

Dec 2013, HAPPIEST MINDS TECHNOLOGIES

# Flow Document – A solution to text readers in .Net

Author(s):  
Nivedita Parihar  
Archita Dash



happiest minds  
The Mindful IT Company

Born **Digital** . Born **Agile**

## Copyright Information

This document is an exclusive property of Happiest Minds Technologies. It is intended for limited circulation.

## Contents

<b>Copyright Information .....</b>	<b>2</b>
<b>Abstract .....</b>	<b>4</b>
<b>Introduction .....</b>	<b>5</b>
<b>Use of Flow Document in several scenarios:.....</b>	<b>6</b>
<b>Sample Application Screenshot.....</b>	<b>9</b>
<b>Conclusion:.....</b>	<b>10</b>
<b>References: .....</b>	<b>10</b>

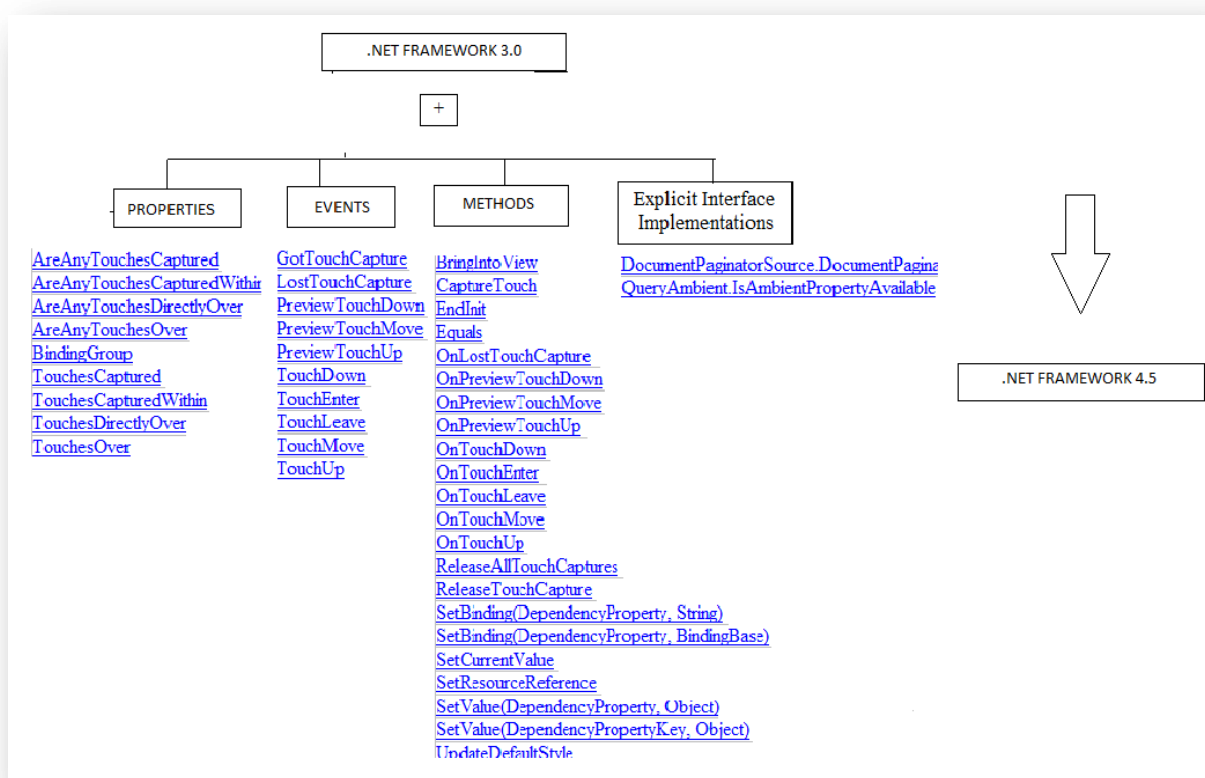
## Abstract

.Net provides a wide variety of features. At times some of the features don't get due credit from developers. One of these features is Flow Document. Flow Documents were introduced with .Net Framework 3.0. It is a kind of container that can work for almost everything related to text content. It can be used as a text reader, for example, an offline reader for drop box. There are many reasons why Flow Document did not get its required exposure. One is the fact that it was considered as an option to XPS. Many developers used to customize the Panels existing in .Net to achieve the functionalities equivalent to Flow Document. However, Flow Documents are richer and provide enhanced layout options. This document will discuss XAML Flow Document in different contextual scenarios, such as comparison with HTML5 tags and its usability. The paper also describes the scenarios where Flow Documents can provide a significant edge over using the Panels/XPS, the significance of using them in mobile applications and how can the same application be created for different platforms like desktop native applications or mobile applications using Flow Documents and its equivalent tags in HTML. The white paper will also discuss the best usage scenarios, disadvantages of Flow Document and how we can overcome them by customizing it.

## Introduction

Flow Documents are designed to optimize readability and it dynamically reflows and adjusts the contents based on runtime requirements such as window size device resolution etc. Flow Document has evolved since the time it got first introduced as .Net Framework 3.0. The changes have made Flow Document more usable and portable. Now Flow Document can be used in native apps on desktops, tablets and mobile.

This diagram below shows the various features added in Flow Document after it was introduced in .Net framework version 3.0.



The diagram explains that from .Net framework 3.0 to 4.5, after introduction of Flow Document there are several features added to enable it to be used in mobile applications.

<b>Properties</b>	Touch properties are added in the Flow Document and it's also binds each and every property.
<b>Events</b>	Touch events are added to Flow Documents for previewing it in mobile applications.
<b>Methods</b>	Methods also include the touch logic implementation to work with touch mobile applications and to set resources and bindings.
<b>Explicit Interface Implementations</b>	Implements content pagination.

Looking at the diagram it becomes very easy to conclude that with introduction of methods and properties that support touch, orientation changes and effective data binding Flow Documents now are ready to be used in next gen applications like tabs and mobiles.

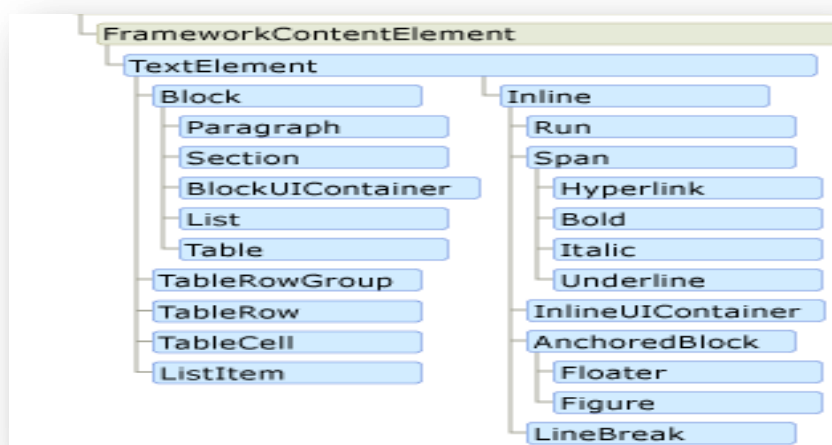
Let us now discuss on how Flow Documents are currently used in various scenarios.

## Use of Flow Document in several scenarios:

### 1. Flow doc use as a native app - WPF / Silverlight/Mobile app

There are many mobile applications which use Flow Document, for example: In Google book reader app for Windows Phone 8 which uses Google Books API for fetching data, we can make use of Flow Document to go through the pages of selected books.

### 2. Flow doc to HTML5



These are the markup tags of XAML Flow Document.

Let us consider an example:

#### XAML Code

```

<Flow Document>
<Paragraph>Text<Run FontWeight="bold">Element</Run>that needs to be converted to
HTML. </Paragraph>
</Flow Document>
  
```

Let us now see how the same code can be ported in HTML

#### HTML Code

```

<p> Text <b> Element </b> that needs to be converted to HTML.</p>
  
```

Notice, that even if someone develops a native app with a Flow Document it just takes a few seconds to convert it into a HTML5 document.

Now let us consider some of the equivalent tags of XAML Flow Document converted into HTML tags

XAML Flow Document Tags	HTML5 Tags
<Paragraph>	<p>
<Section>	<Blockquote>, <div>,<section>
<List>	<ul>
<ListItem>	<li>
<Bold>	<b>
<Table>	<Table>
<TableRow>	<tr>
<Tablecell>	<td>
<Figure>	<Figure>
<LineBreak/>	 
<span> , <Run>	<span>

A basic tool is available from Microsoft to convert a XAML document into HTML5 tags. Following is the link.

<http://blogs.msdn.com/b/wpfsdk/archive/2006/05/25/606317.aspx>

Although this tool easily converts HTML5 Documents to XAML documents but when we evaluated this tool we came across multiple limitations:

- It does not have scrollable view for typing the XAML or HTML5 tags.
- If we are typing an internal piece of code for XAML document it cannot convert to its equivalent tags i.e. starting from the tag <Flow Document>.
- It can only convert the small simple XAML document to HTML but it cannot solve the complex XAML document to convert it to HTML.

We can improvise this tool to make it more useful for various business related scenarios.

### **3. How Flow Document can be used for eLearning sites/educational sites/research**

The eLearning sites, Educational Sites and in the field of research, document reader is almost used everywhere. Many eLearning sites for example, Pearson eLearning Portal, Tutor Vista eLearning Portal, are guiding people worldwide with their interactive tutorials. People are able to access several books through these educational sites which are mostly build using XAML Flow Document. Even for research people store their discoveries in applications in the form of documents which can be made using Flow Documents.

- **EBook Reader**

To understand this, consider example of Amazon eBook Reader (Kindle) or any other. People can read and access books and documents from the cloud or service provider datacenter. Similarly we can build custom eBook Reader for an enterprise, easily with the use of XAML Flow Document which includes features of pagination and ambience.

- **Offline drop box**

After browsing eBooks if we want to store those books in DropBox, we can create our own offline DropBox and save the browsed eBooks and access them offline. Any file you save to Drop Box also instantly saves to your computers, phones, and the Drop Box website. Flow Document in offline Drop Box, extracts the content of eBooks and parses it to display the eBook content stored in Drop Box in form of Flow Document.

#### ***4. Flow Document is as a Problem Solver when compared to Panels.***

If we look at the difference between use of Flow Document and Panels for implementation of making document, some developers are not fully cognizant of the capabilities of Flow Document vis-à-vis the Panel. There are several issues which can be resolved by Flow Documents, such as hosting and formatting flow content with advanced document features such as pagination and columns. Flow Document is a heavy object, therefore if not handled correctly applications run the risk of becoming unnecessarily heavy. If it is possible to use Panel and implement the required features without using Flow Document, then that is an advisable solution.

#### ***5. How Flow Documents are easily printable, no need to use XPS.***

Flow Document Reader already has inbuilt print feature, which is used as below:

```
// Create a PrintDialog
PrintDialog printDlg = new PrintDialog ();

// create a Flow Document dynamically.
Flow Document doc = // creating Flow Document
doc.Name = "FlowDoc";

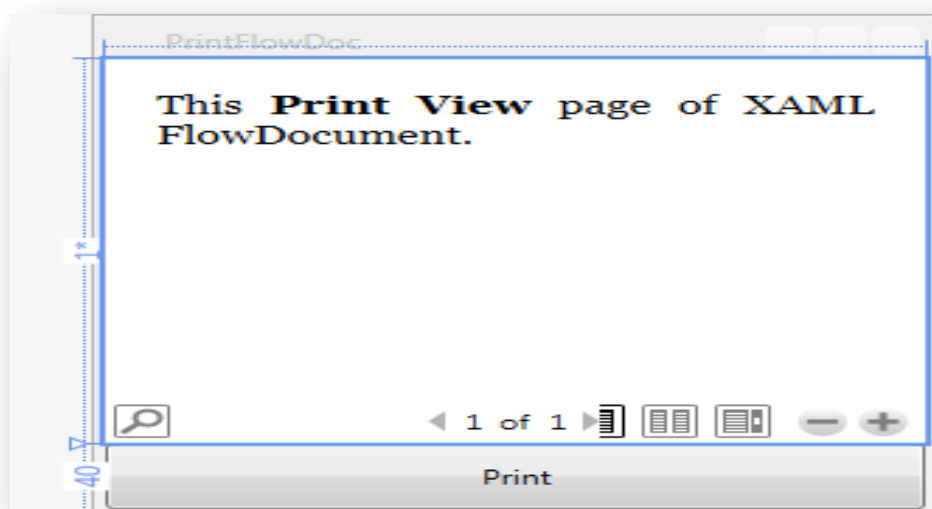
// Create IDocumentPaginatorSource from Flow Document
IDocumentPaginatorSource idpSource = doc;

// Call PrintDocument method to send document to printer
printDlg.PrintDocument (idpSource.DocumentPaginator, "Hello WPF Printing.");
```



## Sample Application Screenshot

Here is the Sample Application Screenshot How to use Flow Document.



## Conclusion:

Flow Documents are best utilized when ease of reading is the primary document consumption scenario. Flow Documents have a number of built in features including search, pagination, viewing modes that optimize readability and the ability to change the size and appearance of fonts. After discussing about above mentioned different scenarios and its usability in mobile applications other than native applications, and other than eBooks there are many other kinds of applications where we can use Flow Documents, especially where we need to implement pagination and zoom features together.

There are various options available to show the content in form of documents like Panel, Canvas etc. However, with the complexity of the code, using the Flow Document is best when the application requirements have been carefully considered, when support for all the platforms either for native application or mobile application using single code application which is the feature of responsive design.

## References:

[http://msdn.microsoft.com/en-us/library/aa970909%28v=vs.110%29.aspx#flow\\_document\\_types](http://msdn.microsoft.com/en-us/library/aa970909%28v=vs.110%29.aspx#flow_document_types)  
<http://msdn.microsoft.com/en-us/magazine/cc163371.aspx>  
<http://msdn.microsoft.com/en-us/library/system.windows.documents.FlowDocument%28v=vs.110%29.aspx>  
<https://www.dropbox.com/features>

**MSDN posted tool in this blog:** <http://blogs.msdn.com/b/wpfsdk/archive/2006/05/25/606317.aspx>