



# The StemSnap Hint Operator for Type 1 Font Programs

---

*Adobe Developer Support*

Technical Note #5049

31 March 1992

## Adobe Systems Incorporated

Corporate Headquarters  
1585 Charleston Road PO Box 7900  
Mountain View, CA 94039-7900  
(415) 961-4400 Main Number  
(415) 961-4111 Developer Support  
Fax: (415) 961-3769

Adobe Systems Europe B.V.  
Europlaza  
Hoogoorddreef 54a  
1101 BE Amsterdam Z-O, Netherlands  
+31-20-6511 200  
Fax: +31-20-6511 300

Adobe Systems Eastern Region  
24 New England  
Executive Park  
Burlington, MA 01803  
(617) 273-2120  
Fax: (617) 273-2336

Adobe Systems Japan  
Swiss Bank House 7F  
4-1-8 Toranomon, Minato-ku  
Tokyo 105, Japan  
81-3-3437-8950  
Fax: 81-3-3437-8968

Copyright ©1991-1992 by Adobe Systems Incorporated. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the publisher. Any software referred to herein is furnished under license and may only be used or copied in accordance with the terms of such license.

PostScript, the PostScript logo, the Adobe logo, and Adobe Type Manager are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions. Other brand or product names are the trademarks or registered trademarks of their respective holders.

*This publication and the information herein is furnished AS IS, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes and noninfringement of third party rights.*



# Contents

---

## **The StemSnap Hint Operator for Type 1 Font Programs 1**

- 1 Introduction 1
- 2 The StemSnap(H,V) Operators 2
  - Determining Printer Capabilities 2
  - Creating Device-Independent Font Programs Using StemSnap(H,V) 3

## **Appendix: Changes Since Earlier Versions 5**

## **Index 7**



# The StemSnap Hint Operator for Type 1 Font Programs

---

## 1 Introduction

This document discusses the **StemSnapH** and **StemSnapV** hint operators (hereafter referred to as the **StemSnap(H,V)** operators) in the Type 1 font format.

The **StemSnap(H,V)** commands were designed to cause stems of similar widths to be rendered at the same pixel width when their unrounded pixel widths differ by less than a threshold distance. This has the advantage of allowing fonts with slight differences in stem widths to be converted to a consistent width for smaller sizes, yet allow their true widths to be rendered when imaged for higher resolutions.

There are three ways PostScript™ interpreters handle the **StemSnap(H,V)** hint operator:

- Early versions of PostScript interpreters ignore this operator.
- Some recent versions of PostScript interpreters (in laser printers only) allow only two values for the array. This excludes all imagesetters.
- Newer printers, imagesetters, and all versions of Adobe Type Manager™ software accept 12 values for this array. Interpreters that can accept 12 values contain a new keyword **StemSnapLength**, which has a value of 12.

The purpose of this document is to explain how font vendors can create device-independent font programs that can best utilize the functionality of the device on which they are used.

## 2 The StemSnap(H,V) Operators

The **StemSnap(H,V)** operators were designed to cause all character stems whose widths fall within an interpreter-supplied pixel tolerance to be rendered at the same pixel width. This has the benefit of facilitating the production and improving the quality of fonts with inconsistencies due to digitization errors. It also allows designers to create Type 1 fonts with subtleties that would otherwise not be rendered optimally at low resolutions.

For example, if the mean width of a font's upper case vertical stems was 122 character-space-units, and if lower case vertical stems averaged 114 units, the following operator could be included in the font program:

```
StemSnapV [114 122]
```

If the two vertical stems of the H measured 121 and 124 units, these stems would snap to the pixel-width of the specified 122 unit stem if they were within the pixel tolerance allowed by the interpreter.

Although few fonts require 12 values for the **StemSnap(H,V)** array, many fonts benefit from having, for example, 3 to 6 values. These values are typically derived from the mean values for upper and lower case vertical stems as well as from other groups such as the figures, superiors, symbols, and math characters. The most likely need for more than 3 to 6 values is a font with a set of symbols that has significantly different widths than those of a standard character set.

It is important that only the mean value of groups of stems be entered in the array. Entering values that are too close together, such as [ 121, 122, ... 172, 174...] might produce undesirable results. Hence, it is recommended that values be a minimum of five units apart.

### 2.1 Determining Printer Capabilities

During development, font developers can query a printer's ability to accept 12 values for the **StemSnap(H,V)** array by downloading the following PostScript language code:

```
%!  
/Helvetica findfont 24 scalefont setfont  
100 600 moveto  
(StemSnapLength: ) show  
systemdict /internaldict known  
  { 1183615869 systemdict /internaldict get exec  
    /StemSnapLength known  
  {(Known) show } { (Not known) show } ifelse }  
{(Not known) show} ifelse  
showpage
```

If the printer can accept 12 values, this procedure prints the message “StemSnapLength: Known.” If the keyword is not found, it means that the printer will accept at most two values. Earlier versions of the PostScript interpreter will not be capable of interpreting the **StemSnap(H,V)** operator, in which case the hint operator will be ignored.

## 2.2 Creating Device-Independent Font Programs Using StemSnap(H,V)

The goal is to create smart font programs that can query the printer for the existence of the **StemSnapLength** keyword. If this entry is missing, the code in the font installs a **StemSnap(H,V)** array of only two elements. If this font program is installed in a printer that does not recognize the **StemSnap(H,V)** operator, this hint is ignored. If **StemSnapLength** is present, the full **StemSnap(H,V)** array is implemented.

For each **StemSnap(H,V)** array in the font file that has more than two elements, replace the accompanying **def** operator with the sequence

```
systemdict /internaldict known
{ 1183615869 systemdict /internaldict get exec
/StemSnapLength 2 copy known { get XXXX lt } { pop pop true } ifelse
}
{ true } ifelse {pop [YYYY ZZZZ]} if def
```

where

XXXX is the length of the **StemSnap(H,V)** array for this font program.

YYYY and ZZZZ are the two elements of the **StemSnap(H,V)** array to be used if the printer cannot accept 12 values. One of these should be the same as the corresponding **StdVW** or **StdHW** array entry.

Adobe Type Manager software ignores the additional PostScript language code and uses only the initial statement of values, which can be up to 12 values.

The installation of a **StemSnap(H,V)** array with two values slightly limits the effectiveness of the hint operator on those printers. However, even two values achieve a significant improvement in quality, and the full effect of the hint will be realized on the majority of new printers.

It is recommended that font-creation software allow users to specify more than two values for the **StemSnap(H,V)** array, but should also allow them to specify the two values considered the most important. This is usually the mean values for upper and lower case stems for latin typefaces. The program should then insert the above code into the font program for each array.

A font might need to be converted to the above device-independent form, but the translating software does not have the information to know which two values of the **StemSnap(H,V)** array to insert in the code. In this situation, it is recommended that the single value in the appropriate **StdVW** or **StdHW** array be inserted for the YYYY argument shown above, and omit the second value (ZZZZ). This means using only the most significant stem value for printers that cannot accept 12 values, but the **StemSnap(H,V)** hint will be fully functional for printers that do accept 12 values.

**Example:** *Converting to a device-independent font program*

If the original entries in a font program are

```
/StemSnapH [ 74 107 117 132 158 ] def
/StemSnapV [ 85 130 172 199 ] def
```

then to work on all printers, these lines should be replaced by the following code:

```
/StemSnapH [ 74 107 117 132 158 ]
  systemdict /internaldict known
  { 1183615869 systemdict /internaldict get exec
    /StemSnapLength 2 copy known { get 5 lt } { pop pop true } ifelse
  }
  { true } ifelse {pop [107 132]} if def
/StemSnapV [ 85 130 172 199 ]
  systemdict /internaldict known
  { 1183615869 systemdict /internaldict get exec
    /StemSnapLength 2 copy known { get 4 lt } { pop pop true } ifelse
  }
  { true } ifelse {pop [130 172]} if def
```





# Appendix: Changes Since Earlier Versions

---

## Changes since February 21, 1991 version

- Document was reformatted in the new document layout and minor editorial changes were made.





# Index

---

## S

- StemSnap(H,V)
  - creating device-independent font programs 3
  - example 4
- StemSnap(H,V)** 1–4
  - determining printer capabilities 2
- StemSnapLength 3

## T

- Type 1 font format
  - StemSnap(H,V) 1–4