One vital piece of information is missing from the present .AL font format—the baseline of the font. The format is redefined to accommodate a 7-bit field "baseline," which is the distance (in bits) from the baseline to the top of the highest character. Baseline=0 means that you are reading a font file created before this format change, and you must either guess at the baseline or suffer.

The format change only involves the first two words of the font:

```
structure AL :
    Height word       //Height of font
    Proportional bit  //True if font is proportionally spaced
    Baseline bit 7    //Here is the new baseline info
    MaxWidth byte     //Maximum character width
```

Note: The Height entry is the honest truth about the font: if the user allocates Height scan-lines for a text line, no character will overflow either above or below the line of text. Note that characters with 0 words of bit-map (e.g., space) may exceed the Height number without trouble.

The MAXC directory <ALTOFONTS> will contain only .AL fonts with baselines correctly incorporated. This directory is at present empty; I will build it up in time. Matt Heiler's wonderful new fonts coordinated with EARS fonts will shortly begin to appear on <ALTOFONTS>.

<fonts> contains .AL files in old format
**Alto Font**

**Fontbase**

- Height
- Width
- \(\text{Ptr}_0\)
- \(\text{Ptr}_1\)
- \(\text{Ptr}_2\)
- ... 

**ExtPerm**

- \(\text{Ptr}_{0376}\)
- \(\text{Ptr}_{0377}\)
- ExtPerm
- ExtPerm

**Char. Desc. Block**

- Char. Desc. Block
- ... 

**ExtPerm**

- Char. Desc. Block
- ... 

**Structure DBL**

```c
typedef struct DBL
{
    WORD next_word;
    WORD resolution_bit_1;
    WORD background_bit_1;
    WORD indentation_bit_6;
    WORD width_bit_8;
    WORD bitMapAddress_word;
    WORD height_word;
} DBL;
```

- **Width** : If font is fixed pitch then font character width.
- else \(\pm 00000 \land \) widest character width.

- **Char. Width**: If char width < 16, \(2 \times \text{width}\) else \(2 \times \) pseudocharcode, where Fontbase + pseudocharcode point to the extension block of the character.

- **Line**
  - **Line 1** of bitmap
  - **Line 2** of bitmap
  - **Line 3** of bitmap
  - ... 
  - **Line \(X_H\) of bitmap**

- **HD** : Number of scan lines to skip before displaying bit map.
- **X_H** : Height of bit map.