**Institut für Softwaretechnik**

Favoritenstr. 9 - 11 / 188
A - 1040 Wien
Tel.: (+43) 1 58801 18801, Fax.: (+43) 1 5040532

Arbeitsgruppe: Information & Software Engineering

# SOMLib Data Files

## Technical Report TR-IR98-1

### Ver. 1.3.5 - 18. 07. 2000 (internal)

(History)

# SOMLib Data Files - General Information

Basically there are 5 different types of data files which are used to create 6 different files namely:

1. SOMLib Map Description File: SOMLib-Map-Descr
2. SOMLib Weight Vector File:
   - SOMLib Weigth Vectors
   - SOMLib Quantization Error Map
   - SOMLib Input Vector File
3. SOMLib Template Vector File: SOMLib Template Vector File
4. SOMLib Unit Description File: SOMLib Unit Descriptions
5. SOMLib Vector Description File: SOMLib Vector Descriptions

Alle of these files are built around the same basic structure which is defined as follows:

- Entries can be comments or parameter values
- **Comments** are indicated by a **#** character at the beginning of a line
- **Parameters** are indicated by a **$** character at the beginning of the line
- Comments are allowed
  - as a block of comment lines at the beginning of every file
  - after a parameter introduced by ' #' and running till end of line
    (e.g. *$TYPE vec # input vector file*)

  Note: no comment lines are allowed after the initial block of comments.
- Parameters are identified by a certain KEYWORD, followed by a blank and the according value, which can be either a **real, integer,** or **string** value or a **list** of these values separated by blanks
- For real numbers the separator character is a dot.
- If a value is not available, a default NULL value is given which is in the case of
  - string: VOID

    - real, integer: -1
- It is suggested to follow the order of the entries in the data files. If parameters are given in a different order, a warning shall be printed to stdout/log when trying to read the file - however, it should not be relied upon.
- Some of the parameters to be read are **mandatory**. When mandatory parameters are missing, reading fails with an error message.
- Some of the parameters to be read are **optional**. When optional parameters are missing, a warning shall be printed to stdout/log when trying to read the file with the reading process continuing.

In the following sections the 6 files are described in more detail, giving an idea of the contents and the intention of the file as well as its very structure in terms of the order of parameters as well as the distinction between mandatory (M) and optional (O) parameters. Furthermore, the relationships between the parameters are listed.

---

# SOMLib Map Description File

**Standard filename:** XXX.map
**Produced by:** SOM training program
**Modified by:** SOM mapping program, SOM quant-error program

This file describes the basic structure of the Self-Organizing Map, giving all the parameters used in the training process. It is initially written as result of the training process of the SOM. Additional Information attributes may be added as required by various programs.

Parameter Entries:

- **# Block of Comments**: (optional) several lines of comments each starting with #
- **$TYPE** : string, mandatory
  describes the topology of the map, currently used values: **descr**
- **$TOPOLOGY** : describes the topology of the map. Currently used values **rect, hex, hfm, gcs, gg, ghsom**
- **$XDIM** : integer, mandatory
  number of units in x-direction
- **$YDIM** : integer, mandatory
  number of units in y-direction
- **$VEC_DIM** : integer, mandatory
  dimensionality of weight vectors of map
- **$STORAGE_DATE** : string (or date, format tbd), optional
  date of storage time of trained map
- **$STORAGE_TIME** : string (or time, format tbd), optional
  time of storage time of trained map, probably combined with $STORAGE_DATE in one string?
- **$TRAINING_TIME** : integer, optional
  training time for map in seconds
- **$LEARNRATE_TYPE** : string, optional
  type of learn rate given as free text string

- **$LEARNRATE_INIT** : real, optional
  initial learn rate a0
- **$NEIGHBORHOOD_TYPE** : string, optional
  type of neighborhood region as free text string
- **$NEIGHBORHOOD_INIT** : real, optional
  initial neighborhood range e0
- **$RAND_INIT** : integer, optional
  init value for random number generator
- **$ITERATIONS_TOTAL** : integer, optional
  number of iterations of training process
- **$ITERATIONS_BUFFERED** : integer, optional
  number of iterations of one training process cycle when using buffered reading
- **$NR_TRAINVEC_TOTAL** : integer, optional
  number of input vectors used for training in total
- **$NR_TRAINVEC_BUFFERED** : integer, optional
  number of input vectors used for training on cycle when using buffered reading of input vectors
- **$VEC_NORMALIZED** : integer, optional
  indicator whether input vectors were normalized prior to the training process. permitted values 0, 1
- **$QUANTERROR_MAP** : real, optional
  quantization error of map
- **$QUANTERROR_VEC** : real, optional
  average input vector quantization error of map, i.e. the quantization error of the map divided by the number of vectors mapped onto the SOM ($QUANTERROR_MAP / $NR_TRAINVEC_TOTAL)
- **$URL_TRAINING_VEC** : string, optional
  URL of file containing input vectors used for training (Input Vector File, XXX.in)
- **$URL_TRAINING_VEC_DESCR** : string, optional
  URL of file containing description of input vectors used for training (Input Vector Description File, XXX.vec)
- **$URL_WEIGHT_VEC** : string, optional
  URL of file containing weight vectors of trained map (Weight Vector File, XXX.wgt)
- **$URL_QUANTERR_MAP** : string, optional
  URL of file containing quantization error vectors of trained map (Quantization Error File, XXX.err) written by SOM quant-error program
- **$URL_MAPPED_INPUT_VEC** : list of strings, optional
  URLs of files containing input vectors mapped onto trained map (Input Vector File, XXX.in) written by SOM mapping program
- **$URL_MAPPED_INPUT_VEC_DESCR** : list of strings, optional
  URLs of files containing descriptions of input vectors mapped onto trained map (Input Vector Description File, XXX.vec) written by SOM mapping program
- **$URL_UNIT_DESCR** : string, optional
  URL of file containing description of units of trained map (Unit Description File, XXX.unit)
- **$DESCRIPTION**: string or memo, optional free form text description of map to be used for display. Read to TO_EOF, i.e. description may span multiple lines.

Back to Top.

# SOMLib Weigth Vector File

**Standard filename:** XXX.wgt
**Produced by:** SOM init program, SOM training program
**Modified by:** -
**Demo-File:** animals_som.wgt

This file describes the weight vectors of the trained Self-Organizing Map. It is initially written as result of the SOM init program, read by the SOM training program as initialized map and finally written by the SOM training program after training
The files consists of two blocks, the first one describing the general SOM structure, the second giving the weight vectors of the SOM
The first 4 parameter entries are given as a sanity check to find out whether the given SOM map file and weight vector file match. If any of the 4 first parameters does not match the program should print a detailed error message and exit.

Parameter Entries:

- **# Block of Comments**: (optional) several lines of comments each starting with **#**
- **$TYPE** : string, mandatory
  describes the filetype and/or topology of the map, currently used values: **hex, rect, som, ghsom, rect_som, hex_som**
- **$XDIM** : integer, mandatory
  number of units in x-direction
- **$YDIM** : integer, mandatory
  number of units in y-direction
- **$VEC_DIM** : integer, mandatory
  dimensionality of weight vectors of map, = n
- **<x_1_1> ..... <x_1_n> <label_1>**
- ::::::::::::::::::::::::::::::::::: :::::::
- **<x_m_1> ..... <x_m_n> <label_m>**
  lists the n vector elements (n dimensions, i.e. n entries per line) of m weight vectors where m = XDIM x YDIM, being real values, followed by the label of the weight vector, being a string value like "SOM_MAP_Name_(X/Y)". All values are mandatory.
  If the number of weight vectors m is smaller than XDIM x YDIM the program reading this file should print a warning message.
  the order of vectors should be line by line, i.e. (0/0), (1/0), (2/0), from left to right, starting with (0/0) in the upper left corner of the map.
  If the number of vector elements does not match the given dimensionality VEC_DIM the program should print a detailed error message and exit.

Back to Top.

# SOMLib Quantization Error Map File

**Standard filename:** XXX.err

**Produced by:** SOM quantization error program
**Modified by:** -
**Demo-File:** animals_som.err

This file describes the quantization error vectors of the trained Self-Organizing Map. It is written by the SOM quantization error program based on a trained map and given input vectors
The files consists of two blocks, the first one describing the general SOM structure, the second giving the quantization error vectors of the SOM.
The file structure is identical to the general weight vector description file. The first 4 parameter entries are given as a sanity check to find out whether the given SOM map file and weight vector file match. If any of the 4 first parameters does not match the program should print a detailed error message and exit.

Parameter Entries:

- the parameters and the file structure is identical to the SOMLib Weigth Vector File, with the $TYPE Parameter being set to **qerr, qerr_rect, qerr_hex, err** etc.

Back to Top.

# SOMLib Input Vector File

**Standard filename:** XXX.in
**Produced by:** Parser, Vector Generator
**Modified by:** -
**Demo-File:** animals.in

This file describes the input vectors to be used for the training process of a Self-Organizing Map. It is written by the parser or vector generator program creating the vector structure
The files consists of two blocks, the first one describing the input vectors in order to follow the general file structure of weight vector files, the second giving the input vectors
The file structure is identical to the SOMLib Weight Vector File. However, some semantical changes of the first 4 vector entries are as follows

Parameter Entries:

- **# Block of Comments**: (optional) several lines of comments each starting with **#**
- **$TYPE** : string, mandatory
  **vec, vec_tf, vec_tfxidf, vec_bin, vec_structure** to indicate input vector file further information about the type of quantization and encoding used can be packed into this string
- **$XDIM** : integer, mandatory
  number of input vectors in file
- **$YDIM** : integer, mandatory
  1; this allows again for XDIM x YDIM to give the total number of vectors to be read from file.
  **NOTE:** for any program reading this file: the number of vectors listed in the file is given by **XDIM * YDIM**, and not by XDIM alone!
- **$VEC_DIM** : integer, mandatory
  dimensionality of weight vectors of map, = n

The remainder of the file is identical to the SOMLib Weigth Vector File:

- **<x_1_1> ..... <x_1_n> <VEC_ID_1>**
- :::::::::::::::::::::::::::::::::: :::::::
- **<x_m_1> ..... <x_m_n> <VEC_ID_m>**
  lists the n vector elements (n dimensions, i.e. n entries per line) of m weight vectors where m = XDIM (i.e. = XDIM x YDIM with YDIM being 1), being real values, followed by the **<VEC_ID>**, i.e. the label of the weight vector, being a string value. All values are mandatory.
  If the number of weight vectors m is smaller than XDIM x YDIM the program reading this file should print a warning message.
  If the number of vector elements does not match the given dimensionality VEC_DIM the program should print a detailed error message and exit.

Back to Top.

# SOMLib Template Vector File

**Standard filename:** XXX.tv
**Produced by:** Parser, Vector Generator
**Modified by:** -
**Demo-File:** animals.tv

This file describes the template vectors providing the attribute structure of the input vectors used for the training process of a Self-Organizing Map. It is written by the parser or vector generator program creating the vector structure

Parameter Entries:

- **# Block of Comments**: (optional) several lines of comments each starting with **#**
- **$TYPE** : string, mandatory
  **template** to indicate template vector file further information may be packed into this string
- **$XDIM** : integer, mandatory
  nr. of columns used in layout, min.: 2 (Nr. and Attribute), max. currently 7
- **$YDIM** : integer, mandatory
  number of feature vectors in corresponding SOMLib Input Vector File
- **$VEC_DIM** : integer, mandatory
  dimensionality of weight vectors of map, = n

The remainder of this files lists the attributes of the vectors by 7 columns of information as follows

- **<nr> <attr> [<df> <tf_coll> <max_tf> <min_tf> <mean_tf> # comment]**
- :::::::::::::::::::::::::::::::::: :::::::
- **<nr> <attr> [<df> <tf_coll> <max_tf> <min_tf> <mean_tf> # comment]**
- with
  - **<nr>**: int, consecutive numbering of attributes, starting with 0, up to VEC_DIM-1
  - **<attr>**: string, label or name of the attribute, i.e. keyword etc.

- **<df>**: int, document frequency - in how many documents or feature vectors is this attribute present, i.e. has an input vector value <> 0
- **<tf_coll>**: real, term frequency in the whole collection - how often does this attribute show up in the whole collection of feature vectors, ako of counter for the attribute, sum of all values of the attribute (sum across all feature vectors)
- **<min_tf>**: real, minimal value of this attribute in the collection of feature vectors
- **<max_tf>**: real, maximum value of this attribute in the collection of feature vectors
- **<mean_tf>**: real, mean value of this attribute in the collection of feature vectors
- **# comment**: optional comment for attributes till end of line

Back to Top.

# SOMLib Unit Description File

**Standard filename:** XXX.unit
**Produced by:** SOM training program
**Modified by:** SOM mapping program, LabelSOM program

This file describes the units of the trained Self-Organizing Map. It is written by the SOM training program.
The files consists of two blocks, the first one describing the general SOM structure, the second giving a specific description of every unit
The first 3 parameter entries are given as a sanity check to find out whether the given SOM map file and weight vector file match. If any of the 3 first parameters does not match the program should print a detailed error message and exit.

Parameter Entries:

- **# Block of Comments**: (optional) several lines of comments each starting with #
- **$TYPE** : string, mandatory
  describes the topology of the map, currently used values: hex, rect
- **$XDIM** : integer, mandatory
  number of units in x-direction
- **$YDIM** : integer, mandatory
  number of units in y-direction

This header describes the general SOM structure.
Following this block, the second block contains the following set of attributes per unit:

- **# Block of Comments**: (optional) several lines of comments each starting with #
- **$POS_X** : integer, mandatory
  x coordinate of unit in standard visualization of SOM (column)
- **$POS_Y** : integer, mandatory
  y coordinate of unit in standard visualization of SOM (line)
- **$UNIT_ID** : string, optional
  short label / id of unit as free text string, e.g. (0/0), (1/0), etc.

- **$QUANTERROR_UNIT** : real, optional
  quantization error of unit
- **$QUANTERROR_UNIT_AVG** : real, optional
  average input vector quantization error of unit, i.e. QUANTERROR_UNIT divided by the number of weight vectors mapped onto this unit (NR_VEC_MAPPED)
- **$AC_POS_X** : real, optional
  x coordinate of unit in AC visualization of SOM
- **$AC_POS_Y** : real, optional
  y coordinate of unit in AC visualization of SOM
- **$UMAT_UNIT** : real, optional
  averaged distance for U-Matrix representation for unit
- **$UMAT_RIGHT** : real, optional
  distance to the right neighbor for U-Matrix representation
- **$UMAT_DOWN** : real, optional
  averaged distance to the lower neighbor for U-Matrix representation in case of hexagonal map arrangement distance to the lower neighbor for U-Matrix representation in case of rectangle map arrangement
- **$UMAT_DOWN_LEFT** : real, optional
  averaged distance to the left lower neighbor for U-Matrix representation in case of hexagonal map arrangement averaged distance to the lower left neighbor for U-Matrix representation in case of rectangle map arrangement
- **$UMAT_DOWN_RIGHT** : real, optional
  averaged distance to the right lower neighbor for U-Matrix representation in case of hexagonal map arrangement averaged distance to the lower right neighbor for U-Matrix representation in case of rectangle map arrangement
- **$NR_VEC_MAPPED** : integer, optional
  number of input vectors mapped onto this unit written by SOM training program
- **$MAPPED_VECS** : list of string, optional
  list of strings giving the VEC_ID's of input vectors (labels) mapped onto this unit. Used for static referencing The number should be identical to NR_VEC_MAPPED. IF not a warning should be printed. Written by SOM mapping program
- **$MAPPED_VECS_DIST** : list of real, optional
  distances by which vectors are mapped onto the unit
- **$NR_SOMS_MAPPED** : integer, optional
  number of other SOMs mapped onto this unit written by hierarchical SOM training program or by integrating SOM training program
- **$URL_MAPPED_SOMS** : list of strings, optional
  list of strings giving the URL's of SOM Map Description Files (filename XXX.map) Used for dynamic referencing The number should be identical to NR_SOMs_MAPPED. IF not a warning should be printed. Written by SOM mapping program
- **$MAPPED_SOM_DIST** : list of real, optional
  distances by which SOM vectors are mapped onto the unit. for GHSOM: mqe of that unit
- **$NR_UNIT_LABELS** : int, optional
  number of labels for this unit, written by LabelSOM program
- **$UNIT_LABELS** : list of strings, optional
  list of labels for unit, written by LabelSOM program
- **$UNIT_LABELS_QE** : list of real, optional
  quantization error of the labels

- **$UNIT_LABELS_WGT** : list of real, optional
  weight of the labels
- **$UNIT_LABELS_LEFT** : list of strings, optional
  list of labels for unit, written by LabelSOM program
- **$UNIT_LABELS_LEFT_DIFF** : list of real, optional
  difference to left neighbor labels of the labels
- **$UNIT_LABELS_RIGHT** : list of strings, optional
  list of labels for unit, written by LabelSOM program
- **$UNIT_LABELS_RIGHT_DIFF** : list of real, optional
  difference to right neighbor labels of the labels
- **$UNIT_LABELS_UP** : list of strings, optional
  list of labels for unit, written by LabelSOM program
- **$UNIT_LABELS_UP_DIFF** : list of real, optional
  difference to upper neighbor labels of the labels
- **$UNIT_LABELS_DOWN** : list of strings, optional
  list of labels for unit, written by LabelSOM program
- **$UNIT_LABELS_DOWN_DIFF** : list of real, optional
  difference to down neighbor labels of the labels
- **$URL_RELATED_UNITS** : list of string, optional
  list of strings giving URLs of related units. These can be links to units within the same map or to units on other SOM maps. The URL willmost probably consist of the URL of the SOM map file (XXX.map) plus the unit location within the map given as '#(x/y)', details tbd.
- **$DESCRIPTION** : string, optional
  free form text description of unit, terminated by newline

Back to Top.

# SOMLib Vector Description File

**Standard filename:** XXX.vec
**Produced by:** Parser or vector generator program
**Modified by:** SOM browsing software

This file describes the input vectors for a self-organizing map. It is written by the parser or vector generator program and describes the properties of each vector
The file consists of one set of attributes per vector with the very attributes still being subject to modification, or rather, extension. The structure of the description of the vectors follows in general the structure of the unit description file. Further attributes will be added as the necessity arises, especially in the context of metaphor graphics. Furthermore, the question whether each of the description files should be kept as an independet file or be part of one lare file comrising the whole collection has not been fully decided upon.
The attributes considered so far are:

Parameter Entries:

- **# Block of Comments**: (optional) several lines of comments each starting with #
- **$TYPE**: string, mandatory

**vecdescr** to indicate input vector description file, further information may be packed into this string
- **$NR_Files**: int, mandatory
  number of files / vectors described in this file
- **$NR_METADATA_ATTR**: int, mandatory
  number of metadata attributes per entry

The header above describes the general file structure.
following this block, the second block contains the following set of attributes per vector/file:

- **# Block of Comments**: (optional) several lines of comments each starting with #
- **$VEC_ID** : string, mandatory
  ID of vector, a kind of short label or unique ID specially for documents split into several vectors
- **$LABEL** : string, optional
  label of vector, full name, file name, possibly identical to $VEC_ID
- **$URL_DOC** : string, optional
  URL of document being the basis for the vector
- **$TYPE** : string, mandatory
  giving the type of the vector with currently supported types being DOC, SOM and VEC, with **DOC** for vectors describing documents (that can be referenced), **SOM** for other SOMs (that can be referenced) and **VEC** for general vectors (that cannot be referenced)
- **DUBLIN-CORE Metadata Attribute Set**: All Attributes of the Dublin Core Metadata Set - recommended set of attributes for Input Vector Description, such as **creator, subject, keywords** etc.
  Additional attributes only where necessary, e.g. price, domain, etc.
  May also be used to directly accomodate libViewer attributes.
- **$SIZE** : integer, optional
  length of document in bytes, size of SOM in terms of units or documents mapped, details tbd.
- **$PRICE**: price for the document etc.
- **$NR_TIMES_REFERENCED** : integer, optional
  number of times this vector was referenced. initialized to 0 by parser program, modified by SOM browsing software
- **$LAST_REFERENCED** : string / date, optional
  date of last reference to vector modified by SOM browsing software
- **$DESCRIPTION** : string, optional
  free form text description of unit, terminated by newline

Back to Top.

# History

- Vers. 1.3.5: (18.7.2000):
  - fixed formatting
  - changed SOMLib Input Vector Description File structure and setup
  - renamed file to SOMLib Vector Description File
- Vers. 1.3.4: (17.7.2000):
  - fixed formatting (added $)

    - changed structure and occurrence of comments
    - added Dublin Core recommendation to Input Vector Description File
- Vers. 1.3.3: (11.7.2000):
  * fixed formatting errors
  * added Distances in Unit Description Files for mapped vecs, soms, and labels
  * added $NR_UNIT_LABELS
- Vers. 1.3.2: (10.7.2000):
  * fixed formatting errors
  * added 7. attribute for template vector file: mean
- Vers. 1.3.1: (8.7.2000):
  * fixed formatting errors
- Vers. 1.3: (6.7.2000):
  * adapted SOM Input Vector File
  * changed $TOPOLOGY into $TYPE
  * changed NODE into UNIT
  * removed $SIGNATURE
  * added template vector file description
  * added some demo-files (artificially created - real ones to be added)
  * removed hex-SOM required condition for x/y-Pos values
- Vers. 1.2: (18.11.1998):
  * SOM Unit Description File: X_POS, Y_POS mandatory instead of optional
  * 0 <= POS_X < 2*XDIM and 0 <= POS_Y < 2*YDIM to accomodate hex-location
  * 0 <= AC_POS_X < 2*XDIM and 0 <= AC_POS_Y < 2*YDIM to accomodate hex-location
- Vers. 1.1: (03.11.1998):
  * added UMAT_RIGHT, UMAT_UNIT, UMAT_DOWNRIGHT, UMAT_DOWNLEFT to SOM Unit Description
  * changed URL_VEC to URL_DOC in Input Vector Descriptions
  * added (keyword) to SOM Map Description File to indicate whether a description follows
  * spelling
- Vers. 1.0: (17.09.1998):
  * basic Datafile Structure

Back to Top.

---

IfS Home

---

Comments: rauber@ifs.tuwien.ac.at