Print Preview Architecture

Article • 08/03/2021 • 4 minutes to read

This article explains how the MFC framework implements print preview functionality. Topics covered include:

- Print preview process
- Modifying print preview

Print preview is somewhat different from screen display and printing because, instead of directly drawing an image on a device, the application must simulate the printer using the screen. To accommodate this, the Microsoft Foundation Class Library defines a special (undocumented) class derived from CDC Class, called `CPreviewDC`. All `CDC` objects contain two device contexts, but usually they are identical. In a `CPreviewDC` object, they are different: the first represents the printer being simulated, and the second represents the screen on which output is actually displayed.

The Print Preview Process

When the user selects the Print Preview command from the **File** menu, the framework creates a `CPreviewDC` object. Whenever your application performs an operation that sets a characteristic of the printer device context, the framework also performs a similar operation on the screen device context. For example, if your application selects a font for printing, the framework selects a font for screen display that simulates the printer font. Whenever your application would send output to the printer, the framework instead sends the output to the screen.

Print preview also differs from printing in the order that each draws the pages of a document. During printing, the framework continues a print loop until a certain range of pages has been rendered. During print preview, one or two pages are displayed at any time, and then the application waits; no further pages are displayed until the user responds. During print preview, the application must also respond to WM_PAINT messages, just as it does during ordinary screen display.

The CView::OnPreparePrinting function is called when preview mode is invoked, just as it is at the beginning of a print job. The CPrintInfo Structure structure passed to the

function contains several members whose values you can set to adjust certain characteristics of the print preview operation. For example, you can set the *m_nNumPreviewPages* member to specify whether you want to preview the document in one-page or two-page mode.

Modifying Print Preview

You can modify the behavior and appearance of print preview in a number of ways rather easily. For example, you can, among other things:

- Cause the print preview window to display a scroll bar for easy access to any page of the document.
- Cause print preview to maintain the user's position in the document by beginning its display at the current page.
- Cause different initialization to be performed for print preview and printing.
- Cause print preview to display page numbers in your own formats.

If you know how long the document is and call `SetMaxPage` with the appropriate value, the framework can use this information in preview mode as well as during printing. Once the framework knows the length of the document, it can provide the preview window with a scroll bar, allowing the user to page back and forth through the document in preview mode. If you haven't set the length of the document, the framework cannot position the scroll box to indicate the current position, so the framework doesn't add a scroll bar. In this case, the user must use the Next Page and Previous Page buttons on the preview window's control bar to page through the document.

For print preview, you may find it useful to assign a value to the *m_nCurPage* member of `CPrintInfo`, even though you would never do so for ordinary printing. During ordinary printing, this member carries information from the framework to your view class. This is how the framework tells the view which page should be printed.

By contrast, when print preview mode is started, the *m_nCurPage* member carries information in the opposite direction: from the view to the framework. The framework uses the value of this member to determine which page should be previewed first. The default value of this member is 1, so the first page of the document is displayed initially.

You can override `OnPreparePrinting` to set this member to the number of the page being viewed at the time the Print Preview command was invoked. This way, the application maintains the user's current position when moving from normal display mode to print preview mode.

Sometimes you may want `OnPreparePrinting` to perform different initialization depending on whether it is called for a print job or for print preview. You can determine this by examining the *m_bPreview* member variable in the `CPrintInfo` structure. This member is set to **TRUE** when print preview is invoked.

The `CPrintInfo` structure also contains a member named *m_strPageDesc*, which is used to format the strings displayed at the bottom of the screen in single-page and multiple-page modes. By default these strings are of the form "Page *n*" and "Pages *n* - *m*," but you can modify *m_strPageDesc* from within `OnPreparePrinting` and set the strings to something more elaborate. See CPrintInfo Structure in the *MFC Reference* for more information.

See also

Printing and Print Preview
Printing
CView Class
CDC Class