



# OPEN MATRIX (OMX) FILE SPECIFICATION

V 0.3

## Table of Content

1. [Standard File Extension.](#)
2. [File Structure.](#)
  - [2.1. Root Node.](#)
  - [2.2. Matrices.](#)
  - [2.3. ZonalReferences.](#)
3. [OMX Compression](#)

## **1. Standard file extension**

An OMX file is referred to as an "open modeling matrix" or simply "open matrix" and it is recommended that it uses the file extension ".omx"

## **2. File Structure**

An OMX file must be a valid HDF5 format file, with these properties:

1. An HDF5 Group node with the path "/matrices" containing all the matrices in the OMX file. Each individual matrix (or "table") will be stored as an HDF5 Dataset under this node.
2. An HDF5 Group node with the path "/zonalReferences" containing indexing information about each zone in the OMX file. Each "index" under this group node must be an HDF5 Dataset. Each of these datasets are 1 dimensional arrays of length equal to the number of zones as defined by the OMX\_ZONES attribute.

All official OMX attributes that will be listed below must be of the following HDF5 data types<sup>1</sup>:

- H5T\_STD\_I8LE
- H5T\_STD\_U8LE
- H5T\_STD\_I16LE
- H5T\_STD\_U16LE
- H5T\_STD\_I32LE
- H5T\_STD\_U32LE
- H5T\_STD\_I64LE
- H5T\_STD\_U64LE
- H5T\_IEEE\_F32LE
- H5T\_IEEE\_F64LE
- H5T\_STD\_U64LE
- HDF5 H5T\_C\_S1 (All strings must be stored as this HDF5 type)

An attribute called "OMX\_TITLE", may be added to any [matrix](#) or [zonal reference](#). When present it must be of type H5T\_C\_S1.

### **2.1 Root node**

The root node of an OMX matrix must include the following standard attributes:

1. OMX\_VERSION: A string of type H5T\_C\_S1 representing the version number of the OMX matrix file standard that has been implemented. This document specifies OMX\_VERSION "0.3".
2. OMX\_ZONES: scalar value of type H5T\_STD\_U64LE denoting the number of zones supported in the current OMX file. Each matrix (table) under the /matrices node will have this structure. Each matrix is a square matrix.
3. OMX\_CAPABILITIES: An optional attribute that if present must be a scalar value of type H5T\_C\_S1. This attribute is used to provide for future enhancements through

---

<sup>1</sup> Only the listed data types are supported to avoid having to account for issues related to endian-ness and to ensure that all API implementations will be compatible.

feature detection. It is a semicolon delimited list of features that are support or implemented in an OMX file. This allows for new features to be added to OMX files without requiring new official OMX versions if they features are simply additional capabilities.

## 2.2 Matrices

- This node contains the datasets associated with each matrix.
- Every matrix must be child of this node (“/matrices” node).
- No leaf nodes under this node are allowed.
- The matrices are stored as two-dimensional HDF5 datasets in row-major order.
- All matrices within an OMX file must have the same number of zones.
- All zonal numbers used in matrices row/column or zonalReference must be 0-based.
- Only the following data types will be supported for matrices:
  - H5T\_STD\_I8LE
  - H5T\_STD\_U8LE
  - H5T\_STD\_I16LE
  - H5T\_STD\_U16LE
  - H5T\_STD\_I32LE
  - H5T\_STD\_U32LE
  - H5T\_STD\_I64LE
  - H5T\_STD\_U64LE
  - H5T\_IEEE\_F32LE
  - H5T\_IEEE\_F64LE
  - H5T\_STD\_U64LE
- One OMX file may have several matrices, each using a different supported data type.
- The only HDF5 data type allowed for strings is HDF5 H5T\_C\_S1.
- The only HDF5 data type allowed for “ZONES” attribute is H5T\_STD\_U64LE.
- Each OMX matrix:
  1. Is stored under a unique Name. The Name is required. The name is used to retrieve a specific matrix. With the root node and the name established, the matrix can be identified by its full path, e.g. “/matrices/m1”.
  2. May have the optional attribute “OMX\_TITLE” of type H5T\_C\_S1.
- Matrices can have additional attributes as scalar key-value pairs, such that each value is one of the support OMX data types. An OMX API is only required to read and write scalar key-value pairs of the supported OMX data types.

## 2.3 zonalReferences

- This node contains 1-dimensional datasets of length equal to OMX\_ZONES and may only be any of the supported OMX HDF5 data types, including H5T\_C\_S1. These zonalReference datasets allow the mapping of metadata, associating a scalar value with each zone. For instance, one may associate a county identifier with each zone using this facility or simply plain language name for each zone<sup>2</sup>.

---

<sup>2</sup> A zonal reference is a one-dimensional array (vector) with the same number of simple data types such as integer or string elements as the corresponding matrix dimension (one of the numbers in the omx zones attribute). The elements of the vector are populated with the corresponding index value. For example, for a matrix of zones (1500, 1500) values of up to 1500 can be used as index for both dimensions. As you have

- This node must use a 0-based indexing<sup>3</sup> <sup>4</sup>.
- Datasets under this node may have attributes just as with the OMX file and individual matrices.
- This node use OMX lookup index that must be stored under a unique name that is used to identify the zonal reference.
- Each zonal reference may have the optional attribute “OMX\_TITLE” of type H5T\_C\_S1.

### **3.OMX Compression**

When compressing OMX, only use the "zlib" compression filter<sup>5</sup> (5).

Only HDF5 zlib compression is supported for matrices and zonalReferences. OMX APIs must support writing levels 1-9 compression as well as support uncompressed matrices and zonal references.

It is recommended not to compress data unless storage is a concern and if so it is recommended not to use compression levels above 4 or even 1 in many cases for performance reasons.

It is acceptable to mix compression levels for different matrices and zonalReferences within the same file.

---

noticed, only squared matrixes are allowed.

<sup>3</sup> HDF5 natively users a 0 based indexing.

<sup>4</sup> This feature can be used to associate taz "numbers" (which may not be contiguous, and may not even be numbers) with a set of ordinal integers corresponding to row or column index positions in the data set. Each element of the lookup map will contain the taz number associated with the corresponding index position in the matrix data.

Another common application would be to identify districts or jurisdictions such as cities or counties. In that case, each element of the zonal reference will contain the district identifier for the corresponding index position. The same district identifier may be used for multiple index positions.

<sup>5</sup> Zlib is the fastest and universal compression filter across all HDF5 implementations. Note that the compression filter slows down file writing.