
**Information technology — Font
information interchange —**

**Part 3:
Glyph shape representation**

**AMENDMENT 2: Additional shape
representation technology for Open Font
Format**

iTeh STANDARD PREVIEW
(standards.iteh.ai)

<https://standards.iteh.org/standards/sist/59214e82-02cc-4972-91ee-63a2c6a07981/iso-iec-9541-3-1994-amd-2-2009>
Technologies de l'information — Échange d'informations sur les
fontes —
Partie 3: Représentation de la forme de glyphes

AMENDEMENT 2: Technologie de représentation de forme
additionnelle pour format de fontes ouvert

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 9541-3:1994/Amd 2:2009](https://standards.iteh.ai/catalog/standards/sist/59214e82-02cc-4972-91ee-63a2c6a07981/iso-iec-9541-3-1994-amd-2-2009)

<https://standards.iteh.ai/catalog/standards/sist/59214e82-02cc-4972-91ee-63a2c6a07981/iso-iec-9541-3-1994-amd-2-2009>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2009

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 2 to ISO/IEC 9541-3:2008 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, *Document description and processing languages*.

This Amendment appends an additional glyph shape representation technology for the harmonization of ISO/IEC 9541-3 to Open Font Format (ISO/IEC 14496-22). In Open Font Format file, the scalable glyph shape representations by TrueType instructions or Adobe Type2 CharStrings are possible, but ISO/IEC 14496-22 does not include it. For the harmonization between ISO/IEC 9541 and ISO/IEC 14496-22, this Amendment covers Adobe Type2 CharStrings that is the half of the scalable glyph shape representation in ISO/IEC 14496-22. It is an extended version of ISO/IEC 9541-3 Type 1 glyph shape representation.

The additional interchange format is described by SGML (Standard Generalized Markup Language) conforming to ISO 8879:1986 and ISO 8879:1986/Cor.2:1999 (Annex K: Web SGML Adaptations).

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 9541-3:1994/Amd 2:2009

<https://standards.iteh.ai/catalog/standards/sist/59214e82-02cc-4972-91ee-63a2c6a07981/iso-iec-9541-3-1994-amd-2-2009>

Information technology — Font information interchange —

Part 3: Glyph shape representation

AMENDMENT 2: Additional shape representation technology for Open Font Format

Page 3, 1.7.2

Replace the declaration with the following:

```

<!-- (c) International Organization for Standardization 2004 Permission
to copy in any form is granted for use with conforming WebSGML
systems and applications as defined in ISO 8879:1986(WWW),
provided this notice is included in all copies. -->
<!-- Public document type definition. Typical invocation:
<!DOCTYPE gshapes PUBLIC "ISO 9541-3:1994 AM1:2004 and AM2:2008//DTD Glyph Shapes//EN">
-->
https://standards.iteh.ai/catalog/standards/sist/59214e82-02cc-4972-91ee-63a2c6a07981/iso-iec-9541-3-1994-amd-2-2009
<!-- GLYPHSHAPES -->
<!ELEMENT gshapes (t1shapes | t2shapes | ot3shapes | nprop)+ >

<!-- Type 1 shape information. Typical invocation:
<!DOCTYPE t1shapes PUBLIC "ISO 9541-3:1994//DTD Type 1 Glyph Shapes//EN">
-->

<!-- Type 2 shape information. Typical invocation:
<!DOCTYPE t2shapes PUBLIC "ISO 9541-3:1994 AM1:2004//DTD Type 2 Glyph Shapes//EN">
-->

<!-- Open Type 3 shape information. Typical invocation:
<!DOCTYPE ot3shapes PUBLIC "ISO 9541-3:1994 AM2:2008//DTD Open Type 3 Glyph Shapes//EN">
-->

```

Add the following after Section 3:

Section 4: Open Type 3 glyph shape representation

4.1 Scope

This section specifies the architecture and interchange format of one standard Glyph Shape Representation: ISO/IEC 9541 Standard OPEN TYPE 3. The Open Type 3 glyph shape representation is designed for the harmonization to the glyph shape representation used in “CFF” table in Open Font Format (ISO/IEC 14496-22). It is an extended version of ISO/IEC 9541-3 Type 1 glyph shape representation.

4.2 Definitions

This section uses the terms defined by 2.2. Extra terms in the following definitions are specific to this section.

4.3 Overview of Open Type 3 glyph shape representation architecture

4.3.1 Difference from Type 1 glyph shape representation

The original ISO/IEC 9541-3 Type 1 glyph shape representation used graphical drawing operators that take single set of operands. For example, relative lineto operator (`rlneto`, described in 2.7.3.2.5) takes 2 number object operands of `dx` and `dy` that specifies relative displacements. To compress the glyph procedures, the Open Type 3 glyph shape representation enhances Type 1 glyph procedure operators to take multiple set of operands. The relative lineto operator in the Open Type 3 glyph shape representation recognizes the operand list as an array of pairs of `dx` and `dy`. By this enhancement, the glyph procedure repeating `rlneto` operator can be compressed to multiple sets of relative displacements and single `rlneto` operator. By the operator to be interpreted and the length of the operand list, the glyph procedure interpreter dynamically determines how many sets of operands are taken from the tail of operand list.

4.3.2 Extended virtual machine

The interpretation of Open Type 3 glyph procedure is modelled by the virtual machine that is described in 2.7.1. To illustrate the interpretation of Open Type 3 glyph shape representation, “a transient array” to store any objects is introduced in state variables (described in 2.7.1.4). The Open Type 3 glyph procedure has no operators to allocate, free, initialize the transient array explicitly, it must be allocated dynamically or pre-allocated before the interpretation. The size of the transient array must at least 32 elements, although individual implementations may have longer array. As other state variables, the entries of the transient array are persistent only during the interpretation of each glyph procedures. The transient array has no default values. Therefore, it is possible that individual implementation resets all entries to number 0, or to random number, or keeps the objects stored in previous interpretation.

4.4 Open Type 3 glyph shape representation

By the comparison with original Type 1 glyph shape representation in the section 2, the Open Type 3 glyph procedure is classified into 4 groups.

- Original Type 1 glyph procedure operators that are not modified from the definition in section 2
- Enhanced Type 1 glyph procedure operators that the syntaxes are enhanced for Open Type 3
- Additional operators that are not defined in original Type 1 glyph shape representation
- Obsolete operators that described in section 2 but deprecated in Open Type 3

Some operators in the Open Type 3 glyph procedure take the multiple sets of the operands, and the number how many sets are taken is calculated dynamically from the length of the operand list when the operator is interpreted. For the description of such operators' syntax, following notation is used for enhanced operators. Other notations follow to the conventions defined in 2.7.3.

Table 3 —Symbols preceding glyph procedure notation for Open Type 3 glyph shape representation

operand-list notation	Meaning
{...}	indicates grouping for set of operands
?	takes zero or one operand or set of operands if available in the operand list.
+	takes the array of operand or set of operands as many as available from the operand list.

4.4.1 Unchanged Type 1 glyph procedure operators

The following operators are same as original Type 1 glyph procedures.

4.4.1.1 Unchanged operators for starting and finishing

4.4.1.1.1 Reference point and escapement (rpe)

This operator is same as that described in 2.7.3.1.1.

4.4.1.1.2 Horizontal reference point and escapement (xrpe)

This operator is same as that described in 2.7.3.1.2.

4.4.1.1.3 End glyph (endglyph)

This operator is same as that described in 2.7.3.1.3.

4.4.1.1.4 Standard indexed accented glyph (siag)

This operator is same as that described in 2.7.3.1.4.

4.4.1.2 Unchanged path construction operators

For the following graphical operators, the expected result by giving multiple sets of operands is identical to the result by the final set of operands only. The detailed behaviours of the operators are described in 2.7.3.

4.4.1.2.1 Closepath (closepath)

This operator is same as that described in 2.7.3.2.1.

4.4.1.2.2 Horizontal moveto (hmoveto)

This operator is same as that described in 2.7.3.2.3.

4.4.1.2.3 Relative moveto (rmoveto)

This operator is same as that described in 2.7.3.2.6.

4.4.1.2.4 Vertical moveto (vmoveto)

This operator is same as that described in 2.7.3.2.10.

4.4.1.2.5 Set current point (setcurrentpoint)

This operator is same as that described in 2.7.3.2.11.

4.4.1.3 Unchanged arithmetic operator

For the following arithmetic operator, giving multiple sets of operators is impossible because the result is stacked per each set.

4.4.1.3.1 Divide (div)

This operator is same as that described in 2.7.3.4.1.

4.4.1.4 Unchanged subroutine and subroutine-related operators

The following operators switch the sequence of glyph procedure tokens to be interpreted. Therefore it is impossible to take multiple sets of operands.

4.4.1.4.1 Call subroutine (callsubr)

This operator is same as that described in 2.7.3.5.1.

4.4.1.4.2 Return (return)

This operator is same as that described in 2.7.3.5.2.

4.4.1.4.3 Call utility subroutine (callutilsubr)

This operator is same as that described in 2.7.3.5.3.

ISO/IEC 9541-3:1994/Amd 2:2009
<https://standards.iteh.ai/catalog/standards/sist/59214e82-02cc-4972-91ee-63a2c6a07981/iso-iec-9541-3-1994-amd-2-2009>

4.4.2 Enhanced Type 1 glyph procedure operators

The following operators are enhanced to take multiple set of operands.

4.4.2.1 Enhanced path construction operators

The following operators are designed to draw a line or curve between the current point memorized by interpreter and the parameters given by the set of operands. The interpretation updates the current point in the interpreter. In the description of glyph outline, they are most frequently used. Collecting all control point and omitting similar operator can reduce the size of the procedure. In most enhancements, the operators are enhanced to draw the zig-zag kinked line.

4.4.2.1.1 Horizontal lineto (hlineto)

The original syntax of this operator in Type 1 glyph shape representation is described in 2.7.3.2.3. This operator in Open Type 3 glyph shape representation is interpreted by most appropriate syntax in following syntaxes:

- * dx1 dy2 ... dxN **hlineto** (00/06) *
- * dx1 dy2 ... dxN dy(N+1) **hlineto** (00/06) *

and appends the alternating horizontal and vertical line from the current point. The first line is horizontal and the second line is vertical (specified by only *dy1*).

4.4.2.1.2 Horizontal-vertical curveto (hvcurveto)

The original syntax of this operator in Type 1 glyph shape representation is described in 2.7.3.2.4. This operator in Open Type 3 glyph shape representation is interpreted by most appropriate syntax in following syntaxes:

- * dx1 dx2 dy2 dy3 dx3? **hvcurveto** (01/15) *
- * dx1 dx2 dy2 dy3 {dya dxb dyb dxc dxd dx e dye dyf}+ dx f? **hvcurveto** (01/15) *
- * {dxa dx b dx b dyc dyd dx e dye dx f}+ dy f? **hvcurveto** (01/15) *

and appends one or more Bézier curves to the current point. The specification of a Bézier curve from currentpoint requires 6 operands. They are $x_1, y_1, x_2, y_2, x_3, y_3$ in figure 6, x_0 and y_0 are given by currentpoint. In the first and second syntax of this operator, the tangent of beginning of first Bézier curve must be horizontal ($y_1=y_0$), and that of ending of first Bézier curve must be vertical ($x_3=x_2$). By this restriction, the number of operands for the first Bézier curve is reduced to 4 (dx_1, dx_2, dy_2, dy_3). The restriction of beginning and ending tangents are alternating. The second Bézier curve must start with vertical tangent and finish with horizontal tangent specified by 4 operands ($dya, dx b, dy b, dxc$), because the first Bézier curve finishes with vertical tangent. The ending tangent of the final Bézier curve is not restricted. In basic syntax, although the final Bézier curve is specified by 4 operands ($dxd, dx e, dye, dyf$ in the first and second syntax, or $dyd, dx e, dye, dx f$ in the third syntax), extra operand ($dx f$ in the first and second syntax, or dyf in the third syntax) makes sloping end of final Bézier curve. The standard order of 2 operands is x and y to specify a point, but the order of extra operand is non-standard y and x order in the first and second syntax.

4.4.2.1.3 Relative lineto (rlineto)

The original syntax of this operator in Type 1 glyph shape representation is described in 2.7.3.2.5.

- * {dxa dya}+ **rlineto** (00/05) *

This operator appends one or more lines to the current point. Each set of 2 operands is interpreted by original **rlineto** operator syntax.

4.4.2.1.4 Relative-relative lineto (rrcurveto)

The original syntax of this operator in Type 1 glyph shape representation is described in 2.7.3.2.7.

- * {dxa dya dx b dy b dxc dyc}+ **rrcurveto** (00/08) *

This operator in Open Type 3 glyph shape representation appends one or more Bézier curves to the current point. Each set of 6 operands is interpreted by original **rrcurveto** operator syntax described in 2.7.3.2.7.

4.4.2.1.5 Vertical-horizontal curveto (vhcurveto)

The original syntax of this operator in Type 1 glyph shape representation is described in 2.7.3.2.8. This operator in Open Type 3 glyph shape representation is interpreted by most appropriate syntax in following syntaxes:

- * dy1 dx2 dy2 dx3 dyf? **vhcurveto** (01/14) *
- * dy1 dx2 dy2 dx3 {dxa dx b dy b dyc dyd dx e dye dx f}+ dy f? **vhcurveto** (01/14) *
- * {dya dx b dy b dxc dx d dx e dye dy f}+ dx f? **vhcurveto** (01/14)

and appends one or more Bézier curves to the current point. The syntax is same as enhanced **hvcurveto** except x, y coordinates are exchanged for initial and final curves.