



# App Designer Overview

## Table of Contents

Introduction .....	1
Designer overview .....	1
Getting started .....	1
Concepts used throughout the designer .....	1
Action buttons .....	1
Save/undo buttons .....	1
Tick-selecting.....	2
The sample SOP app .....	2
Selecting an app.....	2
Creating an app.....	2
Other app/version level actions.....	2
Data modelling.....	3
Data entities.....	3
Properties.....	3
Classification types.....	3
Defining menus .....	4
Defining pages series .....	4
Page navigation.....	5
Data sources.....	5
Pages .....	6
Designing pages .....	6
Layout designer.....	6
Page segment designer .....	6
General page settings .....	7

# Introduction

This document contains a high-level overview of the functionality provided by Evoke's App Designer.

## Designer overview

The purpose of Evoke's App Designer is to allow you to create platform independent definitions of various aspects of your app's UI. These definitions can then be used by the App Designer's sister component, the App Generator, to drive the creation/maintenance of multiple Visual Studio projects that implement the designs using a range of technologies.

The app designer runs inside the desktop web browser. Currently, you will need to use Internet Explorer version 10 or higher. In the very near future you will also be able to run the designer in other desktop browsers, such as Chrome, Safari and Firefox.

The website hosting the app designer is located within Microsoft's Azure cloud infrastructure. All of the app designs that you create are securely stored within Azure.

## Getting started

The first step in getting started with Evoke is to submit a request for an Evoke developer subscription. Details of how to do this and associated costs can be found on the Evoke website [www.bluefinity.com/evoke](http://www.bluefinity.com/evoke)

When your subscription is activated, you will be able to login to the app designer by browsing to <https://www.appevoke.com/evokedesigner>

The login page allows you to enter the details that you will have specified in your developer subscription request. After entering these details clicking the Login button will display the Account page of the designer. From here you can begin your app creation journey using Evoke!

## Concepts used throughout the designer

There are a number of common concepts used throughout the designer. In fact, because the app designer has been created using the app designer itself, these common concepts also hold true for the apps that you create using the Evoke.

### Action buttons

Throughout the designer you will see the following icon . The generic term for this icon is an "action button". This icon indicates that there are some actions associated with the item or region next to or containing the icon. If you click it a pop-up menu listing the available actions will be displayed.

### Save/undo buttons

Whenever you make a change to something within the designer, two icons will be displayed in the top left and right hand corners of the screen: the save button  and the undo button . Clicking the save button indicates that you would like to commit all of your unsaved changes; clicking the undo button will cancel all unsaved changes. Related to these buttons, when you start making changes to something within the designer, a red pencil icon  is displayed in the top-right corner

of the associated menu option at the top of the designer screen. This allows you to see which menu options contain unsaved changes even if you navigate away from them.

## Tick-selecting

On many of the menus associated with action buttons that relate to a list of items there are options which allow you to operate a display of checkboxes alongside each item in the list. This feature is called *tick-selecting* and allows one or more items to be selected ready for a subsequent action to be performed on all ticked items.

## The sample SOP app

As part of the initial setup that BlueFinity performs when you first activate your Evoke developer subscription, a sample app will be created within your designer account – the SOP (sales order processing) app. This app demonstrates a wide range of Evoke features and there is a document that you can download that is dedicated to walking you through the constituent parts of this app. This is a great learning resource and we strongly recommend taking some time to look at how this app is built in order to kick-start your Evoke learning process.

## Selecting an app

The initial display of the Account page within the designer presents a selection widget that allows you to select an app already created within your designer account. Once you select an app, a separate area is displayed to the right that allows you to select a particular version of this app. The app versioning feature of Evoke allows you to have multiple independent versions of your app in play at any one time. This is useful if you, say, need to partition the current production version of an app from the version that is being actively developed ready for the next release.

Once you have selected an app version, the design details of the app are loaded into the browser and additional menu options at the top of the designer window will be displayed.

## Creating an app

When the time comes to create a brand new app, you can click the action button at the right edge of the app selection widget and click the “New” option. You’ll be asked to confirm the action and then the app, along with its first version will be created. You can adjust the name and other details of the app and version as necessary. When you have done this you can save your changes by clicking the save button in the top right corner of the designer window. Alternatively, you can click the undo changes button in the top left corner to cancel your unsaved amendments.

## Other app/version level actions

The action button at the right edge of both the app and version selection widgets can be used to perform a range of actions against the app as a whole or against the currently selected version.

**App Delete** deletes the currently selected app along with all of its versions. Obviously to be used with caution because once it’s gone – it’s gone!

**Version Refresh** Reloads all of the design data for an app. This can be useful if you want to load up amendments that another developer has just made.

- Version New** Creates a new empty app version.
- Version Duplicate** This option takes a snapshot of the currently selected app and uses it as the starting point for a new version.
- Version Delete** deletes the currently selected version and all of its content. Again, use with caution.

## Data modelling

The **Data** menu option at the top of the designer window allows you to create an entity data model specifically designed to supply data to your app's UI. Thus, when you first sit down and start thinking about what the structure of your data model should be like, you should be guided by the needs of the UI, not by the structure of data within your database server.

Your UI data model will be made up of 2 main parts, a list of data entities and a list of classification types.

### Data entities

A data entity represents a "thing" within your app. The entity will have a series of data fields (known as "properties") associated with it and typically the values of some or all of these properties will be displayed within the UI of the app.

One way of viewing the concept of a data entity is as the transport mechanism between the UI and the back-end data repository.

The designer allows you to create any number of data entity types. When you come to do your UI design, the app designer allows you to indicate where data from any of the entities is to be mapped or "bound" onto the UI.

The action button within the Entities tab-ear can be used to create and delete entities.

### Properties

Each entity type can contain one or more properties. These represent the elements of data that an entity contains.

Each property has a number of pieces of definition information, one of the main ones being the data type. One of the data types available is "Entity" – this allows you to construct connections or "relationships" between the different types of entities within your app.

The action button within the Properties panel heading can be used to create and delete properties within the selected entity.

### Classification types

Within the Data page of the designer, as well as creating entity types you can also create classification types. A classification type is a general purpose mechanism provided by Evoke to cater for the very common requirement within business apps to hold reasonably static lists of status codes, categories and other classification type information.

Classification types are very similar to entities, except that every classification type has exactly the same list of properties. In fact, all classification instances are an instance of the Classification entity

type and you can view the properties of this entity type within the app designer (though you can't modify it) just like any other entity type.

The sample SOP app contains some classification types so take a look at it to get a better idea of the kind of thing you might use them for.

The action button within the Classifications tab-bar can be used to create and delete classification types.

## Defining menus

The [Menu](#) option within the designer's top level menu allows you to define the structure of the menu of your app.

Defining menus is a pretty straight forward process of creating a series of menu nodes, each one having a display caption along with a few other pieces of definition information. If your app contains a lot of different aspects of functionality, you may need to create a menu structure with nested levels in order to break down the overall navigation experience for the end user.

If a menu node does not contain any child menu nodes, you may associate a page series with the node in order to launch page UI when the user selects the menu option. More on page series later.

The menu page allows you to define, if necessary, a separate menu structure for each generic type of run-time platform – these are called the design-time platform types. The list of design-time platform types is listed on the left side of the menu page: Desktop, Tablet, Phone and Modern UI. When you come to use the App generator, it will select the appropriate design-time menu structure for the various types of run-time platforms that you can request code generation to handle.

In the centre left of the Menu page there are a list of action panels that you can use to manipulate your currently selected menu structure.

Your menu structure is displayed in the centre right of the page and as you select nodes from within your menu structure, the definition of the selected node is shown on the right of the menu structure panel.

## Defining pages series

The [Pages](#) option within the designer's top level menu allows you to define the bulk of the UI for your app.

The basic concept within Evoke is that you can create many "page series". Each page series will consist of one or more individual pages (or screens) of UI content. Most page series are accessed via the menu system, although a few special ones are accessed via other methods.

As in the Menu page, the Page definition page of Evoke allows you to create, if necessary, a separate page series structure for each different design-time platform, although much of the time you may be able to use the same page series definition across multiple platform types.

Your list of page series is shown on the left of the screen. You can use the action button in the top panel heading to create and delete page series.

When you select a page series from the left-hand list, its general details are displayed in the top-right of the page. In the lower right portion of the page you are able to create the individual pages of UI within the series for each of the design-time platforms – as necessary.

## Page navigation

Evoke supports a number of different ways for you to manage page navigation within a page series:

- Child navigation** Child page navigation is where the user navigates to a page which contains more detail relating to something on the current page (sometimes known as drill-down navigation). When child page navigation occurs, Evoke automatically displays a *back to parent page* button in the top left corner. The styling of this button will be altered to fit the visual culture of the run-time platform.
- Parent navigation** Parent navigation is where the user clicks the back button displayed when child navigation has occurred. If the parent page has not itself been displayed as a result of child navigation, the back button will be hidden.
- Sibling navigation** Sibling page navigation is where the user navigates to a page which contains some alternative information relating to that displayed on the current page. Thus, the new page can be logically view as a sibling rather than a child. The display (of non-display) of the back button is not affected by sibling navigation. The user initiates sibling navigation via a drop down list associated with a button in the top right of the screen . The presence of this button is controlled by the sibling navigation settings of the current page (see below).

## Data sources

The Data sources tab allows you to define which entity types are associated with each of your page series. This is done by defining one or more data sources. Each data source has a number of characteristics:

- Name** The logical name (or ID) of the data source.
- Entity type** The type of entity associated with the data source.
- Collection** Indicates whether the data source relates to a single entity instance or a collection (list) of instances.
- Auto saved** Indicates whether Evoke is to automatically watch out for updates within the data source at run-time within your app and automatically display the save/undo buttons at the top of the screen if changes are detected.
- Load style** Indicates at what point the data source is to be populated with data at run-time.
- Global** Indicates whether a data source is to be made available to other page series as soon as its data is loaded

Later, you will reference your data sources within the page designer in order populate your UI with data.

## Pages

The Pages tab allows you to define the individual pages within your page series. The app designer displays a separate list of pages for each design-time platform. You can use the action button in the header of each of these lists to perform a range of actions on your page definitions.

New	Creates a new page definition.
Delete	Deletes the pages that you have tick-selected.
Copy	Copies the pages that you have tick-selected.
Paste	Creates new pages based on the pages that have been previously been copied.

## Designing pages

When you click on a page name within one of the design-time page lists, a new page will be displayed within the designer – this is the page designer.

The page designer is the most complex aspect of the app designer. This guide only presents an overview of the functionality supported by the page designer. For detailed information on a range of page designer features, please use the video snippet library via the BlueFinity web site.

The functionality and data collected by the page designer is divided into a series of separate tab panels, accessed at the top of the page designer's page. Each of these tabs is explained below.

### Layout designer

The first aspect of page design is to divide the overall surface area of the page into the desired sections in order to apply a structure and visual aesthetics.

The mechanism for dividing up the surface area of a page is by splitting it into rows and columns. You need to first decide whether the top level division is by rows or by columns. You can then further sub-divide the top level rows into a series of columns, or the top level columns into a series of rows. You can carry on sub-dividing rows and columns as necessary to any level of nesting.

The layout designer provides a simple yet very flexible method of sub-dividing your page area into whatever rows and columns are necessary in order to give the correct breakdown of the overall area. The generic term for a row or column is *page segment*.

At some point during the process of dividing the page area, you will reach a point where you will no longer need to sub-divide a page segment. At this point you can define what kind of content this non-divided page segment is to hold. If the page segment is to display entity data, you can define whether the data content is to come from a single entity instance or from a collection of instances.

### Page segment designer

Once you have divided your page area into segments, you can use the Settings tab to add more detailed styling and definition data to each segment as necessary. When you click on a page segment within the designer its current settings are displayed within the tabbed area to the right of the designer screen. There are a number of different setting types available, as summarized below:

Content	Allows you define whether the segment should contain a header and/or footer and also allows you to select what kind of entity data should be displayed (if relevant).
---------	---

- Sizes** Allows you to define spacing between the page segment and neighboring segments along with the spacing between various internal areas of the segment.
- Color** Allows you to apply color to the various parts of the page segment.
- Navigation** Allows you to define how navigation to child pages can be initiated from within the page segment.
- Adaptations** Allows you to define advanced UI adaptations to be activated when the size (height or width) of the app display area crosses a set threshold.

## General page settings

The General details tab allows you set edit various general details of the page. The Entry page checkbox allows you to flag which page is to be displayed when its associated menu option is first selected.

On the right of the general details page, you are able to define two sets of page level options:

- Sibling page navigation** Allows you to define the pages that are to be included in the sibling page navigation button's drop down list.
- Page level actions** Allows you to define what actions are to be associated with an action button displayed in the top left corner of the screen. This action button will only be displayed if you enter one or more actions in this grid. The use of page level actions caters for situations where you want to allow the user to initiate an action on the current page as a whole, e.g. Print, Email, etc.