


# **Building Canvas Apps using Standard Guidelines and Best Practices**



# Our Newest PowerApps Training Course

- Advanced PowerApps and Flow training
  - Designed for students who have already attended Microsoft's App-in-a-day
  - Three days filled with plenty of hands-on lab exercises
  - Learn advanced builder skills & best practices
  - More info at <https://CriticalPathTraining.com>



[Training](#)[Courses](#)[Schedule](#)[Video Gallery](#)[Competition](#)[Contact Us](#)

[Home](#) > [Training Courses](#) > [Business Users](#)

## Building Business Solutions with PowerApps and Flow

[Course Overview \(PDF\)](#)

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-world 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](#)

### Upcoming Offerings

Date	Location	Instructor	Action
Apr-15	Tampa, FL	Ted Pattison	<a href="#">Register</a>
Jun-24	Tampa, FL	Ted Pattison	<a href="#">Register</a>

### Course Details

Course Code	BBSPA
Course Version	1.0
Course Length	3 Days

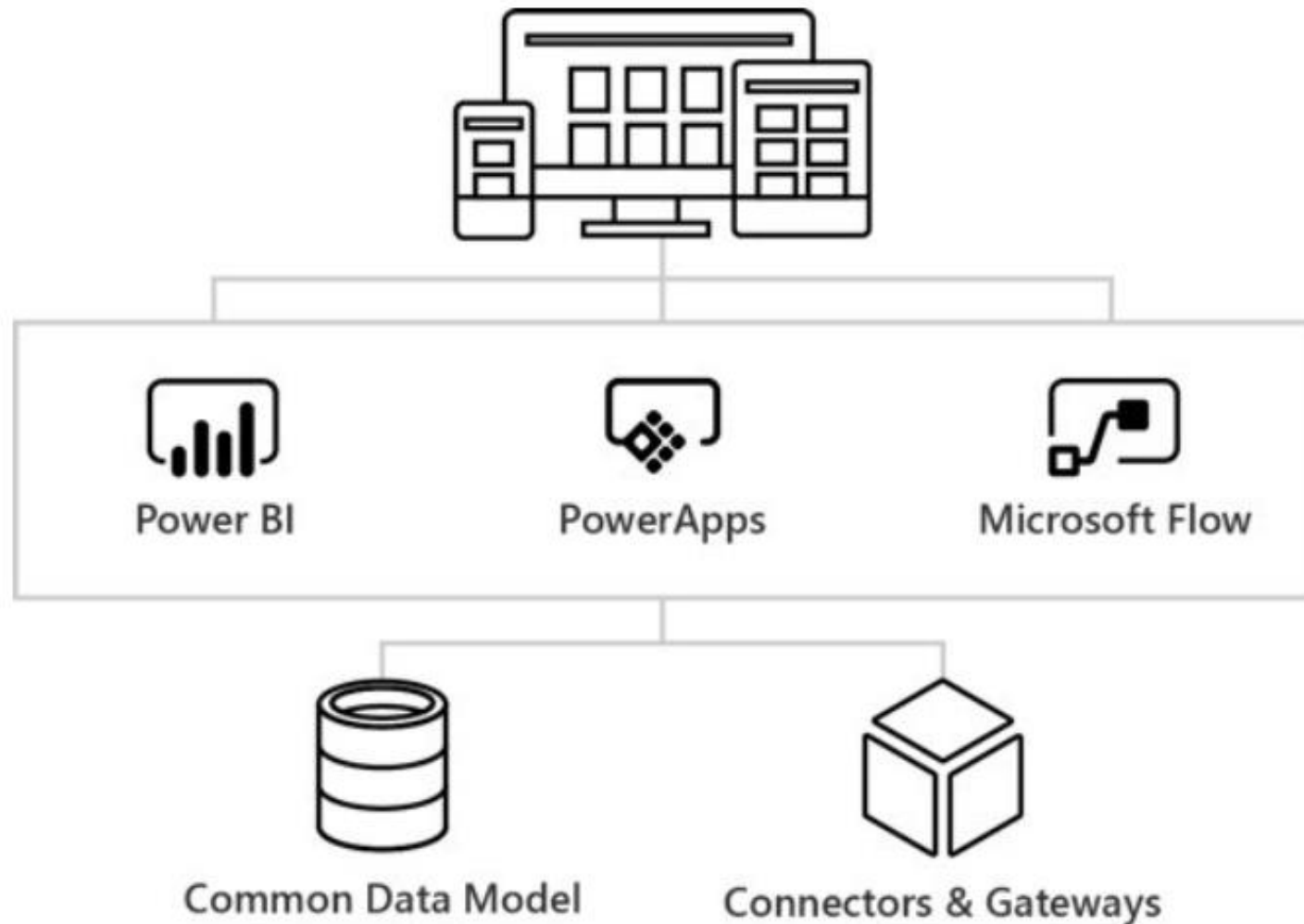


# Agenda

- PowerApps Challenges
- Naming Conventions
- Screen Design
- Delegation and Performance
- State Management
- Components



# What is the Power platform?



# 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)
  - Used to create business-centric data solutions
- Model-driven Apps
  - Application platform built on top of CDS for Apps



# Challenges with Building Canvas Apps

- Who is the typical PowerApps application maker?
  - A developer without a software development background
- What is the maker is responsible for?
  - designing, building, testing, deploying
- What are the challenges in deploying canvas apps?
  - Building projects that are easy to maintain and extend
  - Building consistency across team members
  - Building canvas apps with better performance





# Read the Canvas Apps Whitepaper

- PowerApps canvas app coding standards and guidelines

<https://powerapps.microsoft.com/en-us/blog/powerapps-canvas-app-coding-standards-and-guidelines/>



Todd Baginski



Pat Dunn



# Goals for the Canvas App Builder

- As a builder, you should strive for...
  - Simplicity
  - Readability
  - Supportability
  - Ease of deployment and administration
  - Performance
  - Accessibility





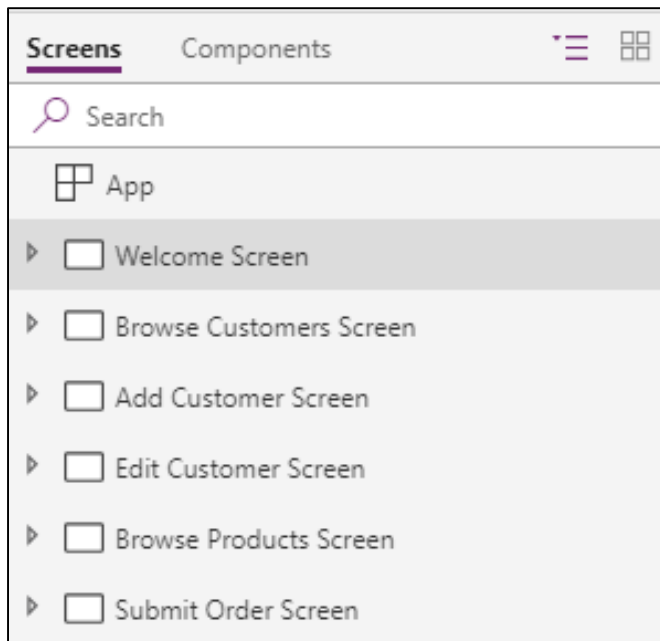
# Agenda

- ✓ PowerApps Challenges
- Naming Conventions
  - Screen Design
  - Delegation and Performance
  - State Management
  - Components



# Creating Screen Names

- It's import to create screen names correctly
  - Screen names read aloud by screen readers
  - Names should include spaces and avoid abbreviations
  - Screen name should end with the word "Screen"
  - Screen name should reflect purpose of screen



# Control Naming Convention

- Define and document a naming convention
  - Use this list as a starting point

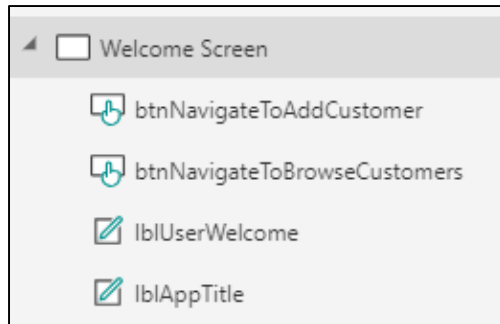
button	btn
camera control	cam
canvas	can
card	crd
collection	col
combo box	cmb
date picker	dte
drop down	drp
radio button	rad
form	frm
gallery	gal

group	grp
header page shape	fdr
html text	html
icon	ico
image	img
label	lbl
page section shape	sec
shapes (rectangle, etc.)	shp
table data	tbl
text input	txt
timer	tim



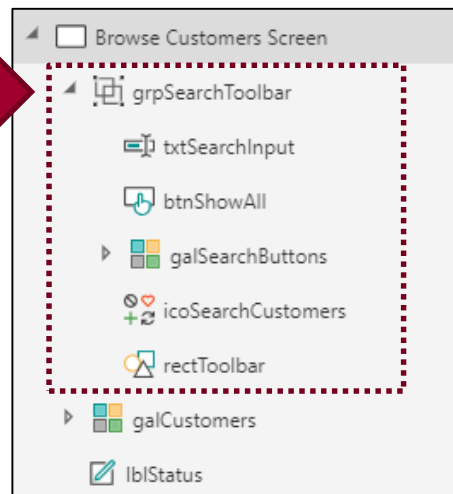
# Keeping Screens Maintainable

- Each control name should be easy to read & understand

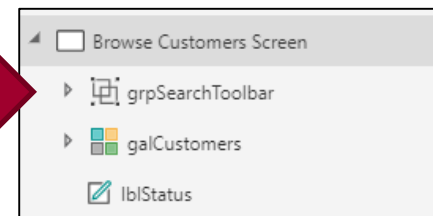


- Use grouping to isolate set of controls with single purpose

Expanded Group








Collapsed Group



# Code Naming Conventions

- Consistent naming conventions for variables & collections
  - Names should indicate type, purpose and scope
  - Collection names start with **col** such as **colProductCatalog**
  - Global variable names start with **gbl** such as **gblCustomApiUrl**
  - Context variables start with **loc** such as **locCustomerFilter**

## Collections

-  colLastOrderDetails
-  colNavigationTable
-  **colProductCatalog**
-  colShoppingCart
-  colLastOrder

## colProductCatalog

Here's a preview of the first 5 items in this collection  
[Learn about working with collections](#)

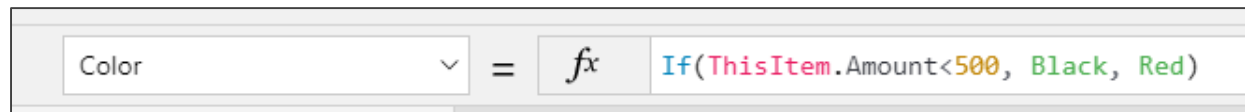
Category	Description	ListPr...	Product
Action Figures	A super hero who sometimes plays the role of a dark knight.	14.95	Batman Action Figure
Action Figures	A super action figure that protects freedom and the American way of life.	12.95	Captain America Action F

# PowerApps Formula Language

- PowerApps provides its own Formula Language
  - Designed to be as similar as possible to Excel Formula language
  - PowerApps Formula Language includes built-in set of functions
- You write formulas for specific properties
  - Set the Text property for a label



- Set the Color property of the label text



- Write an formula to filter the items shown in a gallery



# 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 (;)

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





# 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	Removelf	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	



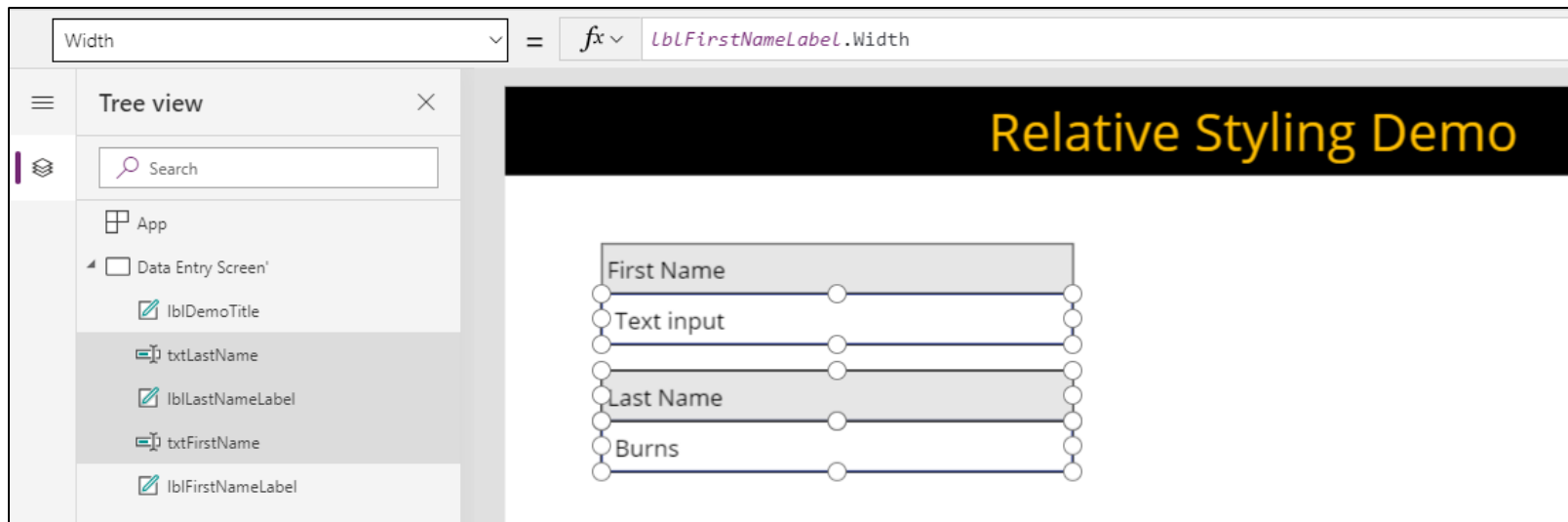
# Agenda

- ✓ PowerApps Challenges
- ✓ Naming Conventions
- Screen Design
  - Delegation and Performance
  - State Management
  - Components



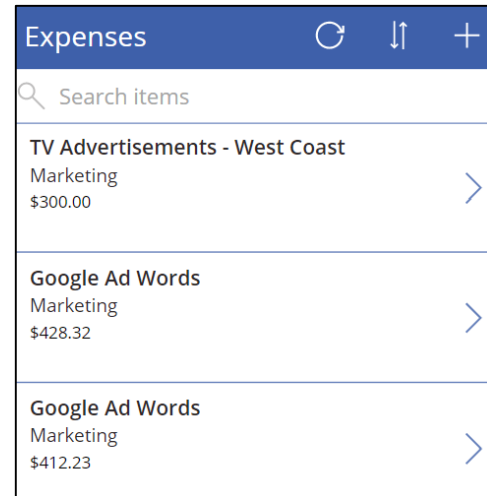
# Smart Designing Screen

- Use relative styling
  - Calculate control property values from other controls
  - Common to use properties like X, Y, Width, Height, Size, Fill, etc.

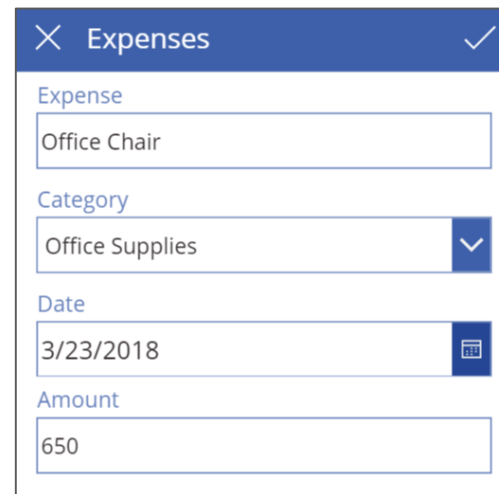


# Data Binding with Galleries and Forms

- Table binding
  - Gallery control
  - DataTable control
- Single-record binding
  - Display form control
  - Edit form control










Expenses	
Search items	
TV Advertisements - West Coast Marketing \$300.00	>
Google Ad Words Marketing \$428.32	>
Google Ad Words Marketing \$412.23	>


















Expenses	
Expense	
Office Chair	
Category	
Office Supplies	
Date	
3/23/2018	
Amount	
650	



# Use Galleries As Much as Possible

 <b>Lionel Witt</b>    email: Lionel.Witt@SoylentCorporation.com phone: 1(305)888-6666	 <b>Casandra Vega</b>    email: Casandra.Vega@TailSpinToys.com phone: 1(480)444-8888	 <b>Jude Melton</b>    email: Jude.Melton@JupiterMiningCorp.com phone: 1(707)222-5555
 <b>Suzette Fernandez</b>    email: Suzette.Fernandez@GraysonSkyDomes.com phone: 1(404)333-1111	 <b>Dee Santana</b>    email: Dee.Santana@TradeFederation.com phone: 1(541)555-6666	 <b>Corrine Ortega</b>    email: Corrine.Ortega@TyrellCorporation.com phone: 1(210)333-5555
 <b>Gaston Knight</b>    email: Gaston.Knight@LucernePublishing.com phone: 1(801)333-3333	 <b>Elvis Knight</b>    email: Elvis.Knight@DoublemeatPalace.com phone: 1(505)111-2222	

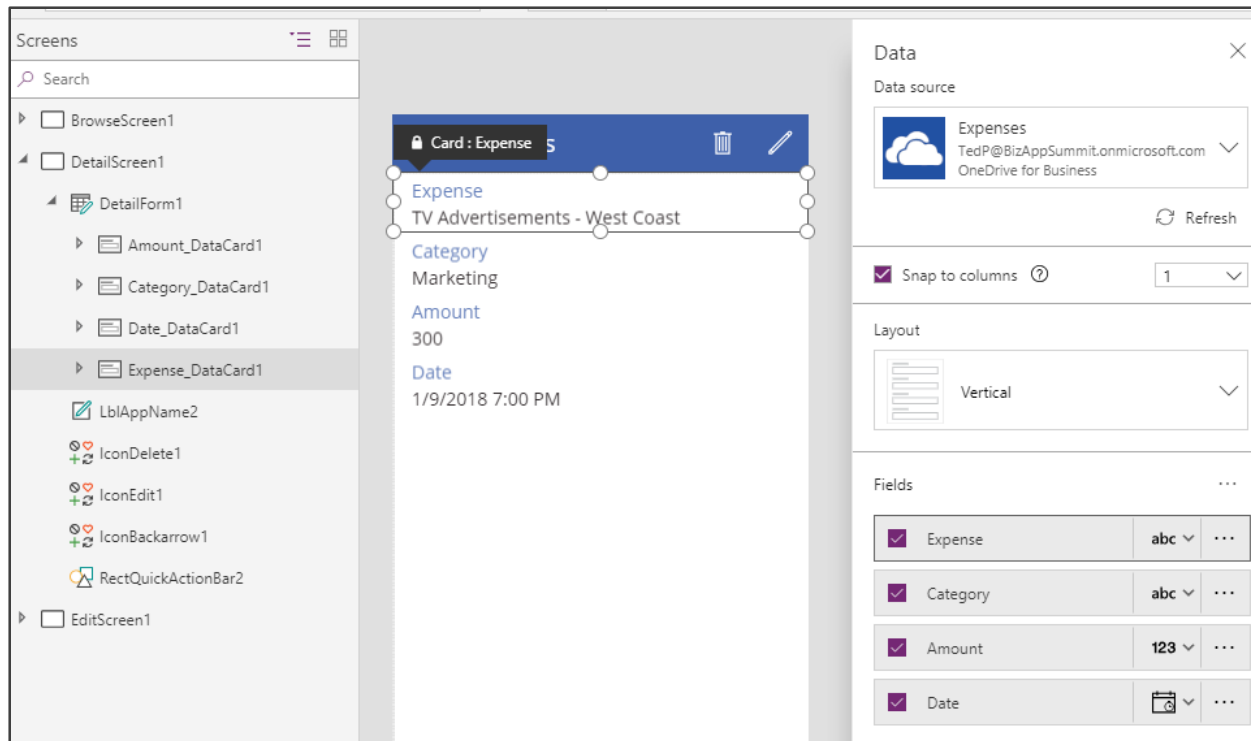
	<b>Batman Action Figure</b> A super hero who sometimes plays the role of a dark knight.	Quantity 1	
	<b>Captain America Action Figure</b> A super action figure that protects freedom and the American way of life.	Quantity 1	
	<b>GI Joe Action Figure</b> A classic action figure from the 1970s.	Quantity 1	
	<b>Green Hulk Action Figure</b> An overly muscular action figure that strips naked when angry.	Quantity 2	
	<b>Godzilla Action Figure</b> The classic and adorable action figure from those old Japanese movies.	Quantity 1	
	<b>Twitter Follower Action Figure</b> An inexpensive action figure you can never have too many of.	Quantity 10	

Shopping Cart for Lionel Witt			
No	Product	List Price	Total
1	Batman Action Figure	\$14.95	\$14.95 
2	Green Hulk Action Figure	\$9.95	\$19.90 
10	Twitter Follower Action Figure	\$1.00	\$10.00 



# Understanding Forms and Data Cards

- Form acts as a container for data cards
  - Each form binds to a single record
  - Within a form, each data card binds to an underlying field
  - Each data card contains an encapsulated set of child controls



# Using Patch Instead of an Edit Form

- Sometimes edit forms are not the best option
  - **Patch** function used to create and update records in data source

```
Patch(  
    '[dbo].[Orders]',  
    Defaults('[dbo].[Orders]'),  
    {  
        CustomerId: galCustomers.Selected.CustomerId,  
        OrderAmount: Sum(  
            colShoppingCart,  
            Total  
        ),  
        OrderDate: Today()  
    }  
)
```

- **Patch** function can be used with **ForAll** to insert multiple records

```
ForAll(  
    colShoppingCart,  
    Patch(  
        '[dbo].[OrderDetails]',  
        Defaults('[dbo].[OrderDetails]'),  
        {  
            OrderId: First(colLastOrder).OrderId,  
            ProductId: ProductId,  
            Quantity: Quantity,  
            Total: Total  
        }  
    )  
)
```





# Agenda

- ✓ PowerApps Challenges
- ✓ Naming Conventions
- ✓ Screen Design
- Delegation and Performance
  - State Management
  - Components



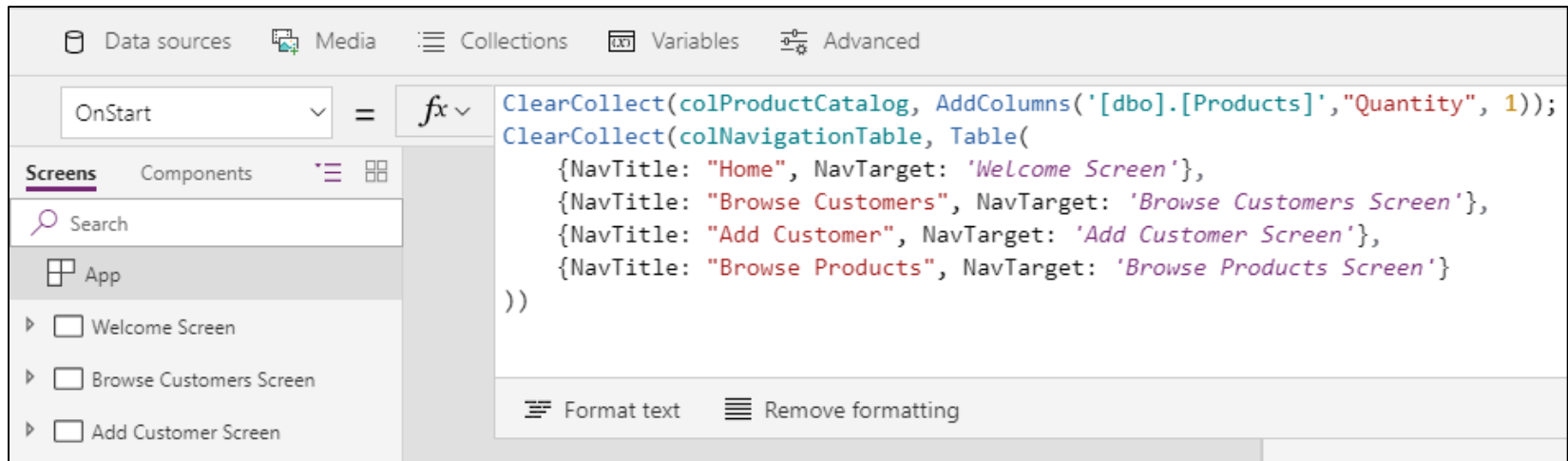
# Understanding Delegation

- **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



# App OnStart

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



# Famous PowerApps Blog Post

- Performance considerations with PowerApps
  - by Mehdi Slaoui Andaloussi of Microsoft
  - <https://powerapps.microsoft.com/en-us/blog/performance-considerations-with-powerapps/>

Blog > Common Patterns > General > Networking > Tutorials

## Performance considerations with PowerApps



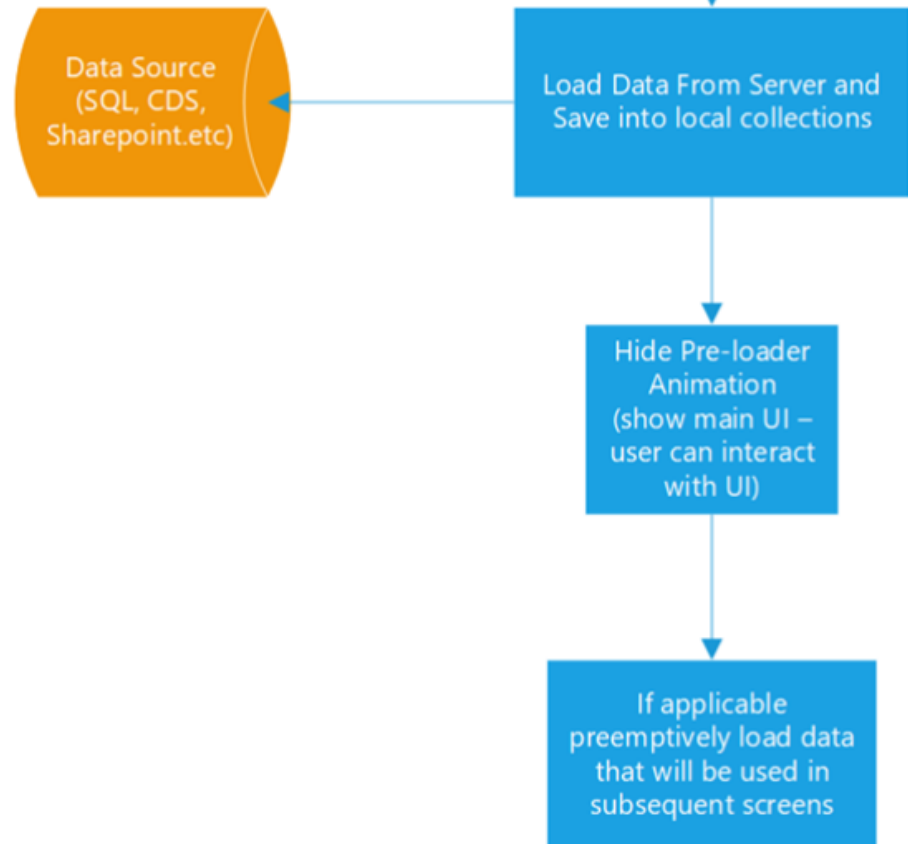
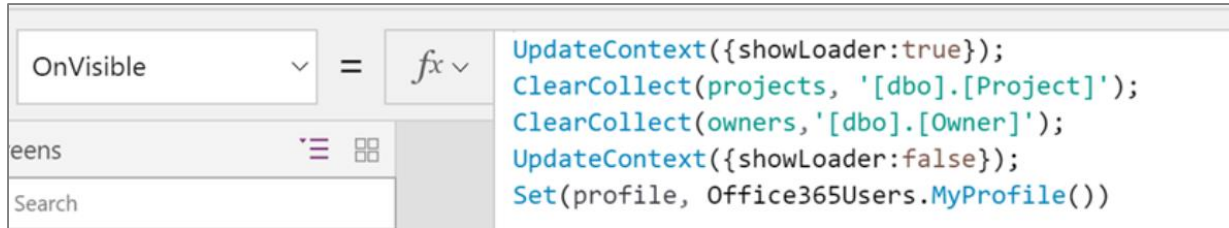
**Mehdi Slaoui Andaloussi**, Principle Program Manager, Wednesday, January 17, 2018



Performance is a top priority for the PowerApps engineering team. There are major efforts currently underway to improve performance around app load time, the designer experience in PowerApps Studio, connectors response time...etc. While the infrastructure will continue to improve, there are a number of things you can do as an app maker to improve the performance of your app. In this blog post I will touch on few concepts that are universal when talking about building native or web apps but apply equally well to apps built with PowerApps.

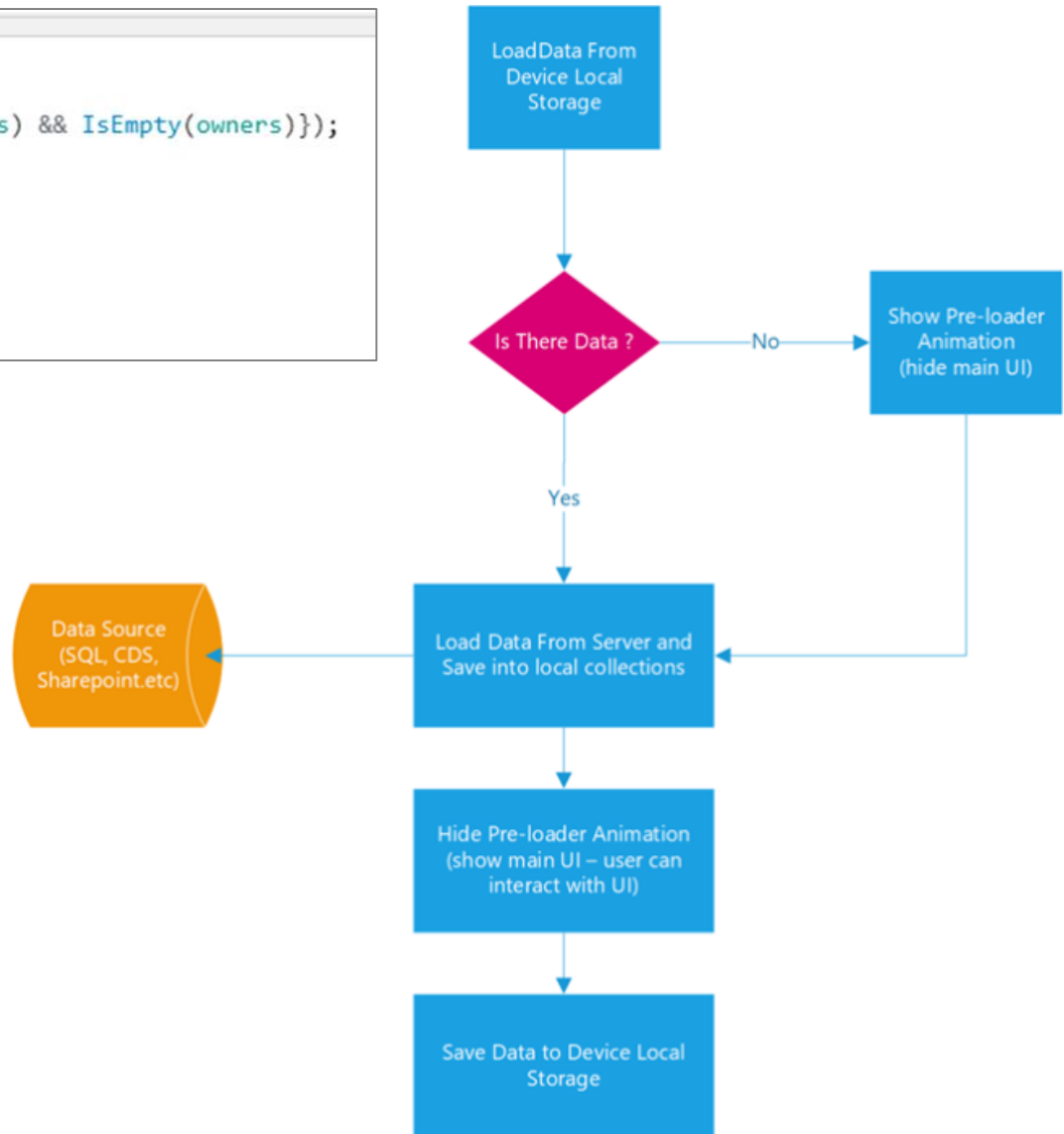


# Pattern 1: Load Data From Server



# Pattern 2: Load Data with Local Caching

```
LoadData(projects, "projects",true);
LoadData(owners, "owners",true);
UpdateContext({showLoader:true && IsEmpty(projects) && IsEmpty(owners)});
ClearCollect(projects, '[dbo].[Project]');
ClearCollect(owners, '[dbo].[Owner]');
UpdateContext({showLoader:false});
SaveData(projects, "projects");
SaveData(owners, "owners");
Set(profile, Office365Users.MyProfile())
```



# Pattern 3: Concurrent Calls

- Concurrent function used to execute queries in parallel
  - Leads to faster load times than sequential execution

```
Concurrent(  
    ClearCollect( Product, '[SalesLT].[Product]' ),  
    ClearCollect( Customer, '[SalesLT].[Customer]' ),  
    ClearCollect( SalesOrderDetail, '[SalesLT].[SalesOrderDetail]' ),  
    ClearCollect( SalesOrderHeader, '[SalesLT].[SalesOrderHeader]' )  
)
```





# Agenda

- ✓ PowerApps Challenges
- ✓ Naming Conventions
- ✓ Screen Design
- ✓ Delegation and Performance
- State Management
- Components



# State Variables



- Collections
  - Created as tables at app scope
  - Managed using `Collect`, `Clear` and `ClearCollect`
  - Can be stored to local device using `SaveData` & `LoadData`
- Context variables
  - Created as primitive, record or table at screen scope
  - Managed using `UpdateContext` and `Navigate`
- Global variables
  - Created as primitive, record or table at app scope
  - Created and managed using `Set` function



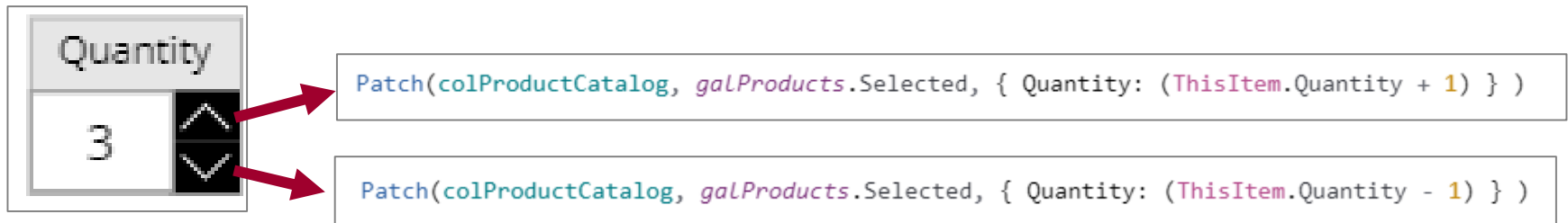
# Updatable Collection Columns

- Often helpful to add updatable column to collections

```
ClearCollect(colProductCatalog, AddColumns('[dbo].[Products]', "Quantity", 1));
```

	<b>Batman Action Figure</b> A super hero who sometimes plays the role of a dark knight.	Quantity 1	ADD TO CART
	<b>Captain America Action Figure</b> A super action figure that protects freedom and the American way of life.	Quantity 3	ADD TO CART

- Columns updated using calls to Patch



# Capturing Return Value from Patch

```
ClearCollect(
    collastOrder,
    Patch(
        '[dbo].[Orders]',
        Defaults('[dbo].[Orders]'),
        {
            CustomerId: galCustomers.Selected.CustomerId,
            OrderAmount: Sum(
                colShoppingCart,
                Total
            ),
            OrderDate: Today()
        }
    )
);
ClearCollect(
    collastOrderDetails,
    ForAll(
        colShoppingCart,
        Patch(
            '[dbo].[OrderDetails]',
            Defaults('[dbo].[OrderDetails]'),
            {
                OrderId: First(collastOrder).OrderId,
                ProductId: ProductId,
                Quantity: Quantity,
                Total: Total
            }
        )
    )
);
```

CustomerId	OrderAmount	OrderDate	OrderId
10	44.85	4/3/2019	14

**collastOrder**

Id	OrderId	ProductId	Quantity	Total
36	14	1	1	14.95
37	14	4	2	19.9
38	14	6	10	10

**collastOrderDetails**




# Populating UI from Cached State

CustomerId	OrderAmount	OrderDate	OrderId
10	44.85	4/3/2019	14

collastOrder

Id	OrderId	ProductId	Quantity	Total
36	14	1	1	14.95
37	14	4	2	19.9
38	14	6	10	10

collastOrderDetails



## Order Confirmation

[Home](#)[Browse Customers](#)[Add Customer](#)[Browse Products](#)

The Order Has Been Submitted Successfully

### Order Information




Order ID:

Order Date:

Order Amount:

Customer Shipping Address:

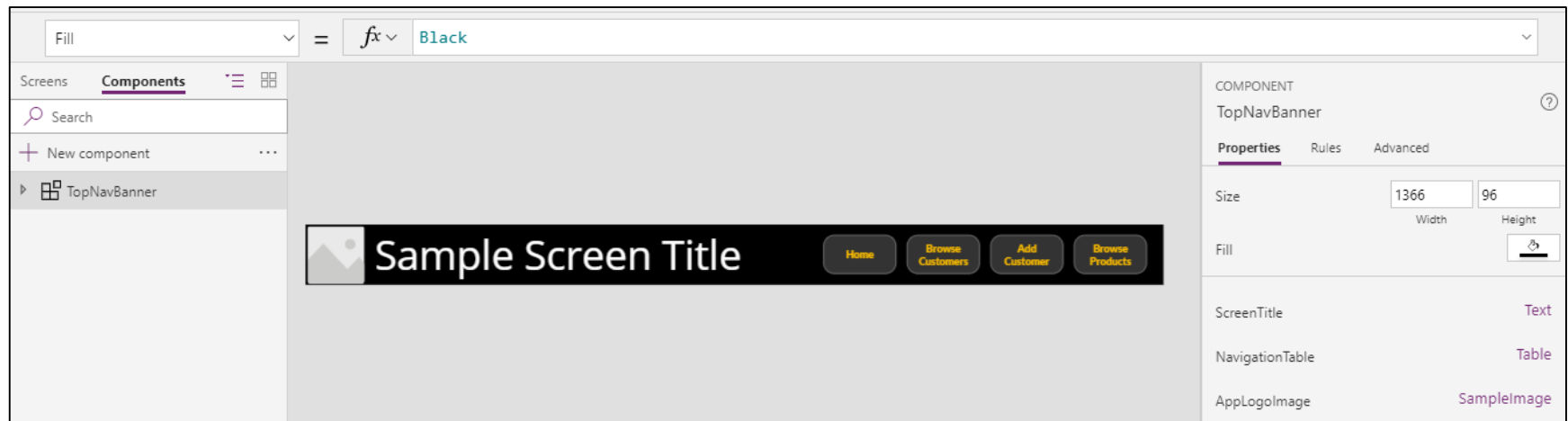
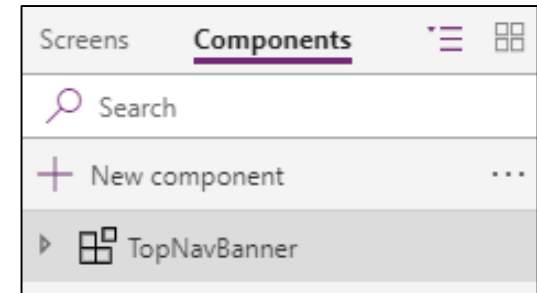
### Items purchased in Order #0014

No	Product	List Price	Total	
1	Batman Action Figure	\$14.95	\$14.95	
2	Green Hulk Action Figure	\$9.95	\$19.90	
10	Twitter Follower Action Figure	\$1.00	\$10.00	

Shopping Cart Total: \$44.85

# Designing Components

- Steps to using components
  - Create new component
  - Add component properties
  - Implement component UI and behavior
  - Add component to screens in your canvas apps



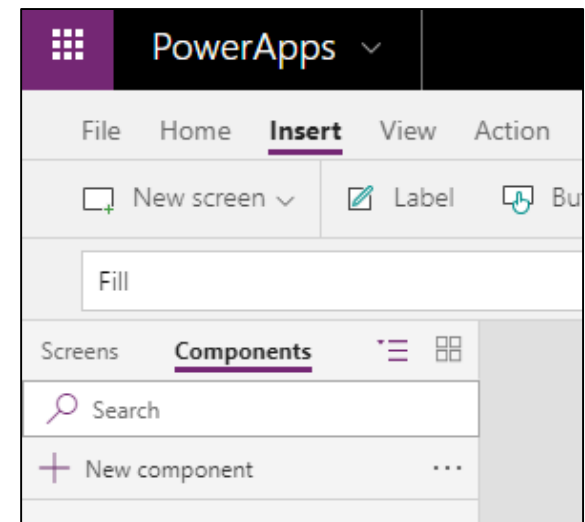
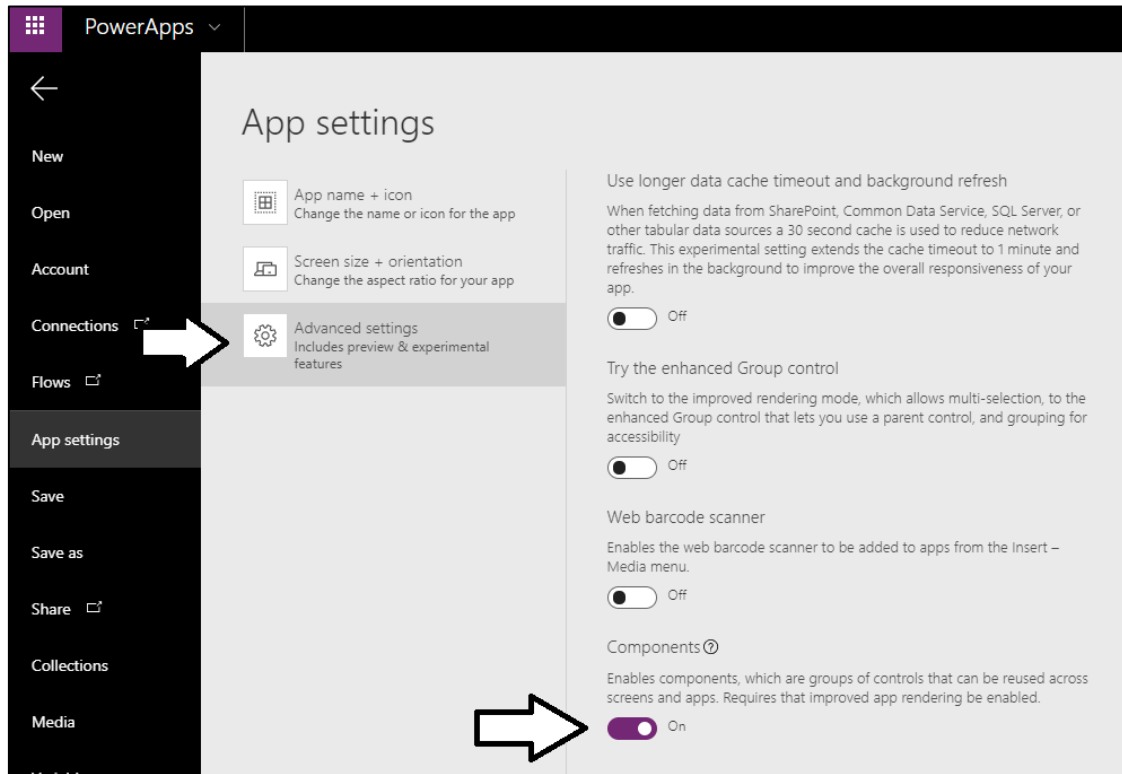
# Agenda

- ✓ PowerApps Challenges
- ✓ Naming Conventions
- ✓ Screen Design
- ✓ Delegation and Performance
- ✓ State Management
- Components






# Enabling Components in a Canvas App



# Our Newest PowerApps Training Course

- Advanced PowerApps and Flow training
  - Designed for students who have already attended Microsoft's App-in-a-day
  - Three days filled with plenty of hands-on lab exercises
  - Learn advanced builder skills & best practices
  - More info at <https://CriticalPathTraining.com>



[Training](#)[Courses](#)[Schedule](#)[Video Gallery](#)[Competition](#)[Contact Us](#)

[Home](#) > [Training Courses](#) > [Business Users](#)

## Building Business Solutions with PowerApps and Flow

[Course Overview \(PDF\)](#)

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-world 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](#)

### Upcoming Offerings

Date	Location	Instructor	Action
Apr-15	Tampa, FL	Ted Pattison	<a href="#">Register</a>
Jun-24	Tampa, FL	Ted Pattison	<a href="#">Register</a>

### Course Details

Course Code	BBSPA
Course Version	1.0
Course Length	3 Days



# Summary

- ✓ PowerApps Challenges
- ✓ Naming Conventions
- ✓ Screen Design
- ✓ Delegation and Performance
- ✓ State Management
- ✓ Components

