#  WOFF File Format 2.0

[Rekomendasyon ng W3C](https://www.w3.org/standards/types#REC) ,10 Marso 2022

**Ang bersyon na ito:**

<https://www.w3.org/TR/2022/REC-WOFF2-20220310/>

**Pinakabagong bersyon:**

<https://www.w3.org/TR/WOFF2/>

**Nakaraang Bersyon:**

<https://www.w3.org/TR/2021/REC-WOFF2-20210706/>

**Pinakabagong draft ng editor:**

<https://w3c.github.io/woff/woff2/>

**Kasaysayan:**

<https://www.w3.org/standards/history/WOFF2>

**Ulat sa Pagpapatupad:**

<https://www.w3.org/Fonts/WG/WOFF2/Implementation.html>

**Mga editor:**

Vladimir Levantovsky (Mga Inimbitahang Eksperto ng W3C)

Raph Levien (Google)

**Feedback:**

[Mga isyu sa WOFF at WOFF2](https://github.com/w3c/woff/issues)

**Errata:**

<https://www.w3.org/Fonts/REC-WOFF2-20220310-errata.html>

Pakisuri ang [errata](https://www.w3.org/Fonts/REC-WOFF2-20220310-errata.html) para sa anumang mga error o isyu na iniulat mula nang ma-publish.

[Copyright](https://www.w3.org/Consortium/Legal/ipr-notice#Copyright) © 2022 [W3C](https://www.w3.org/)® ( [MIT](https://www.csail.mit.edu/) , [ERCIM](https://www.ercim.eu/) , [Keio](https://www.keio.ac.jp/) , [Beihang](https://ev.buaa.edu.cn/) ). [Nalalapat ang pananagutan](https://www.w3.org/Consortium/Legal/ipr-notice#Legal_Disclaimer) ng W3C , [trademark](https://www.w3.org/Consortium/Legal/ipr-notice#W3C_Trademarks) at pinahihintulutang mga panuntunan sa [lisensya ng dokumento .](https://www.w3.org/Consortium/Legal/2015/copyright-software-and-document)

## **Abstract**

Batay sa karanasan sa WOFF 1.0, na malawakang na-deploy, ang detalyeng ito ay binuo upang magbigay ng pinahusay na compression at sa gayon ay mas mababang paggamit ng bandwidth ng network, habang pinapayagan pa rin ang mabilis na decompression kahit sa mga mobile device. Ito ay nakakamit sa pamamagitan ng pagsasama-sama ng isang content-aware na hakbang sa preprocessing at pinahusay na entropy coding, kumpara sa Flate compression na ginamit sa WOFF 1.0.

## **Katayuan ng Dokumentong Ito**

Inilalarawan ng seksyong ito ang katayuan ng dokumentong ito sa oras ng paglalathala nito. Ang isang listahan ng kasalukuyang mga publikasyong W3C at ang pinakabagong rebisyon ng teknikal na ulat na ito ay makikita sa [*W3C technical reports index*](https://www.w3.org/TR/) sa https://www.w3.org/TR/.

Ang dokumentong ito ay nai-publish ng [Web Fonts Working Group](https://www.w3.org/groups/wg/webfonts) bilang Rekomendasyon gamit ang Recommendation track. Kasama dito [ang mga iminungkahing pagwawasto](https://www.w3.org/2021/Process-20211102/#proposed-corrections) .

Inirerekomenda ng W3C ang malawak na pag-deploy ng detalyeng ito bilang pamantayan para sa Web.

Ang Rekomendasyon ng W3C ay isang detalye na, pagkatapos ng malawakang pagbuo ng pinagkasunduan, ay ineendorso ng W3C at ng mga Miyembro nito, at may mga pangako mula sa mga miyembro ng Working Group sa [walang royalty na paglilisensya](https://www.w3.org/Consortium/Patent-Policy/%22%20%5Cl%20%22sec-Requirements) para sa mga pagpapatupad.

Ang pagtutukoy ng WOFF 2.0 ay ipinatupad sa lahat ng pangunahing browser, at malawakang ginagamit sa mga website ng produksyon. Sinusuportahan nito ang kabuuan ng mga detalye ng TrueType at OpenType, kabilang ang mga Variable font, Chromatic font, at Mga Koleksyon ng font.

Isang kumpletong [listahan ng mga pagbabago](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22changes-since-2021) sa dokumentong ito mula noong available ang huling publikasyon.

Ang sumusuportang materyal, kabilang ang mga resulta ng mga sukat ng compression, ay maaaring matagpuan sa kasamang [WOFF 2.0 Evaluation Report](https://www.w3.org/TR/WOFF20ER/) . Ang Working Group ay [gumawa ng mga pagsubok](https://github.com/w3c/woff2-compiled-tests) para sa mga WOFF2 [encoder](https://www.w3.org/Fonts/WG/wiki/TestPlan20-AuthoringTool) , [decoder](https://www.w3.org/Fonts/WG/wiki/TestPlan20-Decoder) , [browser](https://www.w3.org/Fonts/WG/wiki/TestPlan20-UserAgent) , at para [sa mismong format](https://www.w3.org/Fonts/WG/wiki/TestPlan20-Format%22%20%5Co%20%22TestPlan20-Format) (isang WOFF2 validator). Ang source para bumuo ng mga pagsubok na ito ay [available sa GitHub](https://github.com/w3c/woff2-tests) .

Available ang [Implementation Report](https://www.w3.org/Fonts/WG/WOFF2/Implementation.html) . Tandaan na, para sa OpenType Collections (dati, TrueType Collections) habang may dalawang nasubok na pagpapatupad ng pag-encode at ng pag-decode, wala pang user agent na ipinapakita na gagamitin ang mga ito pagkatapos ng pag-decode. Ang suporta sa browser para sa mga koleksyon ng font ay inaasahang bubuti, at kapag nangyari na ang WOFF 2.0 ay mayroon nang kinakailangang suporta.

Mangyaring magpadala ng mga komento tungkol sa dokumentong ito bilang mga [isyu sa GitHub](https://github.com/w3c/woff/issues) . Ang mga komento ay dapat bayaran ni10 Mayo 2022.

Ang dokumentong ito ay ginawa ng isang pangkat na tumatakbo sa ilalim ng [W3C Patent Policy](https://www.w3.org/Consortium/Patent-Policy-20200915/) . Ang W3C ay nagpapanatili ng isang [pampublikong listahan ng anumang mga pagsisiwalat ng patent na](https://www.w3.org/2004/01/pp-impl/44556/status) ginawa kaugnay ng mga naihatid ng grupo; Kasama rin sa page na iyon ang mga tagubilin para sa pagsisiwalat ng patent. Ang isang indibidwal na may aktwal na kaalaman sa isang patent na pinaniniwalaan ng indibidwal na naglalaman ng [(mga) Mahahalagang Claim](https://www.w3.org/Consortium/Patent-Policy/#def-essential) ay dapat ibunyag ang impormasyon alinsunod sa [seksyon 6 ng Patakaran sa Patent ng W3C](https://www.w3.org/Consortium/Patent-Policy/%22%20%5Cl%20%22sec-Disclosure) .

Para sa Brotli compression scheme na ginamit sa WOFF 2.0, gumawa ang Google ng [Royalty Free licensing commitment](https://www.w3.org/standards/licensing/#3) para magamit sa detalyeng ito.

Para sa WOFF 2.0, ang Monotype Imaging ay gumawa ng [Royalty Free licensing commitment](https://www.w3.org/standards/licensing/#2) para magamit sa detalyeng ito.

Ang dokumentong ito ay pinamamahalaan ng [2 Nobyembre 2021 W3C Process Document](https://www.w3.org/2021/Process-20211102/) .

Ang mga iminungkahing pagwawasto ay minarkahan sa dokumento.

## **1. Panimula**

Tinutukoy ng dokumentong ito ang format ng packaging ng WOFF2 ng font. Ang format na ito ay idinisenyo upang magbigay ng isang makatwirang madaling ipatupad na compression ng data ng font na may makabuluhang mas mahusay na compression kaysa sa mga nakaraang diskarte, na angkop para sa paggamit sa CSS @font-face na mga panuntunan. Ang pagpapabuti sa mga rate ng compression, kumpara sa dating binuong WOFF 1.0 na format [ [*WOFF1*](https://www.w3.org/TR/WOFF2/#ref-woff10) ] ay natanto dahil sa pinahusay na entropy coding at hakbang sa preprocessing at pag-optimize ng data ng font na binabawasan ang built-in na redundancy ng iba't ibang istruktura ng data ng font. Ang mga detalye tungkol sa kasaysayan ng pag-unlad ng WOFF 2.0 ay matatagpuan sa WOFF 2.0 Evaluation Report [ [*WOFF2ER*](https://www.w3.org/TR/WOFF2/#ref-woff20er) ].

### **1.1 Notational Convention**

Ang lahat-ng-uppercase na susing salita na "DAPAT", "HINDI DAPAT", "KINAKAILANGANG", "HINDI DAPAT", "HINDI DAPAT", "HINDI DAPAT", "INIREREKOMENDASYON", "MAY" at "OPTIONAL" sa ang dokumentong ito ay dapat bigyang-kahulugan gaya ng inilarawan sa RFC 2119 [ [*RFC2119*](https://www.w3.org/TR/WOFF2/#ref-RFC-2119) ]. Kung ang mga salitang ito ay nangyayari sa maliit o halo-halong titik, dapat silang bigyang-kahulugan alinsunod sa kanilang normal na kahulugan sa Ingles.

Kasama sa dokumentong ito ang mga seksyon ng teksto na tinatawag na "Mga Tala" at itinatakda mula sa pangunahing teksto ng detalye. Ang mga talang ito ay nilayon bilang mga paliwanag o paglilinaw ng impormasyon, upang magsilbing mga pahiwatig o gabay sa mga nagpapatupad at gumagamit, ngunit hindi bahagi ng normatibong teksto.

Gumagamit din ang dokumentong ito ng hexadecimal numerical notation na "0x..." para tukuyin ang mga pare-parehong binary value gaya ng magic number o tag values, bitfields / flags format at mask value na ginagamit sa pseudo-code bitwise operator.

## **2. Pangkalahatang Pangangailangan**

Ang pangunahing layunin ng format na WOFF2 ay ang mahusay na pakete ng mga font na naka-link sa mga dokumento sa Web sa pamamagitan ng CSS @font-face rules. Ang mga ahente ng gumagamit na sumusuporta sa format ng WOFF2 file para sa mga naka-link na font ay dapat igalang ang mga kinakailangan ng detalye ng CSS3 Fonts ([ [*CSS3-Fonts*](https://www.w3.org/TR/WOFF2/#ref-CSS3-Fonts) ] [Seksyon 4.1: Ang @font-face rule](https://www.w3.org/TR/css-fonts-3/%22%20%5Cl%20%22font-face-rule) ). Sa partikular, ang mga naka-link na font ay magagamit lamang sa mga dokumentong tumutukoy sa kanila; HINDI sila DAPAT gawing available sa ibang mga application o dokumento sa system ng user .

Ang input ng font file ay maaaring maglaman ng maraming iba't ibang mga talahanayan ng data ng font na inilarawan sa sugnay 5 ng detalye ng [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ]. Ang pagkakasunud-sunod ng mga talahanayan sa isang font file ay maaaring mag-iba at, habang ang mga entry sa direktoryo ng talahanayan ay kailangang ayusin sa pataas na pagkakasunud-sunod ayon sa mga tag, ang aktwal na mga talahanayan ay maaaring ipakita sa isang font file sa arbitrary na pagkakasunud-sunod.

Kapag na-decompress ang WOFF2 file, dapat ayusin ng decoder ang mga entry sa direktoryo ng talahanayan sa pataas na alpabetikong pagkakasunud-sunod ng mga tag ng talahanayan at dapat ayusin ang pagkakasunud-sunod ng talahanayan sa isang font file ayon sa Mga Rekomendasyon (subclause 8.1 "Optimized na pag-order ng talahanayan") ng [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ].

## **3. Pangkalahatang istraktura ng file at mga pangunahing uri ng data**

Ang istraktura ng mga WOFF2 file ay katulad ng sa SFNT at WOFF 1.0 na mga font file, dahil mayroong isang header na sinusundan ng isang direktoryo ng talahanayan, na sinusundan ng data para sa mga talahanayang iyon. Ang istraktura ng SFNT ay ganap na inilalarawan sa TrueType [ [*TrueType*](https://www.w3.org/TR/WOFF2/#ref-TTF) ], OpenType [ [*OpenType*](https://www.w3.org/TR/WOFF2/#ref-OTF) ], at ISO "Open Font Format" [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF)] mga pagtutukoy. Gayunpaman, ang WOFF2 ay naiiba sa ilang mahahalagang aspeto mula sa SFNT. Kapansin-pansin, ang data para sa mga talahanayan ng font ay na-compress sa isang stream ng data na binubuo ng lahat ng mga talahanayan ng font. Katulad ng WOFF 1.0 na format, ang opsyonal na pinalawak na metadata at pribadong data block ay ipinapakita bilang hiwalay na mga bloke ng data, at pinagsama-sama sa WOFF 2.0 file. Ang compression algorithm na ginagamit para sa parehong naka-compress na font data stream at pinahabang metadata block ay Brotli [ *[Brotli](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22ref-Brotli)* ].

Ang isang kumpletong WOFF2 file ay binubuo ng ilang mga bloke ng data: isang 48-byte na header, kaagad na sinundan (sa ganitong pagkakasunud-sunod) ng isang variable-size na Table Directory, isang opsyonal na Collection Directory (naroroon lamang kung ang input font file ay isang font collection), isang naka-compress na bloke ng data ng font, isang opsyonal na bloke ng Pinalawak na Metadata, at isang opsyonal na bloke ng Pribadong Data. Maliban sa padding na may maximum na tatlong null byte sa mga lugar kung saan tinukoy ang 4-byte na alignment ng haba ng talahanayan o block offset, HINDI DAPAT magkaroon ng anumang extraneous na data sa pagitan ng mga bloke ng data o WOFF2 header at direktoryo ng talahanayan, o higit pa sa huling bloke o mesa . Kung may ganoong extraneous na data , DAPAT tanggihan ng isang umaayon na user agent ang file bilang hindi wasto . Ang file DAPAT ding tanggihan bilang di-wasto kung ang mga offset at haba ng anumang mga bloke ng data o mga talahanayan ng font ay nagpapahiwatig ng magkakapatong na mga hanay ng byte ng file, o mga saklaw na lalampas sa dulo ng file .

|  |
| --- |
| **WOFF2 File** |
| WOFF2Header  | Header ng file na may pangunahing uri ng font at bersyon, kasama ang mga offset sa metadata at pribadong data block. |
| TableDirectory  | Direktoryo ng mga talahanayan ng font, na naglalaman ng laki at iba pang impormasyon.   |
| CollectionDirectory  | Isang opsyonal na talahanayan na naglalaman ng mga paglalarawan ng fragment ng font ng mga entry sa koleksyon ng font.   |
| CompressedFontData  | Mga nilalaman ng mga talahanayan ng font, na-compress para sa imbakan sa WOFF2 file.   |
| ExtendedMetadata  | Isang opsyonal na bloke ng pinahabang metadata, na kinakatawan sa XML na format at naka-compress para sa storage sa WOFF2 file. |
| PrivateData  | Isang opsyonal na bloke ng pribadong data para sa font designer, foundry, o vendor na gagamitin.   |

### **3.1. Uri ng data**

|  |
| --- |
| **Uri ng data** |
| UInt8  | 8-bit na unsigned integer.  |
| Int16  | 16-bit signed integer sa 2's complement format, na nakaimbak ng big-endian.  |
| UInt16  | 16-bit na unsigned integer, nakaimbak ng big-endian.  |
| UInt32  | 32-bit unsigned integer, nakaimbak ng big-endian.  |
| 255UInt16  | Variable-length na pag-encode ng 16-bit na unsigned integer para sa na-optimize na intermediate font data storage.  |
| UIntBase128  | Variable-length encoding ng 32-bit unsigned integers.  |

#### 255UInt16 Uri ng Data

Ang 255UInt16 ay isang variable-length na encoding ng isang unsigned integer sa range na 0 hanggang 65535 kasama. Ang uri ng data na ito ay inilaan na gamitin bilang intermediate na representasyon ng iba't ibang mga halaga ng font, na karaniwang ipinahayag bilang UInt16 ngunit kumakatawan sa mga medyo maliit na halaga. Depende sa naka-encode na halaga, ang haba ng field ng data ay maaaring isa hanggang tatlong byte, kung saan ang halaga ng unang byte ay kumakatawan sa maliit na halaga mismo o itinuturing bilang isang code na tumutukoy sa format ng karagdagang (mga) byte. Ang "C-like" pseudo-code na naglalarawan kung paano basahin ang 255UInt16 na format ay ipinakita sa ibaba:

 Basahin 255UShort( data )

 {

 UInt8 code;

 UInt16 value, value2;

 const oneMoreByteCode1 = 255;

 const oneMoreByteCode2 = 254;

 const wordCode = 253;

 const lowestUCode = 253;

 code = data.getNextUInt8();

 kung ( code == wordCode ) {

 /\* Magbasa ng dalawa pang byte at pagdugtungin ang mga ito upang bumuo ng UInt16 value\*/

 value = data.getNextUInt8();

 halaga <<= 8;

 halaga &= 0xff00;

 value2 = data.getNextUInt8();

 halaga |= halaga2 & 0x00ff;

 }

 else if ( code == oneMoreByteCode1 ) {

 value = data.getNextUInt8();

 value = (value + lowestUCode);

 }

 else if ( code == oneMoreByteCode2 ) {

 value = data.getNextUInt8();

 value = (value + lowestUCode\*2);

 }

 iba {

 halaga = code;

 }

 ibalik ang halaga;

 }

Tandaan na ang pag-encode ay hindi natatangi. Halimbawa, ang value na 506 ay maaaring i-encode bilang [255, 253], [254, 0], at [253, 1, 250]. Ang isang encoder ay maaaring gumawa ng alinman sa mga ito, at ang isang decoder ay DAPAT tanggapin silang lahat . Ang isang encoder ay dapat pumili ng mas maikling mga pag-encode, at dapat na pare-pareho sa pagpili ng pag-encode para sa parehong halaga, dahil ito ay malamang na mag-compress ng mas mahusay.

#### Uri ng Data ng UIntBase128

Ang UIntBase128 ay isang iba't ibang variable na haba ng pag-encode ng mga unsigned integer, na angkop para sa mga value na hanggang 2 32 -1. Ang isang UIntBase128 na naka-encode na numero ay isang pagkakasunud-sunod ng mga byte kung saan ang pinaka makabuluhang bit ay itinakda para sa lahat maliban sa huling byte, at malinaw para sa huling byte. Ang numero mismo ay base 128 na naka-encode sa mas mababang 7 bits ng bawat byte. Kaya, ang isang pamamaraan ng pag-decode para sa isang UIntBase128 ay: magsimula sa halaga = 0. Kumonsumo ng isang byte, ang pagtatakda ng halaga = ang lumang halaga ay beses na 128 + (byte bitwise-at 127). Ulitin ang huling hakbang hanggang sa ang pinaka makabuluhang bit ng byte ay mali.

Ang format ng pag-encode ng UIntBase128 ay nagbibigay-daan sa posibilidad ng sub-optimal na pag-encode, kung saan hal. ang parehong numerical na halaga ay maaaring katawanin na may variable na bilang ng mga byte (gamit ang nangungunang 'zero'). Halimbawa, ang value 63 ay maaaring i-encode bilang alinman sa isang byte 0x3F o dalawa (o higit pa) byte: [0x80, 0x3f]. Hindi dapat payagan ng isang encoder na mangyari ito at dapat gumawa ng pinakamaikling posibleng pag-encode. DAPAT tanggihan ng decoder ang font file kung makatagpo ito ng UintBase128-encoded value na may mga leading zero (isang value na nagsisimula sa byte na 0x80), kung UintBase128-encoded sequence ay mas mahaba sa 5 bytes, o kung ang UintBase128-encoded value ay lumampas sa 2 32 -1 . Ang "C-like" pseudo-code na naglalarawan kung paano basahin ang UIntBase128 na format ay ipinakita sa ibaba:

 bool ReadUIntBase128( data, \*resulta )

 {

 UInt32 accum = 0;

 para sa (i = 0; i < 5; i++) {

 UInt8 data\_byte = data.getNextUInt8();

 // Walang leading 0's

 kung (i == 0 && data\_byte == 0x80) ay nagbabalik ng false;

 // Kung ang alinman sa nangungunang 7 bits ay nakatakda pagkatapos ay << 7 ay umaapaw

 kung (accum & 0xFE000000) ay nagbabalik ng false;

 \*accum = (accum << 7) | (data\_byte & 0x7F);

 // Paikutin hanggang sa ang pinaka makabuluhang bit ng data byte ay false

 kung ((data\_byte & 0x80) == 0) {

 \*resulta = accum;

 bumalik ng totoo;

 }

 }

 // Ang UIntBase128 sequence ay lumampas sa 5 bytes

 ibalik ang mali;

 }

### **3.2. WOFF2 Header**

Ang WOFF 2.0 na header ay may kasamang pirmang nagpapakilala at nagbibigay ng impormasyon tungkol sa mga naka-compress at hindi naka-compress na laki ng naka-encapsulate na data ng font. Ipinapahiwatig din nito ang partikular na uri ng data ng font na kasama sa WOFF 2.0 file, numero ng bersyon ng font at nagbibigay ng mga offset sa karagdagang mga bloke ng data na kasama sa file.

|  |
| --- |
| **WOFF2 Header** |
| UInt32 | pirma | 0x774F4632 'wOF2' |
| UInt32 | lasa | Ang "sfnt na bersyon" ng input na font. |
| UInt32 | haba | Kabuuang laki ng WOFF file. |
| UInt16 | numTables | Bilang ng mga entry sa direktoryo ng mga talahanayan ng font. |
| UInt16 | nakalaan | Nakalaan; itakda sa 0. |
| UInt32 | kabuuangSfntSize | Kabuuang sukat na kailangan para sa hindi na-compress na data ng font, kabilang ang sfnt header,direktoryo, at mga talahanayan ng font (kabilang ang padding). |
| UInt32 | kabuuangCompressedSize | Kabuuang haba ng naka-compress na bloke ng data. |
| UInt16 | majorVersion | Pangunahing bersyon ng WOFF file. |
| UInt16 | minorVersion | Minor na bersyon ng WOFF file. |
| UInt32 | metaOffset | Offset sa metadata block, mula sa simula ng WOFF file. |
| UInt32 | metaLength | Haba ng naka-compress na metadata block. |
| UInt32 | metaOrigLength | Hindi naka-compress na laki ng metadata block. |
| UInt32 | privOffset | Offset sa pribadong data block, mula sa simula ng WOFF file. |
| UInt32 | privLength | Haba ng pribadong data block. |

Ang interpretasyon ng WOFF2 Header ay kapareho ng WOFF Header sa [ [*WOFF1*](https://www.w3.org/TR/WOFF2/#ref-woff10) ], kasama ang pagdaragdag ng isang bagong totalCompressedSize field. Ang signature field sa WOFF2 header ay DAPAT maglaman ng value na 0x774F4632 ('wOF2') , na nakikilala ito sa WOFF 1.0 na mga file. Kung ang field ay hindi naglalaman ng halagang ito, DAPAT tanggihan ng mga user agent ang file bilang hindi wasto . Ang isang wastong WOFF 2.0 file ay DAPAT nakatakda ang nakalaan na field sa 'zero' , HINDI DAPAT tanggihan ng decoder ang isang na-download na font file kung ang nakareserbang halaga ng header ay hindi zero . Mga ahente ng gumagamit HINDI DAPAT tanggihan ang wastong na-decode na font file kung ang resultang laki ng font file ay hindi tumutugma sa kabuuangSfntSize value na naka-encode bilang bahagi ng WOFF2 header .

Ang halaga ng "totalSfntSize" sa WOFF2 Header ay nilayon na gamitin para sa mga layuning sanggunian lamang. Maaaring ito ay kumakatawan sa laki ng hindi naka-compress na input ng font file, ngunit kung ang binagong 'glyf' at 'loca' na mga talahanayan ay naroroon, ang hindi naka-compress na laki ng mga muling itinayong talahanayan at ang kabuuang na-decompress na laki ng font ay maaaring mag-iba nang malaki mula sa orihinal na kabuuang sukat na tinukoy sa ang WOFF2 Header.

## **4. Direktoryo ng font**

Ang seksyon ng direktoryo ng font ay binubuo ng direktoryo ng talahanayan, na naglalaman ng isang entry para sa bawat natatanging talahanayan ng font sa file ng font ng input, at maaari ring magsama ng isang opsyonal na talahanayan ng direktoryo ng koleksyon, kapag ang file ng font ng input ay isang koleksyon ng font.

### **4.1. Format ng direktoryo ng talahanayan**

Ang direktoryo ng talahanayan ay isang hanay ng mga entry sa direktoryo ng talahanayan ng WOFF2, gaya ng tinukoy sa ibaba. Ang direktoryo ay sumusunod kaagad pagkatapos ng WOFF2 file header; samakatuwid, walang tahasang offset sa header na tumuturo sa block na ito. Ang laki nito ay nakasalalay sa eksaktong nilalaman; kaya, ang pinakamahusay na diskarte para sa pag-decode ay iproseso ang file bilang isang stream, sa halip na subukang i-access ito nang random. Ang bawat entry sa direktoryo ng talahanayan ay tumutukoy sa orihinal na laki ng isang talahanayan ng data ng font at, para sa mga talahanayan kung saan inilapat ang karagdagang pagbabago, ang laki ng binagong talahanayan. Ang mga nabagong talahanayan ay nakikilala sa pamamagitan ng kanilang field na 'mga flag' na pinagsasama ang mga kilalang tag ng talahanayan (bits 0-5, tingnan ang "Mga Kilalang Tag ng Talahanayan") at numero ng bersyon ng pagbabagong-anyo (mga bit 6 at 7, tingnan ang paglalarawan sa ibaba) na tumutukoy sa inilapat na pagbabago. Ang kumbinasyon ng tag ng talahanayan at ang numero ng bersyon ng pagbabago ay tumutukoy sa eksaktong pagbabagong inilapat sa isang talahanayan, ang kasalukuyang tinukoy na mga pagbabago ay tinukoy sa mga detalye sa[sugnay 5](https://www.w3.org/TR/WOFF2/#table_format) ng detalyeng ito.

Taliwas sa paraan kung paano tinukoy ang mga entry ng direktoryo ng talahanayan sa orihinal na file ng input ng font (kung saan ang mga entry ng direktoryo ng talahanayan ay pinagsunod-sunod sa pataas na pagkakasunud-sunod ng alpabeto), ang mga entry sa direktoryo ng talahanayan ng WOFF2 ay tumutukoy sa pisikal na pagkakasunud-sunod ng mga talahanayan kung saan sila naproseso at na-encode bilang bahagi ng compressed font data stream. Responsibilidad ng decoder na pagbukud-bukurin at muling ayusin ang muling itinayong direktoryo ng talahanayan kapag na-decompress ang font file.

Upang pasimplehin ang proseso ng pag-decode ng font DAPAT muling isaayos ng isang encoder ang mga binagong talahanayan upang ang mga nauugnay na talahanayang 'glyf' at 'loca' ay palaging naroroon bilang isang pares, na may talahanayang 'loca' na sumusunod sa talahanayang 'glyf' . Pakitandaan na ang entry para sa binagong 'loca' na talahanayan ay isang placeholder lamang na nagsasaad ng laki na kinakailangan upang muling buuin ang talahanayan habang hindi gumagamit ng mga byte ng stream ng data. Samakatuwid, ang pangangailangan para sa 'loca' table entry na agad na sumunod sa transformed 'glyf' table sa table directory ay nagpapadali sa pagpapatupad ng 'loca' table reconstruction step kapag ang reverse transform (tingnan ang [subclause 5.1](https://www.w3.org/TR/WOFF2/#glyf_table_format)para sa mga detalye) ay inilalapat sa talahanayan ng 'glyf'. Para sa higit pang mga detalye sa mga hadlang sa pagkakasunud-sunod ng talahanayan tingnan [ang subclause 5.5](https://www.w3.org/TR/WOFF2/#table_order) .

Ang format ng bawat indibidwal na entry sa direktoryo ng talahanayan ay ang mga sumusunod:

|  |
| --- |
| **TableDirectoryEntry** |
| UInt8 | mga watawat | uri ng talahanayan at mga watawat |
| UInt32 | tag | 4-byte na tag (opsyonal) |
| UIntBase128 | origLength | haba ng orihinal na talahanayan |
| UIntBase128 | transformLength | nabagong haba (kung naaangkop) |

Ang interpretasyon ng field ng mga flag ay ang mga sumusunod. Ang mga bit [0..5] ay naglalaman ng index sa talahanayang "kilalang tag", na kumakatawan sa mga tag na malamang na lumabas sa mga font. Kung ang tag ay wala sa talahanayang ito, ang halaga ng bit field na ito ay 63. Ang mga bit 6 at 7 ay nagpapahiwatig ng preprocessing transformation version number (0-3) na inilapat sa bawat talahanayan. Para sa lahat ng mga talahanayan sa isang font, maliban sa mga talahanayang 'glyf' at 'loca', ang bersyon 0 ng pagbabagong-anyo ay nagpapahiwatig ng null transform kung saan ang orihinal na data ng talahanayan ay direktang ipinapasa sa Brotli compressor para isama sa naka-compress na stream ng data. Para sa mga talahanayang 'glyf' at 'loca', ang pagbabagong bersyon 3 ay nagpapahiwatig ng null transform kung saan direktang ipinasa ang orihinal na data ng talahanayan sa Brotli compressor nang hindi inilalapat ang anumang paunang pagproseso na tinukoy sa [subclause 5.1](https://www.w3.org/TR/WOFF2/#glyf_table_format) at [subclause 5.3](https://www.w3.org/TR/WOFF2/#loca_table_format) . Ang binagong mga format ng talahanayan at ang kanilang nauugnay na mga numero ng bersyon ng pagbabago ay inilarawan sa mga detalye sa [sugnay 5](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22table_format) ng detalyeng ito.

Kung maraming bersyon ng pagbabagong-anyo ang tinukoy para sa isang partikular na talahanayan, karaniwang dapat piliin ng isang encoder ang bersyon ng pagbabagong-anyo na nagreresulta sa pinakamaliit na binagong haba ng talahanayan, maliban kung ang pagbawas ay napakaliit na hindi gaanong mahalaga.

Kung saan higit sa isang bersyon ng pagbabago ang tinukoy para sa isang ibinigay na tag ng talahanayan, maaaring gamitin ng isang encoder ang alinman sa mga tinukoy na pagbabagong naaangkop sa data ng pag-input; DAPAT maging handa ang isang decoder na tanggapin ang alinman sa mga tinukoy na pagbabago . Kung ang isang decoder ay nakatagpo ng isang entry sa talahanayan na tumutukoy sa isang hindi kilalang numero ng bersyon ng pagbabagong-anyo DAPAT na tanggihan ang buong font dahil hindi ito ma-decode nang tama.

Kung ang isang font table ay naka-encode ng isang kilalang tag ng talahanayan o tahasang kasama ang apat na byte na tag ay walang semantical significance; isa lamang itong pagpipilian ng pag-encode na nilayon upang mapabuti ang kahusayan ng compression. Katulad nito, kung ang isang partikular na tag na may apat na byte ay naroroon o wala sa talahanayang "Mga Kilalang Tag ng Talahanayan" ay hindi isang normatibong pahayag tungkol sa kung ang mga naturang tag ay dapat isama sa mga web font. DAPAT gumamit ang encoder ng kilalang flag ng talahanayan kapag nag-e-encode ng isang input font table na may tag na nakalista sa talahanayang "Mga Kilalang Tag ng Talahanayan" .

Mayroong paunang natukoy na mekanismo ng extension para sa anumang custom na tag ng talahanayan na hindi kasama sa kilalang listahan ng tag (o para sa anumang bagong karaniwang mga tag na maaaring tukuyin sa hinaharap). Kung ang mga bit [0..5] ng flags byte ay may value na 63 (0x3f), pagkatapos ay ang pagsunod sa flag byte ay isang 4-byte na arbitrary na halaga ng tag. Kung hindi, ang field ng tag ay tinanggal sa istraktura ng TableDirectoryEntry, at nagmula sa mga bit [0..5] ng flag byte mula sa nakapirming talahanayan ng Mga Kilalang Tag ng Talahanayan, na ibinigay sa ibaba. MAAARING tanggapin ng decoder ang isang entry sa talahanayan na tinukoy gamit ang isang custom na tag ng talahanayan kahit na ang tag na sumusunod ay natukoy bilang isa sa mga tag ng kown table .

|  |
| --- |
| **Mga Kilalang Tag ng Talahanayan** |
| **Bandila** | **Tag** | **Bandila** | **Tag** | **Bandila** | **Tag** | **Bandila** | **Tag** |
| 0 | cmap | 16 | EBLC | 32 | CBDT | 48 | gvar |
| 1 | ulo | 17 | hingal | 33 | CBLC | 49 | hsty |
| 2 | hhea | 18 | hdmx | 34 | COLR | 50 | basta |
| 3 | hmtx | 19 | kern | 35 | CPAL | 51 | lcar |
| 4 | maxp | 20 | LTSH | 36 | SVG | 52 | mort |
| 5 | pangalan | 21 | PCLT | 37 | sbix | 53 | morx |
| 6 | OS/2 | 22 | VDMX | 38 | acnt | 54 | opbd |
| 7 | post | 23 | vhea | 39 | avar | 55 | prop |
| 8 | cvt | 24 | vmtx | 40 | bdat | 56 | trak |
| 9 | fpgm | 25 | BASE | 41 | bloc | 57 | Zapf |
| 10 | glyf | 26 | GDEF | 42 | bsln | 58 | Silf |
| 11 | lugar | 27 | GPOS | 43 | cvar | 59 | Glat |
| 12 | paghahanda | 28 | GSUB | 44 | fdsc | 60 | Sinabi ni Gloc |
| 13 | CFF | 29 | EBSC | 45 | feat | 61 | Feat |
| 14 | VORG | 30 | JSTF | 46 | fmtx | 62 | Sill |
| 15 | EBDT | 31 | MATH | 47 | fvar | 63 | sumusunod ang arbitrary na tag |

Pakitandaan na ayon sa mga [detalye ng format ng font na nakabatay sa SFNT](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22References) ang lahat ng mga tag ng talahanayan ay dapat na binubuo ng apat na character. Ang mga tag ng talahanayan na may mas mababa sa apat na letra, gaya ng ' cvt ' (tag value 0x63767420) ay nilagyan ng mga trailing space.

Ang pagsunod sa mga flag (at ang opsyonal na field ng tag) ay isa o dalawang halaga ng haba, bawat isa ay nasa UIntBase128 unsigned integer encoding. Tinutukoy ng field na origLength ang haba ng talahanayan sa isang hindi naka-compress na bersyon ng font. Para sa mga talahanayan na sumasailalim sa mga karagdagang [pagbabago](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22table_format) , DAPAT isama ang field na transformLength na tumutukoy sa haba ng binagong bersyon ng talahanayan .

Ang field ng transformLength ay naroroon sa entry ng direktoryo ng talahanayan kung, at kung, ang talahanayan ay naproseso ng isang non-null na pagbabago bago ang Brotli compression. Para sa mga talahanayan na hindi binago, walang field ng transformLength ang naroroon sa entry ng direktoryo.

Ang field na naglalaman ng binagong haba ay nagbibigay ng impormasyon tungkol sa binagong laki ng talahanayan bago ito i-compress ng [ *[Brotli](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22ref-Brotli)* ]. Opsyonal ang field na ito at naroroon lamang para sa ilang partikular na talahanayan (tingnan sa ibaba ang paglalarawan ng "Format ng naka-compress na data"). Pakitandaan na habang ang binagong haba ay maaasahan sa pagtukoy sa na-decompress na laki ng talahanayan, ang orihinal na haba ng talahanayan ng binagong talahanayan ng font ay dapat tratuhin nang may pag-iingat.

Ang proseso ng muling pagtatayo ng mga binagong talahanayan ay ginagarantiyahan ang pagpapanatili ng paggana ng mga talahanayan ngunit maaaring magdulot ng mga binary na resulta na iba sa orihinal na data. Halimbawa, ang mga talaan ng talahanayan ng 'glyf' ay maaaring may mga outline point na coordinate na naka-encode gamit ang isa- o dalawang-byte na format, at ang mga paulit-ulit na halaga ay maaaring tahasang ma-duplicate sa coordinate stream o tanggalin (na ipapahiwatig ng flags byte ng control. punto). Samakatuwid, posible ang iba't ibang representasyon ng mga glyph outline point na magbubunga ng magkaparehong resulta ng pag-render; gayunpaman, ang binary data ng mga na-reconstruct na glyph record ay maaaring malaki ang pagkakaiba sa orihinal na data. Tingnan ang subclause 5.3.3 ng ISO "Open Font Format" [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF)] detalye para sa mga detalye.

Para sa kadahilanang ito, ang halaga ng field ng origLength ng binagong talahanayan ay dapat na ituring lamang bilang isang sanggunian at hindi dapat umasa sa paggawa ng mga desisyon sa paglalaan ng memorya kapag ang data ng WOFF2 ay na-decode.

### **4.2. Format ng direktoryo ng koleksyon**

Ang direktoryo ng koleksyon ay naroroon lamang kung ang input na font ay isang koleksyon; ibig sabihin, kung ang halaga ng [WOFF2 file header](https://www.w3.org/TR/WOFF2/#woff20Header) na "flavor" ay nakatakda sa 0x74746366 ('ttcf'). Kung mayroon, ang direktoryo ng koleksyon ay sumusunod kaagad pagkatapos ng direktoryo ng talahanayan. Ang direktoryo ng koleksyon ay binubuo ng isang CollectionHeader at isa o higit pang mga tala ng CollectionFontEntry.

Kung ang input ng font ay isang koleksyon, ang [direktoryo ng talahanayan](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22table_dir_format) ay naglalaman ng isang entry para sa bawat natatanging talahanayan sa buong koleksyon, hindi alintana kung ang isang talahanayan ay nakabahagi sa iba't ibang mga font sa isang koleksyon o hindi. Kaya, ang direktoryo ng talahanayan para sa isang koleksyon ng font na na-compress sa WOFF2 ay binubuo ng lahat ng mga talahanayan mula sa lahat ng mga font na nilalaman sa file ng koleksyon ng font.

Nagsisimula ang direktoryo ng Koleksyon sa Header ng Koleksyon na sinusundan ng hanay ng Mga Entry ng Font ng Koleksyon.

|  |
| --- |
| **CollectionHeader** |
| UInt32 | bersyon | Ang Bersyon ng TTC Header sa orihinal na font. |
| 255UInt16 | numFonts | Ang bilang ng mga font sa koleksyon. |

Tinukoy ng mga entry ng font ng koleksyon ang bawat font sa koleksyon (tinukoy ng field na "numFonts" ng CollectionHeader). Ang format ng bawat indibidwal na entry ng font ng koleksyon ay ang mga sumusunod:

|  |
| --- |
| **CollectionFontEntry** |
| 255UInt16 | numTables | Ang bilang ng mga talahanayan sa font na ito |
| UInt32 | lasa | Ang "sfnt na bersyon" ng font |
| 255UInt16 | index[numTables] | Ang index na tumutukoy sa isang entry sa Table Directory para sa bawat talahanayan sa font na ito (kung saan ang index ng unang entry sa Table Directory ay zero.) |

Ang isang talahanayan na ang offset ay tinutukoy ng maraming Offset na Talahanayan sa orihinal na koleksyon ay itinuturing na "nakabahagi."  DAPAT maglabas ng isang TableDirectoryEntry ang isang encoder para sa bawat natatanging talahanayan ng font na tinukoy ng offset nito , hindi alintana kung ang talahanayang iyon ay nakabahagi sa maraming mga font sa isang koleksyon o hindi. Ibig sabihin, kung ang isang koleksyon ay naglalaman ng maraming font na may katumbas na Talaan ng Talaan ng mga ito na tumutukoy sa mga nakabahaging talahanayan ng font na may parehong mga offset,  HINDI DAPAT na duplicate ang data ng talahanayan sa WOFF2 compressed data stream .

Kung ang isang input na file ng font ay isang koleksyon ng font na naglalaman ng maraming natatanging glyf at loca table, ang mga [pagbabagong](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22glyf_table_format) inilarawan para sa glyf at loca table ay DAPAT ilapat sa bawat pares ng mga talahanayan . Ang pagbabahagi ng mga glyf at loca table ay pinapayagan at hinihikayat (ito ay isa sa mga pangunahing benepisyo ng isang koleksyon ng font); gayunpaman, posible na ang mga koleksyon ng font ay maaaring magkaroon ng dalawa o higit pang mga pares ng mga talahanayan ng glyf / loca na maaaring hindi maibahagi. Kapag ang mga talahanayan ay nakabahagi, isang encoder DAPAT i-verify na ang parehong mga talahanayan ay nakabahagi at na parehong bumubuo ng isang nauugnay na pares (kung higit sa isang pares ng mga talahanayan ng glyf / loca ang naroroon) at DAPAT tanggihan ang isang koleksyon na naglalaman ng mga font na nagbabahagi lamang ng isa sa alinman sa glyf o loca table.

Ang isang encoder, kapag nagpoproseso ng isang koleksyon ng font,  DAPAT panatilihin ang parehong pagkakasunud-sunod ng mga nested na font tulad ng mga ito sa koleksyon ng input ng font at DAPAT itala ang index ng katugmang TableDirectoryEntry sa CollectionFontEntry para sa bawat font .

DAPAT ibalik ng isang decoder ang koleksyon na may parehong bilang ng mga font at ang kanilang mga katumbas na Offset Tables . Ang Offset Tables para sa bawat font na binubuo ng output font collection file DAPAT magkaroon ng parehong numTables bilang input collection, o mas kaunti kung ang DSIG ay nasa orihinal na input Offset Table. Kung ang halaga ng field ng bersyon para sa TTC Header sa CollectionHeader ay nakatakda sa "2.0", DAPAT itakda ng isang decoder ang mga field ng TTC Header na {ulDsigTag, ulDsigLength, ulDsigOffset} sa koleksyon ng output sa null o i-convert ang format ng TTC header sa bersyon 1 (0x00010000) .

Ang isang decoder, kapag nagpoproseso ng koleksyon ng font, DAPAT na muling likhain ang parehong pagkakasunud-sunod ng mga naka-nest na font gaya ng mga ito sa koleksyon ng input . Ang mga talahanayan sa loob ng bawat nested font ay maaaring muling ayusin. Kapag nagpoproseso ng koleksyon ng font na may maraming pares ng mga talahanayang 'glyf'/'loca', DAPAT suriin ng isang decoder ang mga indeks ng CollectionFontEntry upang matiyak na ang mga na-refer na 'glyf' at 'loca' na mga talahanayan ay ipinares sa isa't isa, at DAPAT tanggihan ang isang koleksyon ng font kung may nakitang mismatch sa table pairing . Para sa higit pang mga detalye tungkol sa mga pares ng talahanayan tingnan ang paglalarawan ng format ng [direktoryo ng talahanayan .](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22table_dir_format)DAPAT makapag-load at gumamit ng wastong format na file ng koleksyon ng font na nabuo ng isang WOFF2 decoder ang isang umaayon na user agent .

## **5.  Naka- compress na format ng data**

Ang CompressedFontData field sa isang WOFF2 file ay naglalaman ng pinagsama-samang data para sa bawat talahanayan sa font, sa pagkakasunud-sunod na ang mga entry ay lilitaw sa WOFF2 na [direktoryo ng talahanayan](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22table_dir_format) . HINDI DAPAT magkaroon ng anumang extraneous data sa pagitan ng mga entry sa talahanayan sa decompressed stream ng data gaya ng tinukoy ng direktoryo ng talahanayan . Kung may ganoong extraneous na data , DAPAT tanggihan ng isang umaayon na user agent ang file bilang hindi wasto . DAPAT i-compress ang CompressedFontData stream gamit ang Brotli compression algorithm [ *[Brotli](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22ref-Brotli)* ] . Kung nabigo ang decompression ng data block para sa anumang kadahilanan, ang WOFF2 file ay hindi wasto at HINDI DAPAT i-load .

Ang proseso ng pag-decode ng data ng talahanayan sa isang WOFF2 font file ay maaaring tukuyin sa pamamagitan ng pag-decompress sa byte-level na compression ng CompressedFontData field, na nagbubunga ng isang "table data block", pagkatapos ay paglalapat ng mga karagdagang hakbang sa pag-decode tulad ng inilarawan sa ibaba. Ang isang aktwal na pagpapatupad ay libre upang pagsamahin ang mga hakbang na ito o isagawa ang ilan sa mga hakbang sa isang incremental o streaming na paraan, ngunit ang mga resulta ay dapat na pare-pareho sa sequential na proseso tulad ng tinukoy dito.

Ang ilang partikular na talahanayan (gaya ng mga talahanayan ng glyf , loca at hmtx , na kinilala sa pamamagitan ng kanilang mga kaukulang tag), ay napapailalim sa mga karagdagang pagbabago. Kung ang isang talahanayan ng font ay hindi binago, ang data ng talahanayan ay lilitaw sa naka-compress na stream sa literal na anyo, at sumasakop sa origLength byte ng bloke ng data ng talahanayan. Kung binago man o hindi ang talahanayan, at ang numero ng bersyon ng inilapat na pagbabago ay tinutukoy ng dalawang pinakamahalagang flag bits (tingnan ang [subclause 4.1](https://www.w3.org/TR/WOFF2/#table_dir_format) para sa mga detalye), at ang data ng talahanayan ay dapat na karagdagang iproseso sa pamamagitan ng pagbabagong tinukoy sa ibaba. Sa kasong ito, ang binagong talahanayan ay sumasakop sa transformLength byte ng bloke ng data ng talahanayan.

Ang na-decompress at na-reconstruct na data ng talahanayan ay DAPAT na naka-imbak sa format na tinukoy ng detalye ng [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ]. Ang bawat reconstructed table directory entry DAPAT maglaman ng valid na 'checkSum' value, DAPAT muling kalkulahin ng decoder ang checkSum value para sa bawat decoded table. Gayundin, dahil sa pagbabago ng mga pagbabagong inilapat sa mga talahanayan ng glyf at loca , DAPAT muling kalkulahin ng decoder ang halaga ng checkSumAdjustment ng buong font at DAPAT na iimbak ang na-update na halaga sa head table.

Ang kilalang mga halaga ng flag ng talahanayan ay hindi dapat umasa sa pagtukoy ng pagkakaroon ng mga nabagong talahanayan, posible na hal. ang glyf table ay maaaring katawanin sa direktoryo ng talahanayan na may alinman sa flag = 10 at walang tag, o may flag = 63 at 'glyf ' tag na sumusunod. Ang kumbinasyon ng mga flag na halaga na tumutukoy sa isang numero ng bersyon ng pagbabago at ang tag ng talahanayan ay dapat gamitin upang matukoy ang inilapat na pagbabago.

Ang kabuuan ng mga field ng origLength (para sa mga hindi binagong talahanayan) at transformLength (para sa mga binagong talahanayan) sa direktoryo ng talahanayan ay DAPAT katumbas ng laki ng bloke ng data ng font pagkatapos itong ma-decompress gamit ang [ *[Brotli](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22ref-Brotli)* ] . Kung ang laki ng na-decompress na font Ang data block ay hindi tumutugma sa kabuuan ng mga haba na tinukoy sa direktoryo ng talahanayan tulad ng inilarawan sa itaas, HINDI DAPAT i-load ang WOFF2 file .

Ayon sa kasalukuyang bersyon ng WOFF2, maaaring ilapat ang isang pagbabago sa tatlong uri ng mga talahanayan: glyf , na kumakatawan sa outline data, loca , na kumakatawan sa mga offset ng mga indibidwal na glyph sa loob ng isang glyf table, at hmtx , na kumakatawan sa glyph horizontal metrics, kung ang mga ito ang mga talahanayan ay nasa isang font. Ang isang tool sa pag-akda ay maaaring gumamit ng anumang naaangkop na mga pagbabagong isinasaalang-alang ang mga karagdagang hadlang na tinukoy sa [subclause 5.3](https://www.w3.org/TR/WOFF2/#loca_table_format) at [subclause 5.5](https://www.w3.org/TR/WOFF2/#table_order) . Ang pagbabagong-anyo ng talahanayan ng glyf ay tinukoy sa [subclause 5.1](https://www.w3.org/TR/WOFF2/#glyf_table_format) , ang pagbabagong-anyo ng loca table ay tinukoy sa[subclause 5.3](https://www.w3.org/TR/WOFF2/#loca_table_format) , at ang hmtx table transformation ay tinukoy sa [subclause 5.4](https://www.w3.org/TR/WOFF2/#hmtx_table_format) .

Pagproseso ng CFF table at de-subroutinization.

Karamihan sa mga OpenType na font na may CFF outline ay may CFF data subroutinized para sa mas compact na data storage. Ang mga eksperimento na isinagawa bilang bahagi ng WOFF2 development at evaluation ay nagpakita na ang Brotli compression technology ay gumagana nang mas mahusay at gumagawa ng mas mahusay na compression gain para sa WOFF2 fonts kung ang CFF outline data ay de-subroutinize bago ang WOFF2 encoding - ito ay sa average na bawasan ang compressed data size ng 5-10%. Gayunpaman, ang CFF de-subroutinization ay, sa parehong oras, ay tataas ang laki ng output file ng ~10%.

Dahil ang CFF pre-processing / de-subroutinization ay itinuturing na isang panlabas na hakbang na maaaring ipatupad bago ang WOFF2 encoding, hindi ito saklaw ng detalyeng ito. Ang mga producer ng font ay may pagkakataon na i-optimize ang kanilang proseso ng produksyon upang bawasan ang laki ng naka-compress na font para sa pinakamabilis na paglilipat ng data ng webfont sa gastos ng pagtaas ng laki ng font ng output na CFF (na magpapahusay din sa pagganap ng pag-render) o upang mapanatili ang subroutinized na data ng CFF upang mabawasan ang input/ laki ng font ng output.

Ang mga pagbabagong WOFF 2.0 na inilapat sa ilang mga talahanayan ay idinisenyo upang bawasan at/o alisin ang mga built-in na redundancies ng SFNT na format at muling isaayos ang stream ng data ng font para sa mas mahusay na entropy encoding. Bilang resulta, pananatilihin ng na-reconstruct na data ng font ang eksaktong functionality ng input font file, ngunit dahil sa ilang posibleng variation ng pag-encode (gaya ng iba't ibang antas ng pag-optimize ng mga outline point coordinates sa 'glyf' table, at/o pagkakaiba sa offset mga kalkulasyon ng talahanayang 'loca') ang iba't ibang mga decoder ng WOFF2 ay maaaring gumawa ng isang output file na hindi magiging isang bitwise na tugma sa input ng font file. Ang mga talahanayan ng font ng input na file ng font ay maaari ding sumailalim sa mga operasyon ng muling pagsasaayos,[*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ] detalye – bilang resulta, kahit na hindi apektado ang nilalaman ng mga talahanayan ng data ng font, maaaring magbago ang mga offset ng talahanayan. Ang mga pagkakaiba sa binary font data (gayunpaman maliit, dahil sa makabuluhang lawak sa decoder optimizations at table reordering) sa pagitan ng naka-encode at decoded font data structures ay magpapawalang-bisa sa 'DSIG' table, kung mayroon ito. Samakatuwid, DAPAT alisin ng sumusunod na WOFF2 encoder  ang talahanayan ng DSIG mula sa isang input na data ng font , bago ilapat ang mga pagbabago at hakbang sa entropy coding.

DAPAT ding itakda ng mga WOFF 2.0 encoder ang bit 11 ng field na 'flag' ng head table (tingnan ang detalye ng [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ]) upang ipahiwatig na ang isang muling nilikhang font file ay sumailalim sa lossless modifying transform.

### **5.1. Binago ang format ng glyf table**

Ang data ng talahanayan ng glyf ay maaaring ipakita sa WOFF2 file sa isa sa dalawang format na tinukoy ng numero ng bersyon ng pagbabagong-anyo (naka-encode sa mga bits ng flag ng direktoryo ng talahanayan, tingnan ang [subclause 4.1](https://www.w3.org/TR/WOFF2/#table_dir_format) para sa mga detalye). Ang bersyon ng pagbabagong "3" ay tumutukoy sa isang null na pagbabago kung saan ang nilalaman ng talahanayan ng glyf ay ipinakita sa orihinal at hindi binagong format nito. Ang bersyon ng pagbabagong "0", na tinukoy sa ibaba, ay opsyonal at maaaring ilapat upang alisin ang ilang partikular na redundancies sa data ng talahanayan ng glyf .

Ang bersyon na "0" ng pagbabago ng talahanayan ng glyf (tulad ng tinukoy ng mga bits ng flag ng direktoryo ng talahanayan, tingnan ang [subclause 4.1](https://www.w3.org/TR/WOFF2/#table_dir_format) para sa mga detalye) na tinukoy sa subclause na ito ay nilalayon na bawasan ang kalabisan na impormasyon at magbigay ng mas mahusay na pag-encode ng aktwal na TrueType na mga outline ng mga glyph. Ang binagong pagbabago ay batay sa isang katulad na pagbabagong inilarawan sa detalye ng MicroType Express [ [*MTX*](https://www.w3.org/TR/WOFF2/#ref-mtx) ]. Ang sanggunian sa MTX ay nagbibigay-kaalaman; ang mga detalye ng binagong pagbabago ay nakasaad sa ibaba at ang seksyong ito ay normatibo.

Habang pinapanatili ng pagbabago ng talahanayan ng glyf ang functionality at katapatan ng pag-render para sa bawat glyph sa isang font, malamang na magreresulta ito sa paggawa ng bagong hanay ng mga tala ng glyph na hindi magiging binary na tugma sa orihinal na font. Ayon sa mga panuntunan sa pag-encode na tinukoy sa subclause 5.3.3 ng [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ], maaaring mayroong maraming wastong muling pagtatayo ng isang glyph record. Posible na depende sa konteksto at sa antas ng mga pag-optimize na inilalapat sa na-reconstruct na glyf table data, ang laki ng muling itinayong talahanayan ay maaaring mas malaki kaysa, mas mababa sa, o katumbas ng orihinal na laki ng glyf table ng input font na-compress ng WOFF2. Ni ang detalyeng ito o ang [[*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ] karaniwang mandato ng mga partikular na pag-optimize na inilalapat sa data ng font; samakatuwid, responsibilidad ng mga pagpapatupad ng WOFF2 decoder na tiyaking may sapat na dami ng memorya ang inilalaan kapag ang mga tala ng glyph ay nire-reconstruct mula sa WOFF2 file – ang orihinal na laki ng glyf table na ibinigay bilang isang origLength na halaga para sa binagong glyf table ay maaari lamang ginamit bilang reference point.

Anuman ang mga detalye ng pagpapatupad ng proseso ng muling pagtatayo ng tala ng glyph at ang antas ng mga pag-optimize na inilapat sa muling itinayong data ng talahanayan ng 'glyf' HINDI DAPAT tanggihan ng mga Ahente ng Gumagamit ang wastong na-decode na font file kung hindi tumutugma sa origLength na halaga ang laki ng na-reconstruct na 'glyf' na talahanayan naka-encode bilang bahagi ng 'glyf' table entry sa WOFF2 Table Directory .

Para sa higit na pagiging epektibo ng compression, ang glyf table ay nahahati sa ilang mga substream, upang igrupo ang tulad ng data. Ang binagong talahanayan ay binubuo ng isang bilang ng mga patlang na tumutukoy sa laki ng bawat isa sa mga substream, na sinusundan ng mga substream sa pagkakasunud-sunod. Sa panahon ng proseso ng pag-decode, nagaganap ang reverse transformation, kung saan ang data mula sa iba't ibang magkahiwalay na substream ay muling pinagsama-sama upang lumikha ng kumpletong glyph record para sa bawat entry ng orihinal na glyf table.

IMINUNGKAHING PAGWAWASTO 1: TINUTUKOY NG MGA [simpleng glyph flag](https://docs.microsoft.com/en-us/typography/opentype/spec/glyf) ang format ng x/y coordinate vectors para sa bawat punto ng glyph outline. Sa WOFF2, ang paunang pagproseso na hakbang ay nagko-convert sa parehong mga flag at mga coordinate sa kanilang mga sarili sa iba't ibang mga stream ng data. Maaaring itakda ang anim na magkakaibang flag (bits 0-5) upang tukuyin ang on/off-curve point at ang format para sa bawat hanay ng mga point coordinates, ang mga flag ay maaaring ibahagi sa maraming point coordinates.

Iba ang bit 6 na flag (OVERLAP\_SIMPLE), maaari lang itong itakda nang isang beses sa unang flag ng glyph. Kung itinakda, ipinapahiwatig nito na ang mga contour ng glyph ay \_maaaring\_ nagsasapawan. Ang OpenType spec ay hindi nangangailangan ng flag na ito na itakda, at ang ilan (ngunit hindi lahat) rasterizer ay hindi papansinin ang kaunting ito. Ang Bit7 ay nakalaan / hindi nagamit.

Dahil ang flag bit6 ay hindi kinakailangan at bit 7 ay nakalaan, sa orihinal na WOFF2 Recommendation ay maaari lamang silang tanggalin kapag ang source data ay na-encode, at itakda sa '0' sa muling pagtatayo ng glyph data. Tila, naging sanhi ito ng pagkawala ng data para sa flag bit 6 sa ilang mga font na may mga glyph na may magkakapatong na contour kung saan nakatakda ang opsyonal na bit 6 na ito, at isa itong isyu para sa ilang mga rasterizer na umaasa pa rin sa flag bit 6 na setting upang harapin ang mga contour overlap. .

Niresolba ng Iminungkahing Pagwawasto na ito ang problema, una, sa pamamagitan ng muling pagtukoy [sa dating 32-bit na field ng bersyon](https://www.w3.org/TR/WOFF2/) sa [binagong 'glyf' table](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22glyf_table_format) sa 16 bits, at pagdaragdag ng 16-bit na bagong field na 'optionFlags' sa nabakanteng espasyo. Ang isa sa mga bit na ito ay tinukoy na gagamitin bilang isang flag na nagpapahiwatig ng pagkakaroon ng bagong stream ng data, na ginamit upang i-encode ang pagkakaroon ng mga overlap na halaga ng flag (bit 6). Pangalawa, dahil ang bit 6 na flag ay maaari lamang itakda nang isang beses bawat glyph, isang [bagong numGlyphs-long bit array ang tinukoy at idinagdag](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22overlapSimpleBitmap) sa dulo ng WOFF2 transformed 'glyf' table.

Ang pagbabagong ito ay nasubok, at napag-alamang 100% paatras na tugma - ang mga legacy na WOFF2 decoder ay walang pakialam sa mga nakareserbang bit value, at hindi umaasa o nagbabasa ng karagdagang bit array sa dulo ng glyph table. Kaya, ang mga legacy na decoder ay patuloy na gumagana gaya ng dati, at ang mga na-update na decoder ngayon ay nagpapanatili ng opsyonal na bit na ito.

Ipakita ang PagbabagoIpakita ang KasalukuyanIpakita ang Kinabukasan

|  |
| --- |
| **Binagong glyf Table** |
| **Uri ng datos** | **Semantiko** | **Paglalarawan at uri ng halaga (kung naaangkop)** |
| UInt16 | nakalaan | 0x0000 |
| UInt16 | optionFlags | Bit 0: kung nakatakda, ay nagpapahiwatig ng pagkakaroon ng overlapSimpleBitmap[] bit array. Bits 1-15: Nakalaan. |
| UInt16 | numGlyphs | Bilang ng mga glyph |
| UInt16 | indexFormat | Offset na format para sa loca table, dapat na pare-pareho sa indexToLocFormat ng orihinal na head table (tingnan ang [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ] na detalye) |
| UInt32 | nContourStreamSize | Sukat ng nContour stream sa bytes |
| UInt32 | nPointsStreamSize | Sukat ng nPoints stream sa bytes |
| UInt32 | flagStreamSize | Sukat ng flag stream sa bytes |
| UInt32 | glyphStreamSize | Sukat ng glyph stream sa bytes (isang stream ng variable-length na mga naka-encode na value, tingnan ang paglalarawan sa ibaba) |
| UInt32 | compositeStreamSize | Sukat ng pinagsama-samang stream sa bytes (isang stream ng variable-length na naka-encode na mga halaga, tingnan ang paglalarawan sa ibaba) |
| UInt32 | bboxStreamSize | Sukat ng data ng bbox sa mga byte na kumakatawan sa pinagsamang haba ng bboxBitmap (isang naka-pack na bit array) at bboxStream (isang stream ng mga halaga ng Int16) |
| UInt32 | instructionStreamSize | Laki ng stream ng pagtuturo (isang stream ng mga halaga ng UInt8) |
| Int16 | nContourStream[] | Stream ng mga value ng Int16 na kumakatawan sa bilang ng mga contour para sa bawat tala ng glyph |
| 255UInt16 | nPointsStream[] | Stream ng mga value na kumakatawan sa bilang ng mga outline point para sa bawat contour sa mga tala ng glyph |
| UInt8 | flagStream[] | Stream ng mga value ng UInt8 na kumakatawan sa mga value ng flag para sa bawat outline point. |
| Iba-iba | glyphStream[] | Stream ng mga byte na kumakatawan sa mga value ng point coordinate gamit ang variable length encoding format (tinukoy sa [subclause 5.2](https://www.w3.org/TR/WOFF2/#triplet_decoding) ) |
| Iba-iba | compositeStream[] | Stream ng mga byte na kumakatawan sa mga value ng flag ng component at nauugnay na composite glyph data |
| UInt8 | bboxBitmap[] | Bitmap (isang numGlyphs-long bit array) na nagsasaad ng mga tahasang kahon ng hangganan |
| Int16 | bboxStream[] | Stream ng mga value ng Int16 na kumakatawan sa data ng glyph bounding box |
| UInt8 | instructionStream[] | Stream ng mga value ng UInt8 na kumakatawan sa isang set ng mga tagubilin para sa bawat katumbas na glyph |
| UInt8 | overlapSimpleBitmap[] | Isang numGlyphs-long bit array na nagbibigay ng mga value para sa overlap na flag [bit 6] para sa bawat simpleng glyph. (Naka-encode na ang mga value ng flag para sa mga composite glyph bilang bahagi ng compositeStream[]). |

Ang format ay pinakamahusay na nailalarawan sa pamamagitan ng paglalarawan sa proseso ng pag-decode, lalo na ang mga indikasyon ng kung ano ang wasto at di-wastong data. Tandaan din na tinutukoy ng format na ito ang na-decode na resulta sa antas ng semantiko, hindi mga partikular na byte stream.

Kasama sa Transformed glyf Table ang isang bboxBitmap na nagsasaad para sa bawat glyph kung naglalaman ito ng tahasang naka-encode na bounding box, o kung ang bounding box ay mahihinuha mula sa mga coordinate value. Ang mga nauugnay na pag-compute para matukoy ang status ng bounding box ay dapat gawin ng parehong encoder at decoder. Para sa bawat simpleng glyph,  DAPAT kalkulahin ng isang encoder ang mga value ng xMin, yMin, xMax at yMax na coordinate gamit ang lahat ng outline point (parehong on- at off-curve point) at ihambing ang mga nakalkulang value sa naka-encode na bounding box na impormasyon para sa glyph. Kung ang impormasyon ng glyph bounding box ay tumutugma sa mga kinakalkula na halaga, DAPAT alisin ng isang encoder ang bounding box info . Kung hindi, kung ang kinakalkula na mga halaga ng bounding box ay hindi katumbas ng mga halaga ng glyph na naka-encode sa isang font, DAPAT itakda ng isang encoder ang katumbas na flag ng bboxBitmap at itala ang orihinal na mga halaga ng bounding box sa bboxStream . Para sa mga tala ng glyph na naglalaman ng mga zero contour,  DAPAT suriin ng isang encoder na ang mga value ng glyph bounding box ay mga zero lahat at, kung hindi matugunan ang kundisyong ito, DAPAT tanggihan ang isang input na font bilang hindi wasto. Para sa alinman sa isang walang laman na glyph o isang glyph na walang mga contour,  DAPAT palaging i-clear ng isang encoder ang kaukulang bboxBitmap flag. Para sa isang composite glyph,  DAPAT palaging itakda ng isang encoder ang kaukulang bboxBitmapi -flag at itala ang orihinal na mga halaga ng bounding box sa bboxStream .

Ang isang decoder ay DAPAT magsagawa ng mga katulad na kalkulasyon kung ang bounding box ay mahihinuha. Para sa isang partikular na glyph, kapag ang isang katumbas na bit sa bboxBitmap ay hindi nakatakda, ang mga halaga ng xMin, yMin, xMax at yMax ay DAPAT kalkulahin sa oras ng pag-decode gamit ang lahat ng outline point coordinates . Ang kabuuang bilang ng mga byte sa bboxBitmap ay katumbas ng 4 \* floor((numGlyphs + 31) / 32). Ang mga bit ay naka-pack upang ang glyph number 0 ay tumutugma sa pinaka makabuluhang bit ng unang byte, glyph number 7 ay tumutugma sa hindi bababa sa makabuluhang bit ng unang byte, glyph number 8 ay tumutugma sa pinaka makabuluhang bit ng pangalawang byte, at iba pa sa. Ang isang bit=1 na halaga ay nagpapahiwatig ng isang tahasang itinakda na kahon ng hangganan.

Sa pagbabasa ng Transformed glyf Table, ang proseso ng pag-decode ay umuulit ng isang glyph sa isang pagkakataon. Para sa bawat glyph, nagbabasa ito ng zero o higit pang mga byte mula sa bawat isa sa mga stream na tinutukoy sa Transformed glyf Table. Gayundin, sa punto ng muling pagtatayo ng glyph, kailangang iimbak ng decoder para sa bawat glyph ang katumbas na offset sa bagong reconstructed glyph table, at ang data na ito ay sama-samang magiging nilalaman ng reconstructed loca table (tingnan ang [subclause 5.3](https://www.w3.org/TR/WOFF2/#loca_table_format) sa ibaba para sa higit pang impormasyon tungkol sa ang muling pagtatayo ng loca table).

#### Reconstruction ng Glyph Records

Ang proseso ng muling pagtatayo ay nagsisimula sa pamamagitan ng pagsasagawa ng sumusunod na hakbang upang matukoy ang naka-encode na uri ng glyph (simple, composite o walang laman na glyph):

Magbasa ng Int16 mula sa nContour stream. I-store ito sa field ng numberOfContours sa na-reconstruct na TrueType glyph. Ang interpretasyon ng field ay kapareho ng TrueType spec; kung ito ay zero, ang glyph ay walang laman. Kung ito ay positibo, ang glyph ay simple at ang halaga ay kumakatawan sa bilang ng mga contour sa outline. Kung ang halaga ng nContour ay katumbas ng -1 (0xffff), kung gayon ang glyph ay composite.

#### Pag-decode ng mga Empty Glyph

Ang muling pagtatayo ng isang walang laman na glyph (kapag nContour = 0) ay isang simpleng hakbang na kinabibilangan ng pagtaas ng bilang ng tala ng glyph at paggawa ng bagong entry sa talahanayan ng loca kung saan ang loca[n] = loca[n-1] . Kung ang flag ng bboxBitmap ay nagsasaad na ang mga value ng bounding box ay tahasang naka-encode sa bboxStream DAPAT tanggihan ng decoder ang WOFF2 file bilang hindi wasto .

#### Pag-decode ng Mga Simpleng Glyph

Para sa isang simpleng glyph (kapag nContour > 0), magpapatuloy ang proseso tulad ng sumusunod:

1. Basahin ang mga value ng numberOfContours 255UInt16 mula sa nPoints stream. Ang bawat isa sa mga ito ay ang bilang ng mga punto ng tabas na iyon. I-convert ito sa endPtsOfContours[] array sa pamamagitan ng pag-compute ng pinagsama-samang kabuuan, pagkatapos ay pagbabawas ng isa. Halimbawa, kung ang mga value sa stream ay [2, 4], ang endPtsOfContours array ay [1, 5]. Gayundin, ang kabuuan ng lahat ng mga halaga sa array ay ang kabuuang bilang ng mga puntos sa glyph, nPoints. Sa halimbawang ibinigay, ang halaga ng nPoints ay 6.
2. Basahin ang mga halaga ng nPoints UInt8 mula sa stream ng mga flag. Ang bawat isa ay tumutugma sa isang punto sa muling itinayong glyph outline. Ang interpretasyon ng flag byte ay inilarawan sa mga detalye sa [subclause 5.2](https://www.w3.org/TR/WOFF2/#triplet_decoding) .
3. Para sa bawat punto (ibig sabihin, nPoints times), basahin ang ilang point coordinate byte mula sa glyph stream. Ang bilang ng mga point coordinate byte ay isang function ng flag byte na nabasa sa nakaraang hakbang: para sa (flag < 0x7f) sa saklaw na 0 hanggang 83 kasama, ito ay isang byte. Sa saklaw na 84 hanggang 119 kasama, ito ay dalawang byte. Sa saklaw na 120 hanggang 123 kasama, ito ay tatlong byte, at sa hanay na 124 hanggang 127 kasama, ito ay apat na byte. I-decode ang mga byte na ito ayon sa pamamaraang tinukoy sa [subclause 5.2](https://www.w3.org/TR/WOFF2/#triplet_decoding) upang buuin muli ang mga halaga ng delta-x at delta-y ng mga coordinate ng glyph point. Itago ang mga delta-x at delta-y na halaga na ito sa muling itinayong glyph gamit ang karaniwang TrueType glyph encoding [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ] subclause 5.3.3.
4. Magbasa ng isang 255UInt16 value mula sa glyph stream, na instructionLength, ang bilang ng mga instruction byte.
5. Basahin ang instructionLength bytes mula sa instructionStream, at i-store ang mga ito sa reconstituted glyph bilang mga tagubilin.

Ang optionFlags field bit 0 ay nagpapahiwatig ng pagkakaroon ng karagdagang overlapSimpleBitmap[] bit array. Ang lahat ng iba pang mga piraso ay nakalaan at dapat na i-clear. Kung ang lahat ng simpleng glyph ng input font ay may mga OVERLAP\_SIMPLE na flag na nakatakda sa zero,  DAPAT i-clear ng encoder ang bit 0 ng field ng optionFlags at alisin ang overlapSimpleBitmap[] bit array . Kung hindi bababa sa isa sa mga simpleng glyph sa input font ang may mga flag na bit 6 set,  dapat itakda ng encoder ang optionFlags bit 0 sa "1", at DAPAT gumawa ng karagdagang overlapSimpleBitmap[] bit array kung saan DAPAT itakda ang mga katumbas na bits sa ang mga halaga ng flag bit 6 ng bawat simpleng glyph sa input font .

DAPAT suriin ng decoder ang optionFlags bit0 value upang matukoy kung ang karagdagang overlapSimpleBitmap[] ay kasama bilang bahagi ng binagong 'glyf' na talahanayan. Kung wala ang karagdagang overlapSimpleBitmap[] bit array, sa muling pagtatayo ng mga simpleng glyph na nag-flag DAPAT itakda ng decoder ang lahat ng OVERLAP\_SIMPLE na halaga ng flag sa zero . Kung kasama ang overlapSimpleBitmap[], DAPAT gumamit ang decoder ng mga setting ng overlapSimpleBitmap[] bit upang buuin muli ang OVERLAP\_SIMPLE na mga value ng flag, at itakda ang katumbas na overlapSimpleBitmap[] bit value bilang flag bit 6 sa unang flag byte para sa bawat simpleng glyph ng output font .

#### Pag-decode ng mga Composite Glyph

Para sa isang composite glyph (nContour == -1), ang mga sumusunod na hakbang ay pumapalit sa mga hakbang 1-5 sa itaas:

1a. Magbasa ng UInt16 mula sa compositeStream. Ito ay binibigyang kahulugan bilang isang component na flag na salita tulad ng sa TrueType spec. Batay sa mga value ng flag, mayroong sa pagitan ng 4 at 14 na karagdagang argument byte, na binibigyang kahulugan bilang glyph index, arg1, arg2, at opsyonal na sukat o affine matrix.

2a. Basahin ang bilang ng mga byte ng argumento gaya ng natukoy sa hakbang 1a mula sa composite stream, at iimbak ang mga ito sa muling itinayong glyph. Kung ang salitang flag na binasa sa hakbang 1a ay may set na FLAG\_MORE\_COMPONENTS bit (bit 5), bumalik sa hakbang 1a.

3a. Kung ang alinman sa mga flag na salita ay may set na FLAG\_WE\_HAVE\_INSTRUCTIONS bit (bit 8), pagkatapos ay basahin ang mga tagubilin mula sa glyph at itago ang mga ito sa muling itinayong glyph, gamit ang parehong proseso tulad ng inilarawan sa mga hakbang 4 at 5 sa itaas.

Panghuli, para sa parehong simple at composite glyph, kung ang katumbas na bit sa bounding box bit vector ay nakatakda, pagkatapos ay basahin din ang 4 na Int16 value mula sa bbox stream, na kumakatawan sa xMin, yMin, xMax, at yMax, ayon sa pagkakabanggit, at itala ang mga ito sa kaukulang mga patlang ng muling itinayong glyph. Para sa mga simpleng glyph, kung ang katumbas na bit sa bounding box bit vector ay hindi nakatakda, pagkatapos ay kunin ang bounding box sa pamamagitan ng pag-compute ng minimum at maximum na x at y na mga coordinate sa outline, at iimbak iyon.

Ang isang composite glyph ay DAPAT mayroong isang tahasang ibinigay na bounding box . Ang motibasyon ay ang pag-compute ng mga bounding box ay mas kumplikado, at mangangailangan ng paglutas ng mga sanggunian sa mga component glyph na isinasaalang-alang ang mga composite glyph na tagubilin at ang mga tinukoy na scale ng mga indibidwal na bahagi, na sasalungat sa isang purong streaming na pagpapatupad ng pag-decode ng font. DAPAT suriin ng isang decoder ang pagkakaroon ng impormasyon ng bounding box bilang bahagi ng composite glyph record at HINDI DAPAT mag-load ng font file na may nawawalang data ng composite bounding box .

### **5.2. Pag-decode ng variable-length na X at Y na mga coordinate**

[Ang simpleng](https://www.w3.org/TR/WOFF2/#SimpleGlyph) istraktura ng data ng glyph ay tumutukoy sa lahat ng mga contour na binubuo ng isang glyph outline, na ipinapakita ng isang pagkakasunud-sunod ng on- at off-curve na mga coordinate point. Ang mga point coordinates na ito ay naka-encode bilang mga delta value na kumakatawan sa mga incremental na value sa pagitan ng nakaraan at kasalukuyang katumbas na X at Y na coordinate ng isang punto, ang unang punto ng bawat outline ay nauugnay sa (0,0) point. Upang i-minimize ang laki ng dataset ng mga value ng point coordinate, ang bawat punto ay ipinapakita bilang isang (flag, xCoordinate, yCoordinate) triplet. Ang halaga ng flag ay iniimbak sa isang hiwalay na stream ng data at ang mga halaga ng coordinate ay iniimbak bilang bahagi ng stream ng data ng glyph gamit ang isang variable-length na format ng encoding na kumukonsumo ng kabuuang 2-5 byte bawat punto.

**IMINUNGKAHING PAGWAWASTO 2:** Kung ang on-curve ay ipinahiwatig ng bit na itinakda, o hindi nakatakda, ay hindi malinaw.

Ipakita ang PagbabagoIpakita ang KasalukuyanIpakita ang Kinabukasan

Ang pinakamahalagang bahagi ng isang flag ay nagpapahiwatig kung ang punto ay nasa-o off-curve na punto (kung ang pinaka makabuluhang bit ay 0, kung gayon ang punto ay on-curve), tinutukoy ng natitirang pitong bits ng flag ang format ng mga value ng X at Y coordinate at tumukoy ng 128 posibleng kumbinasyon ng mga indeks na itinalaga na isinasaalang-alang ang tipikal na istatistikal na pamamahagi ng data na makikita sa mga TrueType na font. Kapag ang mga halaga ng X at Y coordinate ay naitala gamit ang mga nibbles (alinman sa 4 bits bawat coordinate o 12 bits bawat coordinate) ang mga bit ay naka-pack sa byte stream na may pinakamaraming makabuluhang bit ng X coordinate muna, na sinusundan ng halaga para sa Y coordinate (pinaka makabuluhang bit una). Bilang resulta, ang laki ng dataset ng glyph ay makabuluhang nabawasan, at ang pagpapangkat ng mga katulad na halaga (mga flag, coordinates) sa magkahiwalay at magkadikit na stream ng data ay nagbibigay-daan sa mas mahusay na aplikasyon ng entropy coding na inilapat bilang ikalawang yugto ng proseso ng pag-encode.

Ang bawat isa sa 128 na mga halaga ng index ay tumutukoy sa mga sumusunod na katangian at tinukoy sa mga detalye sa talahanayan sa ibaba:

* Bilang ng byte (kabuuang bilang ng mga byte na ginamit para sa hanay ng mga halaga ng coordinate kasama ang isang byte para sa halaga ng 'bandila').
* Bilang ng mga bit na ginamit upang kumatawan sa halaga ng X coordinate (X bits).
* Bilang ng mga bit na ginamit upang kumatawan sa halaga ng Y coordinate (Y bits).
* Isang karagdagang incremental na halaga na idaragdag sa halaga ng X bits (delta X).
* Isang karagdagang incremental na halaga na idaragdag sa halaga ng Y bits (delta Y).
* Ang sign ng X coordinate value (X sign).
* Ang sign ng Y coordinate value (Y sign).

Pakitandaan na ang field na “Byte Count” ay sumasalamin sa kabuuang laki ng triplet (flag, xCoordinate, yCoordinate), kasama ang value ng 'flag' na naka-encode sa isang hiwalay na stream.

|  |
| --- |
| **Triplet Encoding** |
| **Index** | **Bilang ng Byte** | **X bit** | **Y bit** | **Delta X** | **Delta Y** | **X sign** | **Y sign** |
| 0 | 2 | 0 | 8 | N/A | 0 | N/A | - |
| 1 |  |  |  |  | 0 |  | + |
| 2 |  |  |  |  | 256 |  | - |
| 3 |  |  |  |  | 256 |  | + |
| 4 |  |  |  |  | 512 |  | - |
| 5 |  |  |  |  | 512 |  | + |
| 6 |  |  |  |  | 768 |  | - |
| 7 |  |  |  |  | 768 |  | + |
| 8 |  |  |  |  | 1024 |  | - |
| 9 |  |  |  |  | 1024 |  | + |
| 10 | 2 | 8 | 0 | 0 | N/A | - | N/A |
| 11 |  |  |  | 0 |  | + |  |
| 12 |  |  |  | 256 |  | - |  |
| 13 |  |  |  | 256 |  | + |  |
| 14 |  |  |  | 512 |  | - |  |
| 15 |  |  |  | 512 |  | + |  |
| 16 |  |  |  | 768 |  | - |  |
| 17 |  |  |  | 768 |  | + |  |
| 18 |  |  |  | 1024 |  | - |  |
| 19 |  |  |  | 1024 |  | + |  |
| 20 | 2 | 4 | 4 | 1 | 1 | - | - |
| 21 |  |  |  |  | 1 | + | - |
| 22 |  |  |  |  | 1 | - | + |
| 23 |  |  |  |  | 1 | + | + |
| 24 |  |  |  |  | 17 | - | - |
| 25 |  |  |  |  | 17 | + | - |
| 26 |  |  |  |  | 17 | - | + |
| 27 |  |  |  |  | 17 | + | + |
| 28 |  |  |  |  | 33 | - | - |
| 29 |  |  |  |  | 33 | + | - |
| 30 |  |  |  |  | 33 | - | + |
| 31 |  |  |  |  | 33 | + | + |
| 32 |  |  |  |  | 49 | - | - |
| 33 |  |  |  |  | 49 | + | - |
| 34 |  |  |  |  | 49 | - | + |
| 35 |  |  |  |  | 49 | + | + |
| 36 | 2 | 4 | 4 | 17 | 1 | - | - |
| 37 |  |  |  |  | 1 | + | - |
| 38 |  |  |  |  | 1 | - | + |
| 39 |  |  |  |  | 1 | + | + |
| 40 |  |  |  |  | 17 | - | - |
| 41 |  |  |  |  | 17 | + | - |
| 42 |  |  |  |  | 17 | - | + |
| 43 |  |  |  |  | 17 | + | + |
| 44 |  |  |  |  | 33 | - | - |
| 45 |  |  |  |  | 33 | + | - |
| 46 |  |  |  |  | 33 | - | + |
| 47 |  |  |  |  | 33 | + | + |
| 48 |  |  |  |  | 49 | - | - |
| 49 |  |  |  |  | 49 | + | - |
| 50 |  |  |  |  | 49 | - | + |
| 51 |  |  |  |  | 49 | + | + |
| 52 | 2 | 4 | 4 | 33 | 1 | - | - |
| 53 |  |  |  |  | 1 | + | - |
| 54 |  |  |  |  | 1 | - | + |
| 55 |  |  |  |  | 1 | + | + |
| 56 |  |  |  |  | 17 | - | - |
| 57 |  |  |  |  | 17 | + | - |
| 58 |  |  |  |  | 17 | - | + |
| 59 |  |  |  |  | 17 | + | + |
| 60 |  |  |  |  | 33 | - | - |
| 61 |  |  |  |  | 33 | + | - |
| 62 |  |  |  |  | 33 | - | + |
| 63 |  |  |  |  | 33 | + | + |
| 64 |  |  |  |  | 49 | - | - |
| 65 |  |  |  |  | 49 | + | - |
| 66 |  |  |  |  | 49 | - | + |
| 67 |  |  |  |  | 49 | + | + |
| 68 | 2 | 4 | 4 | 49 | 1 | - | - |
| 69 |  |  |  |  | 1 | + | - |
| 70 |  |  |  |  | 1 | - | + |
| 71 |  |  |  |  | 1 | + | + |
| 72 |  |  |  |  | 17 | - | - |
| 73 |  |  |  |  | 17 | + | - |
| 74 |  |  |  |  | 17 | - | + |
| 75 |  |  |  |  | 17 | + | + |
| 76 |  |  |  |  | 33 | - | - |
| 77 |  |  |  |  | 33 | + | - |
| 78 |  |  |  |  | 33 | - | + |
| 79 |  |  |  |  | 33 | + | + |
| 80 |  |  |  |  | 49 | - | - |
| 81 |  |  |  |  | 49 | + | - |
| 82 |  |  |  |  | 49 | - | + |
| 83 |  |  |  |  | 49 | + | + |
| 84 | 3 | 8 | 8 | 1 | 1 | - | - |
| 85 |  |  |  |  | 1 | + | - |
| 86 |  |  |  |  | 1 | - | + |
| 87 |  |  |  |  | 1 | + | + |
| 88 |  |  |  |  | 257 | - | - |
| 89 |  |  |  |  | 257 | + | - |
| 90 |  |  |  |  | 257 | - | + |
| 91 |  |  |  |  | 257 | + | + |
| 92 |  |  |  |  | 513 | - | - |
| 93 |  |  |  |  | 513 | + | - |
| 94 |  |  |  |  | 513 | - | + |
| 95 |  |  |  |  | 513 | + | + |
| 96 | 3 | 8 | 8 | 257 | 1 | - | - |
| 97 |  |  |  |  | 1 | + | - |
| 98 |  |  |  |  | 1 | - | + |
| 99 |  |  |  |  | 1 | + | + |
| 100 |  |  |  |  | 257 | - | - |
| 101 |  |  |  |  | 257 | + | - |
| 102 |  |  |  |  | 257 | - | + |
| 103 |  |  |  |  | 257 | + | + |
| 104 |  |  |  |  | 513 | - | - |
| 105 |  |  |  |  | 513 | + | - |
| 106 |  |  |  |  | 513 | - | + |
| 107 |  |  |  |  | 513 | + | + |
| 108 | 3 | 8 | 8 | 513 | 1 | - | - |
| 109 |  |  |  |  | 1 | + | - |
| 110 |  |  |  |  | 1 | - | + |
| 111 |  |  |  |  | 1 | + | + |
| 112 |  |  |  |  | 257 | - | - |
| 113 |  |  |  |  | 257 | + | - |
| 114 |  |  |  |  | 257 | - | + |
| 115 |  |  |  |  | 257 | + | + |
| 116 |  |  |  |  | 513 | - | - |
| 117 |  |  |  |  | 513 | + | - |
| 118 |  |  |  |  | 513 | - | + |
| 119 |  |  |  |  | 513 | + | + |
| 120 | 4 | 12 | 12 | 0 | 0 | - | - |
| 121 |  |  |  |  |  | + | - |
| 122 |  |  |  |  |  | - | + |
| 123 |  |  |  |  |  | + | + |
| 124 | 5 | 16 | 16 | 0 | 0 | - | - |
| 125 |  |  |  |  |  | + | - |
| 126 |  |  |  |  |  | - | + |
| 127 |  |  |  |  |  | + | + |

Para sa karagdagang impormasyon at background sa triplet encoding pakitingnan ang seksyon 5.11 ng MTX proposal [ [*MTX*](https://www.w3.org/TR/WOFF2/#ref-mtx) ].

### **5.3. Binago ang format ng loca table**

Ang data ng loca table ay maaaring ipakita sa WOFF2 file sa isa sa dalawang format na tinukoy ng numero ng bersyon ng pagbabagong-anyo (naka-encode sa mga bits ng flag ng direktoryo ng talahanayan, tingnan ang [subclause 4.1](https://www.w3.org/TR/WOFF2/#table_dir_format) para sa mga detalye). Ang bersyon ng pagbabagong "3" ay tumutukoy sa isang null na pagbabago kung saan ang nilalaman ng talahanayan ng loca ay ipinakita sa orihinal at hindi binagong format nito. Ang bersyon ng pagbabagong "0", bagama't opsyonal, DAPAT ilapat sa data ng loca table sa tuwing mababago ang data ng talahanayan ng glyf . Sa madaling salita, parehong glyf at locaang mga talahanayan ay dapat na naroroon sa kanilang binagong format o may null transform na inilapat sa parehong mga talahanayan.

Ang bersyon na "0" ng pagbabago ng talahanayan ng loca (tulad ng tinukoy ng mga bits ng flag ng direktoryo ng talahanayan, tingnan ang [subclause 4.1](https://www.w3.org/TR/WOFF2/#table_dir_format) para sa mga detalye) ay tinukoy sa ibaba.

DAPAT palaging zero ang transformLength ng binagong loca table . DAPAT ang origLength ay ang naaangkop na laki (tinutukoy ng numGlyphs+1, beses ang laki sa bawat glyph, kung saan ang laki sa bawat glyph ay dalawang byte kapag ang indexFormat (tinukoy sa [subclause 5.1. Transformed glyf table format](https://www.w3.org/TR/WOFF2/#glyf_table_format) ) ay zero, kung hindi man ay apat na byte). Kung ang transformLength ng binagong loca table ay hindi katumbas ng zero, o kung ang naka-encode na origLength ay hindi tumutugma sa kalkuladong laki na tinukoy sa itaas ng decoder DAPAT tanggihan ang WOFF2 file bilang invalid .

Ang loca table ay DAPAT muling buuin kapag ang glyf table ay na-decode . Ang proseso para sa muling pagtatayo ng loca table ay tinukoy sa [subclause 5.1](https://www.w3.org/TR/WOFF2/#glyf_table_format) bilang bahagi ng binagong glyf table decoding na proseso. Para sa mga na-reconstruct na glyph record, DAPAT na iimbak ng isang decoder ang kaukulang mga offset para sa mga indibidwal na glyph gamit ang isang format na isinasaad ng field ng indexFormat ng Transformed glyf Table .

### **5.4. Binago ang format ng talahanayan ng hmtx**

Ang data ng talahanayan ng hmtx ay maaaring ipakita sa WOFF2 file sa isa sa dalawang format na tinukoy ng numero ng bersyon ng pagbabagong-anyo (naka-encode sa mga bits ng flag ng direktoryo ng talahanayan, tingnan ang [subclause 4.1](https://www.w3.org/TR/WOFF2/#table_dir_format) para sa mga detalye). Ang bersyon ng pagbabagong "0" ay tumutukoy sa isang null na pagbabago kung saan ang nilalaman ng talahanayan ng hmtx ay ipinakita sa orihinal at hindi binagong format nito. Ang bersyon ng pagbabagong "1", na tinukoy sa ibaba, ay opsyonal at maaaring ilapat upang alisin ang ilang partikular na redundancies sa data ng talahanayan ng hmtx .

Sinasamantala ng bersyon ng pagbabagong "1" ang built-in na redundancy ng mga TrueType glyph kung saan ang mga outline ng mga glyph na idinisenyo ayon sa mga rekomendasyon ng TrueType ay malamang na ang kaliwang bahagi ay may mga value na katumbas ng xMin value ng glyph bounding box. Ang format ng hmtx table ay tinukoy sa subclause 5.2.4 ng [ [*OFF*](https://www.w3.org/TR/WOFF2/#ref-OFF) ] at may kasamang dalawang array ng left side bearing information para sa proportional at monospaced na glyph run. Ang bawat isa sa mga array na ito ay maaaring alisin mula sa orihinal na data ng input ng font at muling itayo mula sa impormasyon ng glyph bounding box kung matutugunan ang ilang partikular na kundisyon.

|  |
| --- |
| **Binago ang format ng talahanayan ng hmtx** |
| **Uri ng datos** | **Semantiko** | **Paglalarawan at uri ng halaga (kung naaangkop)** |
| UInt8 | Mga bandila | Bitfield na nagsasaad ng mga tukoy na detalye ng inilapat na pagbabago:Bit 0 - kapag itinakda, ay nagpapahiwatig na ang lsb[] array ay wala at kailangang i-reconstruct mula sa mga halaga ng xMin sa glyf table;Bit 1 - kapag itinakda, ay nagpapahiwatig na ang leftSideBearing[] array ay wala at kailangang i-reconstruct mula sa mga halaga ng xMin sa glyf table.Kapag ang hmtx transform ay ipinahiwatig ng direktoryo ng talahanayan, DAPAT itakda ang Flag (bits 0 o 1 o pareho) . Ang mga bits 2-7 ay nakalaan at DAPAT ay zero . |
| UInt16 | advanceWidth[] | Stream ng mga value ng Uint16 na kumakatawan sa mga value ng advanceWidth ng horizontal metrics array para sa mga proporsyonal na glyph. Ang laki ng stream ay katumbas ng numOfHMetrics na tinukoy sa hhea table. |
| Int16 | lsb[] | Stream ng mga value ng Int16 na kumakatawan sa mga value ng bearing sa kaliwang bahagi ng hanay ng horizontal metrics para sa mga proporsyonal na glyph. Ang stream ay naroroon lamang kung ang bit 0 ng field na "Mga Flags" ay hindi nakatakda, kung hindi, ang stream na ito ay nawawala. Ang laki ng stream ay katumbas ng numOfHMetrics na tinukoy ng hhea table. |
| Int16 | leftSideBearing[] | Stream ng mga value ng Int16 na kumakatawan sa mga value ng bearing sa kaliwang bahagi ng horizontal metrics array para sa mga monospaced na glyph. Ang stream ay naroroon lamang kung ang bit 1 ng field na "Mga Flag" ay hindi nakatakda, kung hindi, ang stream na ito ay nawawala. Ang laki ng stream ay katumbas ng (numGlyphs - numOfHMetrics) . |

Ang pagbabagong bersyon 1 na inilarawan sa subclause na ito ay opsyonal at magagamit lamang kapag ang isang input font ay TrueType-flavoured (ibig sabihin ay may glyf table), at kapag ang leftSideBearing value para sa bawat glyph sa proporsyonal o monospaced na glyph ay tumatakbong naka-encode sa hmtx eksaktong tumutugma ang talahanayan sa kaukulang mga halaga ng xMin sa talahanayan ng glyf. Ang isang espesyal na kaso ay kailangang suriin kapag ang isang naka-encode na leftSideBearing na halaga ay tumutugma sa isang walang laman na glyph, kung saan ang hmtx table transform ay maaari lamang ilapat kung leftSideBearing ng isang walang laman na glyph ay nakatakda sa zero.

Kung ang hmtx table transform ay parehong naaangkop at ninanais, DAPAT suriin ng encoder na ang leftSideBearing values ​​ay tumutugma sa mga xMin value ng glyph bounding box para sa bawat glyph sa isang font (o check that leftSideBearing == 0 para sa isang walang laman na glyph) at, para sa font mga koleksyon, DAPAT isagawa ang pagsusuring ito sa bawat katumbas na pares ng mga talahanayan ng glyf at hmtx . Kung natutugunan ang mga kundisyon para sa bawat proporsyonal o monospaced na glyph na tumatakbo , DAPAT itakda ng encoder ang hmtx transform version number sa "1", DAPAT alisin ang kaukulang array mula sa hmtx table at DAPAT itakda ang naaangkop na Flags bits.Sa mga koleksyon ng font na may nakabahaging hmtx table, DAPAT suriin ang tugma sa pagitan ng mga xMin value ng glyph bounding boxes at lsftSideBearing values ​​para sa lahat ng kumbinasyon ng shared hmtx at iba't ibang glyf table , at kung mayroong mismatch ang hmtx table transform ay HINDI DAPAT ilapat .

Kapag ang mga flag ng direktoryo ng talahanayan ay nagpapahiwatig na ang hmtx transform ay inilapat, ang User Agent ay DAPAT suriin ang mga halaga ng Flags at DAPAT tanggihan ang WOFF2 file kung ang mga Flag ay hindi wasto . DAPAT muling buuin ng decoder ang mga array ng lsb[] at/o leftSideBearing[] mula sa mga halaga ng xMin na ibinigay ng impormasyon ng glyph bounding box mula sa glyf table .

### **5.5. Mga hadlang sa pagkakasunud-sunod ng talahanayan**

Ang mga sumusunod na hadlang sa mga wastong WOFF2 file ay nilayon upang mapadali ang isang WOFF 2.0 file transfer at decoding na proseso na mahusay sa memorya.

DAPAT sundin ng loca table ang glyf table sa direktoryo ng talahanayan. Kapag ang WOFF2 file ay naglalaman ng indibidwal na naka-encode na font file, MAAARING maglaman ang direktoryo ng talahanayan ng iba pang mga talahanayan na ipinasok sa pagitan ng mga talahanayan ng glyf at loca ; gayunpaman kapag ang WOFF2 ay naglalaman ng isang file ng koleksyon ng font , DAPAT agad na sundin ng bawat loca table ang kaukulang glyf table nito . Halimbawa, ang sumusunod na pagkakasunud-sunod ng mga talahanayan: 'cmap', 'glyf', 'hhea', 'hmtx', 'loca', 'maxp' ... ay katanggap-tanggap para sa indibidwal na naka-encode na mga file ng font; gayunpaman, ang mga file ng koleksyon ng font ay dapat may glyf / locamga talahanayan na ipinares sa isa't isa, bagama't hindi sila kailangang i-order bilang magkadikit na bloke ng mga talahanayan, hal: 'cmap', 'glyf', 'loca', 'hhea', 'hmtx', 'glyf', 'loca ', 'maxp', 'post' ... ay katanggap-tanggap. Ang mga karagdagang hadlang sa talahanayan ng loca ay ibinibigay sa [subclause 5.3](https://www.w3.org/TR/WOFF2/#loca_table_format) .

## **6. Pinalawak na Metadata Block**

MAAARING maglaman ang WOFF2 file ng isang bloke ng pinahabang metadata . Ang interpretasyon ng bloke na ito ay eksaktong kapareho ng [ [WOFF 1](https://www.w3.org/TR/WOFF2/#ref-woff10) ].

Gayunpaman, habang nasa isang WOFF 1 file ang pinahabang metadata block ay iniimbak na may zlib compression, sa isang WOFF2 file DAPAT na naka-imbak ang metadata na naka-compress gamit ang [ *[Brotli](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22ref-Brotli)* ] bilang isang hiwalay na stream, kung mayroon. . Ang katwiran ng pagbabagong ito ay upang i-minimize ang kabuuang bilang ng mga byte-level na compression algorithm na kailangan para ipatupad ang WOFF2, at dahil din sa Brotli ay inaasahang makakamit ang mas mahusay na compression ratio kaysa sa zlib sa karamihan ng mga kaso.

DAPAT sundin kaagad ang metadata block pagkatapos ng stream ng data ng naka-compress na font . Ang simula ng metadata block ay DAPAT palaging nakahanay sa 4-byte; kapag naroroon ang pinahabang metadata, ang dulo ng naka-compress na bloke ng data ay dapat na may padded ng hanggang tatlong null byte kung kinakailangan upang maabot ang isang 4-byte na hangganan . Kung ang metadata block ay ang huling block sa WOFF file, DAPAT walang karagdagang padding pagkatapos ng dulo ng block .

## **7. Pribadong Data Block**

MAAARING magsama ang WOFF2 file ng isang bloke ng arbitrary na data , na nagpapahintulot sa mga tagalikha ng font na isama ang anumang impormasyong nais nila. Ang nilalaman ng data na ito ay HINDI DAPAT makakaapekto sa paggamit ng font o pag-uugali ng pag-load ng mga ahente ng gumagamit . Ang mga ahente ng gumagamit ay hindi dapat gumawa ng mga pagpapalagay tungkol sa nilalaman ng isang pribadong bloke; maaari itong (halimbawa) naglalaman ng ASCII o Unicode text, o ilang binary data na tinukoy ng vendor, at maaari itong i-compress o i-encrypt, ngunit wala itong pampublikong tinukoy na format. Hindi ipapalagay ng mga umaayon sa user agent ang anumang bagay tungkol sa istruktura ng data na ito. Tanging ang developer ng font o vendor na responsable para sa pribadong block ang inaasahang makakaunawa sa mga nilalaman nito.

Ang pribadong bloke ng data, kung naroroon, DAPAT ang huling bloke sa WOFF file, kasunod ng naka-compress na stream ng data ng font at ang pinalawig na bloke ng metadata, kung naroroon ang isa . DAPAT magsimula ang pribadong bloke ng data sa isang 4-byte na hangganan sa WOFF file, na may hanggang tatlong null byte na ipinasok bilang padding pagkatapos ng anumang naunang bloke, kung kinakailangan upang matiyak ito . Ang dulo ng pribadong data block ay DAPAT tumutugma sa dulo ng WOFF2 file .

## **A. Pagpaparehistro ng Uri ng Internet Media**

Ang apendiks na ito ay nagrerehistro ng bagong uri ng Internet Media, sa ilalim ng uri ng font top level [ [*RFC8081*](https://www.w3.org/TR/WOFF2/#ref-RFC-8081) ]. Ang apendiks na ito ay umaayon sa [BCP 13](http://www.ietf.org/rfc/rfc4288.txt) at [W3CRegMedia](https://www.w3.org/2002/06/registering-mediatype.html) .

**WOFF 2.0**

**Pangalan ng uri:**

font

**Pangalan ng subtype:**

woff2

**Mga kinakailangang parameter:**

wala.

**Opsyonal na mga parameter:**

wala.

**Mga pagsasaalang-alang sa pag-encode:**

Binary.

**Mga pagsasaalang-alang sa interoperability:**

Ang WOFF 2.0 ay isang pagpapabuti sa WOFF 1.0. Ang dalawang format ay may magkaibang Uri ng Internet Media at maaaring gamitin nang magkatulad.

**Nai-publish na detalye:**

Ang pagpaparehistro ng uri ng media na ito ay nakuha mula sa pagtutukoy ng [WOFF 2.0](https://www.w3.org/TR/WOFF2/) sa W3C.

**Mga application na gumagamit ng ganitong uri ng media:**

Ang WOFF 2.0 ay ginagamit ng mga Web browser, kadalasang kasama ng HTML at CSS.

**Karagdagang impormasyon:**

**(mga) magic number:**

Ang signature field sa WOFF header ay DAPAT maglaman ng "magic number" 0x774F4632 ('wOF2')

**(mga) extension ng file:**

woff2

**Macintosh file type code(s):**

(walang tinukoy na code)

**Macintosh Universal Type Identifier code:**

org.w3c.woff2

**@font-face Format:**

woff2

**Mga Identifier ng Fragment**

Tingnan ang Seksyon 4.2 ng [ [RFC8081](https://www.w3.org/TR/WOFF2/#ref-RFC-8081) ]

**Tao at email address na makontak para sa karagdagang impormasyon:**

Chris Lilley (www-font@w3.org).

**Nilalayon na paggamit:**

KARANIWAN

**Mga paghihigpit sa paggamit:**

wala

**May-akda:**

Ang pagtutukoy ng WOFF2 ay isang produkto ng trabaho ng WebFonts Working Group ng World Wide Web Consortium.

**Baguhin ang controller:**

Ang W3C ay may kontrol sa pagbabago sa detalyeng ito.

## **B. Mga Pagbabago**

Ang apendiks na ito ay nagbibigay kaalaman .

### **Mga pagbabago mula noong [Rekomendasyon ng W3C](https://www.w3.org/TR/2021/REC-WOFF2-20210706/) noong Hulyo 6, 2021**

* Ang mga nakaraang Pagwawasto ng Kandidato 1, 3 5 at 6 ay itinuring na editoryal at pinagsama-sama
* Nakaraang Pagwawasto ng Kandidato 2, hinggil sa flag na nagsasapawan ng symple glyph, ay [Iminungkahing Pagwawasto 1 na ngayon](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22p1)
* Nakaraang Pagwawasto ng Kandidato 4, tungkol sa polarity ng on-curve flag bit, ngayon ay [Iminungkahing Pagwawasto 2](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22p2)
* Na-update ang mga sanggunian sa pinakabagong mga bersyon

### **Mga pagbabago mula noong [Rekomendasyon](https://www.w3.org/TR/2018/REC-WOFF2-20180301/) noong Marso 1, 2018**

* Pagwawasto ng Kandidato 1: [para sa isang typo](https://www.w3.org/TR/2021/REC-WOFF2-20210706/#c1)
* Pagwawasto 2 ng Kandidato: [panatilihin ang opsyonal na symple glyph overlap na flag](https://www.w3.org/TR/2021/REC-WOFF2-20210706/%22%20%5Cl%20%22c2)
* Pagwawasto ng Kandidato 3: [ang kahulugan ng uri ng UInt32 ay tinanggal](https://www.w3.org/TR/2021/REC-WOFF2-20210706/#c3)
* Pagwawasto ng Kandidato 4: [polarity ng on-curve flag bit](https://www.w3.org/TR/2021/REC-WOFF2-20210706/#c4)
* Pagwawasto ng Kandidato 5: [typo sa list numbering](https://www.w3.org/TR/2021/REC-WOFF2-20210706/#c5)
* (Editoryal) na-update ang mga link ng IETF

### **Mga pagbabago mula noong [Rekomendasyon](https://www.w3.org/TR/2016/CR-WOFF2-20160315/) ng Kandidato noong Marso 15, 2016**

* Na-update ang sanggunian ng Brotli mula sa isang Draft sa Internet patungo sa Informational RFC.
* Na-update ang uri ng top-level na sanggunian ng font mula sa isang Draft sa internet patungo sa Iminungkahing Standard RFC.
* Pagbanggit ng mga fragment identifier na na-update upang umayon sa RFC 8081 (ang normative definition), na nagreresulta sa pagbabago sa Internet Media Type appendix at pagtanggal ng isang kinakailangan upang mapanatili ang kaayusan sa mga koleksyon ng font.
* Ang CSS Fonts 3 ay ginamit nang normatibo ( dapat ), kaya inilipat mula sa Informative tungo sa Normative references section.

### **Mga pagbabago mula noong**[**Working Draft**](https://www.w3.org/TR/2015/WD-WOFF2-20150414/)**noong Abril 14, 2015**

* Nagdagdag ng tala na naglalarawan ng mga dahilan para sa mga posibleng variation ng na-reconstruct na glyf table data at nilinaw na ang pagtukoy ng sapat na laki ng memory allocations para sa reconstructed glyf table data ay isang responsibilidad ng mga pagpapatupad ng WOFF2 decoder.
* Nagdagdag ng tala na naglalarawan sa opsyonal na hakbang sa pre-processing na maaaring ilapat sa OpenType CFF font, kung saan ang proseso ng CFF data de-subroutinization ay maaaring ilapat bago ang WOFF2 encoding upang mabawasan ang naka-compress na laki ng font.
* Inalis ang ideya ng "nominal na laki" para sa na-decompress na WOFF 2.0 na font dahil ang laki ng font ay maaaring mag-iba nang malaki batay sa mga inilapat na hakbang sa pag-optimize ng data, at inalis ang mga kinakailangan sa pagsunod na nauugnay sa mga paglalaan ng memorya ng decoder at ang orihinal na kahulugan ng laki ng glyf table.
* Nagdagdag ng indikasyon ng bersyon ng pagbabagong-anyo sa mga flag ng direktoryo ng Talahanayan at nagdagdag ng bagong subclause 5.4 na may paglalarawan ng pagbabago sa talahanayan na 'hmtx'.
* Na-update na indikasyon ng bersyon ng pagbabago sa mga flag ng direktoryo ng Table na nagdaragdag ng mga null trnsform na probisyon para sa mga talahanayang 'glyf' at 'loca', nagdagdag ng mga paglilinaw para sa pagbabago ng nakabahaging talahanayang 'hmtx'.
* Nagdagdag ng pseudo-code para sa uri ng data ng UIntBase128, iba't ibang mga update sa editoryal.
* Na-update ang paglalarawan ng walang laman na proseso ng reconstruction ng glyph sa 'glyf' table transform, nagdagdag ng karagdagang check sa 'hmtx' transform para sa left side bearing value ng isang walang laman na glyph, minor editorial updates.
* Nagdagdag ng mga bagong kinakailangan sa paghaharap para sa pagsubok ng mga kundisyon ng mismatch kung saan ang kabuuang laki ng font at ang orihinal na laki ng talahanayang 'glyf' ay naiiba sa mga sukat na idineklara bilang bahagi ng WOFF2 file header o direktoryo ng talahanayan.
* Mga Idinagdag na Pagsasaalang-alang para sa Apendise ng Seguridad at Pagkapribado.
* Na-refer ang bagong font/\* uri ng pinakamataas na antas na binuo sa IETF, na-trimmed ang Internet Media Type registration appendix sa WOFF 2.0 lang.

### **Mga pagbabago mula noong**[**First Public Working Draft**](https://www.w3.org/TR/2014/WD-WOFF2-20140508/)**noong Mayo 8, 2014**

* Nilinaw na ang spec ay nag-uutos sa pag-uuri ng direktoryo ng talahanayan, habang ang mismong pagkakasunud-sunod ng talahanayan ay dapat sumunod sa inirekumendang order ng OpenType/OFF.
* Nagdagdag ng mga kinakailangan para sa mga decoder upang tanggapin ang anumang posibleng mga variation ng pag-encode para sa 255UInt16 uri ng data, inirerekomenda na ang mga encoder ay gumawa ng mas maiikling 255UInt16 byte na pagkakasunud-sunod hangga't maaari.
* Pinalitan ng pseudo-code ang paglalarawang nakabatay sa talahanayan ng 255UInt16 na uri ng data.
* Nagdagdag ng mga kinakailangan sa pagsunod para sa uri ng data ng UIntBase128.
* Nagdagdag ng mga kinakailangan sa pagsunod sa paglalarawan ng WOFF 2.0 Header na nag-uutos sa nakareserbang halaga na itakda sa zero.
* Nagdagdag ng rekomendasyon at kinakailangan sa pagsunod para sa mga WOFF 2.0 na encoder upang alisin ang DSIG table at itakda ang flag (bit 11) ng font na 'head' table.
* Nagdagdag ng kinakailangan sa pagsunod para sa mga decoder upang mag-imbak ng mga glyph record offset bilang pagsunod sa tinukoy na indexFormat.
* Nagdagdag ng mga kinakailangan sa pagsunod upang mag-utos ng mga implicit na kalkulasyon ng glyph bounding box.
* Nilinaw ang nilalamang "Nabagong 'glyf' table" na nagdaragdag ng mga tahasang sanggunian sa bawat stream ng data at sa kanilang pangkalahatang pagkakasunud-sunod.
* Binago ang subclause na "Nabagong 'glyf' table" na nagdaragdag ng detalyadong paglalarawan ng proseso ng muling pagtatayo ng glyph para sa mga walang laman, simple at pinagsama-samang mga glyph.
* Nagdagdag ng suporta para sa mga koleksyon ng font sa WOFF 2.0. Bilang bahagi ng suporta sa pagkolekta ng font, nagdagdag ng bagong kinakailangan upang i-encode ang 'glyf' at 'loca' na mga talahanayan bilang mga pares, na may 'loca' na kaagad na sumusunod sa 'glyf' na talahanayan.
* Inalis ang mga kinakailangan sa pagsunod para sa mga kahulugan ng uri ng data at muling inuri ang ilang partikular na kinakailangan ng user agent bilang partikular sa decoder.
* Nilinaw ang kahulugan ng hindi naka-compress na laki ng data ng font.
* Nilinaw ang mga pagkakaiba sa mga kalkulasyon ng bounding box para sa simple at composite na mga glyph.
* Nagdagdag ng 4-byte na mga kinakailangan sa alignment para sa Extended Metadata at Private data blocks upang tumugma sa mga kinakailangan sa Rekomendasyon ng WOFF 1.0.
* Nagdagdag ng bagong appendix A na may draft na text ng top-level na pagpaparehistro ng uri ng media para sa bagong uri ng 'font'.
* Idinagdag itong "Mga Pagbabago" na apendise.

## **C. Mga Pagsasaalang-alang para sa Seguridad at Pagkapribado**

Ang apendiks na ito ay nagbibigay- kaalaman .

Ang W3C TAG ay bumubuo ng isang [Self-Review Questionnaire: Seguridad at Privacy](https://w3ctag.github.io/security-questionnaire/) para sa mga editor ng mga detalye upang masagot nang may kaalaman.

Alinsunod sa Mga [Tanong na Dapat Isaalang-alang](https://w3ctag.github.io/security-questionnaire/%22%20%5Cl%20%22questions)

1. Nakikitungo ba ang detalyeng ito sa impormasyong nagbibigay ng personal na pagkakakilanlan?

Hindi.

1. Nakikitungo ba ang detalyeng ito sa data na may mataas na halaga?

Hindi. Ang impormasyon ng credit card at mga katulad nito ay hindi nakaimbak sa WOFF2.

1. Nagpapakilala ba ang detalyeng ito ng bagong estado para sa isang pinagmulan na nagpapatuloy sa mga session ng pagba-browse?

Hindi. Sa partikular, ang mga WOFF2 na font ay magagamit lamang sa mga dokumentong tumutukoy sa kanila; hindi sila [nagpapatuloy](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22conform-css3font-available) para sa paggamit ng iba pang mga application o dokumento sa system ng user.

1. Inilalantad ba ng detalyeng ito ang tuluy-tuloy, cross-origin state sa web?

Hindi.

1. Inilalantad ba ng detalyeng ito ang anumang iba pang data sa isang pinanggalingan na kasalukuyang wala itong access?

Hindi.

1. Pinapagana ba ng detalyeng ito ang mga bagong mekanismo ng pagpapatupad/paglo-load ng script?

Hindi. Ang mga WOFF2 na font ay hindi naglalaman, o naka-link sa, mga script.

1. Nagbibigay-daan ba ang detalyeng ito sa isang pinanggalingang pag-access sa lokasyon ng isang user?

Hindi.

1. Nagbibigay-daan ba ang detalyeng ito sa isang pinanggalingang pag-access sa mga sensor sa device ng isang user?

Hindi.

1. Nagbibigay-daan ba ang pagtutukoy na ito sa pinagmulang pag-access sa mga aspeto ng lokal na kapaligiran sa pag-compute ng user?

Hindi. Ang mga WOFF2 na font ay karaniwang binubuo at iniimbak sa isang server, at walang access sa kapaligiran ng user kapag nabuo. Kapag na-decompress na, wala silang aktibong code at hindi masusuri ang kapaligiran ng user.

1. Nagbibigay-daan ba ang pagtutukoy na ito sa pinagmulang pag-access sa iba pang mga device?

Hindi.

1. Pinapayagan ba ng detalyeng ito ang pinagmulan ng ilang sukat ng kontrol sa native UI ng isang user agent?

Hindi.

1. Inilalantad ba ng detalyeng ito ang mga pansamantalang identifier sa web?

Hindi.

1. Nakikilala ba ng detalyeng ito ang pag-uugali sa mga konteksto ng first-party at third-party?

Hindi.

1. Paano dapat gumana ang detalyeng ito sa konteksto ng mode na "incognito" ng isang user agent?

Walang iba.

1. Nagpatuloy ba ang detalyeng ito ng data sa lokal na device ng isang user?

Hindi.

1. Mayroon bang seksyong "Mga Pagsasaalang-alang sa Seguridad" at "Mga Pagsasaalang-alang sa Privacy" ang detalyeng ito?

Oo.

1. Pinapayagan ba ng detalyeng ito ang pag-downgrade ng mga default na katangian ng seguridad?

Hindi.

Ang seksyong Seguridad ng uri ng nangungunang antas ng font [ [*RFC8081*](https://www.w3.org/TR/WOFF2/#ref-RFC-8081) ] ay naglalaman ng karagdagang impormasyong nauugnay sa seguridad na nauugnay sa WOFF2.

Gumagamit ang WOFF2 ng Brotli compression. Ang seksyong Mga Pagsasaalang-alang sa Seguridad ng detalye ng Brotli [ *[Brotli](https://www.w3.org/TR/WOFF2/%22%20%5Cl%20%22ref-Brotli)* ] ay dapat konsultahin, bilang karagdagan sa apendiks na ito.

## **D. Mga Sanggunian**

### **D.1 Mga Sanggunian sa Normatibo**

**[WOFF1]**

[*WOFF File Format 1.0*](https://www.w3.org/TR/2012/REC-WOFF-20121213/) Jonathan Kew, Tal Leming, Erik van Blokland, Mga Editor. World Wide Web Consortium, 13 Disyembre 2012. Ang pinakabagong bersyon ng pagtutukoy ng WOFF ay makukuha sa [*http://www.w3.org/TR/WOFF/*](https://www.w3.org/TR/WOFF/)

**[OFF]**

[*Buksan ang detalye ng Font Format*](https://standards.iso.org/ittf/PubliclyAvailableStandards/c074461_ISO_IEC_14496-22_2019.zip) (ISO/IEC 14496-22:2019). kasama [*ng Amendment 1: Color font technology at iba pang update*](https://standards.iso.org/ittf/PubliclyAvailableStandards/c078865_ISO_IEC_14496-22_2019_Amd_1_2020%28en%29.zip)

**[Brotli]**

[*Brotli Compressed Data Format, RFC 7932*](https://datatracker.ietf.org/doc/html/rfc7932) J. Alakuijala at Z. Szabadka, Mga Editor. Internet Engineering Task Force, Informational, Hulyo 2016.

**[RFC2119]**

[*RFC 2119: Mga*](https://datatracker.ietf.org/doc/html/rfc2119) pangunahing salita para sa paggamit sa mga RFC upang Ipahiwatig ang Mga Antas ng Kinakailangan. S. Bradner, Editor. Internet Engineering Task Force, Pinakamahusay na Kasalukuyang kasanayan, Marso 1997.

**[RFC8081]**

[*Ang uri ng font na Top Level, RFC 8081*](https://datatracker.ietf.org/doc/html/rfc8081) C. Lilley, (Editor). Internet Engineering Task Force, Iminungkahing Pamantayan, Pebrero 2017.

 **[CSS3-Mga Font]**

[*CSS Fonts Module Level 3*](https://www.w3.org/TR/css-fonts-3/) , John Daggett, Myles C. Maxfield, Chris Lilley (Mga Editor). World Wide Web Consortium Recommendation, 2018. Ang pinakabagong bersyon ng CSS3 Fonts ay available sa [*http://www.w3.org/TR/css-fonts-3/*](https://www.w3.org/TR/css-fonts-3/)

### **D.2 Mga Sanggunian sa Impormasyon**

**[WOFF2ER]**

[*Ulat sa Pagsusuri ng WOFF 2.0*](https://www.w3.org/TR/WOFF20ER/) Chris Lilley, Editor. World Wide Web Consortium, 2015.

**[OpenType]**

[*Pagtutukoy ng Microsoft OpenType*](http://www.microsoft.com/typography/otspec/) . Ang OpenType ay isang rehistradong trademark ng Microsoft Corporation.

**[TrueType]**

[*Apple TrueType Reference manual*](https://developer.apple.com/fonts/TrueType-Reference-Manual/) . Apple, 2014. Ang TrueType ay isang rehistradong trademark ng Apple, Inc.

**[MTX]**

[*Format ng Font ng MicroType® Express (MTX)*](https://www.w3.org/Submission/2008/SUBM-MTX-20080305/) Sarah Martin, Al Ristow, Steve Martin, Vladimir Levantovsky, at Christopher Chapman. Pagsusumite ng Miyembro ng W3C, 5 Marso 2008.