

TDftBitsBlk structure

```
#include <PrintTraps.h>
```

typedef struct TDftBitsBlk {		<u>Size</u>	<u>Offset</u>	<u>Description</u>
<u>short</u> iOpCode;		2	0	6= opcode for DraftBits
<u>short</u> iError;		2	2	0=noErr; 1=noSuchRsl; 2=opNotImpl
<u>long</u> IReserved;		4	4	(not used)
THPrint hPrint;		4	8	<u>Handle</u> to a valid print record
} TDftBitsBlk ;		12		

Notes: **TDftBlk** defines a subsidiary record used by the **TGnlData** record in **PrGeneral** calls. The first 8 bytes are common for all **PrGeneral** calls and, in fact, constitute the **TGnlData** record. It forces draft-mode printing and allows screen dumps through a call to **CopyBits**. **DraftBits** printing is immediate, thereby avoiding print spooling and producing faster results. While it is implemented on the LaserWriter, the call does nothing. LaserWriters are always in draft mode and always capable of printing screen dumps.

HPrint is the handle to a **TPrint** structure that has already been passed by **PrValidate**.

Although speedy and versatile, certain caveats apply when you use **TDftBlk**. For example, make this call before bringing up dialog boxes as it affects their appearance. Also, on the ImageWriter, it disables the Landscape icon in the Style dialog (Landscape format is not supported) and the Best, Faster and Draft buttons in the Job dialog.

TDftBlk will do nothing if the printer doesn't have a draft mode, already prints screen dumps in draft mode, or doesn't print them at all.

You can't use it to print anything but text and screen dumps and everything on the page has to be oriented along the Y-axis (meaning that you can't go back over an area, you can't have two objects next to each other, and the top of one object can't intrude into the bottom of the preceding object).