

gbmx Agent Reference Manual

Hiroshisa Mori
joshua@globalbase.org

Translated by
Reiko Inoue Bendtsen c/o Suzaku Translations
(<http://www.suzaku-translations.com/>)

Edition 2010-12-01

Contents

1	Overview	2
1.1	Abstract and Goal	2
1.2	Human Requirements	2
1.3	System Requirements	2
2	gbmx XL Function Reference	3
2.1	Abstract	3
2.2	Inheritance Information	3
2.3	Functions	4
2.3.1	gmxGetPoint	4
2.3.2	gmxCreate	6
2.3.3	gmxCreate:gmxMatrixParam	7
2.3.4	gmxCreate:gmxMatrixParam:gmxPriority	8
2.3.5	gmxCreate:gmxMatrixDimDivide	9
2.3.6	gmxCreate:gmxMatrixBlockSize	10
2.3.7	gmxCreate:gmxMatrixPixelSize	11
2.3.8	gmxCreate:gmxChannelInfo	12
2.3.9	gmxCreate:gmxPhase	14
2.3.10	gmxClose	15
2.3.11	gmxPNMStatus	16
2.3.12	gmxImportPNM	17
2.3.13	gmxScan	18
2.3.14	gmxCopy	19
2.3.15	gmxStatus	20
2.3.16	Get	21
2.3.17	gmxPgPlotField	23
3	gmxStatus List Type Reference	27
3.1	Abstract	27
3.2	XML Elements	28
3.2.1	gmxStatus	28
3.2.2	gmxStatus:gmxMatrixParam	29
3.2.3	gmxStatus:gmxMatrixParam:gmxPriority	30
3.2.4	gmxStatus:gmxMatrixDimDivide	31
3.2.5	gmxStatus:gmxMatrixBlockSize	32
3.2.6	gmxStatus:gmxMatrixPixelSize	33
3.2.7	gmxStatus:gmxChannelInfo	34
3.2.8	gmxStatus:gmxPhase	36

Chapter 1

Overview

1.1 Abstract and Goal

gbmx provides functions and mathematical formulas for manipulating matrix data, which is a common format of bitmap and vector data in GLOBALBASE. Please refer to [UNDEF REF (matrix)] for the detailed explanation on the structure of matrix files.

1.2 Human Requirements

In order to understand and make use of this manual, you must be familiar with xl scripts [UNDEF REF (xl)]. Moreover, please refer to "Protocol Reference Model of GLOBALBASE Architecture: The Autonomous Decentralized GIS" [?] for the explanation on the XL protocol. It is also necessary that you have at least a basic understanding of the principles involved in searching and overlapping GLOBALBASE coordinate systems. GLOBALBASE Technical Materials [UNDEF REF (GLOBALBASE-tech)] provides useful background information on these technical issues. We also recommend you try using COSMOS GLOBALBASE VIEWER [1] in order to ensure that you have first-hand understanding of the GLOBALBASE functions.

1.3 System Requirements

A POSIX-compliant system is required. You will be able to use the xlsv functions after installing LANDSCAPE GLOBALBASE VIEWER [2] on a POSIX machine.

Chapter 2

gbmx XL Function Reference

2.1 Abstract

gbmx XL functions refer to various XL language functions for operations on matrices.

2.2 Inheritance Information

gbmx agents inherit the functions of standard agent [UNDEF REF (xl)]as is.

2.3 Functions

2.3.1 gmxGetPoint

■ Prototype

<gmxGetPoint> *filename pointdata* </gmxGetPoint>

■ Arguments

filename [1] XLT_STRING [UNDEF REF (XLT_STRING)]

pointdata [1] XLT_PAIR [UNDEF REF (XLT_PAIR)]

■ Attributes

path-type [arbitrary]: How a filename path is given. absolute/related/network/key default="absolute"

mode [arbitrary]: How a return value is given. normal/block default="normal"

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1,Env0

■ Explanation

This function acquires data of coordinate positions, *pointdata*, from the matrix file *filename*. If the channel type is vector data, the return value varies depending on how the mode attribute is given. If mode="normal," the value of dim_code corresponding to the vector is returned. If mode="block," the vector data itself is returned. Note that if the channel type is not vector data, mode must be set to "block."

filename can be given as absolute path or relative path from the root of the xl server. If you specify the attribute path-type="absolute," an absolute path is specified and if you specify path-type="related," a relative path is specified. Note that the method of specifying absolute path cannot be used without access right to Env1 (XL administrator privilege).

The data format of *pointdata* is a list of XLT_INTEGER [UNDEF REF (XLT_INTEGER)]or XLT_FLOAT [UNDEF REF (XLT_FLOAT)]with a length of the dimensions indicated by *filename* plus 1,if only one point is given. One element is added because one integer value that gives the access level is placed at the head. If two or more points are added, multiple lists indicating coordinate values shall be specified.

■ Return Value

If no errors occur, an S formula that lists (channel number [XLT_INTEGER [UNDEF REF (XLT_INTEGER)]] coordinate position data) for the number of channels is returned. The coordinate position data takes the value of the corresponding type depending on channel data type, either XLT_INTEGER [UNDEF REF (XLT_INTEGER)]or XLT_FLOAT [UNDEF REF (XLT_FLOAT)]. If mode="block," it takes the value of the corresponding type.

■ Error

- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

1. "param"

The format of *pointdata* is illegal.

2. "attribute::mode"

The value of the mode attribute is not supported.

3. "attribute:path-type"

The value of path-type is not supported.

- XLE_PROTO_INV_OBJECT [UNDEF REF (XLE_PROTO_INV_OBJECT)]

No pixels exist for the coordinate position given by *pointdata*.

Connection is not allowed as a result of authentication.

- XLE_PROTO_OPEN_FILE [UNDEF REF (XLE_PROTO_OPEN_FILE)]

Unable to open the file specified by *filename*.

- XLE.SEMANTICS_TYPE.MISSMATCH [UNDEF REF (XLE.SEMANTICS_TYPE.MISSMATCH)]
The data types of pointdata and filename do not match.
- XLE.PROTO_PERMISSION_DENIED [UNDEF REF (XLE.PROTO_PERMISSION_DENIED)]
It was attempted to execute path-type="absolute" in the Env0 environment.

■ **References**

■ **Bugs**

2.3.2 gmxCreate

■ Prototype

<gmxGetPoint> *gmxCreateElements* </gmxGetPoint>

■ Arguments

gmxCreate

gmxCreate / gmxMatrixParam (Section 2.3.3)

gmxCreate / gmxMatrixParam/ gmxPriority (Section 2.3.4)

gmxCreate / gmxMatrixDimDivide (Section 2.3.5)

gmxCreate / gmxMatrixBlockSize (Section 2.3.6)

gmxCreate / gmxMatrixPixelSize (Section 2.3.7)

gmxCreate / gmxChannelInfo (Section 2.3.8)

gmxCreate / gmxPhase (Section 2.3.9)

■ Attributes

filename [mandatory]: The name of the file to be created

■ Evaluation Type

normal

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1

■ Explanation

This function creates a matrix file and makes settings for the matrix file to be created.

■ Return Value

MX_ENTRY ID of the generated matrix file (XLT_INTEGER [UNDEF REF (XLT_INTEGER)])

■ Error

- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]
The attribute filename is not specified.
- XLE_PROTO_OPEN_FILE [UNDEF REF (XLE_PROTO_OPEN_FILE)]
Failed to open the file.

■ References

■ Bugs

2.3.3 gmxCreat:gmXMatrixParam

■ Prototype

```
<gmXMatrixParam> <gmXPriority/> .... </gmXMatrixParam>
```

■ Arguments

gmxCreat (Section 2.3.2)

gmxCreat/ **gmXMatrixParam**

gmxCreat/ **gmXMatrixParam** / gmXPriority (Section 2.3.4)

gmxCreat/ gmXMatrixDimDivide (Section 2.3.5)

gmxCreat/ gmXMatrixBlockSize (Section 2.3.6)

gmxCreat/ gmXMatrixPixelSize (Section 2.3.7)

gmxCreat/ gmXChannelInfo (Section 2.3.8)

gmxCreat/ gmXPhase (Section 2.3.9)

■ Attributes

channel_nos [mandatory]: Number of channels

dim [mandatory]: Dimensions

dim [mandatory]: Dimensions

index_hem [arbitrary]: Enables the on/off HEM function (MPF_INDEX_HEM). The default is "off."

cache_file [arbitrary]: Enable the on/off cache file function (MPF_CACHE_FILE). The default is default="off."

■ Evaluation Type

normal

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1/gmxCreat

■ Explanation

This function sets the basic parameters and priorities of the matrix to be generated.

■ Return Value

XLT_NULL [UNDEF REF (XLT_NULL)]or error

■ Error

- XLE_SEMANTICS_INV_FORMAT [UNDEF REF (XLE_SEMANTICS_INV_FORMAT)]

The corresponding element was used outside the gmxCreat function.

- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

- "channel_nos is required"

The channel_nos attribute is not specified.

- "invalid channel_nos value"

The value of the channel_nos attribute is 0or less than

- "dim is required"

The dim attribute is not specified.

- "invalid dim value"

The value of dim is 0or less.

- XLE_PROTO_UNSUTISFIED_INFO [UNDEF REF (XLE_PROTO_UNSUTISFIED_INFO)]

set_matrix_param caused an error. The number indicates an error code.

■ References

■ Bugs

2.3.4 gmxCreat:gmXMatrixParam:gmXPriority

■ Prototype

<gmXPriority/>

■ Arguments

gmxCreat (Section 2.3.2)

gmxCreat/ gmXMatrixParam (Section 2.3.3)

gmxCreat/gmXMatrixParam/ **gmXPriority**

gmxCreat/ gmXMatrixDimDivide (Section 2.3.5)

gmxCreat/ gmXMatrixBlockSize (Section 2.3.6)

gmxCreat/ gmXMatrixPixelSize (Section 2.3.7)

gmxCreat/ gmXChannelInfo (Section 2.3.8)

gmxCreat/ gmXPhase (Section 2.3.9)

■ Attributes

area [mandatory] Integer: Priority setting area

pri [mandatory] Integer: Priority

mode [arbitrary] Character string: active/direct/server/client mode; the default is "active."

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1/gmxCreat/gmXMatrixParam

■ Explanation

This function sets the priority of the matrix to be generated.

■ Return Value

XLT_NULL [UNDEF REF (XLT_NULL)]or error.

■ Error

- XLE.SEMANTICS_INV_FORMAT [UNDEF REF (XLE.SEMANTICS_INV_FORMAT)]
The corresponding element was used outside gmXMatrixParam.
- XLE.PROTO_INV_PARAM [UNDEF REF (XLE.PROTO_INV_PARAM)]
 - "area"
The area attribute is not specified.
 - "pri"
The pri attribute is not specified.

■ References

■ Bugs

2.3.5 gmxCreate:gmxMatrixDimDivide

■ Prototype

<gmxMatrixDimDivide/>

■ Arguments

gmxCreate (Section 2.3.2)

gmxCreate/ gmxMatrixParam (Section 2.3.3)

gmxCreate/gmxMatrixParam/ gmxPriority (Section 2.3.4)

gmxCreate/ **gmxMatrixDimDivide**

gmxCreate/ gmxMatrixBlockSize (Section 2.3.6)

gmxCreate/ gmxMatrixPixelSize (Section 2.3.7)

gmxCreate/ gmxChannelInfo (Section 2.3.8)

gmxCreate/ gmxPhase (Section 2.3.9)

■ Attributes

dim [mandatory] Integer: dim to be set

value [mandatory] Integer: Number of divisions to be set

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1/gmxCreate

■ Explanation

This function specifies the number of divisions for each dimension of the matrix to be created. The actual number of divisions given is 2^{divide} .

■ Return Value

XLT_NULL [UNDEF REF (XLT_NULL)]or error

■ Error

- XLE.SEMANTICS_INV_FORMAT [UNDEF REF (XLE.SEMANTICS_INV_FORMAT)]

The corresponding element was used outside gmxCreate.

- XLE.PROTO_INV_PARAM [UNDEF REF (XLE.PROTO_INV_PARAM)]

- "dim"

- The dim attribute is not specified.

- "divide"

- The divide attribute is not specified.

- XLE.PROTO_UNSUTISFIED_INFO [UNDEF REF (XLE.PROTO_UNSUTISFIED_INFO)]

set_matrix_dim_divide caused an error. The number indicastes an error code.

■ References

■ Bugs

2.3.6 gmxCreate:gmxMatrixBlockSize

■ Prototype

<gmxMatrixBlockSize/>

■ Arguments

gmxCreate (Section 2.3.2)

gmxCreate/ gmxMatrixParam (Section 2.3.3)

gmxCreate/gmxMatrixParam/ gmxPriority (Section 2.3.4)

gmxCreate/ gmxMatrixDimDivide (Section 2.3.5)

gmxCreate/ **gmxMatrixBlockSize**

gmxCreate/ gmxMatrixPixelSize (Section 2.3.7)

gmxCreate/ gmxChannelInfo (Section 2.3.8)

gmxCreate/ gmxPhase (Section 2.3.9)

■ Attributes

dim [mandatory] Integer: dim to be set

value [mandatory] Integer: Block size to be set

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1/gmxCreate

■ Explanation

This function specifies the size of each block of the matrix to be generated in the direction of each dimension. The actual block size is $2^{\text{block_size}}$.

■ Return Value

XLT_NULL [UNDEF REF (XLT_NULL)]or error

■ Error

- XLE.SEMANTICS_INV_FORMAT [UNDEF REF (XLE.SEMANTICS_INV_FORMAT)]

The corresponding element was used outside gmxCreate.

- XLE.PROTO_INV_PARAM [UNDEF REF (XLE.PROTO_INV_PARAM)]

– "dim"

The dim attribute is not specified.

– "block_size"

The divide attribute is not specified.

- XLE.PROTO_UNSUTISFIED_INFO [UNDEF REF (XLE.PROTO_UNSUTISFIED_INFO)]

set_matrix_dim_divide caused an error. The number indicates an error code.

■ References

■ Bugs

2.3.7 gmxCreate:gmxMatrixPixelSize

■ Prototype

<gmxMatrixPixelSize/>

■ Arguments

gmxCreate (Section 2.3.2)

gmxCreate/ gmxMatrixParam (Section 2.3.3)

gmxCreate/gmxMatrixParam/ gmxPriority (Section 2.3.4)

gmxCreate/ gmxMatrixDimDivide (Section 2.3.5)

gmxCreate/ gmxMatrixBlockSize (Section 2.3.6)

gmxCreate/ **gmxMatrixPixelSize**

gmxCreate/ gmxChannelInfo (Section 2.3.8)

gmxCreate/ gmxPhase (Section 2.3.9)

■ Attributes

dim [mandatory] Integer: dim to be set

value [mandatory] Integer: Block size to be set

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1/gmxCreate

■ Explanation

This function specifies the pixel size of the entire matrix to be generated in the direction of each dimension. The actual pixel size is $2^{\text{pixel_size}}$.

■ Return Value

XLT_NULL [UNDEF REF (XLT_NULL)]or error

■ Error

- XLE.SEMANTICS_INV_FORMAT [UNDEF REF (XLE.SEMANTICS_INV_FORMAT)]
The corresponding element was used outside gmxCreate.
- XLE.PROTO_INV_PARAM [UNDEF REF (XLE.PROTO_INV_PARAM)]
 - "dim"
The dim attribute is not specified.
 - "pixel_size"
The divide attribute is not specified.
- XLE.PROTO_UNSUTISFIED_INFO [UNDEF REF (XLE.PROTO_UNSUTISFIED_INFO)]
The set_matrix_dim_divide caused an error. The number indicates an error code.

■ References

■ Bugs

2.3.8 gmxCreate:gmxCChannelInfo

■ Prototype

<gmxCChannelInfo/> <gmxCChannelInfo> *initialize_data* <gmxCChannelInfo>

■ Arguments

gmxCreate (Section 2.3.2)

gmxCreate/ gmxCMatrixParam (Section 2.3.3)

gmxCreate/gmxCMatrixParam/ gmxCPriority (Section 2.3.4)

gmxCreate/ gmxCMatrixDimDivide (Section 2.3.5)

gmxCreate/ gmxCMatrixBlockSize (Section 2.3.6)

gmxCreate/ gmxCMatrixPixelSize (Section 2.3.7)

gmxCreate/ **gmxCChannelInfo**

gmxCreate/ gmxCPhase (Section 2.3.9)

■ Attributes

channel [mandatory] Integer: Target channel number

type [mandatory] Character string: Data type of a channel

send [arbitrary] Character string: Turns on the on/off MF_SEND flag. The default is "off."

file [arbitrary] Character string: Turns on the on/off MF_SFILE flag. The default is "off."

visu [arbitrary] Character string: Turns on the on/off MF_VISU flag. The default is "off."

send_file [arbitrary] Character string: Turns on the on/off MF_SEND_FILE flag. The default is "off."

send_visu [arbitrary] Character string: Turns on the on/off MF_SEND_VISU flag. The default is "off."

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1/gmxCreate

■ Explanation

This function specifies information of each channel of the matrix to be generated. The attributes specify data type and flags. The list of data types that can be set is shown below.

- Basic data types
 - bit - Bit type
 - int8 - 8-bitsigned integer type
 - int16 - 16-bitsigned integer type
 - int32 - 32-bitsigned integer type
 - int64 - 64-bitsigned integer type
 - uint8 - 8-bitunsigned integer type
 - uint16 - 16-bitunsigned integer type
 - uint32 - 32-bitunsigned integer type
 - uint64 - 64-bitunsigned integer type
 - float - Floating point type
 - double - Double-precision floating point type
 - string - Character string type
 - block - Block type
 - sexp - XL S formula type
 - rgb8 - 8-bitRGB type
- Vector types
 - int8_v - 8-bitsigned integer type
 - int16_v - 16-bitsigned integer type

- int32_v - 32-bit signed integer type
- int64_v - 64-bit signed integer type
- uint8_v - 8-bit unsigned integer type
- uint16_v - 16-bit unsigned integer type
- uint32_v - 32-bit unsigned integer type
- uint64_v - 64-bit unsigned integer type
- float_v - Floating point type
- double_v - Double-precision floating point type
- string_v - Character string type
- rgb8_v - 8-bit RGB type

The `gmxChannelInfo` function is able to specify the initial data corresponding to the data types above as internal elements. Integer type and floating point type data is converted to the relevant types above and initialized. If no internal elements are given, data is not initialized.

■ **Return Value**

XLT_NULL [UNDEF REF (XLT_NULL)] or error

■ **Error**

- XLE_SEMANTICS_INV_FORMAT [UNDEF REF (XLE_SEMANTICS_INV_FORMAT)]
The corresponding element was used outside `gmxCreate`.
- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]
 - "channel"
The channel attribute is not specified.
 - "type"
The type attribute is not specified.
 - "invalid type (type)"
The value of the type attribute is invalid.
- XLE_PROTO_UNSATISFIED_INFO [UNDEF REF (XLE_PROTO_UNSATISFIED_INFO)]
`set_matrix_channel_info` caused an error. The number indicates an error code.
- XLE_SEMANTICS_TYPE_MISMATCH [UNDEF REF (XLE_SEMANTICS_TYPE_MISMATCH)]
The data type given by the type attribute and the value type given during initialization do not match.

■ **References**

■ **Bugs**

2.3.9 gmxCreate:gmxPhase

■ Prototype

<gmxPhase> *Phase_commands* <gmxPhase> <gmxPhase/>

■ Arguments

gmxCreate (Section 2.3.2)

gmxCreate/ gmxMatrixParam (Section 2.3.3)

gmxCreate/gmxMatrixParam/ gmxPriority (Section 2.3.4)

gmxCreate/ gmxMatrixDimDivide (Section 2.3.5)

gmxCreate/ gmxMatrixBlockSize (Section 2.3.6)

gmxCreate/ gmxMatrixPixelSize (Section 2.3.7)

gmxCreate/ gmxChannelInfo (Section 2.3.8)

gmxCreate/ **gmxPhase**

■ Attributes

area [mandatory] Integer: Phase area to be set.

type [arbitrary] Character string: Specification of normal/direct/equ Phase_commands; the default is "normal."

equ [arbitrary] Integer: Target phase area of equivalent

mode [arbitrary] Character string: active/direct/server/client mode; the default is "active."

■ Evaluation Type

normal

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1/gmxCreate

■ Explanation

This function specifies the phase script of the matrix to be generated.

This function specifies the mode to be set for the attribute mode. This function specifies the phase area code to be set for the attribute area.

The attribute type can take the values normal or direct, which is the same value as the type attribute of the quote function, as well as equ. If normal is specified, an evaluated \$ element is set as the phase command in the same way as for interpretation of the normal attribute of the quote function. On the other hand, if direct is specified, \$ elements are not evaluated. If equ is specified, the same phase command as the area code given by the equ attribute is set for the script.

■ Return Value

XLT_NULL [UNDEF REF (XLT_NULL)]or error

■ Error

- XLE_SEMANTICS_INV_FORMAT [UNDEF REF (XLE_SEMANTICS_INV_FORMAT)]

The corresponding element was used outside gmxCreate.

- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

– "area"

The area attribute is not specified.

– "equ"

The equ attribute is not specified, although the type is set to "equ."

- XLE_PROTO_UNSUTISFIED_INFO [UNDEF REF (XLE_PROTO_UNSUTISFIED_INFO)]

set_matrix_cal caused an error. The number indicates an error code.

■ References

■ Bugs

2.3.10 gmxClose

■ Prototype

<gmxClose/>

■ Arguments

■ Attributes

id [mandatory] Integer: A matrix's MX_ENTRY ID

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env0

■ Explanation

The function closes the target matrix file.

■ Return Value

XLT_NULL [UNDEF REF (XLT_NULL)]or error

■ Error

- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

1. "id"

The id attribute is not specified.

2. "id invalid value"

The id value is outside the valid range.

■ References

■ Bugs

2.3.11 gmxPNMStatus

■ Prototype

<gmxPNMStatus> *filename* <gmxPNMStatus>

■ Arguments

filename [mandatory] XLT_STRING [UNDEF REF (XLT_STRING)]File name

■ Attributes

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1

■ Explanation

This function returns the header information of a PNM file. The header information has the following format.

■ Return Value

```
<gmxPNMStatus>
  type
  width
  height
  max
</gmxPNMStatus>
```

type is a character string indicating the PNM file type, such as P1 and P2. *width* and *height* indicate numbers of vertical and horizontal pixels of an image. *max* indicates color resolution. All of these are defined in a PNM file.

■ Error

- XLE_SEMANTICS_TYPE_MISMATCH [UNDEF REF (XLE_SEMANTICS_TYPE_MISMATCH)]
- XLE_PROTO_OPEN_FILE [UNDEF REF (XLE_PROTO_OPEN_FILE)]

■ References

■ Bugs

2.3.12 gmxImportPNM

■ Prototype

<gmxImportPNM> *filename channel level x y* <gmxImportPNM>

■ Arguments

filename [1] XLT_STRING [UNDEF REF (XLT_STRING)]File name

channel [1] XLT_INTEGER [UNDEF REF (XLT_INTEGER)]Import destination channel number

level [1] XLT_INTEGER [UNDEF REF (XLT_INTEGER)]Import destination level

x [1] XLT_INTEGER [UNDEF REF (XLT_INTEGER)], XLT_FLOAT [UNDEF REF (XLT_FLOAT)]x coordinate value where drawing is started

y [1] XLT_INTEGER [UNDEF REF (XLT_INTEGER)], XLT_FLOAT [UNDEF REF (XLT_FLOAT)]y coordinate value where drawing is started

■ Attributes

id [mandatory] XLT_INTEGER [UNDEF REF (XLT_INTEGER)]Matrix descriptor

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1

■ Explanation

This function writes the PNM file given by *filename* from the coordinate position (x y) from the level and channel indicated by the matrix file's level and channel. The x and y axis directions of drawing must match with the directions specified in the PNM file.

■ Return Value

XLT_NULL [UNDEF REF (XLT_NULL)]

■ Error

- XLE_SEMANTICS_TYPE_MISMATCH [UNDEF REF (XLE_SEMANTICS_TYPE_MISMATCH)]
- XLE_PROTO_OPEN_FILE [UNDEF REF (XLE_PROTO_OPEN_FILE)]
- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

■ References

■ Bugs

2.3.13 gmxScan

■ Prototype

<gmxScan> *start_dim_code* *end_dim_code* <gmxScan>

■ Arguments

start_dim_code [1] List of integers [UNDEF REF (XLT_PAIR)]*dim_code* at the scan start position

end_dim_code [1] List of integers [UNDEF REF (XLT_PAIR)]*dim_code* at the scan end position

■ Attributes

id [mandatory] XLT_INTEGER [UNDEF REF (XLT_INTEGER)]Matrix descriptor

call_dir [arbitrary] XLT_STRING [UNDEF REF (XLT_STRING)]Pre/post scan direction. The default is "pre."

area [mandatory] XLT_INTEGER [UNDEF REF (XLT_INTEGER)]Executed area code

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1

■ Explanation

Synchronized matrix scan function This function executes phases of the area code *area* on all existing nodes having the appropriate dimension codes in the range from *start_dim_code* to *end_dim_code* of the matrix given by *id*.

call_dir indicates the order of calling nodes. The scan is pushed from the highest layer in the matrix to the stack. If pre is specified, all child nodes are evaluated before evaluating this node (child nodes are pushed). If post is specified, child nodes are evaluated after evaluating the node itself is evaluated (depth first scan).

■ Return Value

XLT_INTEGER [UNDEF REF (XLT_INTEGER)]Error codes of matrix generated during execution

■ Error

- XLE_SEMANTICS_TYPE_MISMATCH [UNDEF REF (XLE_SEMANTICS_TYPE_MISMATCH)]
- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

■ References

■ Bugs

In versions ver.B.b14.04 or earlier, there were some problems in synchronization of file data before scanning, and the entire data was not scanned. This problem was solved in ver.B.b15.

2.3.14 gmxCopy

■ Prototype

<gmxCopy> *channel...* <gmxCopy>
<gmxCopy> *channelsList* <gmxCopy>

■ Arguments

channel [1+] Integer [UNDEF REF (XLT_INTEGER)]Channel numbers or character string [UNDEF REF (XLT_STRING)]channel flag

channelsList [1] List of integers [UNDEF REF (XLT_PAIR)]Specify a list of channel numbers or the section of "Channel Flag (Matrix)" in the manual "COSMOS Reference Manual" [3] to be copied

■ Attributes

src [mandatory] XLT_INTEGER [UNDEF REF (XLT_INTEGER)]Copy source matrix descriptor

dest [mandatory] XLT_INTEGER [UNDEF REF (XLT_INTEGER)]Copy destination matrix descriptor

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1

■ Explanation

This function copies the *channel* of all nodes or channels specified by *channelList* from *src* to *dest* . The following channel flags can be specified.

- MF_VISU
- MF_SEND
- MF_FILE
- MF_SEND_VISU
- MF_SEND_FILE

It is allowed to specify multiple channel flags and channel nummbers. It is also allowed to mix them.

■ Return Value

XLT_NULL [UNDEF REF (XLT_INTEGER)]

■ Error

- XLE_SEMANTICS_TYPE_MISMATCH [UNDEF REF (XLE_SEMANTICS_TYPE_MISMATCH)]
- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

■ References

■ Bugs

2.3.15 gmxStatus

■ Prototype

<gmxStatus> *filename* <gmxStatus>

■ Arguments

filename [0-1] XLT_STRING [UNDEF REF (XLT_STRING)] or key etc.

end_dim_code [1] List of integers [UNDEF REF (XLT_PAIR)] Import destination channel numbers

■ Attributes

id [mandatory] XLT_INTEGER [UNDEF REF (XLT_INTEGER)] matrix descriptor

path-type [arbitrary] Method for specifying a file name. absolute/related/network/key default="absolute"

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env1, Env0

■ Explanation

This function returns the status information of the matrix given by *id* or *path-type* and *filename*. If *id* is specified, it is not necessary to specify *filename*.

Moreover, the method of specifying *path-type*="absolute" and *id* must be accessible by Env1.

■ Return Value

List of gmxStatus. Please refer to section 3 for the detailed explanation.

■ Error

● XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

1. "param"
The format of the pointdata attribute is illegal.
2. "attribute::mode"
The value of the mode attribute is not supported.
3. "attribute:path-type"
The value of path-type is not supported.

● XLE_PROTO_INV_OBJECT [UNDEF REF (XLE_PROTO_INV_OBJECT)]

No pixels exist at the coordinate position given by pointdata.

Connection was refused due to authentication failure

● XLE_PROTO_OPEN_FILE [UNDEF REF (XLE_PROTO_OPEN_FILE)]

The file specified by filename could not be opened.

● XLE_SEMANTICS_TYPE_MISMATCH [UNDEF REF (XLE_SEMANTICS_TYPE_MISMATCH)]

The data types of pointdata and filename do not match.

● XLE_PROTO_PERMISSION_DENIED [UNDEF REF (XLE_PROTO_PERMISSION_DENIED)]

It is attempted to execute *path-type*="absolute" in the Env0 environment.

■ References

section 3

■ Bugs

2.3.16 Get

■ Prototype

<Get> *filename dim_code* </Get>

■ Arguments

filename [1] XLT_STRING [UNDEF REF (XLT_STRING)]File path

dim_code [0-1] List of integers [UNDEF REF (XLT_PAIR)]Dimensional code

■ Attributes

mode [arbitrary] Character string; Meta

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env0

■ Explanation

This function is used by the XL server. It acquires matrix file information.

If *mode* is omitted, the function acquires the information from the file itself. It returns a list of node information corresponding to *dim_code*.

If you specify *mode* = "meta," the function returns the status information of the file. This is the same function as *path_type* = "related" in the *gmxStatus* function (Section 2.3.15) . At this point, the format of the return value is given by the *gmxStatus* list (Section 3) .

■ Return Value

In case of *mode* = "meta," a *gmxStatus* list (Section 3) is returned.

If *mode* is omitted, XLT_RAW [UNDEF REF (XLT_RAW)]type node information is returned. The XLT_RAW contains the following information.

1. *nlist_type* 1byte

Format type of a child node list

- 0:Bit field type
- 1:*dim_code* enumeration type

2. Child node: In case of bit field type

(a) *nlist_dim_bit_field_len*

Length of bit field: *compressed_code64* format integer

(b) *nlist_dim_bit_field*

Bit field of length *nlist_dim_bit_field_len*

3. Child node: In case of *dim_code* enumeration type

(a) *nlist_dim_addr_len*

Number of *dim_codes* counted: *compressed_code64* format integer

(b) *dim_codes*

dim_code for the number of *nlist_dim_addr_len*. Each *dim_code* is a row of *compressed_code64* format integers

4. *channel_data*

channel data Each channel data is formatted in the following order, repeated for the number of required channels.

(a) *id*

Channel ID: *compressed_code64* format

(b) *len*

Channel data length: *compressed_code64* format

(c) **data**

The data itself. The format conforms to the format of each data type.

■ **Error**

- XLE_SEMANTICS_TYPE_MISMATCH [UNDEF REF (XLE_SEMANTICS_TYPE_MISMATCH)]
The type of *filename* or *point-data* do not match.
- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

■ **References**

■ **Bugs**

2.3.17 gmxPgPlotField

■ Prototype

- (1) <gmxPgPlotField/>
 (2) <gmxPgPlotField> *field-opt-list* ... </gmxPgPlotField>

■ Arguments

field-opt-list [1] XLT_PTR [UNDEF REF (XLT_PTR)]: Pointer to the work area of matrix field data ... A data group determined by the [1+] merge-type attribute

■ Attributes

- id* (1)(2) [mandatory] Integer: Matrix identifier
csv (1) [not required] (2) [arbitrary] Character string: A,B,C,... Row in a CSVfile
label (1) [not required] (2) [arbitrary] Number: Matrix attribute number
dim (1) [不要] (2) [arbitrary] Number: Matrix dimension number
merge-type (1) [not required] (2) [mandatory] Character string: Attribute detailed type

■ Evaluation Type

applicative

■ Agents Belongs

gbmx

■ Environemnts Belongs

Env0

■ Explanation

This function match rows in a CSV file to a matrix's plot data attribute numbers or dimensions. The correspondence table is stored at an address given by *field-opt-list*. The prototype (1) generates a work area for the matrix field data and returns a pointer to it. The progotype (2) specifies the pointer to the work area *field-opt-list* and creates a correspondence table there.

The prototype (1) requires only the matrix identifier *id* as an attribute value. The prototype (2) requires the following combinations of attribute values. Attribute values indicated as "mandatory" are required, while attribute values indicated as "x" are not required.

Table 2.1: Necessity of attribute values

	id	csv	label	dim	merge-
[1] (1)	mandatory	x	x	x	x
[2] (2) Correspond to dimension numbers	Mandatory	Mandatory	x	Mandatory	Manda
[3] (2) To attribute numbersattribute, constant value	Mandatory	x	Mandatory	x	Manda
[4] (2) To attribute numbers, CSV	Mandatory	Mandatory	Mandatory	x	Manda

Some usage examples are shown below.

[2]

```
<gmxPgPlotField id="~mtx" csv="A" merge-type="~mxPgTYPE_INT" dim="0">
  ^Fields ^mxPgFDIT_DD ^reso (* 180 reso) 0</gmxPgPlotField>
```

[4]

```
<gmxPgPlotField id="~mtx" csv="C" merge-type="~mxPgTYPE_STRING" label="5">
  ^Fields utf-8 ([quote type="direct"] '([field name="LABEL"] ($ ___convert)))</gmxPgPlotField>
```

[3]

```
<gmxPgPlotField id="~mtx" merge-type="~mxPgTYPE_RGBA" label="0">
  ^Fields 0 0xff000000</gmxPgPlotField>
```

[4]

```
<gmxPgPlotField id="~mtx" csv="D" merge-type="~mxPgTYPE_STRING" label="1">
  ^Fields utf-8 () test-http </gmxPgPlotField>
```

As already explained, the instruction [1] generates a work area. The method [2] acquires coordinate values from a CSV file. The method [3] sets a constant value for all the plots. The function [4] processes CSV values and assigns them to attribute numbers of plots.

- **id**
Specify a matrix ID
- **csv**
Specify numbers of rows A, B, C... corresponding to attributes of csv. This XML attribute can be omitted.
- **label**
This XML attribute is used in patterns [3] and [4]; it specifies a matrix's attribute number. Rows in the csv XML attribute are matched with the plot attribute numbers specified by the label XML attribute. If the csv XML attribute is omitted, data value given by the element data of gmXPgPlotField is saved commonly in labelattribute numbers of all plots.
Although the label XML attribute can be omitted, either the label XML or dim XML attribute must exist.
- **dim**
This XML attribute is used in pattern [2] and specifies a matrix's coordinate axis number. As an original function of a matrix, values up to the number of dimensions of a matrix minus 1 can be specified; however, as plot-csv.xl supports only two-dimensional matrices, only 0 or 1 can be specified. 0 refers to the x axis while 1 refers to the y axis.
Although the dim XML attribute can be omitted, either the label XML or dim XML attribute must exist.
- **merge-type** This attribute specifies a matrix's detailed attribute type.

Lastly, the element data of gmXPgPlotField is given differently depending on the data type given by the merge-type XML attribute.

- **mxPgNONE**
None
- **mxPgSTRING**
[Pointer to work area] [saved character code name] [conversion formula] [initialization character string: arbitrary]
If [conversion formula] is () , no special processing is performed on the CSV data, which is set as attribute values. If [conversion formula] is specified as a conversion formula given as an XL formula, CSV values converted by the formula is set as attribute values. The example [4] corresponds to this case. The CSV values are stored in the area of __convert. The actual conversion is executed when a CSV table is created for the first time; a \$ symbol cannot be interpreted at this point and it is thus necessary to use direct quotes.
If the csv attribute is not specified, [initialization character string: arbitrary] becomes necessary. This character string is passed to [conversion formula] and set as the attribute of all plots as a constant value.
- **mxPgINT_ADD mxPgINT_MAX mxPgINT_MIN**
[Pointer to work area] [data type in CSV] [conversion resolution] [conversion offset] [count] [initialization character string: arbitrary]
Data in a CSV file is first interpreted as the data type indicated by [data type in CSV] , which is read as double-precision floating point. The detailed explanation on [data type in CSV] will be given later. The data read is converted by the formula ([read data]x [conversion resolution]+ [conversion offset]). Any fractions after the conversion are omitted, so that the converted output is an

integer, and the result is set as the attribute data or coordinate value of the matrix. [**Conversion resolution**] and [**conversion offset**] must be given as double-precision floating point values.

[**count**] is ignored in these three data types; it is sufficient to specify 0 for the attribute.

If no csv attributes are specified, [**initialization character string: arbitrary**] becomes necessary; this character string is given to [**conversion formula**] and set for all plot attributes as a constant value.

- **mxPgINT_AVG**

[Pointer to work area] [data type in CSV] [conversion resolution] [conversion offset] [count] [initialization character string: arbitrary]

The method of calculating the attribute data is the same as above. **mxPgINT_ADD mxPgINT_MAX mxPgINT_MIN** This data and [**count**] are saved as attribute data, as a pair of integer values. In case of this type, in order to merge plots with higher layers of a matrix, the average of each plot is taken and set as a new plot attribute value. This average is calculated weighted by [**count**] and the [**count**] in the higher layers becomes the total value of [**count**] of lower layers.

If no csv attributes are specified, [**initialization character string: arbitrary**] becomes necessary. This value is passed to the conversion formula above and set as the attribute of all plots as a constant value.

- **mxPgRGBA**

[Pointer to work area] [count] [initialization RGBA: arbitrary]

This function acquires RGBA data from a CSV file and assigns it to plot attributes. If it is given as hexadecimal, the format used in the C language to specify pixel values, for example -16777216, must be used. [**count**] works in the same way as integer type [count] and used for weighting when calculating colors of higher layers. No weighting is performed if 0 is specified.

If no csv attributes are specified, [**initialization RGBA: arbitrary**] is required. This value is passed to the conversion formula above and set for all plot attributes as a constant value.

Integer type [data type in CSV] The following values can be specified for this type.

- **mxPgFDIT_INT**

The CSV data type is integer (64-bit integer).

- **mxPgFDIT_FLOAT**

The CSV data type is double-precision floating point.

- **mxPgFDIT_DD**

The CSV data type is decimal point expression of latitude and longitude, for example means 135degrees 30minutes.

- **mxPgFDIT_DDMSS**

The CSV data type is sexagesimal expression of latitude and longitude, for example means 135degrees 50minutes 12seconds.

- **Return Value**

If *mode* = "meta," a gmxStatus list (Section 3) is returned.

If *mode* is omitted, XLT_RAW [UNDEF REF (XLT_RAW)]-type node information is returned. The XLT_RAW information includes the followings.

1. **nlist_type** 1byte

Format type of child node list

- 0:Bit field type
- 1:dim_code enumeration type

2. **Child node list: In case of bit field type**

- (a) **nlist_dim_bit_field_len**
Bit field length: compressed_code 64format integer
- (b) **nlist_dim_bit_field**
Bit field with a length of nlist_dim_bit_field_len

3. Child node list: In case of dim_code enumeration type

- (a) **nlist_dim_addr_len**
Number of dim_codes counted: compressed_code64 format integer
- (b) **dim_codes**
dim_code for the number of nlist_dim_addr_len. Each dim_code is a row of compressed_code64 format integers

4. channel_data

Channel data Each channel data is formatted in the following order, repeated for the number of required channels.

- (a) **id**
Channel ID: compressed_code 64format
- (b) **len**
The length of the channel data itself, compressed code 64format
- (c) **data**
The data itself. The format conform to the format of each data type.

■ Error

- XLE_SEMANTICS_TYPE_MISMATCH [UNDEF REF (XLE_SEMANTICS_TYPE_MISMATCH)]
The types of *filename* or *point-data* do not match.
- XLE_PROTO_INV_PARAM [UNDEF REF (XLE_PROTO_INV_PARAM)]

■ References

■ Bugs

Chapter 3

gmxStatus List Type Reference

3.1 Abstract

This chapter defines the return value of the gmxStatus (Section 2.3.15) function.

3.2 XML Elements

3.2.1 gmxStatus

■ Prototype

```
<gmxGetPoint> gmxStatusElements .... </gmxGetPoint>
```

■ Arguments

gmxStatus

gmxStatus / gmxMatrixParam (Section 3.2.2)

gmxStatus /gmxMatrixParam/ gmxPriority (Section 3.2.3)

gmxStatus / gmxMatrixDimDivide (Section 3.2.4)

gmxStatus / gmxMatrixBlockSize (Section 3.2.5)

gmxStatus / gmxMatrixPixelSize (Section 3.2.6)

gmxStatus / gmxChannelInfo (Section 3.2.7)

gmxStatus / gmxPhase (Section 3.2.8)

■ Attributes

filenamee [mandatory]: Name of generated file

■ Agents Belongs

gbmx

■ Element Path Expression

gmxStatus

■ Explanation

This function generates a matrix file and expresses the settings of the matrix file to be generated.

■ References

section 2.3.15 , section 2.3.2

■ Bugs

3.2.2 gmxStatus:gmxMatrixParam

■ Prototype

```
<gmxMatrixParam> <gmxPriority/> .... </gmxMatrixParam>
```

■ Arguments

gmxStatus (Section 3.2.1)

gmxStatus/ **gmxMatrixParam**

gmxStatus/ **gmxMatrixParam** / gmxPriority (Section 3.2.3)

gmxStatus/ gmxMatrixDimDivide (Section 3.2.4)

gmxStatus/ gmxMatrixBlockSize (Section 3.2.5)

gmxStatus/ gmxMatrixPixelSize (Section 3.2.6)

gmxStatus/ gmxChannelInfo (Section 3.2.7)

gmxStatus/ gmxPhase (Section 3.2.8)

■ Attributes

channel_nos [mandatory]: Number of channels

]: Dimensions

index_hem [arbitrary]: Enables the on/off HEM function (MPF_INDEX_HEM). The default is "off."

cache_file [arbitrary]: Enables the on/off cache file function (MPF_CACHE_FILE). The default is "off."

■ Agents Belongs

gbmx

■ Element Path Expression

gmxStatus/gmxMatrixParam

■ Explanation

This function sets the basic parameters and the priorities of the matrix to be generated.

■ References

section 2.3.15 , section 2.3.2

■ Bugs

3.2.3 gmxStatus:gmxMatrixParam:gmxPriority

■ Prototype

<gmxPriority/>

■ Arguments

gmxStatus (Section 3.2.1)

gmxStatus/ gmxMatrixParam (Section 3.2.2)

gmxStatus/gmxMatrixParam/ **gmxPriority**

gmxStatus/ gmxMatrixDimDivide (Section 3.2.4)

gmxStatus/ gmxMatrixBlockSize (Section 3.2.5)

gmxStatus/ gmxMatrixPixelSize (Section 3.2.6)

gmxStatus/ gmxChannelInfo (Section 3.2.7)

gmxStatus/ gmxPhase (Section 3.2.8)

■ Attributes

area [mandatory] Integer: Priority setting area

pri [mandatory] Integer: Priority

mode [arbitrary] Character string: active/direct/server/client mode. The default is "active."

■ Agents Belongs

gbmx

■ Element Path Expression

gmxStatus/gmxMatrixParam/gmxPriority

■ Explanation

This function sets the priority of the matrix to be generated.

■ References

section 2.3.15 , section 2.3.2

■ Bugs

3.2.4 gmxStatus:gmxMatrixDimDivide

■ Prototype

<gmxMatrixDimDivide/>

■ Arguments

gmxStatus (Section 3.2.1)

gmxStatus/ gmxMatrixParam (Section 3.2.2)

gmxStatus/gmxMatrixParam/ gmxPriority (Section 3.2.3)

gmxStatus/ **gmxMatrixDimDivide**

gmxStatus/ gmxMatrixBlockSize (Section 3.2.5)

gmxStatus/ gmxMatrixPixelSize (Section 3.2.6)

gmxStatus/ gmxChannelInfo (Section 3.2.7)

gmxStatus/ gmxPhase (Section 3.2.8)

■ Attributes

dim [mandatory] Integer: dim to be set

value [mandatory] Integer: Number of divisions to be set

■ Agents Belongs

gbmx

■ Element Path Expression

gmxStatus/gmxMatrixDimDivide

■ Explanation

This function specifies the number of divisions for each dimension of the matrix to be generated. The actual number of divisions specified is 2^{divide} .

■ References

section 2.3.15 , section 2.3.2

■ Bugs

3.2.5 gmxStatus:gmxMatrixBlockSize

■ Prototype

<gmxMatrixBlockSize/>

■ Arguments

gmxStatus (Section 3.2.1)

gmxStatus/ gmxMatrixParam (Section 3.2.2)

gmxStatus/gmxMatrixParam/ gmxPriority (Section 3.2.3)

gmxStatus/ gmxMatrixDimDivide (Section 3.2.4)

gmxStatus/ **gmxMatrixBlockSize**

gmxStatus/ gmxMatrixPixelSize (Section 3.2.6)

gmxStatus/ gmxChannelInfo (Section 3.2.7)

gmxStatus/ gmxPhase (Section 3.2.8)

■ Attributes

dim [mandatory] Integer: dim to be set

value [mandatory] Integer: Block size to be set

■ Agents Belongs

gbmx

■ Element Path Expression

gmxStatus/gmxMatrixBlockSize

■ Explanation

This function specifies the block size of each dimensional direction of each block of the matrix to be generated. The actual block size is $2^{\text{block_size}}$.

■ References

section 2.3.15 , section 2.3.2

■ Bugs

3.2.6 gmxStatus:gmxMatrixPixelSize

■ Prototype

<gmxMatrixPixelSize/>

■ Arguments

gmxStatus (Section 3.2.1)

gmxStatus/ gmxMatrixParam (Section 3.2.2)

gmxStatus/gmxMatrixParam/ gmxPriority (Section 3.2.3)

gmxStatus/ gmxMatrixDimDivide (Section 3.2.4)

gmxStatus/ gmxMatrixBlockSize (Section 3.2.5)

gmxStatus/ **gmxMatrixPixelSize**

gmxStatus/ gmxChannelInfo (Section 3.2.7)

gmxStatus/ gmxPhase (Section 3.2.8)

■ Attributes

dim [mandatory] Integer: dim to be set

value [mandatory] Integer: Block size to be set

■ Agents Belongs

gbmx

■ Element Path Expression

gmxStatus/gmxMatrixPixelSize

■ Explanation

This function specifies the pixel size of each dimensional direction of the entire matrix to be generated.

The actual pixel size is $2^{\text{pixel_size}}$.

■ References

section 2.3.15 , section 2.3.2

■ Bugs

3.2.7 gmxStatus:gmxChannelInfo

■ Prototype

<gmxChannelInfo/> <gmxChannelInfo> *initialize_data* <gmxChannelInfo>

■ Arguments

gmxStatus (Section 3.2.1)

gmxStatus/ gmxMatrixParam (Section 3.2.2)

gmxStatus/gmxMatrixParam/ gmxPriority (Section 3.2.3)

gmxStatus/ gmxMatrixDimDivide (Section 3.2.4)

gmxStatus/ gmxMatrixBlockSize (Section 3.2.5)

gmxStatus/ gmxMatrixPixelSize (Section 3.2.6)

gmxStatus/ **gmxChannelInfo**

gmxStatus/ gmxPhase (Section 3.2.8)

■ Attributes

channel [mandatory] Integer: Target channel number

type [mandatory] Character string: Channel data type

send [arbitrary] Character string: Turns on the on/off MF_SEND flag. The default is "off."

file [arbitrary] Character string: Turns on the MF_SFILE flag. The default is "off."

visu [arbitrary] Character string: Turns on the on/off MF_VISU flag. The default is "off."

send_file [arbitrary] Character string: Turns on the on/off MF_SEND_FILE flag. The default is "off."

send_visu [arbitrary] Character string: Turns on the on/off MF_SEND_VISU flag. The default is "off."

■ Agents Belongs

gbmx

■ Element Path Expression

gmxStatus/gmxChannelInfo

■ Explanation

This function specifies information of each channel of the matrix to be generated. The attributes determines the data type and flags. The list of data types that can be set are as follows.

- Basis data type
 - bit - Bit type
 - int8 - 8-bit signed integer type
 - int16 - 16-bit signed integer type
 - int32 - 32-bit signed integer type
 - int64 - 64-bit signed integer type
 - uint8 - 8-bit unsigned integer type
 - uint16 - 16-bit unsigned integer type
 - uint32 - 32-bit unsigned integer type
 - uint64 - 64-bit unsigned integer type
 - float - floating point type
 - double - double-precision floating point type
 - string - character string type
 - block - block type
 - sexp - XL S formula type
 - rgb8 - 8-bit RGB type
- Vector type
 - int8.v - 8-bit signed integer type
 - int16.v - 16-bit signed integer type
 - int32.v - 32-bit signed integer type

- `int64_v` - 64-bit signed integer type
- `uint8_v` - 8-bit unsigned integer type
- `uint16_v` - 16-bit unsigned integer type
- `uint32_v` - 32-bit unsigned integer type
- `uint64_v` - 64-bit unsigned integer type
- `float_v` - floating point type
- `double_v` - double-precision floating point type
- `string_v` - character string type
- `rgb8_v` - 8-bit RGB type

`gmxChannelInfo` can specify initialization data corresponding to the data types above as internal elements. Integer and floating point type data are converted to the given type above and initialized. If no internal elements are given, it means initialization data is not set.

If `gmxStatus` (Section 2.3.15) is evaluated in the `Env1` environment variable, the `send`, `file`, `visu`, `send_file` and `send_visu` attributes are all acquired as specified. If it is evaluated in the `Env0` environment variable, `send_file` and `send_visu` are each set to the values of `file` and `visu` and the original `file` and `visu` attributes are hidden. The `send` attribute is visible in both cases.

■ **References**

section 2.3.15 , section 2.3.2

■ **Bugs**

3.2.8 gmxStatus:gmxPhase

■ Prototype

<gmxPhase> *Phase_commands* <gmxPhase> <gmxPhase/>

■ Arguments

gmxStatus (Section 3.2.1)

gmxStatus/ gmxMatrixParam (Section 3.2.2)

gmxStatus/gmxMatrixParam/ gmxPriority (Section 3.2.3)

gmxStatus/ gmxMatrixDimDivide (Section 3.2.4)

gmxStatus/ gmxMatrixBlockSize (Section 3.2.5)

gmxStatus/ gmxMatrixPixelSize (Section 3.2.6)

gmxStatus/ gmxChannelInfo (Section 3.2.7)

gmxStatus/ **gmxPhase**

■ Attributes

area [mandatory] Integer: Phase area to be set

equ [arbitrary] Integer: Target phase area of equivalent

mode [arbitrary] Character string: active/direct/server/client mode; the default is "active."

■ Agents Belongs

gbmx

■ Element Path Expression

gmxStatus/gmxPhase

■ Explanation

This function specifies the phase script of the matrix to be generated.

It indicates the phase area script stored in the area of the mode indicated by mode.

If gmxStatus (Section 2.3.15) is evaluated in the Env0 environment variable, this gmxPhase is ignored.

■ References

section 2.3.15 , section 2.3.2

■ Bugs

Bibliography

- [1] Mori Hirohisa. *COSMOS Startup Manual*. GLOBALBASE PROJECT, 2006.
- [2] Mori Hirohisa and Fujita Haruhiro. *LANDSCAPE Startup Manual*. GLOBALBASE PROJECT, 2006.
- [3] Mori Hirohisa. *COSMOS Reference Manual*. GLOBALBASE PROJECT, 2006.

History

1. **Date:** 2010-12-01
Generatting this manual. (Edition 2010-12-01)
2. **Date:** 2006-08-03
Author: Hiroshisa Mori **TargetVersion:** ver.B.b13
Created this manual.
3. **Date:** 2006-10-15
Author: Hiroshisa Mori **TargetVersion:** ver.B.b13
Changed the specification of gmxGetPoint.
4. **Date:** 2007-04-13
Author: Hiroshisa Mori **TargetVersion:** ver.B.b15
Added gmxCopy. Added explanation about bugs of gmxScan.
5. **Date:** 2009-01-30
Author: Reiko Inoue Bendtsen **TargetVersion:** ver.B.b17.12
Transrated this manual into English