Crystal Reports Technical Document

Subject: Bitmaps in CRW report files
Date: August 24, 1993
Versions: 1.0, 2.0(Std), 2.0(Pro), VB

Situation:
A bitmap value and its presentation options (scaling, cropping etc.) are stored separately in the Crystal Reports report file.
This technote describes the bitmap value only. This value is the same regardless of what presentation options are set elsewhere in the report.
If a bitmap value is replaced in the report (by some external 'patch' program) it is important that the two bitmaps have the same original size (e.g. 3.5 x 4.0 inches)

Solution:
A bitmap value is written out as an uncompressed Windows Device-Independent Bitmap (DIB).
A bitmap value is stored as a series of TLV (type-length-value) records (see other documentation for TLV specification). There are three TLV Types required to describe a bitmap:

Header = 4124 decimal
FileName = 4125
Body = 4126

There is always exactly one Header record written first, with the following value:

```
BitmapHeader
{
   DWORD imageSize;      // 32-bit unsigned
   DWORD maxBlockSize;
   int nBlocks;          // 16-bit signed
};
```

The imageSize indicates the exact number of bytes required to write the DIB, and it is calculated by calling the Windows function GlobalSize () with a pointer to the DIB in memory.

maxBlockSize is 32767 always.

nBlocks is always at least equal to 1. It is (imageSize / maxBlockSize) + 1. This allows for a final partial block, with size imageSize - ((nBlocks - 1) * maxBlockSize).

The second TLV record written is the file name of the original bitmap.
Note: This is for user info only in CRW, for example as a name at the top of the graphic pop-up menu. This file is never attempted to be reopened after the graphic has been saved in the report, and does not need to exist on the machine for the report to be used.

BITMAPS.DOC

Finally, exactly nBlocks Body records are written out containing the DIB value. These are written out directly from the DIB value in memory, from a locked handle to the Windows object. Please consult Windows documentation for the internal DIB structure.

First (nBlocks - 1) Body records are written out with size 32767, then one Body record is written out with the remaining bytes.