

---

---

**CANVASCONVERTER**

---

---

By: Stephen Knowles (Stephen Knowles:49/89/636/13:Siemens AG)

Partly based on work by:

Matthias Schneider-Hufschmidt (Matthias Schneider-Hufschmidt:ZTISOF:SIEMENS)

Giselbert Schramm (Giselbert Schramm:ZTISOF:SIEMENS)

Uses: BITMAPFNS

This document last edited on 19-Sep-1988 13:32:21.

**INTRODUCTION**

This module enables the transfer of bitmaps between the Envos Lisp and Xerox ViewPoint environments. The medium used for the transfer is an NS file server (i.e. a file drawer which can be accessed by both environments). The possibility of transferring Lisp bitmaps into the ViewPoint environment is particularly useful for documenting Lisp applications.

**MODULE EXPLANATIONS**

There are essentially two major functions:

(IL:WRITECANVAS *BITMAP FILE*) [Function]

This function writes the *BITMAP* on to *FILE* and makes *FILE* of type ViewPoint Canvas, whereby *FILE* must be on an NS file server.

(IL:FETCHCANVAS *FILE*) [Function]

This function reads *FILE* into a Lisp bitmap, whereby *FILE* must be on an NS file server.

Additionally there are two auxiliary functions to aid in the use of the above two functions.

(IL:SNAPBM) [Function]

and

(IL:CANVAS-FROM-WINDOW *WINDOW FILE*) [Function]

**EXAMPLES**

All examples must be typed into an INTERLISP exec.

To write a canvas of a Lisp screen region:

(WRITECANVAS (SNAPBM)

```
' {NSfileServer:Domain:Organization}<FileDrawer>Folder>TESTFILE)
```

To write a canvas of a Lisp window:

```
(CANVAS-FROM-WINDOW (WHICHW)
```

```
' {NSfileServer:Domain:Organization}<FileDrawer>Folder>TESTFILE)
```

To read a canvas into a Lisp bitmap:

```
(SETQ X (OPENSTREAM
```

```
' {NSfileServer:Domain:Organization}<FileDrawer>Folder>TESTCANVAS ' INPUT))
```

```
(EDITBM (SETQ LISPBITMAP (FETCHCANVAS X)))
```

```
(CLOSEF X)
```

### CAVEAT

When fetching a canvas, there is a 50-50 chance that the Lisp bitmap will be O.K. It could, however, come out distorted (this is due to the differing ways in which ViewPoint and Lisp handle bitmaps, Lisp uses 16 complement, ViewPoint 32 complement - or something like that). If this should be the case, simply increase the canvas width in ViewPoint by 5 millimeters (approx. 16 pixels) and repeat the fetching process.

Unfortunately in the Lyric version if one repeatedly wrote a canvas with the same name, the file server somehow got mixed up and set the file-info of the folder above the canvas into "type = canvas"! One could put this right with the (SETFILEINFO...) function in Lisp, although under normal circumstances one does not write out a canvas repeatedly with the same name any way. I have been unable to test the behaviour in MEDLEY.

Compatibility has only been tested up to ViewPoint 1.1.