Printing in WASTE 2

What's New

WASTE 1.3 doesn't have any special support for printing. This doesn't mean you can't implement printing with the available APIs, of course, but there are several gotchas you have to watch out for. Printing text with WASTE involves calculating the number of lines that fit in each page, clipping the view rectangle to an integral number of lines, switching grafports, disabling offscreen drawing, etc.

WASTE 2 (starting from revision 2.0a7) introduces some new APIs that are meant to streamline printing code in client applications, especially in applications that have simple printing requirements. These new APIs don't interact directly with the Print Manager, so the print loop is still under your application's control. They are mainly concerned with calculating page boundaries according to a page rectangle you provide. One advantage of this approach is that these APIs work with both the "classic" Mac OS Print Manager and with the new Print Manager in Carbon.

The Print Session Object

The new APIs revolve around the concept of **print session**. To print a WASTE document, you create a print session object based on your print record (page format in Carbon). In your page loop, you call `WEPrintPage` repeatedly on the session object. You get rid of the print session when you're finished.

The line and page layout information maintained by the print session object are relative to the page rectangle you specify, and independent of the line layout information maintained by the main WE instance. So, for example, you can service update events on the screen in the middle of your print loop.

The APIs

WENewPrintSession

```
pascal OSErr WENewPrintSession
  (
    const WEPrintOptions *     inPrintOptions,
    WEReference                inWE,
    WEPrintSession *           outPrintSession
  );
```

This routine creates a new print session object for printing a given WE instance. The `inPrintOptions` parameter points to a block that specifies printing options: currently, the only meaningful field in this block is the **page rectangle**, which represents the printable area in integral points at the standard 72 dpi resolution. Nonetheless, you should zero out the whole block for future compatibility.

`WENewPrintSession` calculates line breaks according to the specified page width: these line breaks are kept in a data structure owned by the session object and don't interfere with the existing line layout information. Once line breaks have been calculated, `WENewPrintSession` goes on to find page breaks. Page breaks are calculated so that lines don't get split across pages.

WECountPages

```
pascal SInt32 WECountPages
  (
    WEPrintSession              inPrintSession
  );
```

This routine returns the number of pages needed to print the whole text according to the layout information associated with the print session.

WEPrintPage

```
pascal OSErr WEPrintPage
  (
    SInt32                      inPageIndex,
    GrafPtr                     inPrintPort,
    const Rect *                inPageRect,
    WEPrintSession              inPrintSession
  );
```

This routine prints a page. It should be called between the Print Manager calls `PrOpenPage` and `PrClosePage` (or their Carbon counterparts `PMBeginPage` and `PMEndPage`). The `inPageIndex` parameter must be in the range `0` to `WECountPages(inPrintSession) - 1`. You pass the printing graphics port (obtained from `PrOpenDoc` in the classic Print Manager or `PMGetGrafPtr` in Carbon) in the `inPrintPort` parameter.

The `inPageRect` parameter is optional: if you don't specify a rectangle (and pass `nil`), `WEPrintPage` will use the page rectangle originally passed to `WENewPrintSession`. If you do specify a different page rectangle, it must be the same width as the page rectangle originally passed to `WENewPrintSession`. A possible use of this parameter is to swap left/right (or top/bottom) margins for even/odd pages.

WEDisposePrintSession

```
pascal OSErr WEDisposePrintSession
  (
    WEPrintSession              inPrintSession
  );
```

Call this routine when you're finished with a print session.