



# Tech Info Library

## PostScript: Language change incorporates packed arrays

Revised: 9/18/87  
Security: Everyone

PostScript: Language change incorporates packed arrays

=====

This article last reviewed: 4 September 1987

Several additions have been made to the standard PostScript language. These additions are upward-compatible and do not affect the function of any existing PostScript programs. The changes are included in the new LaserWriter, LaserWriter Plus, and other PostScript printers; they will be documented in a future edition of the PostScript Language Reference Manual.

### -- General comments

PostScript programs that are intended to be compatible with all PostScript printers should not make use of the new features. However, it is possible for a program to determine whether or not the new features are present and to invoke them conditionally. The descriptions below suggest how to determine whether a particular feature is present or not.

### -- Packed array representation

PostScript procedures are represented as executable arrays which, until now, have been stored in the same fashion as literal data. The representation, while offering maximum flexibility, is very costly in space (8 bytes per element). Large PostScript programs, such as the built-in server program, and downloaded preambles, consume considerable amounts of Virtual Memory.

Since most programs do not require the ability to be treated as data but only the ability to be executed, a more compact representation has been introduced: the packed array. Programs represented by packed arrays are typically 50 to 75 percent smaller than the same programs represented as ordinary arrays.

The packed array object has a type different from an ordinary array object ('packedarraytype' versus arraytype'); but in most respects it behaves the same as an ordinary array.

The differences between a packed array and an ordinary array are:

1. Packed arrays are always read-only: you can't use 'put', 'putinterval', etc. to store into a packed array.
2. Packed arrays are created differently from ordinary array (see below).
3. Accessing arbitrary elements of a packed array can be quite slow; however, accessing the elements sequentially (as is done for the PostScript interpreter and by the 'forall' operator) is approximately as efficient as accessing an ordinary array.
4. The 'copy' operator cannot copy into a packed array since it is read-only; however, it can copy the value of a packed array to an ordinary array of at least the packed array's length.

To create packed arrays, you may employ two methods, the PostScript input scanner and the operator 'packedarray'. More commonly, the PostScript input scanner creates packed arrays automatically for all executable arrays that it reads. That is, whenever the scanner encounters a "{" while reading a file or string, the scanner accumulates all tokens up to the matching "}" and turns them into a packed array instead of an ordinary array. To build a packed array explicitly, invoke the 'packedarray' operator with a list of operands to be incorporated into the new packed array.