

## **EDITBITMAP documentation**

D. Austin Henderson, Jr.

Initially released: April 10, 1983

Last revised: April 23, 1983

<**LISPUSERS**>**EDITBITMAP** (and **.DCOM**) provides an interface (**EDIT.BITMAP**) for manipulating bitmaps. It puts up a menu of bitmap manipulation commands, one of which is **HAND.EDIT** which accesses **EDITBM**, the Interlisp-D bitmap editor. Other commands include shifting (in four directions), rotation (left and right 90 degree), inverting (horizontally, vertically, about diagonals), interchanging black and white, adding a border. A sequence of commands terminated by **QUIT** produces a new bitmap; the original bitmap is not effected. **UNDO** permits backing up. Also works on cursors (produces new cursor) and **ATOMS** (works on their values and resets the value with the result). Comments on both interface and functionality are welcomed.

**EDIT.BITMAP (BITMAP)** **BITMAP** may be a bitmap, a cursor or an **ATOM**. Returns: if **BITAMP** is a bitmap, then a new bitmap; if a cursor, then a new cursor; if an **ATOM**, then the **ATOM** whose bvalue will have been reset with the result of calling **EDIT.BITMAP** on the value of that atom.

The internal functions used by this package are also available for use. They are:

**ADD.BORDER.TO.BITMAP (BITMAP NBITS TEXTURE)** Returns a new bitmap which is **BITMAP** extended by **NBITS** in all four directions, the border being filled in with **TEXTURE**.

**BIT.IN.COLUMN (BITMAP COLUMN)** Returns **T** if any bit in column numbered **COLUMN** (left = 0) is not 0, **NIL** otherwise.

**BIT.IN.ROW (BITMAP ROW)** Returns **T** if any bit in row numbered **ROW** (bottom = 0) is not 0, **NIL** otherwise.

**INVERT.BITMAP.B/W (BITMAP)** Returns a new bitmap which is **BITMAP** with all its bits inverted (black for white).

**INVERT.BITMAP.DIAGONALLY (BITMAP)** Returns a new bitmap which is **BITMAP** flipped about **X=Y** diagonal. (The resulting bitmap's width will be **BITMAP**'s height.)

**INVERT.BITMAP.HORIZONTALLY (BITMAP)** Returns a new bitmap which is **BITMAP** flipped about its vertical centerline.

**INVERT.BITMAP.VERTICALLY (BITMAP)** Returns a new bitmap which is **BITMAP** flipped about its horizontal centerline.

**ROTATE.BITMAP.LEFT (BITMAP)** Returns a new bitmap which is BITMAP rotated 90 degrees counter-clockwise. (The resulting bitmap's width will be BITMAP's height.).

**ROTATE.BITMAP.RIGHT (BITMAP)** Returns a new bitmap which is BITMAP rotated 90 degrees clockwise. (The resulting bitmap's width will be BITMAP's height.).

**SHIFT.BITMAP.DOWN (BITMAP NBITS)** Returns a new bitmap which is BITMAP extended by NBITS in the upward direction, the new space being filled in with white.

**SHIFT.BITMAP.UP (BITMAP NBITS)** Returns a new bitmap which is BITMAP extended by NBITS in the downwards direction, the new space being filled in with white.

**SHIFT.BITMAP.LEFT (BITMAP NBITS)** Returns a new bitmap which is BITMAP extended by NBITS to the right, the new space being filled in with white.

**SHIFT.BITMAP.RIGHT (BITMAP NBITS)** Returns a new bitmap which is BITMAP extended by NBITS to the left, the new space being filled in with white.

**TRIM.BITMAP (BITMAP)** Returns a new bitmap which is BITMAP trimmed at all four edges of all completely white (0) columns and rows.