single record at a time. Display forms and edit forms act as containers that hold another type of control known as a 'data card' which is also called a 'card control'. Each form contains a set of data cards that bind to individual fields from the underlying record. Each data card has a DataField property that determines which field it binds to within the underlying record.

A data card contains its own set of child controls. The child controls inside a data card make up the experience for displaying and editing a single field. For example, a number data card may consist of a Label control to provide the display name of the field and a Text input control to provide an editor for the value of the field. Most of the data cards supplied by PowerApps also provide a label control to display any validation errors that occur due to bad user input.

When you are writing a formula for a Form control property and a data card propertythe current record is available using an object named ThisItem. The ThisItem object for a Form control contains properties for each field in the underlying record.

When you are writing a formula for the properties of a child control inside of a data card, you can use the Parent object. For example, a child control inside a data card should use Parent.Default to read the initial state of the field from the data source. By using Parent instead of directly accessing the information that you want, the data card is better encapsulated. That means you can use the same data card with multiple fields in the same record without breaking internal formulas.

© Critical Path Training. 2019. All Rights Reserved

Changing a Field's Data Card Type

- Fields in Form control get default data card
  - Use Data pane to change data card used by any field
  - Different data cards offer different editing experiences

PowerApps provides several different types of data cards which can be used with various data types to create customized user interface experiences. When you use the 'Start from data' template to create a new app, PowerApps automatically generates a new view form and a new edit form for you with a default set of data cards.

Sometimes PowerApps will create a default set of data cards that isn't exactly what you want. However, you can swap out the default data card used for a field with a different type of data card that improves the user experience. For example, imagine you have an Expenses table with a Category field which only allows 4 predefined category values. You can change the default data card which uses a standard textbox to a different data card that supplies the user with a dropdown list of choices.

Working with data cards can be a little tricky at first because you must select the form in the left navigation and then display the Data pane. Once you've done that, you can then view and edit the type of data card used for each field. The slide above shows an example of changing from a data card with a textbox over to a better data card named 'Allowed Values' that provides a dropdown menu.

© Critical Path Training. 2019. All Rights Reserved

Customizing a Data Card

- By default, data cards are locked and cannot be edited
  - In many scenarios, you should leave data cards locked
  - Some scenarios call for unlocking data cards to customize them

The standard data cards provided by Microsoft are locked by default. When a data card is locked, you can modify a small number of properties of the data card and child controls inside. If a property is locked and can't be modified, it appears in the Property pane with a lock icon next to its name.

In many cases, leaving data cards in their locked state is helpful because it prevents you from making unintended changes that could disrupt the way that data binding is working. In other cases where the data card requires some customization, you'll be required to unlock a data card so you can edit all aspects of its properties and its child controls.

You can unlock a data card by navigating to the Advanced pane on the right with the data card selected and clicking the button with the lock icon with the caption 'Unlock to change properties'. Once the data card is unlocked, you are free to customize the child controls of a data card by resizing them, moving them or hiding them. You might also want to add new child controls to a data card to achieve a particular user interface experience. It's also possible to start with an entirely blank data card known as a 'custom card' which allows you to add all the child controls from scratch.

© Critical Path Training. 2019. All Rights Reserved

Populating a Dropdown Combobox

- Once a data card is unlocked you can customize it
  - Add formula for **Items** property to populate combo box

  - Dropdown list provides better user experience than textbox

Here is a common example of when it is necessary to unlock a data card. Let's say you can change the data card used for a text field to use the Allowed Values data card which provides a user experience with a dropdown menu of predefined choices. When you first update the field to use the Allowed Values data card, all the child controls are locked so you cannot configure the dropdown menu with a set of values.

Once you unlock an Allowed Values data card, you can then write a custom formula for the Items property of the child control which provides the dropdown menu. The screenshot on top in the slide above shows how to write a formula for the Items property to populate the set of choices provided by this dropdown menu. The screenshot on the bottom of the slide shows what the dropdown menu looks like to a user who is a consumer of the app.

© Critical Path Training. 2019. All Rights Reserved

Understanding Delegation

- **Delegation** is act of pushing work to data source
  - Work usually involves filtering and sorting
  - Work can also involve aggregation
  - Delegation minimizes amount of data retrieved by app
- Not all connectors support delegation
  - Non-delegate-able connectors only return 500 items
  - Searching through data becomes unpredictable

© Critical Path Training. 2019. All Rights Reserved

Delegate-able Functions

- **Filter functions**
  - *Filter*, *Search*, and *LookUp* can be delegated
- **Sorting functions**
  - *Sort* and *SortByColumns* can be delegated
- **Aggregate functions**
  - *Sum*, *Average*, *Min*, and *Max* can be delegated
  - Not all data sources support this delegation

https://powerapps.microsoft.com/en-us/tutorials/delegation-list/

© Critical Path Training. 2019. All Rights Reserved

Module 01 Lab: Getting Started with PowerApps Studio

Lab Time: 60 minutes

Lab Folder: C:\Student\Modules\01_GettingStarted\Lab

Lab Overview: This lab covers how to get up and running with the Power Platform by creating a new Microsoft 365 tenant with trial subscriptions to Office 365, PowerApps, Flow, Power BI and SharePoint Online. The act of creating and configuring this new Microsoft 365 tenant will yield an isolated testing and development environment for building and testing the apps and components you can build with PowerApps and Flow. One valuable aspect of creating your own new Microsoft 365 tenant is that you will have Global tenant administrative permissions allowing you to create multiple Microsoft 365 user accounts for testing your apps and flows in isolation from any other existing Microsoft 365 tenant.

Exercise 1: Create a new Microsoft 365 Trial Tenant

In this exercise, you will create a new Microsoft 365 tenant which allows you to create up to 25 user accounts with Office 365 Enterprise E5 trial licenses. Note that the Enterprise E5 trial license provides the benefits of the PowerApps, Flow, Power BI Pro and SharePoint licenses.

1. Navigate to the Office 365 trial sign up page using an Incognito browser window.
   a) Launch the Chrome browser.
   b) Using the dropdown menu in the upper right, select the command to open a **New incognito window**.

   c) Copy and paste the following URL into the address bar of the incognito window to navigate to the sign up page.

```
https://go.microsoft.com/fwlink/p/?LinkID=698279&culture=en-US&country=US
```

It's not always necessary to sign up for an Office 365 trial account using an incognito window. However, most errors that occur when attempting to sign up are caused by cached browser settings from another Office 365 account such as your work account. The solution to overcoming most errors when signing up for a trial account is using an incognito window.

2. Fill out the form with your personal information and click **Next**.

The information you provide on the next page of the signup process will be used to name your new Microsoft 365 tenant.

3. On the Create your user ID page…

© Critical Path Training. 2019. All Rights Reserved

a) Enter a user name

b) Enter a unique company name (you might have to try a few before you get one that's unique)

c) Enter a password that you will remember.

Note that the company name you enter on this page will be used to create the domain name for your new Office 365 trial tenant. For example, if you were to enter a company name of **bbspalabs**, it would result in the creation of a new Microsoft 365 tenant within a domain of **bbspalabs.onMicrosoft.com**. The user name you enter will be used to create the first user account which will be given administrative permissions within the Microsoft 365 tenant. If you enter a user name of **Student**, then the email address as well as user principal name for this account will be **Student@bbspalabs.onMicrosoft.com**.

d) Click **Create my account** to continue and move to the next step.

4. Complete the validation form to prove you are not a robot.

a) Select the **Text me** option and provide the number of your mobile phone.

b) When you go through this process, a Microsoft service will send you a text message that contains an access code.

c) You retrieve the access code form your mobile device and use it to complete the validation process.

5. Once you have completed the validation process, click the **You're ready to go…** link to navigate to the portal welcome page for your new Office 365 trial tenant. Note that you should already be logged on using the user account that was created during the signup process.

6. If you are prompted an Office 365 promotion dialog, click the **x** button in the top right corner to close it.

© Critical Path Training. 2019. All Rights Reserved

7. At this point, you should be located on an Office 365 welcome page. Notice the fancy new icons for all the Office applications!

At this point, you have already created your new Microsoft 365 tenant which can support creating up to 25 user accounts with Office 365 Enterprise E5 trial licenses. Note that some Microsoft 365 services within your new Microsoft 365 tenant such as the Microsoft 365 admin center, PowerApps, Flow and Power BI can be accessed immediately. Other services in your Microsoft 365 tenant such as Outlook, OneDrive for Business and SharePoint Online will not be ready immediately and can take some time to provision.

There is no more need to run the browser in incognito mode anymore because it's only required to get through the signup process. You can now return to using a standard browser window. However, it's always a good thing to check to see who you are logged in as because sometimes the browser may log you on using a different Office 365 account you have instead of your new trial account.

8. Click the **Admin** tile to go to the **Microsoft 365 admin center**

9. If you are presented with the **Welcome to the Microsoft 365 Admin Center** dialog, close it by clicking the **X** in the top right corner.

10. Verify that you are able to access the home page of the **Microsoft 365 admin center** at https://admin.microsoft.com/AdminPortal.
    a) The following screenshot shows the home page of the Microsoft 365 admin center.

© Critical Path Training. 2019. All Rights Reserved

b) Locate the top **Menu** button for the left navigation menu. It's the second button from the top with the arrow icon which sits just beneath the Microsoft 365 app launcher menu button.

c) Click the top **Menu** button several times and see how it toggles the left navigation between a collapsed and expanded mode.

Over the next few steps, you will configure your new Microsoft 365 tenant by creating a secondary user account that you will need later when you begin experimenting with sharing apps and flows with other users.

11. Make sure you are in the browser on the home page of the Microsoft 365 admin center.
12. Inspect the set of Active Users in the current tenancy.
    a) In the left navigation menu, expand the **Users** node and click **Active users** to navigate to the **Active users** page.

b) Once the **Active users** page is displayed, you should be able to verify that the user account you are currently logged on as is the only user account that exists in the current tenancy. Remember that this account has been set up as a global administrator within the Microsoft 365 tenant because it is the primary user account that was created when the tenant was initially created.

© Critical Path Training. 2019. All Rights Reserved

13. Create a new user account.
    a) On the **Active Users** page, click the **Add a user** button to create a new user account

    b) Fill in the **Create new user account** form with information for a new user account. When creating this account, you can use any name you would like. These lab instructions will demonstrate this by creating a user account for a person named **James Bond** with a user name and email of **JamesB@bbspalabs.onmicrosoft.com**.

    c) Expand **Password** section under **Contact Information** section.
        i. Select the option for Let me create the password.
        ii. Enter a password of **pass@word1** into the textboxes labeled **Password** and **Retype Password**.
        iii. Uncheck the checkbox for the option labeled Make this user change their password when they first sign in.

    d) Expand the **Roles** section. You do not need to change anything in this section, although you should note that this new user account will be created as a standard user account without any administrator access or privileges.

© Critical Path Training. 2019. All Rights Reserved

Note that the new account is usually assigned a trial license for **Office 365 Enterprise E5** plan. However, it's a good practice to check and make sure the new user has been assigned a license for **Office 365 Enterprise E5**.

e) In the **Product licenses** section, make sure the **Office 365 Enterprise E5** license is set to **On**..

f) Click the **Add** button at the bottom of the new user form to create the new user account.

g) When you see the **User was added** message, click **Send email and close** to dismiss the **Add new user** task pane.

h) Verify that the new user account has been created and is displayed along with your primary user account.

Exercise 2: Create a Trial Subscription for PowerApps Plan 2

In this exercise, you will configure your new Microsoft 365 tenant by creating a new subscription based on PowerApps Plan 2.

1. Navigate to the home page of the Microsoft 365 Admin center.
2. Create a new subscription for PowerApps Plan 2.

a) Click on **Billing** in the left navigation to expand the menu items underneath.

b) Click on the **Purchase services** navigation link.

© Critical Path Training. 2019. All Rights Reserved

c) Scroll down the page and find the subscription with the name **Microsoft PowerApps Plan 2**.

d) Hover over the ellipse (…) menu at the center on the bottom to see the flyout menu.

e) Click the **Start free trial** button

f) When prompted to confirm your order, click **Try now**.

g) You should see an order receipt to confirm you have created the new trial subscription.

© Critical Path Training. 2019. All Rights Reserved

3. Configure your user account by assigning a PowerApps Plan 2 license.
   a) Navigate back to the Active Users page in the Office 365 Admin center.
   b) Click on your user account to edit it.

   c) Click the **Edit** link for **Product licenses**.

   d) Enable the **Microsoft PowerApps Plan 2** subscription and then click Save below.

After creating a new subscription for PowerApps Plan 2, it might take a minute or two before it shows up in the Product licenses dialog.

   e) You should be able to confirm your user account has been configured with a **Microsoft PowerApps Plan 2** subscription.

You will not actually need the Microsoft PowerApps Plan 2 subscription until you reach the lab exercises which work with the Common Data Service for Apps. However, the PowerApps Plan 2 does provide the extra licensing beyond the Office 365 license such as the ability to use premium connectors and custom connectors.

© Critical Path Training. 2019. All Rights Reserved

Exercise 3: Create a Canvas App using the Start From Data Template

In this exercise you will create a new app using **Start from Data** template.

1. Upload the Excel workbook named **Expenses.xlsx** to OneDrive for Business.
   a) Using Windows Explorer, verify that there is an Excel workbook file named **Expenses.xlsx** located at the following path.

   ```
   C:\Student\Modules\01_GettingStarted\Lab\Expenses.xlsx
   ```

   b) Drop down the Microsoft 365 app launcher menu and select **OneDrive** to navigate to your **Files** collection.

   c) Click the **Upload** button and then select **Expenses.xlsx** to upload this file to OneDrive for Business.

   d) Verify that **Expenses.xlsx** has been uploaded to your **Files** folder.

Next, you will create a new Canvas app in PowerApps Studio that will read and write to the **Expenses** table in this Excel workbook.

2. Create the new app using the data in the **Expenses.xlsx** workbook.
   a) Navigate back to the **Home** page of the PowerApps portal at https://web.powerapps.com.
   b) Click on the **Start from data** tile to begin the process of creating the new app.

© Critical Path Training. 2019. All Rights Reserved

c) Next, click the **Make this app** button.

d) Next, click the **Phone layout** button inside the **OneDrive for Business** tile.

e) Click the **Create** button to create a new connection using the **OneDrive for Business** connector.

f) When prompted to **Choose an Excel file** on the **Connections** page, click the Excel workbook file named **Expenses.xlsx**.

© Critical Path Training. 2019. All Rights Reserved

g) When prompted to **Choose a table** on the **Connections** page, select the **Expenses** table and then click **Connect**.

h) Wait while PowerApps Studio generates the starting point for your app.

i) Once PowerApps Studio has created the new app, it should appear as the one in the following screenshot.

The new app has been created with three different screens. The browse screen shows many expenses at one time. The detail screen and the edit screen are both designed to display only one expense at a time.

j) Collapse the screen nodes in the left navigation for **BrowserScreen1**, **DetailScreen1** and **EditScreen1**.

k) Click on **DetailScreen1** in the left navigation to inspect the detail form.

© Critical Path Training. 2019. All Rights Reserved

l) Click on **EditScreen1** in the left navigation to inspect the edit form.

m) Click on **BrowseScreen1** and expand its node in the left navigation.

n) Click the **BrowserGallery1** control and then locate the Layout section in the Properties pane on the right.

o) You should see that the **BrowseGallery1** control currently has a **Layout** setting of **Title and subtitle**.

p) Update the Layout setting for BrowseGallery1 to a value of Title, subtitle and body.

q) Click on the **Edit** link for the **Fields** property to Display the Data pane.

© Critical Path Training. 2019. All Rights Reserved

You should now see the Data pane allow you to map fields to items in the gallery template for **BrowseGallery1**.

r) Set **Body1** to the **Amount** field.

s) Set **Subtitle1** to the **Category** field.

t) Set **Title1** to the **Expense** field.

u) Close the Data pane by clicking the x in the top right corner.

v) The browse screen should now display its fields ordered by **Expense**, **Category** and **Amount**.

3. Test the app by starting it up and testing the search functionality.

a) Click the Start button with the arrow icon to launch the app for testing.

b) The app should start and appear as shown in the following screenshot.

© Critical Path Training. 2019. All Rights Reserved

c) Test search functionality by typing the word “Cleaning” in the search box.

d) Try a different search by typing the word “Coffee” into the search box.

e) Once you have tested the search functionality, stop the app by clicking the button with the **X** icon at the top right.

4. Configure the formatting of the expense **Amount** field.

a) Select the textbox named **Body1** which displays the **Amount** field for each expense. You should be able to see that the **Text** value of this textbox currently configured with a formula which is **ThisItem.Amount**.

© Critical Path Training. 2019. All Rights Reserved

b) Update the **Text** property of the **Body1** textbox with the following formula.

```
Text(ThisItem.Amount, "$#,##0.00")
```

c) When you update the formula, it will initially match the following screenshot.

d) Note that after you update the formula, PowerApps Studio will automatically update the formula to include **[$-en-US]**.

e) The **Amount** field should now display its value with currency formatting.

5. Configure the **Color** property of **Body1** to display **Amount** values in red when they are $500 or greater.

a) With the **Body1** control selected, use the property drop down to display the **Color** property.

b) Update the **Color** property for **Body1** with the following formula.

```
If(ThisItem.Amount<500, Black, Red)
```

c) The formula bar should match the following screenshot.

d) You should now see that **Amount** values of $500 or greater are displayed with a red font.

© Critical Path Training. 2019. All Rights Reserved

6. Modify the edit screen to streamline data entry for new expenses.
   a) Using the left navigation, move to the edit screen.

   b) Display the **Data** pane so you can see the **Fields** collection of the edit form. At this point, the fields are sorted alphabetically.

   c) Using the mouse, rearrange the fields by moving **Expense** to the top followed by **Category**, **Date** and then **Amount**.

   d) The edit screen should now display its fields using the new sort order.

© Critical Path Training. 2019. All Rights Reserved

7. Update the data card for the **Category** field to provide a dropdown list with allowed values.
   a) Drop down the menu with the **abc** icon to the right of the **Category** field.

   b) Select a control type of **Allowed Values**.

   c) Close the **Fields** pane.

   d) The control which displays the **Category** field should change to a dropdown menu.

© Critical Path Training. 2019. All Rights Reserved

e) Select the dropdown menu and examine the **Items** property in the formula bar.

You will notice that the formula bar is read-only for the **Items** property because the data card is locked by default.

f) In the Advanced pane, click the Unlock to change properties button.

Note that the user interface experience might seem a bit strange when you click the **Unlock to change properties** button. At first it seems like nothing is happening. However, after a few seconds you should see that he **Items** property become editable.

g) Update the **Items** property of the dropdown list with the following formula.

```
["Marketing", "Office Supplies", "Operations", "Research & Development"]
```

h) The formula bar should match the following screenshot.

i) If you start the app, you should be able to test the dropdown list and verify that it provides four allowed values.

8. Update the data card for the **Date** field to make it a date-only.

a) Select the data card for the **Date** field.

© Critical Path Training. 2019. All Rights Reserved

b) In the **Advanced** pane, click the **Unlock to change properties** button for the data card for the **Date** field.

c) Using the left navigation, select and delete the controls named **MinuteValue1**, **Seperator2** and **HourValue1**.

d) After deleting **MinuteValue1** and **HourValue1**, you will notice formula errors due to referencing deleted controls.

e) Click on the red error icon with to the left.

f) At this point, you should see the formula for the **Update** property in the formula bar.

g) Replace the existing **Update** formula with the following formula to remove references to **HourValue1** and **MinuteValue1**.

```
DateValue1.SelectedDate
```

h) The formula for the **Update** property of the data card should now appear as the formula shown in the following screenshot.

i) Click on the one remaining red error icon to display the other formula error.

j) You should see the **Y** property of **ErrorMessage3** contains references to the deleted control named **HourValue1**.

k) Replace the existing formula for the **Y** property with a value of **0** as shown in the following screenshot.

l) At this point, you should no longer see any error indicators.

m) Select he **DateValue1** control and examine its **Width** property.

n) You should see that the formula of the **Width** property has the following value.

```
(Parent.Width - 60) / 2
```

o) Update the formula of the **Width** property to the following formula.

```
Parent.Width - 60
```

p) Your formula bar should match the following screenshot.

© Critical Path Training. 2019. All Rights Reserved

q) The **DateValue1** control should expand to the same width of the other input controls on the edit screen.

9. Update the **DataTimeZone** property of **DateValue1**.
   a) Inspect the **DateTimeZone** property of the **DateValue1** control. Its current value should be **Local**.

   b) Update the **DataTimeZone** property to a value of **DataTimeZone.UTC** as shown in the following screenshot.

   c) All the dates displayed on the edit screen should now move ahead by one day and display their proper value.

The problem with date values in the Local format is that they are offset by the difference between Greenwich Mean Time and your local time zone. For example, if you are in Eastern Daylight Time (EDT), the date of **January 10, 2018** is displayed with a 5-hour offset which is **January 9, 2018 at 7:00 PM**. By setting the **DataTimeZone** property to **UTC**, you are effectively removing the offset and the dates are displayed more accurately.

10. Configure the current day as the default value for **DateValue1**.
    a) Make sure the **DateValue1** control is selected.
    b) Inspect the **DefaultDate** property value for **DateValue1**.
    c) Update the **DefaultDate** property using the **Today()** function as shown in the following screenshot.

11. Test out the app by starting it and adding a new expense.
    a) Before starting the app, navigate to the screen named **BrowseScreen1**.

© Critical Path Training. 2019. All Rights Reserved

b) Click the Start button with the arrow icon to launch the app for testing.

c) When the browse screen appears, click to button with the **+** icon to add a new expense.

d) You should now see the edit form into which you can enter a new expense.

e) Fill in the edit form for the new expense using the data shown in the following screenshot and then click the button with the checkmark icon in the upper right to save your work.

f) Once you have saved the new expense, you should be able to see it in the browse screen.

© Critical Path Training. 2019. All Rights Reserved

12. Save the app to the cloud.
    a) Drop down the **File** menu and click the **App settings** link.
    b) Name the app **Expense Tracker** and assign a color, icon and description as shown in the following screenshot.

    c) Click the **Save** link in the left navigation and then click the **Save** button in the lower, right-hand side of the screen.

    d) You should be able to confirm that your app has been saved.

13. Examine the details of the new app.
    a) Return the PowerApps portal at https://web.powerapps.com and click the **Apps** link.
    b) Locate and the new app named **Expense Tracker**.

© Critical Path Training. 2019. All Rights Reserved

c) Click on the ellipse (**…**) dropdown menu to the right of **Expense Tracker** app and select **Details**.

d) On the app **Details** page, locate the **Web link** and click on it to launch the app.

e) The app should start up when you click that Web link.

f) The app should now start up in the usual run mode for end users.

© Critical Path Training. 2019. All Rights Reserved

Exercise 4: Test the Expense Tracker Application From a Mobile Device

In this exercise you will create begin by installing the native PowerApps app on your mobile device. If you have already installed the PowerApps app pn your mobile device, you can skip step 1 and move to step 2 of this exercise. If you do not have a mobile device that supports the PowerApps native app, you can skip the entire exercise and move on to Exercise 5.

1. Install the PowerApps native app on your mobile phone. This lab assumes you have a mobile phone which supports the PowerApps native app which include iPhones, Android phones and Windows 10 phones.
   a) Navigate to the App Store for your mobile device and search for PowerApps.
   b) Locate and installed the PowerApps app on your mobile device.

   c) Once the PowerApps app has been installed, open it.

2. You should now see the app welcome screen..

© Critical Path Training. 2019. All Rights Reserved

3. Sign into the PowerApps native app using the primary user account for the tenant you created at the start of this lab.
   a) Launch the PowerApps native app if it is not already running.
   b) Enter the user name (i.e. email address) of the primary user account for your new Microsoft 365 tenant.
   c) Click the purple button with the arrow to begin the sign in process.

   d) When prompted, enter your password and click **Sign in**.

   e) If it's the first time you have run this app, you will be prompted to allow the app to send you notifications. Click **Allow**.

   f) Once you have logged in, you should be able to see the Expense Track canvas app in the list of apps.

4. Open the Expense Tracker canvas app on your mobile device.

© Critical Path Training. 2019. All Rights Reserved

a) Touch the Expense Tracker app in the apps list to launch it.

b) You should see that app starting up.

c) When the app opens, you should be able to view existing expenses.

5. Use the mobile app to enter a new expense.

a) Touch the button with the **+** sign to add a new expense.

b) You should now see the screen for adding a new expense.

© Critical Path Training. 2019. All Rights Reserved

c) Add new expense data as shown in the following screenshot and touch the button with the checkmark to save your changes.

d) Once the expense has been saved, search for coffee and verify you can see the new expense.

The purpose of this exercise is to ensure you can test canvas apps on your mobile device. Once you have successfully added a new expense, you can close the app on your mobile device.and move ahead to the next exercise.

© Critical Path Training. 2019. All Rights Reserved

Exercise 5: Create a Canvas App using the Start from Blank Template

In this exercise you will create a new canvas app using **Canvas app from blank** template. This will give you experience creating the data binding for a canvas app without having to resort to using the **Start from data** template. In this lab you will also learn how to use the **Filter** function and the **Sort** function to manipulate data in a table. By the end, you will also work through the problems associated with a data source such as a table in an Excel workbook that does not support delegation.

1. Inspect the Excel workbook named **Cusrtomers.xlsx** located in the Student folder.
   a) Using Windows Explorer, verify that there is an Excel workbook file named **Expenses.xlsx** located at the following path.

```
C:\Student\Modules\01_GettingStarted\Lab\Customers.xlsx
```

   b) If you have Excel installed, open up this workbook and inspect what's inside.

CustomerId	FirstName	LastName	Company	EmailAddress	WorkPhone	HomePhone	Address
1	Nina	Diaz	Dunder Mifflin	Nina.Diaz@DunderMifflin.com	1(707)222-2222	1(707)666-4444	24 Route 41
2	Melinda	Carter	Vandelay Industries	Melinda.Carter@VandelayIndustries.com	1(707)111-6666	1(707)777-4444	405 Green Street
3	Pam	Miller	Brown Streak Railroad	Pam.Miller@BrownStreakRailroad.com	1(707)666-3333	1(707)222-0000	60 Mulberry Court
4	Merle	Blackwell	Krusty Burger	Merle.Blackwell@KrustyBurger.com	1(916)222-0000	1(916)333-2222	710 Ann Street
5	Ariel	Hale	ComTron	Ariel.Hale@ComTron.com	1(916)888-8888	1(916)444-3333	70 Walnut Street
6	Randy	Carter	ComTron	Randy.Carter@ComTron.com	1(916)666-1111	1(916)111-4444	9577 Mulberry Street
7	Lillie	Hinton	Blaidd Dwrg Nuclear Power Plant	Lillie.Hinton@BlaiddDwrgNuclearPowerPlant.com	1(707)111-1111	1(707)333-0000	43 Lexington Court
8	Ladonna	Moody	Porter Automobiles	Ladonna.Moody@PorterAutomobiles.com	1(707)888-2222	1(707)888-7777	17 Warren Avenue
9	Buddy	McKay	InGen	Buddy.McKay@InGen.com	1(541)222-2222	1(541)555-5555	1364 Broad Street West
10	Warren	Sykes	Cyberdyne Systems	Warren.Sykes@CyberdyneSystems.com	1(916)444-5555	1(916)444-1111	94 Street Road
11	Jan	Rutledge	Umbrella Corporation	Jan.Rutledge@UmbrellaCorporation.com	1(503)333-2222	1(503)777-5555	6648 Hillside Avenue
12	Dallas	Lester	Strickland Propane	Dallas.Lester@StricklandPropane.com	1(541)111-7777	1(541)333-4444	507 Rose Street

   c) Scroll down until you reach the rows with the **CustomerId** starting with 500.

	CustomerId	FirstName	LastName	Company	EmailAddress
501	500	Trent	Dixon	Michael Scott Paper Company	Trent.Dixon@
502	501	Saul	Snider	Biffco	Saul.Snider@E
503	502	Arron	Barrera	Groovy Smoothie	Arron.Barrera
504	503	Ezra	Bradley	Deon International	Ezra.Bradley@
505	504	Wilbur	Cote	Cyberdyne Systems	Wilbur.Cote@
506	505	Kelley	Gray	Total Bastard Airlines	Kelley.Gray@

A key learning point in this exercise is gain an understanding about the behavior of data sources that do not support delegation. When a connector does not support delegation, it will only return a maximum of 500 records by default. Therefore, you will not see any customer records with an **CustomerId** over 500. Later in this lab you will search for customers with the last name "Barrera". What you will see later in this exercise is that you will not be able to find this customer with the default canvas app settings which limits the tables it returns to a maximum of 500 records.

   d) Close the Excel workbook named **Customers.xlsx** without saving any changes and then close Microsoft Excel.

2. Upload the Excel workbook named **Customers.xlsx** to OneDrive for Business.
   a) Drop down the Microsoft 365 app launcher menu and select **OneDrive** to navigate to your **Files** collection.
   b) Click the **Upload** button and then select **Customers.xlsx** to upload this file to OneDrive for Business.
   c) Verify that **Customers.xlsx** has been uploaded to your **Files** folder.

Name	Modified	Modified By	File Size	Sharing
Customers.xlsx	A few seconds ago	Ted Pattison	148 KB	Private
Expenses.xlsx	4 hours ago	Ted Pattison	12.9 KB	Private

© Critical Path Training. 2019. All Rights Reserved

3. Create a new canvas app using the **Canvas app from blank** data template.
   a) Navigate to the PowerApps portal at https://web.powerapps.com.
   b) Create a new canvas app by clicking the **Canvas app from blank** button.

   c) When prompted with the Canvas app from blank dialog, enter an **App name** of **Customer Search**.
   d) Make sure to select a **Format** of the **Tablet** and then click **Create**.

   e) After a few seconds, you should see your new canvas app in PowerApps Studio.

© Critical Path Training. 2019. All Rights Reserved

4. Configure the App settings and save your new app.
   a) Click the **File** menu and then click **App settings**.
   b) Make sure the **App name** is set to **Customer Search**.
   c) Select an icon and color of your choosing as shown in the following screenshot.

   d) Click Save in the left navigation and then click the Save button in the bottom right corner of the PowerApps Studio window.

   e) Click the back arrow at the top left corner of PowerApps Studio to return to the canvas app in edit mode.

5. Update the name of Screen1 to Customer Search Screen.
   a) Drop down the ellipse menu (**…**) for **Screen1** and select he **Rename** command.

   b) Rename the screen to **Customer Search Screen**.

© Critical Path Training. 2019. All Rights Reserved

6. Add a new **Data table** control to display customer data.
   a) Navigate to the **Insert** tab in the ribbon.
   b) Click the **Data table** button to add a new Data table control to **Customer Search Screen**.

   c) Once the Data table control has been added, inspect its properties in the Properties pan on the right.
   d) Locate the **Items** property which has a dropdown menu with the caption **Select a data source**.

   e) Drop down the menu for the **Items** property and select **Add a data source** to display the **Data** pane.

   f) When the **Data** pane appears, select the connection named **OneDrive for Business**.

The **OneDrive for Business** connection should automatically appear because you created it when building the **Expense Tracker** app.

© Critical Path Training. 2019. All Rights Reserved

g) The **Data** pane should prompt you to **Choose an Excel file** and display the Excel workbooks at your OneDrive root library.

h) Select the Excel workbook named **Customers.xslx**.

i) When prompted to **Choose a table**, select the **Customers** table and then click **Connect**.

At this point, you have bound the **Customers** table to the Data table control but you have not selected any fields to display.

j) Inspect the **Items** property and the **Fields** property of the Data table control.

k) You should be able to verify that **Items** property has a value of **Customers** and the **Fields** property shows **0 selected**.

l) Click the link for the **Fields** property with the caption **0 selected** to display the **Fields** list in the **Data** pane.

© Critical Path Training. 2019. All Rights Reserved

m) Select the fields CustomerId, FirstName, LastName, Company, EmailAddress, WorkPhone and HomePhone.

n) You should now see those fields appear in the Data table control.

o) Using the mouse, resize the Data table to take up and entire height and width of the screen.

p) Move the top of the Data table down leaving a small amount of empty space at the top as shown in the following screenshot.

q) Rename the Data table control to **tblCustomers**.

r) Select the **CustomerId** column and then look at this selection in the property pane on the right.

© Critical Path Training. 2019. All Rights Reserved

s) Set the **Width** property of the **CustomerId** column to **50** and set the **Can grow** property to **Off**.

7. Add a search box

a) Add a search icon onto the **Customer Search Screen**. and rename it to **icoSearch**.

b) Position **icoSearch** in the upper, left as shown in the following screenshot.

c) Add a Text input control to the Customer Search Screen and rename it to txtSearchInput.

d) Set the **Default** property of **txtSearchInput** to **"A"**.

© Critical Path Training. 2019. All Rights Reserved

e) Select **icoSearch** and **txtSearchInput** together at the same time and group them using the **Group > Group** command.

f) You should have created a new group named **Group1**.

g) Rename Group1 to Search Box.

8. Add filtering and sorting behavior.

a) Inspect the **Items** property of **tblCustomers** which should currently have a value of **Customers**.

b) Modify the value for the **Items** property of **tblCustomers** using the following expression to add filtering behavior.

```
Filter( Customers, StartsWith(LastName, txtSearchInput.Text))
```

c) The items displayed by **tblCustomers** should now only include customers whose last name starts with "A".

d) Modify the value for the **Items** property of **tblCustomers** using the following expression to add sorting behavior.

© Critical Path Training. 2019. All Rights Reserved

```
Sort(Filter(Customers,StartsWith(LastName,txtSearchInput.Text)),LastName,Ascending)
```

e) Your formula bar should match the following screenshot.

f) Experiment with formula formatting by dropping down the formula bar vertically and clicking the **Format text** button.

g) You should see that PowerApps Studio is able to format expressions to make them more readable.

9. Test out the filter functionality

a) Start the app and try running a search using a search input string of "Bar". You should see the following results.

b) Search for "Barr" to find customers with the last name Barrera. Note that you cannot find any customers with this last name.

You are only searching through the first 500 records because of the way delegation works in PowerApps. You will never find any records that exist after the first 500. The key point is that you can get into trouble if you don't understand how delegation works.

10. Configure the **Customer Search** app to discover records beyond the first 500 record limitation.

© Critical Path Training. 2019. All Rights Reserved

a) Navigate to Backstage area in PowerApps Studio by clicking the **File** menu.

b) Click **App settings** in the left navigations and then click **Advanced settings**.

c) The **Data row limit for non-delegable queries** setting should have the default value of **500**.

d) Modify the value for the **Data row limit for non-delegable queries** setting to the maximum value of **2000**.

e) Click File to leave the Backstage area.

f) Start up the app and run a search for **Barrera**.

g) You should now find **Barrera** because you are looking through all the records in the underlying Excel table.

Id	FirstName	LastName	Company	EmailAddress	WorkPhone	HomePhone
502	Arron	Barrera	Groovy Smoothie	Arron.Barrera@Groovy...	1(408)555-2222	1(408)222-8888
655	Irene	Barrera	Trioptimum Corporation	Irene.Barrera@Triopti...	1(949)111-1111	1(949)111-0000
919	Amado	Barrera	Prescott Pharmaceutic...	Amado.Barrera@Presc...	1(323)444-2222	1(323)888-0000
938	Shaun	Barrera	The Hanso Foundation	Shaun.Barrera@TheHa...	1(360)555-8888	1(360)888-2222

h) Quit the app from running and then save your changes to the app.

11. Student challenge (if you have time)

© Critical Path Training. 2019. All Rights Reserved

a) Resize the columns in the Data table to make the data more readable.

Id	FirstName	LastName	Company	EmailAddress	WorkPhone	HomePhone
172	Mable	Barber	Yoyodyne Propulsion Systems	Mable.Barber@YoyodynePropulsionSystems.c...	1(206)111-5555	1(206)444-4444
101	Sondra	Barker	Ewing Oil	Sondra.Barker@EwingOil.com	1(916)555-6666	1(916)888-8888
299	Gustavo	Barker	Astromech	Gustavo.Barker@Astromech.com	1(425)777-4444	1(425)777-4444
446	Jerry	Barker	Dinoco	Jerry.Barker@Dinoco.com	1(541)555-4444	1(541)666-5555
10...	Daron	Barker	Zorg Industries	Daron.Barker@ZorgIndustries.com	1(707)444-3333	1(707)666-5555
216	Jermaine	Barlow	Groovy Smoothie	Jermaine.Barlow@GroovySmoothie.com	1(503)222-6666	1(503)555-5555
820	Chris	Barlow	Shelbyville Nuclear Power Pl...	Chris.Barlow@ShelbyvilleNuclearPowerPlant.c...	1(661)444-3333	1(661)666-7777
14...	Angeline	Barlow	Fabrikam	Angeline.Barlow@Fabrikam.com	1(971)666-8888	1(971)555-8888
267	Jessica	Barnes	Metacortex	Jessica.Barnes@Metacortex.com	1(559)666-8888	1(559)111-5555
574	Shelby	Barnes	The Hanso Foundation	Shelby.Barnes@TheHansoFoundation.com	1(415)555-8888	1(415)666-6666
725	Wes	Barnes	Roxxon	Wes.Barnes@Roxxon.com	1(949)777-4444	1(949)444-2222
682	Sang	Barr	Wallaby Airlines	Sang.Barr@WallabyAirlines.com	1(509)666-3333	1(509)444-1111

Congratulations. You have now completed this lab.

© Critical Path Training. 2019. All Rights Reserved