|  |
| --- |
|  |

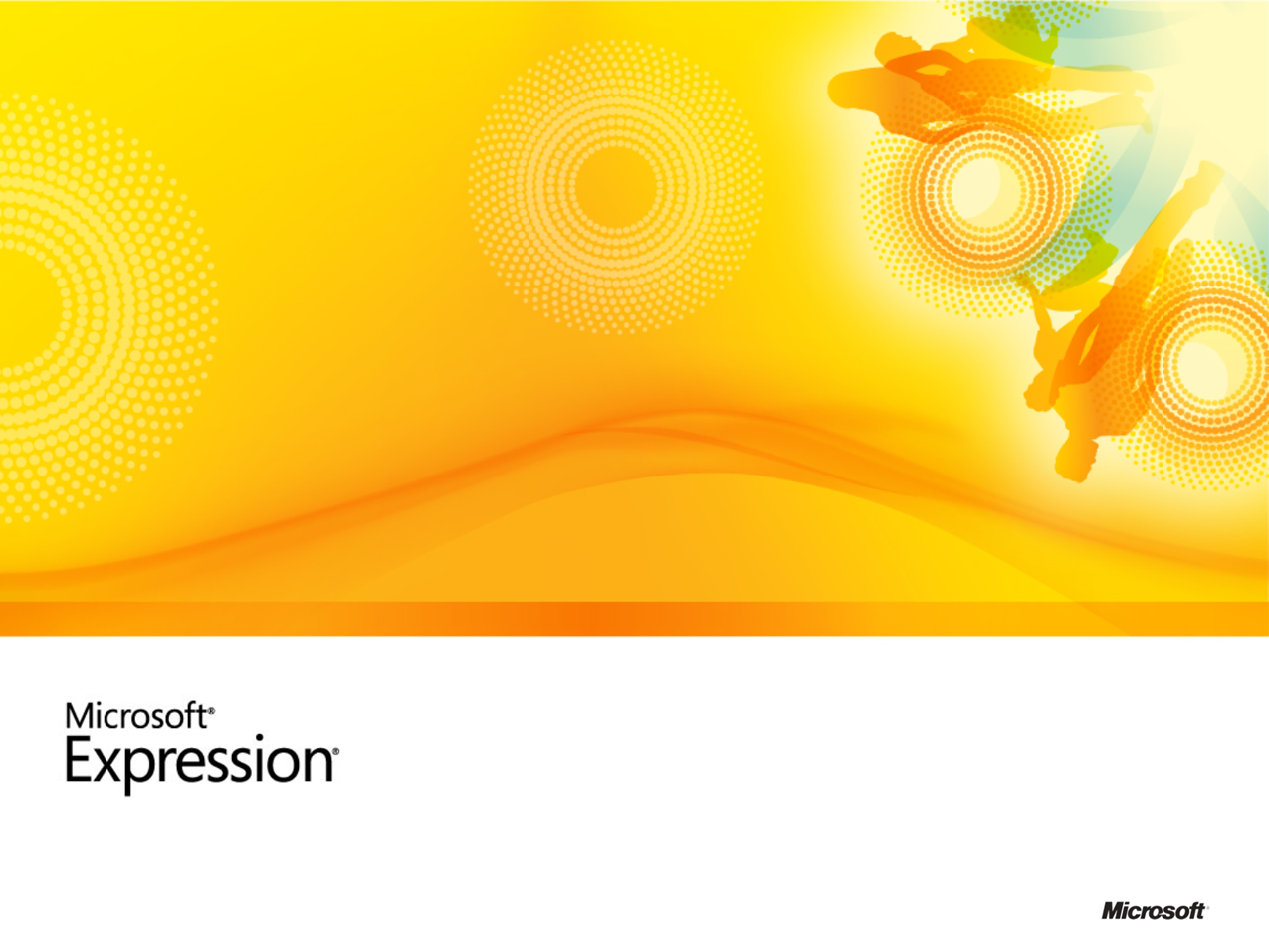
Microsoft® Expression Blend 4 Beta

Technical Feature Overview

Adam Kinney

http://adamkinney.com

March 15 2010



[Microsoft® Expression Blend 4 Beta 1](#_Toc256160545)

[Introduction 3](#_Toc256160546)

[Support for Silverlight 4 and WPF 4 4](#_Toc256160547)

[Create and Edit Silverlight 4 and WPF 4 Projects 4](#_Toc256160548)

[Compatibility with Silverlight 3 and WPF 3.5 Service Pack 1 4](#_Toc256160549)

[Interoperability with Visual Studio 2010 5](#_Toc256160550)

[SketchFlow Improvements for better Prototypes 6](#_Toc256160551)

[SketchStyles 6](#_Toc256160552)

[Naming Screens 6](#_Toc256160553)

[Pausing and Resuming a SketchFlow Animation 6](#_Toc256160554)

[SketchFlow animation picker 6](#_Toc256160555)

[SketchFlow Player Features 7](#_Toc256160556)

[Word Export includes Feedback 8](#_Toc256160557)

[Enabling Business Application Development 9](#_Toc256160558)

[Support for Development using the MVVM Pattern 9](#_Toc256160559)

[MVVM Project and Item Templates 9](#_Toc256160560)

[Design-time data from CLR types 9](#_Toc256160561)

[Design-time Resources 10](#_Toc256160562)

[New Controls and Effects for Dynamic Interfaces 11](#_Toc256160563)

[Improved Control Styling 11](#_Toc256160564)

[Layout States for the ListBoxItem 11](#_Toc256160565)

[PathListBox Control 11](#_Toc256160566)

[Shape Controls 12](#_Toc256160567)

[Common WPF 4 controls are Visual State aware 13](#_Toc256160568)

[Easing Functions 13](#_Toc256160569)

[Pixel Shader Effects 14](#_Toc256160570)

[Transition Effects 14](#_Toc256160571)

[More Interactivity and Less Code with New Behaviors 16](#_Toc256160572)

[Conditional Behaviors 16](#_Toc256160573)

[Bindable Properties 16](#_Toc256160574)

[New Behaviors 17](#_Toc256160575)

[Data Store 18](#_Toc256160576)

[Editor and Tooling Improvements 19](#_Toc256160577)

[Improvements to Photoshop file import 19](#_Toc256160578)

[CompositeTransform 19](#_Toc256160579)

[No modifier key needed for marquee selection of keyframes 19](#_Toc256160580)

[XAML cleanliness 19](#_Toc256160581)

[Run Project with Ctrl+F5 19](#_Toc256160582)

[Zip Support for Samples and Templates 19](#_Toc256160583)

[Expression Blend Resources 20](#_Toc256160584)

[Downloads 20](#_Toc256160585)

[Community Sites 20](#_Toc256160586)

[Social Media Sites 20](#_Toc256160587)

# Introduction

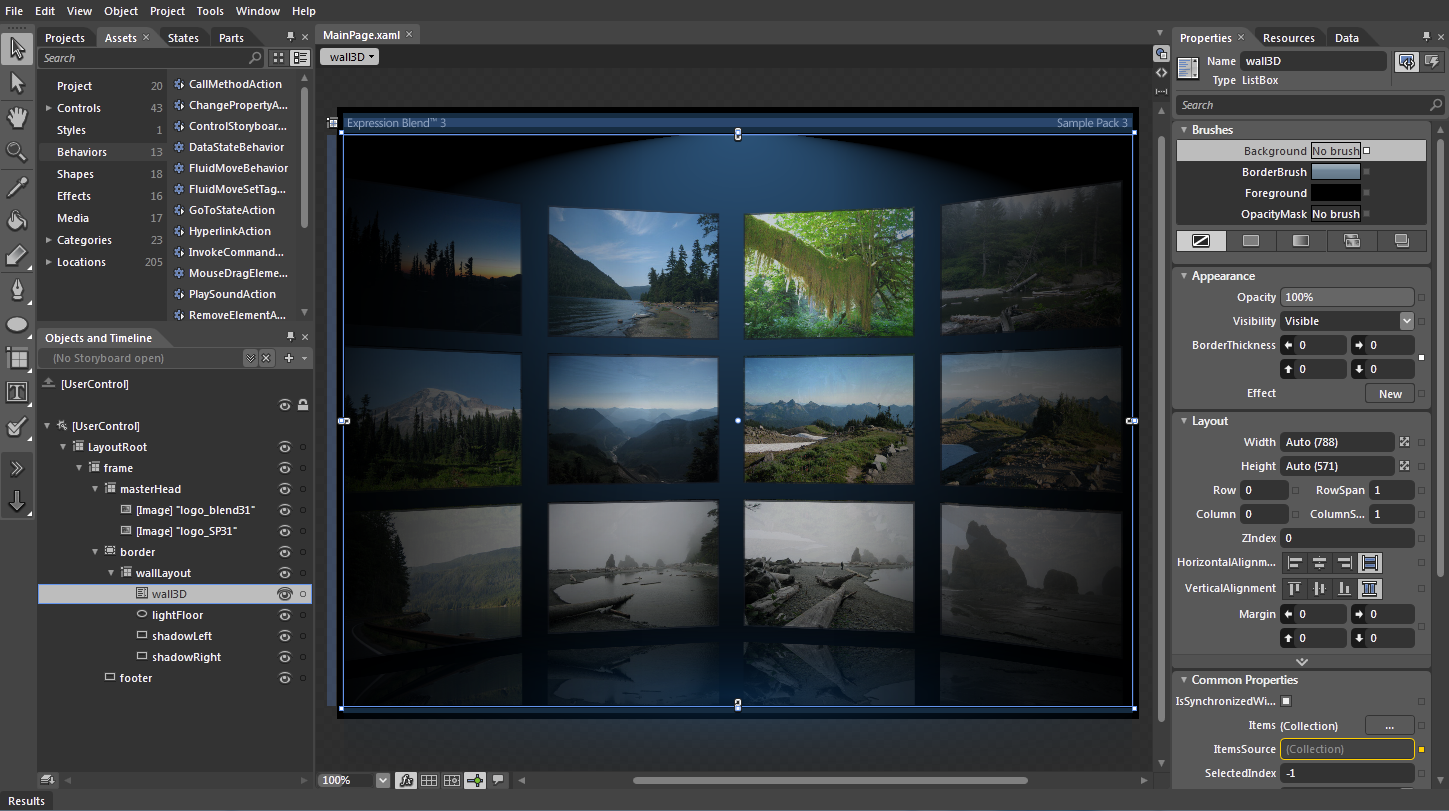
Expression Blend 4 Beta adds support for Silverlight 4 and Windows Presentation Foundation 4, while maintaining the ability to create and edit Silverlight 3 and WPF 3.5 with Service Pack 1(SP1) applications. Interoperability with Visual Studio 2010 has been added, maintaining the seamless workflow between the two tools.

In response to customers’ desires to do more without code, new features in SketchFlow enable more dynamic prototypes to be created and provide an enhanced player experience. Additionally, Expression Blend 4 Beta adds powerful new behaviors for all application types and makes them configurable with conditional statements.

Expression Blend 4 Beta introduces new features that support the Model-View-ViewModel (MVVM) pattern for business application development. New design-time data and resource features, along with behaviors such as CallMethodAction and InvokeCommandAction greatly increase the capability and experience for design tasks while building these applications.

Along with these three areas of focus, other additions include: new controls, such as the PathListBox, new pixel shader effects, transition effects and helpful tooling improvements.

Expression Blend 4 Beta is an exciting update that continues to revolutionize the speed and efficiency with which you can take your ideas from concept to completion on the Silverlight and .NET platforms.



# Support for Silverlight 4 and WPF 4

## Create and Edit Silverlight 4 and WPF 4 Projects

You can create and modify Silverlight 4 and WPF 4 projects in Expression Blend 4 Beta. You can also upgrade a Silverlight 3 project to target Silverlight 4, or upgrade a WPF 3.5 SP1 project to target WPF 4. The project types available include:

**Silverlight Application + Website -** Creates a Solution file, a SilverlightApplication Project and a Website Project. The benefit of the added Website Project is that it allows you to set custom properties on the object tag used to instantiate your Silverlight application.

**Silverlight Application -** Creates a Solution file and a SilverlightApplication Project. When you choose to run your project, an HTML file is generated dynamically to host your application.

**Silverlight Data-driven Application** - Creates a Solution file and a SilverlightApplication Project using the new MVVM Project template. MVVM, or Model-ViewModel-View, is a programming pattern that separates logic from presentation and is discussed in more detail below.

**Silverlight Control Library** -Creates a Solution file and a SilverlightProject. This Project type is designed to define controls that may be used in one or more separate applications.

**Silverlight SketchFlow Application** -Creates a Solution file and two Silverlight Projects using the SketchFlow template. Additionally specific SketchFlow panels and styles are made available.

**WPF Application** -Creates a Solution file and a WPFApplication Project.

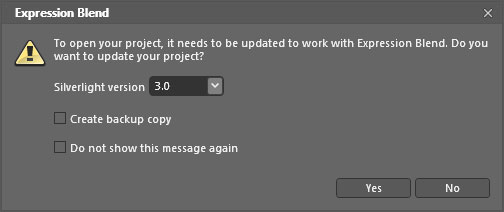
**WPF Data-driven Application** - Creates a Solution file and a WPFApplication Project using the new MVVM Project template. MVVM, or Model-ViewModel-View, is a programming pattern that separates logic from presentation and is discussed in more detail below.

**WPF Control Library** -Creates a Solution file and a WPFProject. This Project type is designed to define controls that may be used in one or more separate applications.

**WPF SketchFlow Application** -Creates a Solution file and two WPF Projects using the SketchFlow template. Additionally specific SketchFlow panels and styles are made available.

## Compatibility with Silverlight 3 and WPF 3.5 Service Pack 1

You can still create and modify Silverlight 3 and WPF 3.5 SP1 application projects in Expression Blend 4 Beta. When you open a Silverlight 3 or WPF 3.5 SP1 project that is using the Visual Studio 2008 project format, you will be prompted to update the solution and project files.

Figure 1

The Solution and Project update dialog that appears when opening a Visual Studio 2008 formatted Solution or Project

## Interoperability with Visual Studio 2010

Expression Blend 4 Beta and Visual Studio 2010 use the same project format and both support Team Foundation Server for source control. For quick access, an “Edit in Visual Studio” option is available when right-clicking Solution or Project files in the Projects panel.

# SketchFlow Improvements for better Prototypes

SketchFlow revolutionizes how quickly and efficiently you can demonstrate your vision for an application. It provides an informal and quick way to explore, iterate and prototype user interface scenarios, allowing you to evolve your concepts from a series of rough ideas into a living and breathing prototype that can be made as real as a particular client or project demands. This rapid, iterative and cost effective approach to prototyping allows you to concentrate on what matters most, being creative and building the best solution for your client, on time and within budget.

## SketchStyles

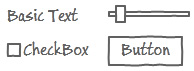
The default Style set for a new SketchFlow project is now SketchStyles. For example, when you add a Button to a screen the SketchFlow style will be applied rather than the default Button style. You can still switch to the default Style or a custom Style at any time.

Figure 2 SketchFlow Styles

## Naming Screens

When creating a new Screen or Component Screen in the SketchFlow map, the screen name is selected by default, allowing you to enter a new name instantly. Hitting the Enter key will then save the new screen name.

## Pausing and Resuming a SketchFlow Animation

Within Blend there is a new pause-on-this-frame toggle in each frame of a SketchFlow Animation. When toggled on, this causes the SketchFlowAnimation to pause at that frame when playing back in the SketchFlow Player. And, in the SketchFlow Player, there is a play/pause button beside each SketchFlow Animation that can be used to play, pause or resume that animation.

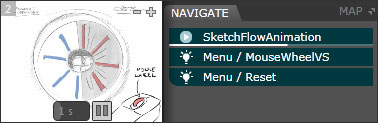


Figure 3

The left section shows how to set a frame to pause during a SketchFlow Animation. The right section shows a SketchFlow Animation paused during playback in the SketchFlow Player.

## SketchFlow animation picker

The PlaySketchFlowAnimationAction is used to play a SketchFlow Animation in response to an event or another type of trigger. Now you can use a new picker in the property inspector to pick a SketchFlow Animation for the action to control.

## SketchFlow Player Features

New features have been added to the SketchFlow Player to allow for panning, scrolling and screen resizing. A subtle border surrounds each screen that a user can click and drag to resize the screen. If the original dimensions or the resized area of the screen is larger than the viewing area, scrollbars appear to allow for horizontal panning and vertical scrolling.

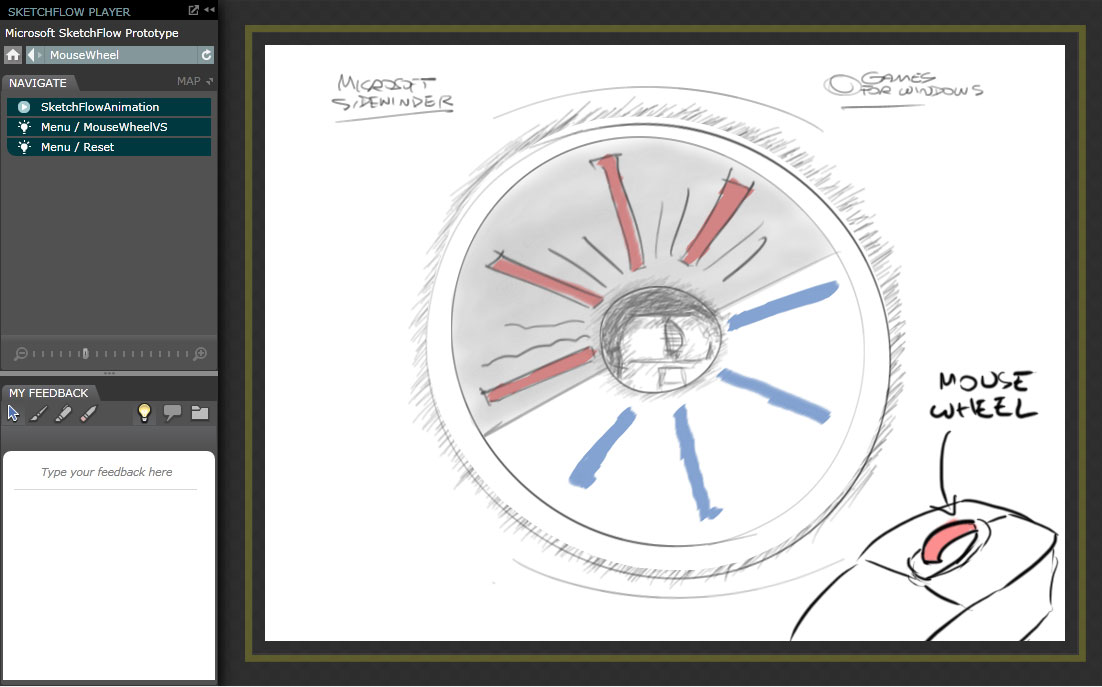


Figure 4

The improved SketchFlow Player with the resize handle highlighted in yellow.

The SketchFlow map is now accessed using the Map button (  ) found next to the Navigation tab. Clicking the button opens the map as an overlay above the current screen.

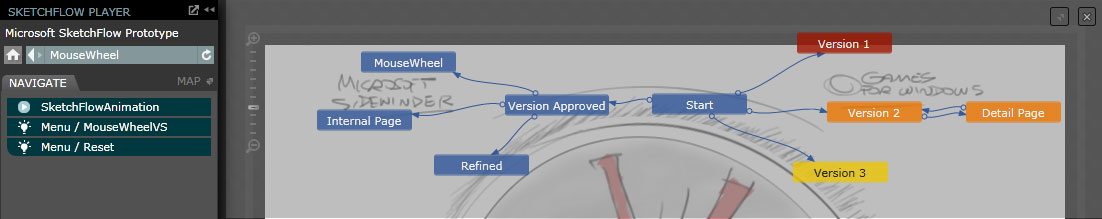


Figure 5

The SketchFlow map opened as an overlay on top of the SketchFlow Player.

## Word Export includes Feedback

When using the Export to Word feature in SketchFlow, a new option has been added called “Include Feedback”. When checked, feedback that is loaded in Blend will be exported in the Word document along with the screens and components.

# Enabling Business Application Development

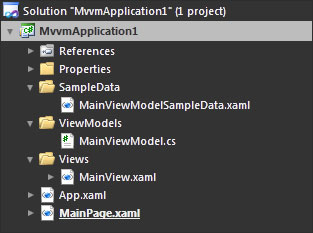
## Support for Development using the MVVM Pattern

Expression Blend 4 Beta offers new sample data, behaviors and project template features to support the MVVM pattern, as will be described in the relevant sections.

The Model-View-ViewModel pattern is a way to structure a Silverlight or WPF application so that UI objects are as decoupled as possible from the application’s data and behavior. This makes it easier for design and development tasks to be performed independently and without breaking one another.

Essentially your UI is the View. You bind objects in the View to properties and commands of the ViewModel, and the View can call methods on the ViewModel.

## MVVM Project and Item Templates

Starting a new project based on the Silverlight Data-driven Application project template creates the standard Silverlight Application files and the following extra items:

**MainViewModel.cs** – A class that defines a property and method for the MainView, providing the data and behavior for the View.

**MainViewModelSampleData.xaml** –A sample instance of the MainViewModel class used as a design-time resource.

**MainView.xaml** – A UserControl with its DataContext property bound to an instance of

Figure 6 A Newly created MVVM Project

the MainViewModel class. The text property of the TextBlock is then bound to a property of the class in order to display the value. Using the new CallMethodAction behavior, the Button is bound to a method of the MainViewModel class.

Additionally, the design-time DataContext property of the UserControl is set to the ModelViewModelSampleData.xaml file in order to provide values for the class properties during design-time, enhancing the ability to design the UI for this dynamic data-driven application.

## Design-time data from CLR types

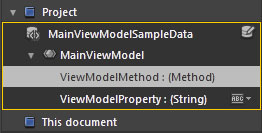
In addition to sample data based on XML, you can now create design-time sample data from your CLR types. Even if a type has no public constructor, or it has properties with no public setter, it can still be made into design-time sample data with the “Create Sample Data from Existing Class” command on the Data panel.

Figure 7 Sample Design-time data based on a CLR type

The Data panel provides a view of an object’s DataContext, whether design-time or run-time, and allows you to drag properties, methods and Commands onto the design surface to bind your UI to them.

## Design-time Resources

Sometimes, resources that will resolve at run-time don’t resolve at design-time. In these cases you can now simply pick a resource dictionary to use at design-time while you’re designing your application.

The Add Design-time Resource Dictionary dialog will open if the current document contains resources that cannot be resolved and the solution contains at least one Resource Dictionary. Once selected, the dictionary will be merged with the Design-time Resources dictionary located under the Properties folder. The resources will now be available during design-time, but will not be available during run-time or compiled with the application. This feature is intended to be used in scenarios where Resource Dictionaries are loaded during run-time in code.

As an example, an application may allow a user to personalize the look of the application by choosing a specific theme. In order to accommodate this feature, the theme, stored in a Resource Dictionary, would be loaded dynamically during run-time. Without changing any code, design tasks could be completed much easier with one of these custom themes selected as a design-time resource.

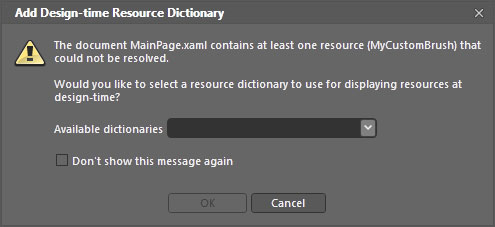


Figure 8

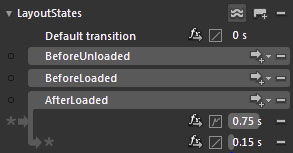
The Add Design-time Resource Dictionary dialog

# New Controls and Effects for Dynamic Interfaces

## Improved Control Styling

Many control Styles and Templates reside in Resource Dictionaries separate from the control instances using them. With Expression Blend 4 Beta, the control instance, its property values and data context, are all carried over to the Style or Template when editing to improve the experience.

## ****Layout States for the ListBoxItem****

In Silverlight 4 RC, you can now animate the process of items being added to and removed from a ListBox, and control all the details of those transitions including easing. You can customize the transition animations by creating your own ItemsContainerStyle.

The new states include:

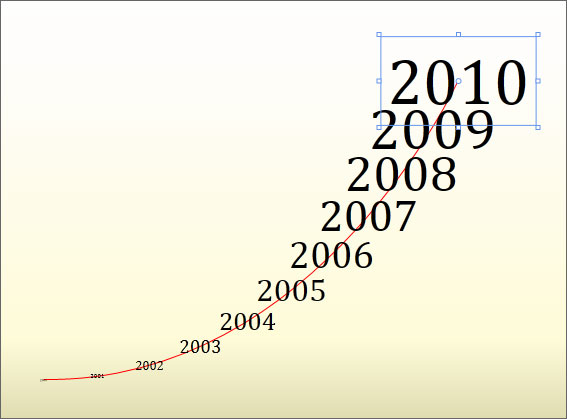
**BeforeLoaded –** Occurs before theListBoxItem is added to the ListBox.

Figure 9 Layout States with custom transitions

**AfterLoaded** – Occurs after the BeforeLoaded state and the ListBoxItem is added to the ListBox.

**BeforeUnloaded** – Occurs before the ListBoxItem is removed from the ListBox.

## PathListBox Control

The PathListBox is a new control that introduces a new and flexible way to layout multiple items. Maintaining the selection and binding features of the traditional ListBox, the layout of the items is determined by defining one or more UIElements as LayoutPaths.

Many properties are made available to customize the positioning and orientation of the items along the paths.

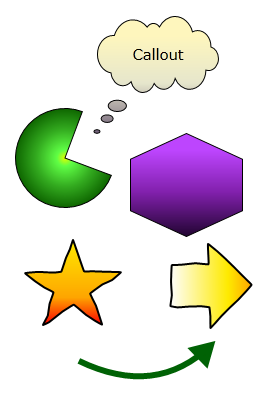
Additionally, individual PathListBoxItems may be modified directly to change their appearance or layout.

Figure 10 A PathListBox with a red arc defined as the LayoutPath

Create your own PathListBox by following the steps below:

1. Add one or two paths to a UserControl. The Path, Ellipse, Rectangle and Shape controls work well as LayoutPaths, but any UIElement will work.
2. Add a PathListBox control from the Assets Browser.
3. Add items to the PathListBox manually or through databinding. Without a LayoutPath defined, you will notice that the items will all be positioned in the same spot.
4. With the PathListBox selected, find the Items Layout pane on the Properties Panel. Use the pick widget ( D:\PlatformEvangelism\Sandbox\adkinn\Blend4\NewInBlend4Images\pickwidget.jpg ) to select one or more objects to use as LayoutPaths.
5. The items of the PathListBox will be arranged along the selected path or paths.

## Shape Controls

The Assets panel in Expression Blend 4 Beta contains a new Shapes category, for easy creation of arcs, arrows, callouts and polygons. The Shape controls provide many properties for customization, including a Sketch style Geometry Effect for use in SketchFlow prototypes.  
The new shapes include:

**Arc –** Available in theAsset browser in three convenient formats: Arc, Pie and Ring.

**BlockArrow -** Available in theAsset browser in four formats: Down, Left, Right and Up.

**Callout -** Available in theAsset browser in four different styles: Cloud, Oval, Rectangle, Rounded Rectangle.

**Pentagon –** Available in theAsset browser in four preconfigured formats: Hexagon, Pentagon, Star and Triangle. This control is easily extended to other shapes by adding more points.

**LineArrow –** A very flexible Arrow control.

Figure 11 New Shape Controls

## Common WPF 4 controls are Visual State aware

When you create or edit a template for some common WPF 4 controls, the States panel now populates with a list of States ready for you to select and design. This is because the following WPF 4 controls are designed to work natively with the Visual State Manager and they advertise the States they work with:

|  |  |
| --- | --- |
| * Button * CalendarButton * CalendarDayButton * CalendarItem * CheckBox * ComboBox * ComboBoxItem * Control * DataGrid * DataGridCell * DataGridColumnHeader * DataGridRow * DataGridRowHeader * DatePicker * DatePickerTextBox * Expander * GridSplitter | * GridViewColumnHeader * ListBoxItem * PasswordBox * ProgressBar * RadioButton * RichTextBox * RepeatButton * ScrollBar * Slider * TabControl * TabItem * TextBox * Thumb * ToggleButton * ToolTip * TreeViewItem |

## Easing Functions

Easing functions for animations and Visual State changes are now available in WPF 4, in the same way as they are accessible in Silverlight 3 and above projects.

## Pixel Shader Effects

Adding to the variety of graphic capabilities, nine new Effects have been made available. Found in the Asset Browser, the Effects can be applied to elements with a simple drag and drop action. Properties of the Effects can be animated with Storyboards or within code behind.

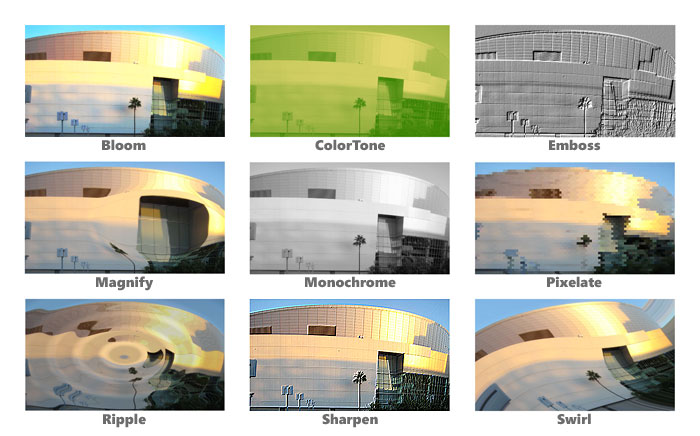
The new effects include:

Figure 12

Nine new pixel shader effects

## Transition Effects

Transition Effects blend two visuals using a pixel effect over time. They are applied in Blend 4 Beta in the States panel. The Transition Effects dropdown ( D:\PlatformEvangelism\Sandbox\adkinn\Blend4\NewInBlend4Images\fx.jpg ) can be found in line with each transition next to the easing function button. The Transition Effect will be visible and animated with other properties during the transition between Visual States.

Transition Effects available include the following:

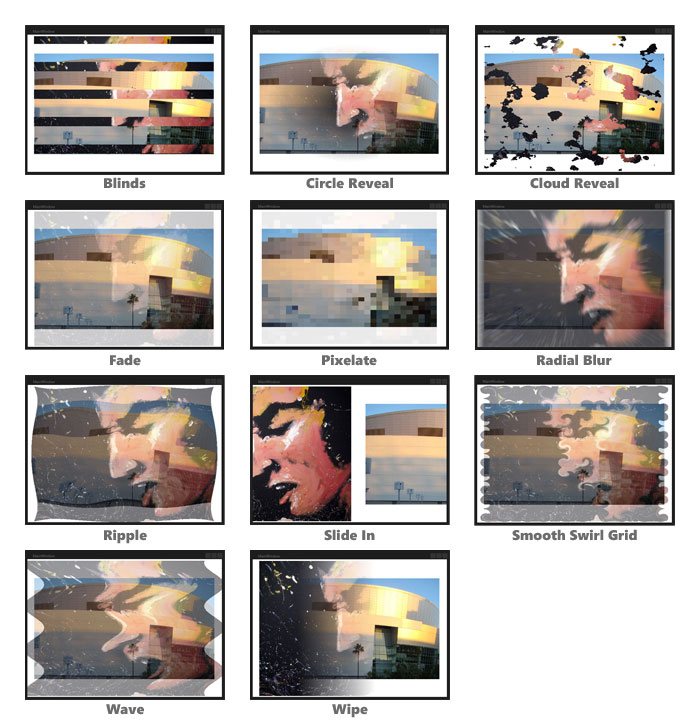
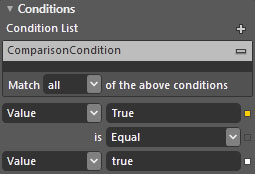


Figure 13

The 11 Transition Effect available

# More Interactivity and Less Code with New Behaviors

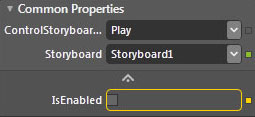
## Conditional Behaviors

Any Action can now be associated with a set of conditions that must be met in order to execute the Action. This means that with Expression Blend 4 Beta you can now build conditional logic into your prototypes and production applications without the need to write code.

In the example to the right, the first operand is using Element binding to bind to the IsChecked property of a Checkbox instance. The second operand is simply set to the string “true” which is then converted to a Boolean during the comparison. In this case, the Action would only happen if the checkbox was checked.

Figure 14 The Conditional pane

## Bindable Properties

Properties of behaviors are now bindable in the same way as FrameworkElement properties. This allows behaviors to be more dynamic and aware of the current context.

To create a quick sample, follow the steps below:

1. Add a Rectangle to a UserControl.

Figure 15 A Behavior property bound to an element

1. Create a Storyboard animating the Rectangle.
2. Add a ControlStoryboardAction to the Rectangle and set the Storyboard property to the one you just created.
3. Add a Checkbox to your UserControl.
4. Select the ControlStoryboardAction and show the advanced properties in the Common Properties pane.
5. Click the Advanced properties button to the right of the IsEnabled property and select the “Element Property Binding…” option.
6. Mouse over and select the Checkbox. From the Create Data Binding dialog select the IsChecked property.
7. Now when you run the project, the Storyboard should only play when you click on the Rectangle and the Checkbox is checked.

## New Behaviors

Adding to the original set of behaviors, new behaviors have been added with a strong focus on interacting with data. Additionally the FluidMoveBehavior has been updated to provide more functionality than before. The new behaviors include the following:

**CallMethodAction** – Calls a method on the target object. This behavior works especially well when using the MVVM pattern for development.

**DataStateBehavior** – When a specified property changes, a comparison statement is run against the new value and when true the current state is set to the specified TrueState, when false the current state is set to the specified FalseState.

**DataTrigger** – When a specified property changes, a comparison statement is run against the new value and if successful the trigger will fire.

**DataStoreChangeTrigger** – Fires when a specified property in the Data Store changes.

**FluidMoveBehavior** – Enhanced with additional properties to expand its scope beyond a single container. By setting a tag property, an element’s current state can be used to dynamically animate discrete properties of the element. FluidMoveBehavior is optimized for two main scenarios; animating list items from one list to another and animating from a master list to a detail view.

**FluidMoveSetTagBehavior** – Sets a tag property to act as a pointer that can be referenced by a FluidMoveBehavior instance. In a Master/Detail scenario, the elements of a master list may only need to broadcast their position. The detail view can then be animated from that position using the recorded values.

**InvokeCommandAction** – Invokes a specified ICommand. This behavior works especially well when using the MVVM pattern for development.

**SetDataStoreValueAction** –Changes the value of a property and optionally animates the change over time. Properties are defined in the new Data Store, explained below.

**TranslateZoomRotateBehavior** - This behavior, available only in WPF 4, enables the target control to be translated, rotated or zoomed in and out using touch gestures.

## Data Store

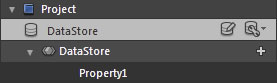
There is also a new feature known as the data store where application variables, such as the current user’s role for example, can be read from and written to so that different parts of your application can communicate via this shared state.

Figure 16 A Data Store instance in the Data panel

New behavior components introduced as part of this feature include the conditions editor which appears in the Properties panel for every action, a SetDataStoreValueAction that allows you to manipulate values in your data store, and a DataStoreChangedTrigger that fires whenever a chosen property inside the data store is changed.

# Editor and Tooling Improvements

## Improvements to Photoshop file import

Expression Blend 4 Beta can now import Adobe Photoshop (.psd) files faster and with less system utilization. Also, the following layer effects can now be merged with their layer: Inner Shadow, Color Overlay, Outer Glow, Inner Glow, Gradient Overlay, Stroke, Bevel & Emboss, Satin, and Pattern Overlay.

## CompositeTransform

In place of a TransformGroup containing the individual ScaleTransform, RotateTransform, SkewTransform and TranslateTransform, Blend now defaults to using the simpler CompositeTransform. Exposing properties that match the desired type of Transform, like ScaleX and Rotation, CompositeTransforms are less verbose and easier to code against.

## No modifier key needed for marquee selection of keyframes

In Expression Blend 4 you can marquee-select keyframes in the Timeline more naturally with the mouse. Previously, marquee selection of keyframes required the Ctrl key to be pressed while dragging.

## XAML cleanliness

Expression Blend 4 now generates less XAML with respect to animations and animation-related properties.

## Run Project with Ctrl+F5

To improve consistency with Visual Studio, the “Run Project” command can now be invoked with Ctrl+F5 as well as F5.

## Zip Support for Samples and Templates

Expression Blend 4 Beta now supports reading item templates, project templates, and samples from Zip files as well as loose files and folders.

# Expression Blend Resources

## Downloads

Expression Blend 4 Beta - <http://go.microsoft.com/fwlink/?linkid=169446>  
  
Silverlight 4 RC - <http://go.microsoft.com/fwlink/?linkid=169417>

Silverlight 4 RC Technical Overview - <http://go.microsoft.com/fwlink/?linkid=177698>

## Community Sites

Expression Forums and Gallery - <http://expression.microsoft.com>

Expression Blend and Design blog - <http://blogs.msdn.com/expression/>

Project Rosetta **-** <http://projectrosetta.com>

## Social Media Sites

Expression Twitter account - <http://twitter.com/msexpression>

A Twitter list of Expression Blend minded folks - <http://twitter.com/adkinn/expressionblend>  
  
SilverlightTV – <http://silverlight.tv>

Expression Studio on Facebook - <http://www.facebook.com/pages/Expression-Studio/108085382071>