If you write your help content in Microsoft Word and save it as an XPS file, then yes, you'd be largely done in terms of preparing your document for display in a DocumentViewer control in WPF. Here’s how this would work:

**Steps:**

1. **Write and Format in Word**:
   * Create your help content in Microsoft Word, formatting it as you see fit with text, images, headings, etc.
2. **Save as XPS**:
   * In Word, go to File > Save As, choose the location, and then in the "Save as type" dropdown, select "XPS Document (\*.xps)". Save the file.
3. **Incorporate in Your WPF Application**:
   * Place the XPS file in your project or in an accessible location on the user's system.
4. **Display in WPF**:
   * Use the DocumentViewer control in your XAML:

xml

<DocumentViewer Name="helpViewer" />

* + And then in your code-behind or ViewModel, load the XPS file:

csharp

private void LoadHelpFile()

{

string xpsFilePath = "path\_to\_your\_xps\_file.xps"; *// This could be a relative path within your project or an absolute path*

if (File.Exists(xpsFilePath))

{

using (var xpsDocument = new XpsDocument(xpsFilePath, FileAccess.Read))

{

helpViewer.Document = xpsDocument.GetFixedDocumentSequence();

}

}

else

{

*// Handle file not found or error*

}

}

**Benefits:**

* **Simplicity**: This approach keeps things straightforward. You leverage Word's rich formatting capabilities and then convert to XPS, which is natively supported by DocumentViewer.
* **Fidelity**: The conversion from Word to XPS generally maintains high fidelity, preserving most of your formatting, including images and complex layouts.
* **No Additional Libraries**: You don't need additional libraries or controls for viewing since WPF's DocumentViewer can handle XPS files directly.

**Considerations:**

* **File Size**: XPS files can be larger than other document formats. This might be a concern for distribution or if you're dealing with bandwidth in online scenarios.
* **Editing**: If you need to update the help file frequently, consider the workflow of having to convert from Word to XPS each time.
* **Compatibility**: Ensure that all the features you use in Word translate well to XPS, as some advanced formatting might not be preserved perfectly.
* **Security**: Be cautious about embedding or linking to external files, ensuring no sensitive information is inadvertently included or accessible.
* **Performance**: For very large documents, loading and displaying might impact performance, especially if not done asynchronously or with some form of pagination/loading strategy.

This method provides a straightforward way to deliver help content to your users within your application using familiar tools like Microsoft Word for content creation. Just remember to test the XPS output to ensure it looks as intended when viewed through DocumentViewer.