ISO/IEC 19794-2:2011/Cor.1:2012 Summary

Fingerprint templates » ISO 19794-2:2011

This is a complete description of fingerprint template format defined in *ISO/IEC 19794-2:2011* spec (often abbreviated *ISO 19794-2:2011* or ambiguously *ISO 19794-2*), including minor fixes described in *Technical Corrigendum 1* (*Cor.1:2012*), except for the following changes:

- Smartcard variation of this format is described separately.
- Essential content from ISO 19794-1:2011 was incorporated to keep this format summary self-contained.
- All content is distributed under permissive CC BY 4.0 license.
- Normative content was completely rewritten from scratch to avoid infringing copyrights.
- Non-normative commentary was dropped. Author added his own non-normative comments.
- Special notes were added for opensource and in-house software developers.
- Differences from other formats and other versions of this format were documented.

This format summary is complete in the sense that implementations written according to this document are also fully conforming implementations of ISO/IEC 19794-2:2011. There is no guarantee though and you should consult the original ISO/IEC 19794-2:2011 spec if in doubt.

This document was written by Robert Važan as part of his work on SourceAFIS fingerprint matcher using his legally obtained copy of ISO/IEC 19794-2:2011 and ISO/IEC 19794-1:2011. Author has no affiliation with ISO. This format summary is distributed free of charge under Creative Commons Attribution 4.0 International License to enable truly open implementation of the format in opensource software. Nevertheless, author discourages use of ISO 19794-2 and other so-called "standard" templates in favor of plain fingerprint images.

Introduction

Human *fingerprint recognition* is usually performed in two separate steps: *feature extraction* and *matching*. Feature extraction takes fingerprint image (also called *fingerprint sample*), usually obtained from *fingerprint reader* (or *fingerprint sensor*) hardware, and produces so-called *fingerprint template* filled with fingerprint *features* useful for matching. Matching takes two fingerprint templates on input, compares fingerprint features contained in them, and produces *similarity score* on output, which is then compared to a *threshold* to produce *match* or *non-match* decision. There are many kinds of fingerprint features: *minutiae* (especially ridge *endings* and *bifurcations*), *core* and *delta* points, ridge shapes or *patterns*, orientation and ridge frequency fields, *edges* between minutiae, and many others.

Most fingerprint recognition systems have their own native fingerprint template format. Some people however insist on exchange of fingerprint templates between fingerprint recognition systems even though it is almost always a bad idea. ISO 19794-2:2011 is one of several fingerprint template formats that can be used for this purpose.

ISO 19794-2:2011 mainly encodes minutiae, including their position, angle, and type (ending, bifurcation, or other). It can optionally encode ridge counts on edges between minutiae, information about cores and deltas, and zonal quality data. It carries some fingerprint *metadata*, notably *finger position* (thumb, index, ...), *resolution*, and image size. It permits encoding several fingerprints in a single template.

Related formats

Smartcard variation of ISO 19794-2:2011 is described separately, because it is substantially different.

This template format supersedes earlier version defined in ISO 19794-2:2005. It can be differentiated from the earlier version by looking at the VERSION field, which is "030\0" in this version and " 20\0" in the previous version.

This format shares MAGIC and VERSION fields with ANSI 378-2009. The only way to differentiate the two formats is to examine other fields, for example check whether FPBYTES fields add up to TOTALBYTES.

This is a modality-specific implementation of common biometric record structure defined in ISO/IEC 19794-1:2011, including optional fields DEVVENDOR, DEVID, QCOUNT, QRECORD, CERTCOUNT, and CERTIFICATE.

When embedded in CBEFF (*Common Biometric Exchange Framework Format*, ISO/IEC 19785-3:2007) *Biometric Data Block* (BDB), this template format will have *CBEFF Format Owner* 0x101 (decimal 257), which is registered to *ISO/IEC JTC 1 SC 37-Biometrics*. *CBEFF Format Type* registered for ISO 19794-2:2011 is 0x1d (decimal 29). Previous version of this format, ISO 19794-2:2005, used CBEFF Format Type of 1 and 2.

Changes

This format is incompatible with its previous version, ISO 19794-2:2005. Differences are listed below:

- VERSION field was changed from " 20\0" to "030\0".

- Fields DEVID, WIDTH, HEIGHT, RESOLUTIONX, and RESOLUTIONY were moved from HEADER to FINGERPRINT.
- Added FPBYTES, DATETIME, DEVTECH, DEVVENDOR, MINBYTES, ENDINGTYPE, ZONEVENDOR, and ZONEALGO.
- Added certification fields HASCERTS, CERTCOUNT, CERTIFICATE (CERTIFIEDBY, CERTTYPE) that replace removed DEVSTAMP.
- FPQUALITY may repeat QCOUNT times as part of new QRECORD that also includes new QALGO field.
- Fields DEVID, VIEWOFFSET, SAMPLETYPE, and FPCOUNT have different size.
- POSITION has more allowed values.
- Reserved byte after HEADER was removed.
- Some fields in FINGERPRINT header were reordered.
- FPCOUNT and MINCOUNT must be at least 1.
- Values in WIDTH and HEIGHT were constrained.
- ENDINGTYPE allows choice between ridge skeleton endpoint and valley skeleton bifurcation as a location for ending minutia (see MINX/MINY). Previous version of the format allowed such choice only for on-card format.
- Meaning of MINTYPE value 00 "other" was clarified.
- There's a recommended algorithm for calculating MINANGLE. Handling of trifurcations is defined.
- MINQUALITY may be omitted from the template (see MINBYTES). If present, it has different range of values and recommended interpretation.
- EXTLEN contains length of the whole EXTENSION instead of just EXTDATA.
- In ridge count extension, quadrant and octant alignment to minutia direction is now specified (see STARTYPE).
- Presence and ordering requirements for EDGEDEF records have changed.
- In ridge count extension, placeholder records set EDGETO and RIDGECOUNT fields to 255 rather than zero.
- Value of the RIDGECOUNT field is higher by one and algorithm to compute it is defined.
- COREANGLE and DELTAANGLE calculation is implementation-defined.
- ZONEBITS has different range of values.

Layout

The template is a binary file consisting of simple fields arranged in field groups. Fields or field groups may repeat. There are no keys or tags and fields are identified only by their position in the template. Numbers are in big-endian byte order.

Repeats	Length	Code	Title
	54+ bytes		ISO/IEC 19794-2:2011 template
once	15 bytes	HEADER	Template header
once	4 bytes	MAGIC	File signature / magic number
once	4 bytes	VERSION	Format version
once	4 bytes	TOTALBYTES	Total template length in bytes
once	2 bytes	FPCOUNT	Number of fingerprints
once	1 byte	HASCERTS	Device certification flag
FPCOUNT	39+ bytes	FINGERPRINT	Fingerprint
once	4 bytes	FPBYTES	Fingerprint length in bytes
once	9 bytes	DATETIME	Capture date and time
once	2 bytes	YEAR	Year
once	1 byte	MONTH	Month
once	1 byte	DAY	Day of month
once	1 byte	HOUR	Hour
once	1 byte	MINUTE	Minute
once	1 byte	SECOND	Second
once	2 bytes	MILLISECOND	Millisecond
once	1 byte	DEVTECH	Sensor technology
once	2 bytes	DEVVENDOR	Sensor vendor ID
once	2 bytes	DEVID	Sensor ID
once	1 byte	QCOUNT	Number of quality records
QCOUNT	5 bytes	QRECORD	Quality record
once	1 byte	FPQUALITY	Fingerprint quality
once	2 bytes	QVENDOR	Quality algorithm vendor
once	2 bytes	QALGO	Quality algorithm
optional	1 byte	CERTCOUNT	Number of certification records
CERTCOUNT	3 bytes	CERTIFICATE	Certification record
once	2 bytes	CERTIFIEDBY	Certification authority
once	1 byte	CERTTYPE	Certification scheme
once	1 byte	POSITION	Finger position on hands
once	1 byte	VIEWOFFSET	Finger view number
once	2 bytes	RESOLUTIONX	Horizontal pixel density

once	2 bytes	RESOLUTIONY	Vertical pixel density
once	1 byte	SAMPLETYPE	Impression type
once	2 bytes	WIDTH	Image width
once	2 bytes	HEIGHT	Image height
once	4 bits	MINBYTES	Length of minutia record
once	4 bits	ENDINGTYPE	Ridge ending type
once	1 byte	MINCOUNT	Number of minutiae
MINCOUNT	6 bytes	MINUTIA	Minutia
once	2 bits	MINTYPE	Minutia type
once	14 bits	MINX	Minutia X position
once	2 bits		Reserved (zeroed)
once	14 bits	MINY	Minutia Y position
once	1 byte	MINANGLE	Minutia angle
optional	1 byte	MINQUALITY	Minutia quality
once	2 bytes	EXTBYTES	Total length of extension data
0 or more	4+ bytes	EXTENSION	Extension data block
once	2 bytes	EXTTYPE	Extension type
once	2 bytes	EXTLEN	Length of extension data
once	EXTLEN - 4	EXTDATA	Extension data

There are three predefined extensions blocks. When EXTTYPE is 1, EXTDATA contains ridge count extension.

Repeats	Length	Code	Title
	3n+1 bytes	RCOUNTEXT	Ridge count extension
once	1 byte	STARTYPE	Edge picking method
0 or more	3 bytes	EDGEDEF	Edge between minutiae
once	1 byte	EDGEFROM	Starting minutia of the edge
once	1 byte	EDGETO	Ending minutia of the edge
once	1 byte	RIDGECOUNT	Number of ridges the edge crosses

When EXTTYPE is 2, EXTDATA contains core and delta extension.

Repeats	Length	Code	Title
	2+ bytes	COREDELTA	Core and delta extension
once	1+ bytes	COREDATA	Core data
once	1 byte	CORENUM	Number of cores

CORENUM	4 or 5 bytes	COREDEF	Core point
once	1 bit		Reserved (zeroed)
once	1 bit	CHASANGLE	Core angle presence flag
once	14 bits	COREX	Core X position
once	2 bits		Reserved (zeroed)
once	14 bits	COREY	Core Y position
optional	1 byte	COREANGLE	Core angle
once	1+ bytes	DELTADATA	Delta data
once	1 byte	DELTANUM	Number of deltas
DELTANUM	4 or 7 bytes	DELTADEF	Delta point
once	1 bit		Reserved (zeroed)
once	1 bit	DHASANGLE	Delta angle presence flag
once	14 bits	DELTAX	Delta X position
once	2 bits		Reserved (zeroed)
once	14 bits	DELTAY	Delta Y position
0 or 3	1 byte	DELTAANGLE	Delta angle

When EXTTYPE is 3, EXTDATA contains zonal quality extension.

Repeats	Length	Code	Title
	4+ bytes	ZONALEXT	Zonal quality extension
once	2 bytes	ZONEVENDOR	Zonal quality vendor
once	2 bytes	ZONEALGO	Zonal quality algorithm
once	1 byte	ZONEWIDTH	Zone width
once	1 byte	ZONEHEIGHT	Zone height
once	1 byte	ZONEBITS	Bits per zone
once	variable	ZONALQUALITY	Zonal quality data

Fields and field groups

Template header (HEADER)

Header contains information common to all fingerprints in the template:

- template format identification: MAGIC, VERSION,
- size information: TOTALBYTES, FPCOUNT, and
- HASCERTS flag.

This field group is identical to HEADER in ISO 19794-1:2011. In ISO 19794-2:2005, fields DEVID, WIDTH, HEIGHT, RESOLUTIONX, and RESOLUTIONY were part of HEADER, but they have been moved to FINGERPRINT block in this version of the format. Reserved byte was removed from the HEADER.

File signature / magic number (MAGIC)

First four bytes of the template contain constant string "FMR\0" where '\0' stands for zero byte. This is a magic number (or file signature) that helps biometric software to automatically distinguish this format from other file formats.

This field is identical to MAGIC in ISO 19794-1:2011 with modality-specific value "FMR\0".

Somewhat confusingly, ANSI 378 uses the same magic number.

Format version (VERSION)

Version field contains 4-byte constant string "030\0" where '\0' stands for zero byte. Every version of this format uses different version string, which makes it easy to distinguish the versions.

This field is identical to VERSION in ISO 19794-1:2011 with value "030\0" defined for this version of ISO 19794-2.

Unfortunately, ANSI 378-2009 uses the same version string "030\0" along with identical MAGIC. One way to distinguish the two formats is to check whether FPBYTES fields add up to TOTALBYTES (taking into account HEADER size). This is of course a completely ridiculous workaround. Spec authors apparently rely too much on external format identification. Either that or there is some format war going on between ANSI and ISO.

Total template length in bytes (TOTALBYTES)

Total length in bytes is redundant, because the template simply ends after the last FINGERPRINT block. It might be nevertheless useful to easily find the end of the template in a stream of bytes.

Template length is encoded in 4 bytes as a big-endian unsigned 32-bit number. Minimum valid value of TOTALBYTES is 54, assuming implementations honor minimum values for FPCOUNT and MINCOUNT.

This field is identical to TOTALBYTES in ISO 19794-1:2011 except for minimum value.

This field can be used to distinguish this format from ANSI 378-2009. In this format, FPBYTES will add up to TOTALBYTES minus HEADER size.

Number of fingerprints (FPCOUNT)

An unsigned big-endian 16-bit number that indicates the number of FINGERPRINT blocks in the template. Minimum value is 1, which means that at least one fingerprint must be present. Since every fingerprint must have a unique combination of POSITION and VIEWOFFSET, there cannot be more than 22 x 16 = 352 fingerprints.

This field is identical to COUNT in ISO 19794-1:2011 except for maximum value. Size of this field changed since ISO 19794-2:2005 and zero FPCOUNT is no longer allowed.

Device certification flag (HASCERTS)

One-byte flag indicating presence of certifications. If this field is set to 1, CERTCOUNT field is present (but it may have zero value). If this field is zero, CERTCOUNT field is omitted from the template and assumed to be zero.

This field is identical to HASCERTS in ISO 19794-1:2011 with value 1 permitted. It is a new addition since ISO 19794-2:2005.

Fingerprint (FINGERPRINT)

A single *fingerprint view*, essentially a single scan from fingerprint reader. Fingerprint block contains MINUTIA records and some metadata about the fingerprint, notably finger's POSITION on hands. Fingerprint section can optionally contain EXTENSION blocks, including ridge count data (RCOUNTEXT), core/delta data (COREDELTA), and zonal quality data (ZONALEXT).

Single template can contain multiple FINGERPRINT blocks. Their number is indicated in the FPCOUNT field. Every template must have at least one FINGERPRINT block.

All fingerprint sections in the template must have unique combination of POSITION and VIEWOFFSET. If there are multiple FINGERPRINT sections with the same POSITION, then unique VIEWOFFSET is assigned to each and they must appear in the template ordered by VIEWOFFSET. The original ISO spec doesn't require it, but it makes sense to list FINGERPRINT sections with the same POSITION together in the template.

This field group is a modality-specific implementation of SAMPLE field group from ISO 19794-1:2011, including optional fields DEVVENDOR, DEVID, QCOUNT, QRECORD, CERTCOUNT, and CERTIFICATE. It contains additional modality-specific fields POSITION, VIEWOFFSET, RESOLUTIONX, RESOLUTIONY, SAMPLETYPE, WIDTH, HEIGHT, MINBYTES, ENDINGTYPE, and MINCOUNT. Modality-specific biometric data consists of MINUTIA records, EXTBYTES, and EXTENSION blocks.

In ISO 19794-2:2005, fields DEVID, WIDTH, HEIGHT, RESOLUTIONX, and RESOLUTIONY were part of HEADER. Some fields were reordered since ISO 19794-2:2005.

Fingerprint length in bytes (FPBYTES)

Fingerprint length in bytes is redundant, because the fingerprint simply ends after its last field. It might be nevertheless useful to easily skip fingerprints in the template without having to parse them.

Fingerprint length is encoded in 4 bytes as a big-endian unsigned 32-bit number. It includes size of fingerprint headers as well as subsequent fingerprint feature data. Minimum valid value of FPBYTES is 39, assuming implementations honor minimum value for MINCOUNT.

This field is identical to SAMPLEBYTES in ISO 19794-1:2011 except for minimum value.

This field can be used to distinguish this format from ANSI 378-2009. In this format, FPBYTES will add up to TOTALBYTES minus HEADER size.

This field is a new addition since ISO 19794-2:2005.

Capture date and time (DATETIME)

This is the time when the fingerprint was captured by fingerprint reader, not the time when the fingerprint template was generated. If fingerprint capture takes some time, this is the time when capture began.

DATETIME is a 9-byte field holding date and time in UTC split into components (year, month, ...).

If date/time is not provided, all bytes should be set to 0xff. Individual date/time fields can be also set to 0xff or 0xffff, indicating that this component of the date/time is not provided. In that case, all fields describing higher resolution date/time components should be also set to 0xff/0xffff. This allows implementations to omit milliseconds or to provide only date without time.

This field group is identical to DATETIME in ISO 19794-1:2011. It is a new addition since ISO 19794-2:2005.

Year (YEAR)

Year encoded as a 16-bit big-endian unsigned number. Value 0 is invalid. Value 0xffff indicates that year is not provided.

This field is identical to YEAR in ISO 19794-1:2011.

Month (MONTH)

Month of year encoded as an 8-bit unsigned number in range 1 through 12. Value 0xff indicates that month is not provided.

This field is identical to MONTH in ISO 19794-1:2011.

Day of month (DAY)

Day of month encoded as an 8-bit unsigned number in range 1 through 31. Value 0xff indicates that day of month is not provided.

This field is identical to DAY in ISO 19794-1:2011.

Hour (HOUR)

Hour of day encoded as an 8-bit unsigned number in range 0 through 23. Value 0xff indicates that hour is not provided.

This field is identical to HOUR in ISO 19794-1:2011.

Minute (MINUTE)

Minute encoded as an 8-bit unsigned number in range 0 through 59. Value 0xff indicates that minute is not provided.

This field is identical to MINUTE in ISO 19794-1:2011.

Second (SECOND)

Second encoded as an 8-bit unsigned number in range 0 through 59. Value 0xff indicates that second is not provided.

This field is identical to SECOND in ISO 19794-1:2011.

Millisecond (MILLISECOND)

Millisecond encoded as a 16-bit big-endian unsigned number in range 0 through 999. Value 0xffff indicates that millisecond is not provided.

This field is identical to MILLISECOND in ISO 19794-1:2011.

Sensor technology (DEVTECH)

An unsigned 8-bit number encoding type of sensor technology. Contrary to the vendor-defined DEVID, this field is actually useful, because its values are specified. Matching algorithms can use this information to adjust error tolerances and other parameters. It must be in range 0-20 with meaning of allowed codes explained by the table below.

Code	Sensor technology
0	Unknown
1	Optical / white light / total internal reflectance (TIR)
2	Optical / white light / direct view on platen
3	Optical / white light / contactless
4	Optical / monochromatic / visible / total internal reflectance (TIR)
5	Optical / monochromatic / visible / direct view on platen
6	Optical / monochromatic / visible / contactless
7	Optical / monochromatic / infrared / total internal reflectance (TIR)
8	Optical / monochromatic / infrared / direct view on platen
9	Optical / monochromatic / infrared / contactless
10	Optical / multispectral / total internal reflectance (TIR)
11	Optical / multispectral / direct view on platen
12	Optical / multispectral / contactless
13	Electro luminescent
14	Semiconductor / capacitive
15	Semiconductor / radio frequency
16	Semiconductor / thermal
17	Pressure
18	Ultrasound
19	Mechanical
20	Glass fiber

Sensor technology types

This field corresponds to to DEVTECH in ISO 19794-1:2011 with modality-specific values enumerated above. This field is a new addition since ISO 19794-2:2005.

Sensor vendor ID (DEVVENDOR)

A big-endian unsigned 16-bit number identifying the vendor of the fingerprint sensor (which is itself identified in DEVID) or zero if sensor vendor is unknown. This field is optional, but interested sensor manufacturers can register with IBIA. See the list of currently registered organizations. If sensor vendor is unknown, this field should be zero rather than 0x103 "Vendor Unknown". Opensource implementations should leave this field zeroed.

This field is identical to DEVVENDOR in ISO 19794-1:2011. It is a new addition since ISO 19794-2:2005.

Sensor ID (DEVID)

A big-endian unsigned 16-bit number identifying the fingerprint sensor used to capture the fingerprint. Device IDs are assigned by sensor vendor identified in DEVVENDOR field. If device ID is unknown, this field should be zero. Opensource implementations should leave this field zeroed.

This field is identical to DEVID in ISO 19794-1:2011.

This field was under HEADER in ISO 19794-2:2005. Its size and value range has changed. Its interpretation is now defined by vendor identified in DEVVENDOR field.

Number of quality records (QCOUNT)

An unsigned 8-bit number indicating the number of QRECORD blocks that follow. Zero value means that quality measurement was not even attempted. Failed attempts are instead indicated by setting FPQUALITY to 255.

This field is identical to QCOUNT in ISO 19794-1:2011. It is always present in this format. It is a new addition since ISO 19794-2:2005.

Quality record (QRECORD)

This 5-byte field group describes one quality measure of the fingerprint. Several different quality measures can be included in separate QRECORD blocks. Number of QRECORD blocks is indicated in QCOUNT field.

This field group is identical to QRECORD in ISO 19794-1:2011. This field group is a generalization of FPQUALITY field from ISO 19794-2:2005.

Fingerprint quality (FPQUALITY)

An unsigned 8-bit number holding quality score of the fingerprint in range 0 (lowest) through 100 (highest) measured according to QALGO. Special value 255 indicates failure to measure fingerprint quality. If quality measurement was not even attempted, the whole QRECORD block should be omitted.

This field is identical to QUALITY in ISO 19794-1:2011. This field was also present in ISO 19794-2:2005, but it could repeat only once and there were no QVENDOR and QALGO fields to identify the algorithm.

Quality algorithm vendor (QVENDOR)

A big-endian 16-bit unsigned number identifying IBIA-registered organization that defined the quality algorithm referenced in QALGO used to interpret values of FPQUALITY.

If quality algorithm vendor is unknown or unreported, this field should be set to zero rather than 0x103 "Vendor Unknown".

When NIST's NFIQ 1.0 is used, this field should be set to NIST's vendor code 0x000f.

This field is identical to QVENDOR in ISO 19794-1:2011. It is a new addition since ISO 19794-2:2005.

Quality algorithm (QALGO)

This is a big-endian 16-bit unsigned number identifying the algorithm used to measure fingerprint quality recorded in FPQUALITY field. Quality algorithm ID is namespaced under QVENDOR, i.e. each vendor has its own algorithm IDs. Quality algorithm can be optionally registered in IBIA's quality algorithm directory.

Expensive QVENDOR registration is hostile to opensource, but opensource implementations can just reuse one of the already defined quality algorithms. It is also possible to set this field to zero to indicate that quality algorithm is not specified.

Unfortunately, nearly all registered quality algorithms require purchase of an additional specification from ISO. There is however one publicly documented quality algorithm: NIST's NFIQ 1.0, defined in NISTIR 7151, which can be used in this field. NFIQ 1.0 has

QVENDOR code 0x000f (NIST) and QALGO code 0x377d. It is however not clear how NFIQ's 5-level scale maps to values in FPQUALITY field that are in range 0-100.

This field is identical to QALGO in ISO 19794-1:2011. It is a new addition since ISO 19794-2:2005.

Number of certification records (CERTCOUNT)

An unsigned 8-bit number indicating the number of CERTIFICATE blocks that follow. This field is only present if HASCERTS is set. Even if this field is present, its value may be zero. If HASCERTS is zero, this field is not present and its value is assumed to be zero.

This field is identical to CERTCOUNT in ISO 19794-1:2011. Together with CERTIFICATE blocks, it replaces DEVSTAMP field from ISO 19794-2:2005.

Certification record (CERTIFICATE)

This 3-byte field group describes one certification of fingerprint sensor used to capture the fingerprint. Several different certifications can be included in separate CERTIFICATE blocks. Number of CERTIFICATE blocks is indicated in CERTCOUNT field. If HASCERTS is zero, this field group is not present.

This field group is identical to CERTIFICATE in ISO 19794-1:2011. Together with CERTCOUNT field, it replaces DEVSTAMP field from ISO 19794-2:2005.

Certification authority (CERTIFIEDBY)

A big-endian 16-bit unsigned number identifying IBIA-registered organization. This field cannot be zero. If there is no certification, the whole CERTIFICATE block should be omitted.

This field is identical to CERTIFIEDBY in ISO 19794-1:2011.

Certification scheme (CERTTYPE)

An unsigned 8-bit number in range 1-3 identifying specific certification scheme according to the table below.

Code	Specification	Description
1	Annex E.1 of ISO 19794-2:2011	Image quality for AFIS
2	Annex E.2 of ISO 19794-2:2011	Image quality for personal verification

Code	Specification	Description
3	Annex E.3 of ISO 19794-2:2011	Requirements for optical fingerprint readers

Certification schemes

This field is identical to CERTTYPE in ISO 19794-1:2011 with modality-specific values defined above.

Finger position on hands (POSITION)

An unsigned 8-bit number encoding hand position of the finger. It must be in range 0-10, 13-15, or 40-50 with meaning of allowed codes explained by the two tables below.

Code	Hand	Finger
0	Unknown	Unknown
1	Right	Thumb
2	Right	Index
3	Right	Middle
4	Right	Ring
5	Right	Little
6	Left	Thumb
7	Left	Index
8	Left	Middle
9	Left	Ring
10	Left	Little

Finger positions for single finger

Codes in the table below are an addition defined by this version of the format. They were not present in ISO 19794-2:2005.

Code	Count	Combination
13	4	Right index, middle, ring, and little
14	4	Left index, middle, ring, and little
15	2	Left and right thumbs
40	2	Right index and middle
41	2	Right middle and ring
42	2	Right ring and little
43	2	Left index and middle

Code	Count	Combination
44	2	Left middle and ring
45	2	Left ring and little
46	2	Left and right index
47	3	Right index, middle, and ring
48	3	Right middle, ring, and little
49	3	Left index, middle, and ring
50	3	Left middle, ring, and little

Finger positions for multiple fingers

Finger view number (VIEWOFFSET)

If multiple fingerprints with the same POSITION are present in the template, unique VIEWOFFSET must be assigned to each, starting with zero and incrementing with each view of the same finger. If there is only one fingerprint with given POSITION, its VIEWOFFSET is zero. VIEWOFFSET is an unsigned 8-bit number with allowed range of 0-15, which limits every finger position to 16 finger views. Finger views of one finger must be listed in the template in order of increasing VIEWOFFSET.

Size of this field changed since ISO 19794-2:2005.

Horizontal pixel density (RESOLUTIONX)

Image resolution (commonly called DPI), specifically its horizontal component, is stored in this field in units of pixels per centimeter. It is encoded as a big-endian unsigned 16-bit number. If sensor resolution is fractional, it is rounded to the nearest integer. Minimum resolution is 250dpi, which translates to minimum RESOLUTIONX value of 99. The common resolution of 500dpi is equal to 197px/cm after rounding.

This field was under HEADER in ISO 19794-2:2005.

Vertical pixel density (RESOLUTIONY)

Like RESOLUTIONX above but for vertical resolution. It's usually equal to RESOLUTIONX.

This field was under HEADER in ISO 19794-2:2005.

Impression type (SAMPLETYPE)

8-bit unsigned number indicating how the fingerprint was captured. Contrary to the vendor-defined DEVID, this field is actually useful, because its values are specified. Matching algorithms can use this information to adjust error tolerances and other parameters. Allowed values in range 0-9, 24, and 28-29 are described in the table below. Code 29 "Unknown" should be used as default.

Code	Impression type
0	Live / plain
1	Live / rolled
2	Non-live / plain
3	Non-live / rolled
4	Latent / impression
5	Latent / tracing
6	Latent / photo
7	Latent / lift
8	Live / swipe
9	Vertical roll
24	Live / contactless (plain)
28	Other
29	Unknown

Impression types

Size of this field changed since ISO 19794-2:2005. List of allowed values has been expanded.

Image width (WIDTH)

Image width in pixels encoded as a big-endian unsigned 16-bit number. Top two bits must be zero.

This field was under HEADER in ISO 19794-2:2005 and its range of values was narrower.

Image height (HEIGHT)

Image height in pixels encoded as a big-endian unsigned 16-bit number. Top two bits must be zero.

This field was under HEADER in ISO 19794-2:2005 and its range of values was narrower.

Length of minutia record (MINBYTES)

A 4-bit unsigned number occupying upper 4 bits of the byte it shares with ENDINGTYPE. It contains length of MINUTIA record in bytes. If MINBYTES is 6, all fields in MINUTIA record are present, including MINQUALITY. If MINBYTES is 5, MINQUALITY field is omitted from all minutiae.

This field is a new addition since ISO 19794-2:2005.

Ridge ending type (ENDINGTYPE)

A 4-bit unsigned number occupying lower 4 bits of the byte it shares with MINBYTES. If zero, ending minutiae (MINTYPE = 01) are located at valley skeleton bifurcations. If one, ending minutiae are located at ridge skeleton endpoints. Skeleton is a 1-pixel thinning of either ridges or valleys.

This field is a new addition since ISO 19794-2:2005.

Number of minutiae (MINCOUNT)

An unsigned 8-bit number that indicates the number of MINUTIA records that follow. MINCOUNT must be at least 1. Fingerprints without any minutiae are not allowed.

ISO 19794-2:2005 allowed zero minutia count, which is no longer allowed.

Minutia (MINUTIA)

Minutiae are interesting points on the fingerprint, usually ridge endings and bifurcations.

MINUTIA records span MINBYTES bytes (5 or 6) each and carry minutia position (MINX, MINY), angle (MINANGLE), type (MINTYPE), and optionally quality (MINQUALITY). MINQUALITY is only present if MINBYTES is 6.

Field MINCOUNT contains the number of MINUTIA records present. There should be at least one MINUTIA record in every fingerprint.

Minutia type (MINTYPE)

Minutia type is packed in the top two bits of the first byte occupied by the MINX field. Three minutia types are supported:

Bits	Type
01	Ridge ending / valley bifurcation
10	Ridge bifurcation
00	Other minutia type

Minutia types

Minutia type 01 represents either ridge skeleton ending or valley skeleton bifurcation depending on value of ENDINGTYPE.

Minutiae of type "other" can represent exotic minutiae that are neither ridge endings nor bifurcations. They can also represent ambiguous minutiae that cannot be reliably classified as either ridge endings or bifurcations. It follows that minutiae of type "other" should match all three minutia types. This is a clarification of wording from ISO 19794-2:2005.

Minutia X position (MINX)

Location of the minutia on the fingerprint in pixel units along X axis. Pixels are counted left-to-right, starting with zero. Minutia is in the center of the pixel specified by MINX and MINY.

MINX is a 14-bit unsigned number that is stored in the lower 14 bits of the big-endian 16-bit number that also contains MINTYPE.

In order to determine minutia position, feature extractor first constructs ridge *skeleton* by thinning ridges down to one pixel wide lines. Valley skeleton is constructed similarly by thinning valleys. Depending on the value of ENDINGTYPE, ridge endings lie either at forking points of valley skeleton or at endpoints of ridge skeleton. Ridge bifurcations lie at forking points of ridge skeleton. Location of minutiae other than ending and bifurcation is implementation-specific, but it should be always present.

Implementation may calculate position differently as long as it approximates the above skeleton-based method. When ridge skeleton endpoints are used (per ENDINGTYPE), the spec recommends (but does not require) placing ridge skeleton endpoint at the very end of the ridge in binarized image, i.e. on centermost border pixel of the ridge.

Placement of ridge endings on endpoints of ridge skeleton is new in this version of the spec. ISO 19794-2:2005 only allowed it in on-card format.

Minutia Y position (MINY)

Like MINX above but for Y axis. Pixels on Y axis are counted top-down, starting with zero. MINY is stored in a 16-bit unsigned big-endian field, but only the lower 14 bits of this field are used for MINY. Upper 2 bits are zero.

Minutia angle (MINANGLE)

An 8-bit unsigned number holding quantized minutia angle in range 0-255. Zero angle points to the right and angle increases counterclockwise. Dequantization to floating-point degrees is performed by multiplying MINANGLE by (360/256). Quantization is performed by dividing the floating-point angle by (360/256), rounding to nearest integer, and taking the lowest 8 bits.

If ridge ending is located at endpoint of ridge skeleton (see ENDINGTYPE), ridge ending angle is the tangent of the ridge skeleton leg that terminates in the minutia. If ridge ending is located at forking point of valley skeleton (per ENDINGTYPE), ridge ending angle is defined as mean angle of two of the three valley skeleton legs that run around the ridge terminated by the ending. Skeleton leg angle is defined as the angle of its tangent. Ridge bifurcation angle is similarly derived from legs of ridge skeleton. Minutiae of type "other" must have an implementation-defined angle.

Skeleton leg tangent is ambiguous, because skeleton legs curve a lot around skeleton forking point. The spec however defines recommended (but not required) method for calculating this tangent. Imagine a circle centered at the minutia with radius of 1.63mm (32px at 500dpi). This circle intersects with skeleton leg if the leg is long enough. Tangent of skeleton leg is then defined as the angle of a ray originating at minutia position and passing through this intersection. If the skeleton leg is not long enough, its end is taken instead of the intersection. If the endpoint is not at least 0.5mm away from minutia position, skeleton leg has no angle and the minutia must be discarded. This recommended procedure is a new addition since ISO 19794-2:2005.

Trifurcations, minutiae with four outgoing ridge skeleton legs, are represented by two minutiae that share the same position (MINX/MINY) and differ in MINANGLE. Handling of trifurcations is a new addition since ISO 19794-2:2005.

Minutia quality (MINQUALITY)

An 8-bit unsigned number encoding minutia quality from 0 (lowest quality) to 100 (highest quality). Special value 254 stands for unreported quality and 255 for failure to measure minutia quality. This field is only present if MINBYTES has value 6.

Meaning of minutia quality is implementation-defined. The spec recommends (but does not require) to set minutia quality to the probability (in percents) that the minutia exists.

Range of values has changed since ISO 19794-2:2005 and recommended meaning was added. It is now also possible to completely omit minutia quality fields (see MINBYTES).

Total length of extension data (EXTBYTES)

A big-endian 16-bit unsigned number holding the total size of all extensions in bytes. If EXTBYTES is zero, there are no extension blocks. If it is non-zero, one or more EXTENSION blocks follow. In that case, EXTBYTES is the sum of EXTLEN fields of all EXTENSION blocks.

Extension data block (EXTENSION)

If EXTBYTES is non-zero, one or more EXTENSION blocks are attached to the fingerprint. Every extension has two compulsory fields: EXTTYPE and EXTLEN. EXTLEN lets implementations skip over unrecognized extensions without having to parse them. EXTTYPE informs the implementation about internal structure of the extension, so that appropriate parser can be chosen. Actual extension data is stored in the EXTDATA field.

There are three predefined extensions: ridge count extension (RCOUNTEXT), core and delta extension (COREDELTA), and zonal quality extension (ZONALEXT). Extensions are intended to store data that cannot be expressed otherwise. Implementations must not abuse extensions to introduce alternative encoding for data that can be already encoded in documented fields of this format, including the predefined extensions.

Extension type (EXTTYPE)

Two bytes encode extension type according to the table below.

Byte 1	Byte 2	Description
0	0	Reserved
0	1	Ridge count extension (RCOUNTEXT)
0	2	Core and delta extension (COREDELTA)
0	3	Zonal quality extension (ZONALEXT)
0	4-255	Reserved
1-255	0	Reserved
1-255	1-255	Implementation-defined extension

Extension types

Interpretation of implementation-defined extension type codes requires knowledge of vendor ID, which can be only provided externally (perhaps via CBEFF header), because this template format has no field for vendor ID. Consequently, when this template format is used alone (without wrapping or context), implementation-defined extension types are meaningless and thus useless in matching. Since opensource implementations usually don't have assigned vendor ID, they should not include any custom extension blocks.

Length of extension data (EXTLEN)

Length of the whole EXTENSION block, including the EXTTYPE field and EXTLEN field itself, is stored in a big-endian 16-bit unsigned number. EXTLEN must be therefore at least 4. Implementations can use this field to skip over unrecognized extensions.

Definition of EXTLEN has changed since ISO 19794-2:2005, which excluded bytes consumed by EXTTYPE field and EXTLEN itself.

Extension data (EXTDATA)

Actual extension data. This field has length of EXTLEN minus the 4 bytes needed for EXTTYPE and EXTLEN. Internal structure of the EXTDATA field depends on the value of EXTTYPE field. This format specifies only structure of ridge count extension (RCOUNTEXT), core and delta extension (COREDELTA), and zonal quality extension (ZONALEXT).

In ISO 19794-2:2005, length of this field is stored in EXTLEN. In this version of the format, length of this field is 4 bytes less, because EXTLEN now includes length of EXTTYPE field and EXTLEN itself.

Ridge count extension (RCOUNTEXT)

Minutia set defined in MINUTIA records can be annotated with ridge counts. Ridges are counted on a line (or *edge*) connecting two minutiae. Edge's ridge count is the number of ridges crossed by the edge plus one as precisely defined in RIDGECOUNT field. RCOUNTEXT extension encodes this ridge count data. RCOUNTEXT has EXTTYPE of 0x0001.

Not all edges in the template need to have ridge count data. Edge picking method is specified in STARTYPE field. Information about every picked edge is then stored in EDGEDEF records that follow. The number of EDGEDEF records can be derived from the length of this extension block (EXTLEN) using expression `(EXTLEN - 5) / 3`. Ordering of EDGEDEF records depends on STARTYPE.

Edge picking method (STARTYPE)

This template format recognizes three edge picking methods: *quadrants*, *octants*, and *custom*. STARTYPE field contains an 8-bit unsigned code according to the table below.

Code	Method
0	Custom
1	Quadrants
2	Octants

Edge picking methods

Custom edge picking (code 0) simply means the implementation is free to choose edges however it wants. It can also choose how to order EDGEDEF records.

Quadrant (code 1) and octant (code 2) edge picking involves splitting the area around central minutia into four or eight sectors respectively. Quadrants and octants are aligned to minutia's direction (MINANGLE) differently. For quadrants, minutia's direction lies between two quadrants. For octants, minutia's direction bisects some octant. For every sector, ridge count is recorded for the edge to the nearest minutia in that sector. Quadrant and octant picking places strict requirements on presence, ordering, and contents of EDGEDEF records as explained below.

Quadrant and octant alignment to minutia direction was added in this version of the format. ISO 19794-2:2005 didn't specify it.

Edge between minutiae (EDGEDEF)

The 3-byte EDGEDEF record contains ridge count (stored in RIDGECOUNT field) for a single edge between minutiae identified by EDGEFROM and EDGETO fields. The number of EDGEDEF records is derived from EXTLEN as `(EXTLEN - 5) / 3`.

Field STARTYPE determines which edges are present. If STARTYPE is custom (code 0), implementation is free to choose which edges to include. If STARTYPE is quadrants (code 1) or octants (code 2), ridge counts must be recorded for every quadrant/octant, but implementation can choose to completely omit some central minutiae with all their quadrants/octants.

EDGEDEF record ordering depends on STARTYPE. If STARTYPE is custom (code 0), EDGEDEF record ordering is implementation-defined. If STARTYPE is set to quadrants (code 1) or octants (code 2), EDGEDEF records for single central minutia are listed together ordered by octant/quadrant number. Both octants and quadrants are numbered counterclockwise. First octant is the one where minutia points. First quadrant is the "north-east" one if minutia points "south". See STARTYPE for description of quadrant and octant alignment.

If STARTYPE is set to quadrants (code 1) or octants (code 2), the structure of EDGEDEF records is defined in more detail. EDGEFROM identifies the central minutia while EDGETO identifies the neighbor minutia. If some quadrant or octant does not have any minutiae, implementation must emit a placeholder record with EDGETO and RIDGECOUNT fields set to 255.

ISO 19794-2:2005 was ambiguous as to whether every minutia should be considered as central minutia when STARTYPE is set to quadrants (code 1) or octants (code 2). This version of the format clarifies that it is not necessary to provide ridge count data for all minutiae. Edge ordering rules are different and less ambiguous. Format of placeholder edge record has changed.

Starting minutia of the edge (EDGEFROM)

An 8-bit unsigned offset into the list of MINUTIA records. It identifies minutia that is on one end of the edge leading to EDGETO. If STARTYPE is quadrants (code 1) or octants (code 2), EDGEFROM is the central minutia.

Ending minutia of the edge (EDGETO)

An 8-bit unsigned offset into the list of MINUTIA records. It identifies minutia that is on the other end of the edge starting in EDGEFROM. If STARTYPE is quadrants (code 1) or octants (code 2), EDGETO is the neighbor minutia. If the current EDGEDEF record is a placeholder for quadrant or octant devoid of minutiae, EDGETO is 255.

In ISO 19794-2:2005, placeholder record had EDGETO set to zero while this format sets it to 255.

Number of ridges the edge crosses (RIDGECOUNT)

An 8-bit unsigned number containing the number of ridges crossed by the edge plus one. Ridges connected to either EDGEFROM or EDGETO minutiae are not considered crossed.

Ridge is crossed when it is crossed in ridge skeleton (single pixel wide representation of ridges). Ridge may be crossed more than once as long as valley skeleton is crossed at least once inbetween.

If the current EDGEDEF record is a placeholder for quadrant or octant devoid of minutiae, RIDGECOUNT is set to 255.

Value of RIDGECOUNT is higher by one compared to ISO 19794-2:2005. Placeholder record has RIDGECOUNT set to 255 instead of zero. Counting of ridge crossings is more precisely defined, including repeated crossings.

Core and delta extension (COREDELTA)

Besides normal minutiae defined in MINUTIA records, this format permits encoding of *cores* and *deltas*, which can be thought of as very special minutiae. COREDELTA extension encodes information about cores and deltas found on the fingerprint. COREDELTA extension has EXTTYPE of 0x0002. COREDELTA extension has two parts: COREDATA and DELTADATA.

Core data (COREDATA)

COREDATA, the first part of COREDELTA extension, contains a list of cores found on the fingerprint. Core is loosely defined as a point that is partially or completely encircled by ridges.

Every core is defined in a COREDEF record. Number of cores is specified by CORENUM field. Presence or absence of COREANGLE field is indicated by CHASANGLE flag.

Number of cores (CORENUM)

Number of COREDEF records in the COREDELTA extension encoded as an unsigned 8-bit number. Valid values are in range 0-15.

Core point (COREDEF)

COREDEF describes one core found on the fingerprint. COREDEF record repeats CORENUM times. Its structure is somewhat similar to MINUTIA record. Every core has position (COREX, COREY). If CHASANGLE is set, every COREDEF also contains COREANGLE.

Core angle presence flag (CHASANGLE)

This 1-bit field occupies bit 14 (counting from zero) of the 16-bit big-endian unsigned number that also contains COREX. If it is set, COREANGLE field is present. If it is zero, COREANGLE is not recorded.

Core X position (COREX)

Equivalent of MINX for cores.

This field occupies lower 14 bits of the 16-bit big-endian unsigned number that also contains CHASANGLE field.

If there are ridge endings enclosed in the core, position of the innermost ridge ending defines position of the core. Innermost ridge ending is defined only as "nearest to maximal curvature" of the enclosing ridge, which is not clear at all. It is probably safe to pick ending closest to extreme (farthest) point of the enclosing ridge.

If there are no enclosed ridge endings, then the end of the enclosed valley defines position of the core.

Core Y position (COREY)

Equivalent of MINY for cores. See notes under COREX.

Core angle (COREANGLE)

This field is present only if the CHASANGLE flag is set. It contains core angle encoded in the same way as MINANGLE.

ISO 19794-2:2005 defined how core angle should be determined. This version of the format leaves it to the implementation arguing that core angle cannot be reliably determined.

Delta data (DELTADATA)

DELTADATA, the second and last part of COREDELTA extension, contains a list of deltas found on the fingerprint. Delta is loosely defined as the point where three distinct ridge flows meet.

Every delta is defined in a DELTADEF record. Number of deltas is specified by DELTANUM field. Presence or absence of DELTAANGLE fields is indicated by DHASANGLE flag.

Number of deltas (DELTANUM)

Number of DELTADEF records in the COREDELTA extension encoded as an unsigned 8-bit number. Valid values are in range 0-15.

Delta point (DELTADEF)

DELTADEF describes one delta found on the fingerprint. DELTADEF record repeats DELTANUM times. Its structure is somewhat similar to MINUTIA record. Every delta has position (DELTAX, DELTAY). If DHASANGLE is set, every DELTADEF also contains three repeats of the DELTAANGLE field.

Delta angle presence flag (DHASANGLE)

This 1-bit field occupies bit 14 (counting from zero) of the 16-bit big-endian unsigned number that also contains DELTAX. If it is set, three repeats of the DELTAANGLE field are present. If it is zero, no DELTAANGLE is recorded.

Delta X position (DELTAX)

Equivalent of MINX for deltas.

This field occupies lower 14 bits of the 16-bit big-endian unsigned number that also contains DHASANGLE field.

Delta position is defined as the center of weight of three points, each of which lies where two ridges approaching the delta diverge. Such definition doesn't cover all types of deltas though.

Delta Y position (DELTAY)

Equivalent of MINY for deltas.

Delta angle (DELTAANGLE)

This field is present three times if DHASANGLE flag is set and zero times otherwise. Every repeat of this field contains one of the three delta angles encoded in the same way as MINANGLE.

If any of the three angles cannot be determined, the unused angle field should be filled with a copy of another angle. When the template is decoded, duplicate delta angles should be discarded.

ISO 19794-2:2005 defined how delta angles should be determined. This version of the format leaves it to the implementation arguing that delta angles cannot be reliably determined.

Zonal quality extension (ZONALEXT)

For the purposes of this extension, original fingerprint image is split into rectangular zones. Zone size is ZONEWIDTH times ZONEHEIGHT pixels. Rightmost and bottom zones may be smaller to fit image width and height exactly. Quality score is assigned to every zone in the image. A two-dimensional array of these scores is stored in this extension using ZONEBITS bits per zone.

Meaning of zone quality scores is defined by algorithm identified in ZONEVENDOR and ZONEALGO fields. Higher values mean higher zone quality. ISO 19794-2:2005 did not have these fields and instead left meaning of zonal quality to the implementation.

Since opensource implementations usually do not have the $500 IBIA registration, they cannot fill ZONEVENDOR with valid value and consequently cannot produce ZONALEXT extension.

Zonal quality vendor (ZONEVENDOR)

A big-endian 16-bit unsigned number identifying IBIA-registered organization that defined the zonal quality algorithm referenced in ZONEALGO used to interpret zone quality stored in ZONALQUALITY.

This field was not present in ISO 19794-2:2005.

Zonal quality algorithm (ZONEALGO)

This is a big-endian 16-bit unsigned number identifying the algorithm used to measure zone quality quality recorded in ZONALQUALITY field. Zonal quality algorithm ID is namespaced under ZONEVENDOR, i.e. each vendor has its own algorithm IDs.

This field was not present in ISO 19794-2:2005.

Zone width (ZONEWIDTH)

Horizontal size of one quality zone in pixels encoded as a non-zero unsigned 8-bit number. If ZONEWIDTH is not a factor of WIDTH, then the rightmost zones will be narrower.

Zone height (ZONEHEIGHT)

Vertical size of one quality zone in pixels encoded as a non-zero unsigned 8-bit number. If ZONEHEIGHT is not a factor of HEIGHT, then the bottom zones will be narrower.

Bits per zone (ZONEBITS)

Number of bits allocated for each quality zone in ZONALQUALITY data. This field is encoded as an unsigned 8-bit number. Allowed range of values is 1-8.

Allowed range of values has been clarified since ISO 19794-2:2005.

Zonal quality data (ZONALQUALITY)

This field encodes a two-dimensional array of unsigned zonal quality values. Meaning of zone quality scores is defined by algorithm identified in ZONEVENDOR and ZONEALGO fields. Higher values mean higher image quality in the zone.

Number of zones can be calculated by dividing WIDTH and HEIGHT by ZONEWIDTH and ZONEHEIGHT respectively and rounding up to get array width and height.

Zones are encoded row by row top-down and left-to-right. Each zone takes up ZONEBITS bits. Zones are aligned to bits rather than bytes, i.e. they are packed without any padding bits, spanning bytes if necessary. Zones in one byte are laid out starting from the most significant bits and ending with the least significant bits. Only the last byte of ZONALQUALITY has unused least significant bits filled with zeroes.

Robert Važan robert.vazan@tutanota.com