MIRAMON RASTER FILE FORMAT

Authors of the document: Xavier Pons and Abel Pau
Initial proposal: 17-01-2009
Last modification and version: 06-05-2025. **1.2**

1. Background and motivation

At the beginning of MiraMon, Xavier Pons adopted, for the software's raster data model, a data structure based on a binary file, with the extension .img, and a small metadata text file, with the extension .doc. This .doc format had nothing to do with the format of the old versions of Microsoft's Word program, but was mirrored in Ronald Eastman's proposal for rasters managed by the Idrisi software. Multiband support was provided through as many img-doc pairs as needed, and compression (lossless) was incorporated through RLE encoding. Soon, however, in the Grumets research group (which contains the people behind MiraMon) it was necessary to sophisticate the model and diverge from the original inspiration. In fact, it was necessary to include much more metadata, and to be able to indicate both the default display (representation scales, color symbolization, etc) and the relationships of the raster cell values with alphanumeric tables. This was achieved by abandoning the old .doc file and adopting a new accompanying text file. This file, with a .rel extension and Windows INI structure, included the necessary properties: metadata, default symbolization and relationships with alphanumeric tables. The new .rel file also allowed multiband processing, in which the bands continued in separate .img files, but the common characteristics (such as the reference system [cartographic projection, etc]) were compiled in this same .rel file. The flexibility of the .rel format allowed specific characteristics to be recorded for the bands where necessary (such as the different number of columns and rows in the panchromatic band of a Landsat ETM+ image, the different symbolization in quality bands of remote sensing images, etc). Joan Masó made important contributions to this design. It was also convenient that the RLE compression did not have the counter of the same numerical type as the data, providing it with an extra-compressed variant (which allows to reduce storage needs in the areas of the raster with values without repetitions) or including an indexing for faster access to the different areas of the file.

The MiraMon .img format **allows the management of the vast majority of raster processing needs within the framework of Geographic Information Science and Technology.** Among the properties that allow this statement to be made, we can highlight:

- **Support for data types, integer or real, typically required.**
- **A format for both .img and .rel files that is relatively simple.** This allows for an **appropriate balance between simple and efficient programming** and the **possibility of addressing any raster geographic analysis.**
- **Virtually infinite possible file size**
- **Complete metadata documentation that:**
    - Provides a **rich and flexible data model** (multiband rasters of different dimensions, which can contain different magnitudes, etc).
    - Allows **rigorous and consistent data processing** by applications that work with the MiraMon raster format. In fact, **each application can obtain the necessary parameters** (radiance converters or precise date and time for calculating the solar position in a radiometric correction of a remote sensing image,

well-defined altimetric and planimetric units for solar radiation calculations from a Digital Elevation Model, etc). This availability **saves space and time for parameter entry**, allows **robust automation**, and **avoids errors.**

The **motivation** of this document is to provide the **technical information** necessary for **reading and writing rasters in MiraMon format**, covering both the .img format and the data contained in the .rel file. The focus of the document is essentially computer programming or be a complement for advanced users.

This document is a companion of the information in the section *"Description of MiraMon raster formats*" < https://www.miramon.cat/help/eng/mm32/ap1.htm> of the software help, where the description of the .img format (and others that the program can access directly) is given from the user's point of view. It may be interesting to read this part of the MiraMon help beforehand if you are not familiar with its data files.

2. Description of the MiraMon raster file format

2.1. Overview

A MiraMon raster layer consists of several files. The main files are always at least 2:

- **A file that contains the graphic part** (the body of the raster) and has the extension **.img**. In the case of multiband rasters there will be as many .img files as necessary. The following section explains the format of these files.

- **A text file, in Windows INI format** , described in the MiraMon help, which **contains** the **layer metadata** (see the "**documentation file**" section on this website or, for more extensive documentation, see https://www.miramon.cat/help/eng/GeMPlus/ClausREL.htm) and also describes both **the default symbolization** (optional) and the possible **relationships of the raster values with alphanumeric tables** (typically DBF which may, in turn, be linked to other tables). **The extension for this file is .rel, and is preceded by the letter 'I', which indicates that it is a REL relative to a file suitable for an image or raster, .img**. The rest of the name is usually the same as that of the raster file and is stored in the same directory. For example, a raster relief**.img** will be accompanied by a relief**I.rel** .

  The most typical case of needing to relate raster values to an alphanumeric table is when it is a categorical raster, such as a land use and land cover layer. In this case the **table acts as a thesaurus of categories**, and is in DBF format, or in extended DBF format if more than 254 fields are required, fields of more than 254 characters, field names of up to 128 characters, a number of records greater than 4.2 billion, etc. The DBF format is a well-known and documented format, and the extended DBF format can also be found documented in [Specification of the "Extended DBF" format](), written by Xavier Pons and Abel Pau . **The extension of these tables is .dbf and their name and location are free (they can be in other directories, drives, etc.)**. The link with these tables is made in the I.rel file and is explained in "**Linking with alphanumeric data tables**".

It is also possible to link the raster to color symbolization tables. See "**Raster visualization**".

Note 1: In this document the noun "double" refers to a data type that allows real numbers to be stored using 64 bits for each number; this type is called *double* in the C programming language and has sufficient precision and numerical range for the storage of, for example, coordinates used in geographic information. On the other hand, the noun "real" refers to a data type that allows real numbers to be stored using 32 bits for each number; this type is called *float* in the C programming language and has sufficient precision and numerical range for the storage of most variables used in geographic information, such as temperature or per capita income.

Note 2: For performance reasons, rasters of real data types should not contain bits combined as Not-a-Number (NAN), infinite, denormal, etc. As explained in this document, the NoData values are correctly described in the raster metadata file (I.rel) and no "strange" bit combinations are needed.

2.2. . img file (raster body)

The **MiraMon raster file format** is based on **binary files**, with **the extension .img**, which represent a two-dimensional surface of data regularly distributed in geographic space. This data (the cells of a raster, the pixels of an image) can refer to continuous quantitative data (orthometric heights, spectral radiances, etc), to categorical data (classes of a layer of land use and land cover, etc), to Boolean masks (which indicate the presence or absence of a phenomenon), etc. Rasters can be **single-band** (for example topographic slopes in degrees) or **multiband** (like a typical Landsat image with data collected, at each pixel, in different regions of the electromagnetic spectrum). **Multiband rasters are made up of several .img files**, as many as bands are necessary.

In a multiband raster we can find .img files with **different numerical types** (byte, single precision real, etc) and also with **different geographic bounding boxes** (necessary, for example, in cases of some Landsat-7 ETM+ images processed according to the Landsat Product Generation System, LPGS).

Data order. An **uncompressed .img** raster contains exclusively the values of the image body, written from left to right and top to bottom, always in **little-endian byte order** (endianness). When the raster contains a bit data type, and the number of columns is a multiple of 8, all bits represent the value of a raster cell; in case the number of columns is not a multiple of 8, the bits corresponding to the raster cells at the beginning of a row always start in a "new byte" (that is, a byte cannot contain bits corresponding to the end of a row and bits corresponding to the beginning of the next row); for neatness, in rasters with bit data types it is preferable to leave any unused bits in the last byte of a row to zero. Uncompressed .img files do not contain additional information.

Compression. All .img raster types, except bit, support compression. Compression is carried out by going through the file in the same order explained in the previous paragraph. In the event that an .img file is compressed, it will contain, in addition to the data to express the values of the cells, information specific to the compression algorithm, as well as, optionally, information to access more quickly the start offset of each row of the compressed raster. As compression strategy, a simple or extra-compressed **RLE** (*run-length encoding*) is used. In both cases, **compression is applied to the values of the cells of the same row** (compression is never shared between rows) and the repeated value counter is always of type byte, and is followed by the value of the raster type (byte, 16-bit integer with or without sign, etc) that presents repetitions in successive columns. In the **extra-compressed variant**, when the repeated value counter is 0, the next value is a second counter, also byte, indicating how many successive values (of the raster type), uncompressed, come next. This is useful for not trying to compress sequences of different values with the RLE criterion, which would result in a file larger than the original; compression starts to be effective for repetitions of 3 or more values.

Indexing. In compressed files it is possible and highly recommended to add an indexing section to the file. The technical information on indexing is explained in section 3 of this document. An indexed file is recognized because it contains special headers at the end. When reading a compressed file consecutively it is not necessary to load the indexing information. Reading this information is only useful if you plan to access directly in areas of the file in a “random” manner.

Where is the information that in other formats is in a header? In other raster formats it is common to have a header containing essential information for correct reading, such as the number of columns and rows, the data type in the cells, etc. As already mentioned, in the MiraMon raster format the .img files only contain the data itself which, in the compressed ones, is complemented with compression information and indexing to speed up reading. The rest of the information is found in the I.rel file (see the sections corresponding to the I.rel file).

2.3. I.rel file: List of files of a multiband raster and “xxxx” designation of the bands inside the I.rel file.

The metadata of an .img file is always in a single I.rel file, whether it is a single-band or multiband dataset. This I.rel file contains basic metadata, such as the columns and rows of each .img, key metadata such as the horizontal reference system or the map coordinates of the bounding box, and other advanced metadata (units of the raster values, radiometry in images of remote sensing sensors, lineage, etc). As indicated, the I.rel file can also contain optional indications of **default visualization** and symbolization (representation scales, color palette) in the [COLOR_TEXT*] and [VISU_LLEGENDA*] sections, and of relationships of the raster values with alphanumeric databases (present in categorical rasters such as land use and land cover maps) in the [ATTRIBUTE_DATA*] and derived sections. The details of this complementary information will be given later.

To be able to read a .img raster you need to obtain the **basic metadata of its companion I.rel file**. I.rel files are files in the popular Windows INI format. **In multiband rasters there is a single I.rel file that contains information both about the multiband dataset** (for example, a general title) and **about each of the bands that make it up** (such as the data type of each band if a particular type needs to be specified for each band). Before explaining how to access any of the information needed to read an .img file, we must explain how to obtain the **list of files in a multiband raster and, especially, the name of the bands inside the I.rel file.** In fact, in the case of an I.rel file that documents a multiband raster, it is necessary to know the list of the various .img files that compose it, such as the name or alias "**xxxx**" that, in the I.rel file, is used to refer to each of the bands. This alias is concatenated to section names when it is necessary to access a “child section” where specific, particular information about the “xxxx” band is stored. For example, an I.rel file typically contains a **[ATTRIBUTE_DATA]** section, but it can also contain **[ATTRIBUTE_DATA:xxxx]** sections, where properties of the “xxxx” band that are specific to it will be documented. In other words, knowing this internal name is important because it allows reading data from each band (such as columns=, or TipusCompressio=) when they are not the same as the modal data (found in sections such as [OVERVIEW:ASPECTES_TECNICS] and [ATTRIBUTE_DATA], respectively). Knowing the internal name “xxxx” is also important because **it allows knowing the names of the .img files that make up the multiband set**.

Determining the number of bands in a raster and the “xxxx” aliases of each band: In the [ATTRIBUTE_DATA] section of every I.rel file we will find a key **IndexsNomsCamps=** in which the internal codes of the different bands are indicated, separated by commas. For example, **IndexsNomsCamps=1,2,3** (alphanumeric ASCII codes can also be used; a second example would be **IndexsNomsCamps=B,G,R,NIR**). **The order of these codes determines the order in which the data will be presented** in, for example, queries by location, or in the "Thematic information" tab of the GeM+ Metadata Manager. With this information, in the same [ATTRIBUTE_DATA] section we will find **keys formed by the concatenation of the string "NomCamp_"** *(field name)* **with the codes obtained**. In the previous example they would be: NomCamp_1= NomCamp_2= NomCamp_3= (or NomCamp_B= NomCamp_G= NomCamp_R= NomCamp_NIR= in the second example). The contents of these concatenated keys will provide **the internal name, "xxxx", of each band**.

Determining the names and locations of the .img files that make up the bands of a raster: In the sections [ATTRIBUTE_DATA:xxxx] we will find the key **NomFitxer=** (*file name*) that allows us to know the name of the .img file. The value of the NomFitxer= key is provided without path because **the datasets of related .img and I.rel files must be in the same directory**. In single-band files with an I.rel that have a name with the simple criterion of concatenating "I.rel" to the name of the .img file without extension (*e. g.*, Veneto.img and VenetoI.rel) the NomFitxer= key is not necessary. Therefore, the absence of this key indicates that the raster is single-band and that it has a “symmetrical” name with the name of the I.rel file. Based on the explanation provided, it is now possible to identify all the .img files associated with an I.rel file. For the opposite operation, see the following point, “Identifying the I.rel file corresponding to an .img file”.

Identifying the I.rel file corresponding to an .img file. The correct reading of an .img file involves **identifying the name of the I.rel file that contains its metadata**. It is worth remembering again that the various .img files that can form a raster dataset (a single .img file in a single-band raster) and the single I.rel file **are always located in the same directory**.

- In the simplest case, of a single-band raster, let's assume that the name is Mallorca**.img**, its metadata file will typically be Mallorca**I.rel**. Therefore, to obtain the I.rel metadata file of the Mallorca.img file, in the first instance we will open, if it exists, the MallorcaI.rel file, we will determine the internal name "xxxx" of the band and we will look, in the [ATTRIBUTE_DATA:xxxx] section, for the key NomFitxer=. If the key does not exist (default criterion) or its content is exactly Mallorca.img we already have the metadata file.

    If the previous procedure does not work we will have to open other I.rel files in the same directory until we find one in which in some section [ATTRIBUTE_DATA:xxxx] the key FileName= contains Mallorca.img.

- In the case of a multiband raster, such as a remote sensing image with 3 bands **ParisR.img, ParisG.img** and **ParisB.img**, the raster may have the metadata of the entire multiband set in a **ParisRI.rel file**, or in a **ParisI.rel file** (recommended criterion to avoid that the REL seems to correspond only to one band, as happens if the metadata is in a Paris**RI.rel file**), or even in a **BadIdea_I.rel file** (not recommended option, but possible). By exploring the I.rel files in the same directory as the file we want to read, preferably those with a similar first part of the name, we can find the I.rel file that contains the metadata for the .img file of interest. We will proceed as in the single-band case, until we find some section [ATTRIBUTE_DATA:xxxx] with a NomFitxer= key that contains the name of the .img file from which we want to obtain the metadata.

2.4. I.rel file: Essential metadata

Once we know whether the file is single-band or multiband, and the "**xxxx**" aliases of each band, we can proceed to read the essential information to process each band:

- **Number of columns:** In single-band rasters it is found in the **columns= key of the [OVERVIEW:ASPECTES_TECNICS]** section. **In multiband rasters this key of this section contains the modal value of the multiband set** . When a band has a number of columns different from this modal value, its number of columns is found in the **columns=** key of the **[ATTRIBUTE_DATA:xxxx] section** .

- **Number of rows:** It is found in the **rows=** key of the same sections explained for the case of the number of columns.

- **Data and compression types:** In single-band rasters, it is found in the key **TipusCompressio=** of the **[ATTRIBUTE_DATA]** section. **In multiband rasters, this key contains the modal value of the multiband dataset**. When a band has a data and/or compression type different from this modal value, the information is found in the key **TipusCompressio=** of the **[ATTRIBUTE_DATA:xxxx]** section. The **supported data and compression types** are bit, byte, byte-RLE, integer, integer-RLE, uinteger, uinteger-RLE, long, long-RLE, real, real-RLE, double and double-RLE, and are expressed in the key TipusCompressio= with the text strings just indicated. The byte data type is always unsigned, "integer" refers to 16-bit integers, "long" to 32-bit integers, the "u" in "uinteger" indicates "unsigned". The extra-compressed RLE variant is not indicated in the text string because it is dynamically deduced from the reading of the data (reading the file following the extra-compressed encoding also allows reading the simple RLE encoding, where a zero-value counter is meaningless).

- **Minimum and maximum values.** The I.rel file indicates the minimum and maximum value of the raster. This value should correspond to the actual minimum and maximum, not to an approximation as is done in some environments. All MiraMon functionalities that write a raster correctly document these values. However, MiraMon is proof to approximate values. The minimum and maximum values are useful, for example, to properly scale colors, etc. These values are indicated through the min= and max= keys in the [ATTRIBUTE_DATA] section. In multiband rasters these keys contain the modal value of the multiband set. When a band has a minimum and/or a maximum different from this modal value, the information is found in the homonymous keys in the [ATTRIBUTE_DATA:xxxx] section.

- **Indication of the value corresponding to the NoData.** Rasters may contain cells in which the value corresponding to the category or magnitude represented by the raster in that cell is not known, or is not intended to be indicated. The value of these cells, which must be outside the true [minimum, maximum] range of the raster data, is indicated with the NODATA= key. Its description is indicated with the NODATADef= key. The two keys are normally in the [ATTRIBUTE_DATA] section or, if necessary, in the [ATTRIBUTE_DATA:xxxx] section. Note, therefore, that in a multiband set each layer can have its NoData values, as appropriate for the data type in the band, etc. See also Note 2 .

- **Reference system and coordinates of the geographic bounding box.** The horizontal reference system is found in the HorizontalSystemIdentifier= key of the [SPATIAL_REFERENCE_SYSTEM:HORIZONTAL] section. See https://www.miramon.cat/help/eng/mm32/AP6.htm for more details on the relational system that allows, from this identifier, to know its properties such as the cartographic projection and its parameters, the datum, the EPSG synonym if it exists, etc. The map coordinates of the geographic bounding box are found in the MinX= MinY= MaxX= MaxY= keys of the [EXTENT] section (or [EXTENT:xxxx] if necessary for some band of a multiband raster). **They always refer to extreme outer coordinates, not to cell centers**.

- **Raster data treatment.** Although it is not essential metadata, it is convenient to read what the default treatment (categorical, quantitative or ordinal) is desired for the raster data. This allows us to know if it makes sense to wait for the link to a thesaurus of categories, which treatment of the color palette in automatic mode is more appropriate, etc. In the case of a single-band raster we will find this indication in the key TractamentVariable= of the [ATTRIBUTE_DATA] section. The key supports the values Categoric, QuantitatiuContinuu (which is the default value) or Ordinal. In the case of a multiband raster it is assumed that all the bands have the treatment assigned in the [ATTRIBUTE_DATA] section, which includes the modal value, but if any band or bands require a different treatment, it can be indicated in the corresponding [ATTRIBUTE_DATA:xxxx] section.

2.5. I.rel file: Linking with alphanumeric data tables

Integer type rasters (typically byte or byte-RLE, but also in other integer types) can be, if they have a categorical treatment (see “Raster data treatment”, in the previous point) linked to **attribute tables**, usually in DBF format. As mentioned, the most typical case of needing to relate raster values to an alphanumeric table is when it is a categorical raster, such as in a land use and land cover layer, in which the raster values correspond to categories such as “Forest” or “Urban”. In this case the **table acts as a thesaurus of categories**. In a multiband raster there can be one or more bands linked to alphanumeric tables. Each section **[ATTRIBUTE_DATA:xxxx]** can contain a key IndexsJoinTaula= that provides a code of the attribute table. This code must be concatenated to “JoinTaula_” in order to obtain the name of the key that will provide, as a value, the name of the section where the details of the linked table are; the section name is preceded by the string “TAULA_”. For example, IndexsJoinTaula=1 --> JoinTaula_1=CATEG --> [TAULA_CATEG]. In the section found ([TAULA_CATEG] in the example) there will be a key NomFitxer= that will allow us to access, through an absolute path, or relative to the .rel file, the name of the table. **If the file is a .dbf table**, in the same section we will find the keys TipusRelacio= and AssociatRel=. The TipusRelacio= key can have the values

RELACIO_1_1, RELACIO_1_N, RELACIO_N_1 and RELACIO_N_N, depending on whether the cardinality between the raster values and the table records is from 1 to 1, from 1 to many (N), etc. If you want to indicate that the relationship is required (the table must contain some record with the raster value, like a complete dictionary), the previous possible values become RELACIO_1_1 _DICC, RELACIO_1_N _DICC, RELACIO_N_1 _DICC and RELACIO_N_N _DICC. In categorical rasters the typical value is RELACIO_N_1 _DICC. The AssociatRel= key provides the name of the table field (indicated by NomFitxer=) that allows the *join to be made* from the raster values. Optionally, there may be sections derived from the current one that provide additional properties for the table fields, such as the field descriptor (descriptor=), the field treatment (TractamentVariable=), etc. The name of these sections is built by concatenating the section name with ':' and with the field name (for example [TAULA_CATEG:COD] in the previous example if the table contains a field named COD with the codes of the categories of the categorical raster, or [TAULA_CATEG:DESC] if the table also contains a field named DESC with the descriptive text of the category).

If the file indicated by FileName= is a file with the extension .rel (with an absolute path, or a path relative to the I.rel file), this file will be a table REL. A table REL is a file in INI format that documents a table using essentially the same resources of section names and keys explained in the previous paragraph. Its usefulness is that, since it usually documents a thesaurus, it allows other categorical .img files to be linked to the same thesaurus without the need to indicate the field descriptors again, etc: The table REL acts as a documented gateway to an alphanumeric table that can be linked from different rasters or even from vector layer tables.

2.6. I.rel file: Raster visualization

The raster display is specified in the [COLOR_TEXT] and [VISU_LLEGENDA] sections. If the raster is multiband, particular specifications for bands may exist in sections [COLOR_TEXT:xxxx] and [VISU_LLEGENDA:xxxx]. If these sections do not exist, a palette of 256 grays (0 black, 255 white) is applied to rasters and scaled linearly if the treatment is quantitative or ordinal, and a palette of different, non-graduated colors, if the treatment is categorical (see “Raster data treatment”); if the raster has NoData, a white or transparent value can be applied to these cells. The [COLOR_TEXT] sections indicate the display characteristics of the raster, for example the definition of the palette to open by default (Color_Paleta=), the treatment of the variable (Color_TractamentVariable=), etc. The [VISU_LLEGENDA] sections indicate the display characteristics of the legend in terms of the number and description of the categories, etc. It is possible that the raster contains advanced display indications, such as the denominators of the representation scales (EscalaMaxima= and EscalaMinima=, assumed 0 and 900000000 by default), or of symbolization (Color_ValorColor_0= and Color_ValorColor_n_1= for the minimum and maximum values to be considered instead of those of the raster itself, Color_EscalatColor= for the type of color scaling, etc). See https://www.miramon.cat/help/eng/mm32/AP4.htm#simb_col for more details (structure of the table that defines the color palette, etc).

RGB compositions from 3 .img files are not indicated in the I.rel files, but rather in MiraMon Map (.mmm) files are used to define them. It is possible to combine .img files from different rasters (single-band or multiband) to generate RGB composites. More details at https://www.miramon.cat/help/cat/mm32/AP3.htm#RASTER_RGB.

3. Technical description of the indexing section of compressed files

The file starts with compressed raster data, using RLE extra-compression (i.e., RLE except when a run of unrepeated values occurs, which are coded with two byte values: a 0 followed by the number of unrepeated values). At the end of the compressed raster data, a structured data block is appended, consisting of three main parts:

1. **Initial header (32 bytes)**
2. **Offset section (variable size, depending on the number of rows in the image and on the size of each offset)**
3. **Final section (32 bytes)**

Suggested reading in v. 1.0: Go to the end of the file and read the last 32 bytes (Final section). Check the type and version. Use the last 64 bytes to know the offset to the Initial Header. Go to the Initial Header and read its 32 bytes. Check the type of file, version and section type. Read the offset size of each offset (offset_size). Continue reading the file in chunks of offset_size, one per each row, to determine the offset of every row from the beginning of the file.

1. Initial header (32 bytes)

This 32-byte header is written at the end of the compressed raster data.

- **Bytes [0,7] (8 bytes) - Type and version**
  - "IMG" followed by a string indicating the version. The string is written as a 4-character sequence terminated with a null character and formatted as ##.# (a leading space is used for versions less than 10).
  - Example: "IMG 1.0\0"
- **Bytes [8,11] (4 bytes) - Section type**
  - Indicates the section type. For offset sections (the only possible type in version 1.0), this value is "2" written as an unsigned __int32.
- **Bytes [12,15] (4 bytes) - Offset size (if section type is 2)**
  - Indicates the size (in bytes) of each offset in the Offsets section, written as a 32-bit unsigned integer.
  - Possible values: 1, 2, 4 or 8.

- **Bytes [16,19] (4 bytes) - Reserved and zero-padded**
- **Bytes [20,23] (4 bytes) - Reserved and zero-padded**
- **Bytes [24,31] (8 bytes) - Offset to another section, written as an unsigned 64-bit integer.**
    - 0 if there are no additional sections.

2. Offsets section (variable size)

This section contains one offset per row in the image. The first offset is written first in the file, and consecutively by the remaining offsets, with the last offset being closest to the end of the file. The number of offsets is equal to nrows.

- **Bytes [0, (offset_size*nrows) - 1] - Offset for row 'n'**
    - Each offset corresponds to the starting position (from the beginning of the file, which has an offset=0) of a row within the body of the raster file.

The size, in bytes, of each offset (**offset_size**) is determined by the value stored in bytes [12,15] of the Initial header.

3. Final section (32 bytes)

A final 32-byte section is added after the last offset.

- **Bytes [0, 15] (16 bytes) – Filled with Zeros**
    - This area is filled with 0. Reserved for future use.
- **Bytes [16, 23] (8 bytes) – Type and version**
    - Same format as bytes [0,7] of the Initial Header.
    - Example: "IMG 1.0\0"
- **Bytes [24, 31] (8 bytes) – Offset to initial header**
    - Written as an unsigned 64-bit integer.