

---

---

**DEBUGGER-CONTEXT**

---

---

By: Herb Jellinek (Jellinek.pa@xerox.com)

This document last edited on August 13, 1987.

## Introduction

When debugging Common Lisp programs, have you ever wished that the Xerox Lisp debugger let you do things in the right lexical context? Ever wish you could access locally-defined functions, return from blocks, or evaluate variables in the debugger without resorting to the inspector? Ever wish you could tell the "big boys" in Washington a thing or two? Well, two out of three ain't bad. `DEBUGGER-CONTEXT` gives you the ability to access and modify the lexical context of code you are debugging in a straightforward way.

## User Interface

`DEBUGGER-CONTEXT` defines two debugger commands that affect the debugger's lexical context.

`lex` [Debugger Command]

Sets the lexical context of the debugger window to that of the stack frame selected in the backtrace window (`il:lastpos`). Henceforth, all evaluation done in this debugger window will be with respect to that lexical environment. The lexical environment of a frame is taken to be the first one encountered as an argument in that frame. Once the `lex` command is given, selecting other stack frames in the backtrace window will not change the lexical environment; you must issue another `lex` command to do so.

The lexical environment of a frame is taken to be the first one encountered as an argument in that frame.

`unlex` [Debugger Command]

Removes the current lexical context; all evaluation will subsequently be done with respect to the current dynamic environment.