MapInfo Native Format Reader/Writer

The MapInfo Native Format Reader and Writer modules provide the Feature Manipulation Engine (FME) with the ability to read and write directly to MapInfo files. The MapInfo Native Format is a proprietary format used by the MapInfo Professional Desktop mapping product. MapInfo native format files are often called Tab files.

The MapInfo Native Format reader and writer are closely patterned after the MapInfo MIF/MID reader and writer. This commonality makes it easy to support both MIF and MapInfo native formats in the same mapping file.

Overview

MapInfo is a two-dimensional system with no provision for transferring elevation data for each vertex in a MapInfo feature. However, point features can define an elevation attribute to store their elevation.

MapInfo files store both feature geometry and attributes. A logical MapInfo file consists of several physical files, having the following file name extensions:

File Name Extension	Contents
`.tab`	The main file for a MapInfo table, it is associated with the appropriate DAT, MAP, ID, and IND files.
`.dat`	Tabular data for a table in MapInfo’s native format.
`.id`	An index to a MapInfo graphical objects (MAP) file.
`.map`	Contains geographic information describing map objects.
`.ind`	An index to a MapInfo tabular (DAT) file.

These extensions are added to the basename of the specified MapInfo file. Throughout the remainder of this chapter, references to “file” are references to the logical MapInfo file, not the multiple physical files that make it up.

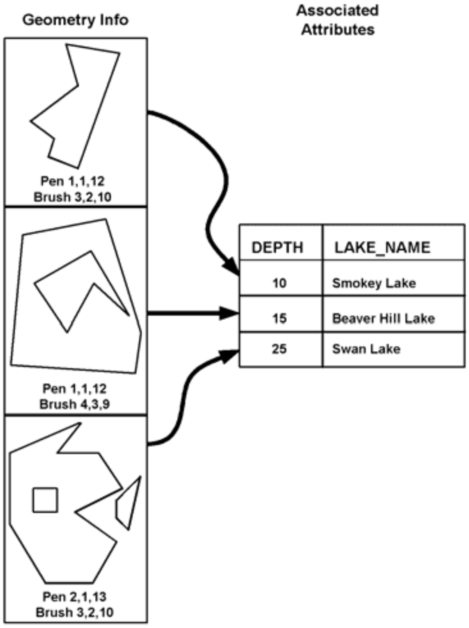
The MapInfo reader and writer support the storage of point, line, polyline, arc, ellipse, rectangle, rounded rectangle, region (polygon), and text geometric data. The MapInfo format also stores features with no geometry. Features having no geometry are referred to as having a geometry of *none*.

Each geometric entity present in MapInfo has display properties, such as pen and brush width, pattern, and color. In addition, each entity has a row of attributes associated with it. A single MapInfo map file can contain many different types of geometry however, the associated attributes must have the same number and type of fields for each entity in the file.

The number and type of attributes associated with each entity is specified by the user. There must be at least one attribute field defined before a MapInfo file can be created.

The following illustration shows a MapInfo file containing three *region* entities. Note that the second polygon contains a hole while the third polygon is an aggregate of two disjoint polygons, one of which contains a hole. Each geometric entity in turn corresponds with one record in the attribute table.

The FME considers a MapInfo dataset to be a collection of tab files and related files in a single directory. The attribute definitions for each MapInfo file set must be defined in the mapping file before it can be read or written.



MapInfo Native Format Quick Facts

Format Type Identifier	MAPINFO
Reader/Writer	Both
Dataset Type	Directory or File
Feature Type	File base name
Typical File Extensions	`.tab (.dat, .id, .map, .ind)`
Automated Translation Support	Yes
User-Defined Attributes	Yes

Coordinate System Support	Yes
Generic Color Support	Yes
Spatial Index	Always
Schema Required	Yes
Transaction Support	No
Geometry Type Attribute	`mapinfo_type`
Geometry Support	

Geometry	Supported	Geometry	Supported
aggregate	yes	polygon	yes
circles	yes	donut polygon	yes
circular arc	yes	line	yes
elliptical arc	yes	point	yes
ellipses	yes	text	yes
none	yes	3D	no

Reader Overview

The MapInfo reader first scans the directory it is given for the MapInfo files which have been defined in the mapping file. For each logical MapInfo file that it finds, it checks to see if it that file is requested by looking at the list of `IDs` specified in the mapping file. If a match is made, or no `IDs` were specified in the mapping file, the MapInfo file is opened. The MapInfo reader then extracts features from the file one at a time, and passes them on to the rest of the FME for further processing. When the file is exhausted, the MapInfo reader starts on the next file in the directory.

Optionally, a single MapInfo file can be given as the dataset. In this case, only that MapInfo file is read.

Reader Keywords

The following table lists the keywords processed by the MapInfo reader. The keywords shown will be prefixed by the current `<ReaderKeyword>_` in a mapping file. By default, the `<ReaderKeyword>` for the MapInfo reader is `MAPINFO`.

Keyword Suffix	Value	Required/Optional
`DATASET`	Contains the directory name of the input MapInfo files, or a single MapInfo file.	Required
`DEF`	Defines a MapInfo file. The definition contains the file’s base name (without any of the extensions), and the definitions of the attributes. There may be many `DEF` lines, one for each file to be read.	Optional
`IDs`	Contains a list of MapInfo files to process. If this is not specified, then all defined MapInfo files in the directory will be read.	Optional

Keyword Suffix	Value	Required/Optional
BREAK_ COLLECTION	Specifies how MapInfo collection features are handled. If YES, collection features are broken down into their component parts before being returned to FME. If NO, then the collections are preserved (usually this is only of use when doing a MapInfo to MapInfo translation). The default is YES.	Optional
SEARCH_ ENVELOPE	Specifies the spatial extent of the feature retrieval. If this is not supplied, all features will be returned. NOTE: this option is only available with the MITAB based MapInfo reader.	Optional

DATASET

The value for this keyword is the directory containing the MapInfo files to be read, or a single MapInfo file. A typical mapping file fragment specifying an input MapInfo dataset looks like:

```
MAPINFO_DATASET /usr/data/mapinfo/92i080
```

DEF

The definition specifies the base name of the file, and the names and types of all attributes. The syntax of a MapInfo DEF line is:

```
<ReaderKeyword>_DEF <baseName> [<attrName>
<attrType>[,indexed]]+
```

The file names of the physical MapInfo files are constructed by using the directory specified by the DATASET keyword, the basename specified on the MapInfo DEF lines, and the file extensions.

MapInfo requires that at least one attribute be defined. The attribute definition given must match the definition of the file being read. If it does not, translation is halted and the true definition of the MapInfo file attributes are logged to the log file. There are no restrictions on the field names of MapInfo attributes. The following table shows the attribute types which are supported.

Field Type	Description
char(<width>)	Character fields store fixed length strings. The width parameter controls the maximum of characters stored by the field. No padding is required for strings shorter than this width.
date	Date fields store dates as character strings with the format dependent on your location. This format is usually YYYYMMDD.

Field Type	Description
decimal(<width>, <decimals>)	Decimal fields store single and double precision floating point values. The width parameter is the total number of characters allocated to the field, including the decimal point. The decimals parameter controls the precision of the data and is the number of digits to the right of the decimal.
float	Float fields store floating point values. There is no ability to specify the precision and width of the field.
integer	Integer fields store 32 bit signed integers.
logical	Logical fields store boolean data. Data read or written from/to such fields must always have a value of either true or false.
smallint	Small integer fields store 16 bit signed integers, and therefore have a range of -32767 to +32767.

The attribute type may also have ,indexed when the definition is specified for a writer. When specified, this results in the writer building an attribute index table for the columns that are indexed thereby making queries in MapInfo faster. This directive is only recognized by the writer module.

The following mapping file fragment defines two MapInfo files. Notice that neither definition specifies the geometric type of the entities it will contain because MapInfo files may contain any of the valid geometry types.

```
MAPINFO_DEF landcover                                      \
  area              decimal(12,3)                          \
  landcoverType     char(11)                               \
  perimeter         float

MAPINFO_DEF roads                                          \
  numberOfLanes     smallint                               \
  roadType          char(5)                                \
  underConstruction logical                                \
  divided           logical                                \
  travelDirection   char(6)
```

IDs

This optional specification is used to limit the MapInfo files that are read. If no IDs are specified, then all defined and available MapInfo files are read. The syntax of the IDs keyword is:

```
<ReaderKeyword>_IDs <baseName1>                            \
                    <baseName2>                            \
                    <baseNameN>
```

The basenames must match those used in DEF lines.

The example below selects only the `roads` MapInfo file for input during a translation:

```
MAPINFO_IDs roads
```

BREAK_COLLECTION

This directive specifies how the MapInfo collections are processed. If no `BREAK_COLLECTION` is specified, then all MapInfo collections are broken down into their component parts before being returned to FME. If a MapInfo-to-MapInfo translation is being performed, then this may be set to `NO` to preserve the collections as single features.

This example shows how collections may be preserved:

```
MAPINFO_BREAK_COLLECTION NO
```

SEARCH_ENVELOPE

This keyword specifies the spatial extent of the feature retrieval. Only features that intersect this bounding box are returned by the reader. If this directive is not specified, then all features are returned.

Note that this directive is only honoured by the MITAB-based MapInfo reader in FME. This is the only MapInfo reader available on the UNIX platforms supported by FME, and can optionally be enabled on Windows platforms by renaming the `mitab.dll` in the FME home directory to `mapinfo.dll`.

The syntax of the `MAPINFO_SEARCH_ENVELOPE` directive is:

```
MAPINFO_SEARCH_ENVELOPE <minX> <minY> <maxX> <maxY>
```

The coordinate values specified are measured in the ground units of the input data.

The example below selects a small area in a lat/long dataset for extraction:

```
MAPINFO_SEARCH_ENVELOPE -130 49 -128 50.1
```

Writer Overview

The MapInfo writer creates and writes feature data to MapInfo files in the directory specified by the `DATASET` keyword. If the directory does not exist, the writer has to create it. Any old MapInfo files in the directory are overwritten with the new feature data. As features are routed to the MapInfo writer, the MapInfo writer determines the file into which the features are to be written and outputs them accordingly. Many MapInfo files can be written during a single FME session.

When the MapInfo writer receives a feature with an fme_color or fme_fill_color attribute, the writer will honor the color values. The only exception is when native MapInfo color settings are also present, in which case the native settings will take precedence.

Writer Keywords

The MapInfo writer processes the DATASET and DEF keywords as described in the *Reader Keywords* section. It does, however, make use of some additional keywords:

Keyword Suffix	Value	Required/ Optional
DATASET	Contains the directory name of the output MapInfo files.	Required
DEF	Defines a MapInfo file. The definition contains the file's base name (without any of the extensions), and the definitions of the attributes. There may be many DEF lines, one for each file to be written.	Required
COORDSYS_STATEMENT	Contains a MapInfo specific coordinate system string to be used for the output files.	Optional
BOUNDS	Specifies the bounds which should be used to limit the range of the output MapInfo files. Because MapInfo has limited precision available for the storage of coordinates, defining a tight bound on the data can preserve more accuracy. The syntax of this directive is: MAPINFO_BOUNDS <xmin> <ymin> <xmax> <ymax>	Optional
USE_SOURCE_BOUNDING_BOX	Indicates if the MapInfo writer should use any available bounding box information that the source reader can provide. If YES, then this bounding box information is looked for and used. If NO, then the writer will extract its own bounding box information, if it needs it. The default is NO.	Optional
FILENAME_PREFIX	The value given this directive is prepended to every file output by the writer.	Optional
WRITE_REGION_ CENTROIDS	Directs the writer to output region centroids. **Values:** yes or no. **Default:** no	Optional
STROKE_ARCS	Indicates whether or not arcs will be vectorized before writing. **Values:** yes or no. **Default:** no	Optional

COORDSYS_STATEMENT

The value for this keyword is the coordinate system statement that should be used in the produced MapInfo files. Normally, FME examines the coordinate system information present on the features written to the files, and outputs the coordinate system based on this information. However, in certain circumstances it is necessary to override this and force a particular coordinate system to be output. This is typically done to force the units of a *non-earth* projection to something other than the default, which is metres.

The syntax of this line is the same as that defined for the `CoordSys` line in the MapInfo MIF/MID documentation. For example, to force a non-earth inches coordinate system, this line would be present in the mapping file:

```
MAPINFO_COORDSYS_STATEMENT CoordSys NonEarth Units \"in\"
```

Notice that the quotes must be escaped, as they are required when the coordinate system statement is interpreted by the MapInfo Writer.

BOUNDS

This directive allows explicit setting of the bounds of the output features. Because MapInfo has limited precision available for the storage of coordinates, defining a tight bound on the range of the data can preserve more accuracy. The syntax of this directive is:

```
MAPINFO_BOUNDS<xmin> <ymin> <xmax> <ymax>
```

USE_SOURCE_BOUNDING_BOX

The `USE_SOURCE_BOUNDING_BOX` directive tells the MapInfo writer to attempt to use any bounding box information that the Reader for the current FME session can provide it to set its bounds. This will only be used when no coordinate system is set for the MapInfo writer. Currently, only the Shape reader in FME provides bounding box information, so setting this directive to `YES` will only have an effect if a Shape to MapInfo translation is being performed without any coordinate system being set.

Note This is a writer directive and applies only to the MapInfo writer. Its use is discouraged.

FILENAME_PREFIX

The value for this keyword is prepended to every output file that is created by the writer.

For example, to have the word `temp` appear on the front of every file name, this line would be present in the mapping file:

```
MAPINFO_FILENAME_PREFIX temp
```

WRITE_REGION_CENTROIDS

To direct the Writer to output region centroids, the syntax of this directive is:

```
WRITE_REGION_CENTROIDS yes
```

STROKE_ARCS

Indicates whether the arcs will be vectorized before writing. If yes, all arcs will be converted to polylines. This option may be useful where sweep angles have precision finer than 0.1 degree.

```
STROKE_ARCS yes
```

Feature Representation

MapInfo features consist of geometry and attributes. The attribute names are defined in the DEF line and there is a value for each attribute in each FME MapInfo feature.

In addition, each MapInfo FME feature contains several special attributes to hold the type of the geometric entity and its display parameters. All MapInfo FME features contain the mapinfo_type attribute, which identifies the geometric type. All MapInfo features may contain either or both of the fme_color and fme_fill_color attributes, which store the color and fill color of the feature respectively.

In addition to the generic FME feature attributes that FME Workbench adds to all features (see *About Feature Attributes* on page 1), this format adds the format-specific attributes described in this section.

Attribute Name	Contents	
mapinfo_type	The MapInfo geometric type of this entity.	
	Range:	mapinfo_point| mapinfo_polyline| mapinfo_region| mapinfo_text| mapinfo_ellipse| mapinfo_arc| mapinfo_rectangle| mapinfo_rounded_rectangle| mapinfo_collection| mapinfo_none
	Default:	No default

Attribute Name	Contents
fme_color	A normalized RGB triplet representing the color of the feature, with format r,g,b. **Range:** 0,0,0 to 1,1,1 **Default:** No default
fme_fill_color	A normalized RGB triplet representing the fill color of the feature, with format r,g,b. **Range:** 0,0,0 to 1,1,1 **Default:** No default

Points

mapinfo_type: mapinfo_point

MapInfo point features specify a single x and y coordinate in addition to any associated user-defined attributes. An aggregate of point features may also be read or written – this corresponds to the MapInfo MULTI_POINT primitive type.

A MapInfo point also specifies a symbol. The symbol is defined by a symbol number, a color, and a size.[1] If no symbol is defined for a point entity, the previous symbol is used. The table below lists the special FME attribute names used to control the MapInfo symbol settings.

Attribute Name	Contents
mapinfo_symbol_color	The color of the symbol. MapInfo colors are defined in relative concentrations of red, green, and blue. Each color ranges from 0 to 255, and the color value is calculated according to the formula: (red*65536) + (green*256) + blue **Range:** 0…2^24 - 1 **Default:** 0 (black)
mapinfo_symbol_shape	The number of the symbol. See the *MapInfo Reference Manual* for a list of the available symbols. **Range:** 31...67 **Default:** 35 (a star)
mapinfo_symbol_size	The point size of the symbol. Note that this size is *not* scaled depending on the zoom level. **Range:** Any integer number > 0 **Default:** 10

[1] MapInfo symbols cannot be rotated. However, some third-party add-ons to MapInfo will rotate symbols based on a user-defined rotation attribute.

Font Points

mapinfo_type: mapinfo_font_point

MapInfo font points are very similar to MapInfo points, but allow a symbol based on a TrueType font to be specified. In addition to the font, a rotation, color, shape number, size, and style may be specified.

The table below lists the special FME attribute names used to control the MapInfo font point settings:

Attribute Name	Contents
mapinfo_symbol_color	The color of the symbol calculated according to the formula: (red*65536) + (green*256) + blue **Range:** 0…2^24 - 1 **Default:** 0 (black)
mapinfo_symbol_shape	The number of the shape within the TrueType font to use as the symbol. **Range:** Integer **Default:** No default
mapinfo_symbol_size	The point size of the symbol. **Range:** Integer **Default:** 12
mapinfo_symbol_font	The name of the TrueType font to be used for the symbol. **Range:** String **Default:** No default
mapinfo_symbol_angle	The rotation angle for the symbol, measured in degrees counterclockwise from horizontal. **Range:** -360.0..360.0 **Default:** 0
mapinfo_symbol_style	The display style for the symbol. **Range:** 0 (Plain text) 1 (Bold text) 16 (Black border around symbol) 32 (Drop Shadow) 256 (White border around symbol) **Default:** 0

Custom Points

mapinfo_type: mapinfo_custom_point

MapInfo custom points are very similar to MapInfo points, but allow a bitmap image to be specified as the symbol to be drawn. In addition to the image, color, size, and style may be specified.

The table below lists the special FME attribute names used to control the MapInfo custom point settings:

Attribute Name	Contents
mapinfo_symbol_color	The color of the symbol calculated according to the formula: (red*65536) + (green*256) + blue Whether or not the color is used, depends on the setting of the style attribute. **Range:** 0…2^24 - 1 **Default:** 0 (black)
mapinfo_symbol_file_name	The name of the bitmap file found in the MapInfo CustSymb directory. **Range:** String **Default:** No default
mapinfo_symbol_size	The point size of the symbol. **Range:** Integer **Default:** 12
mapinfo_symbol_style	The display style for the symbol. **Range:** 0 (White pixels in the image are transparent, allowing whatever is beneath to show through. Non-white pixels are drawn in the same color as they are in the bitmap.) 1 (White pixels in the image are drawn as white. Non-white pixels are drawn in the same color as they are in the bitmap.) 2 (White pixels in the image are transparent. Non-white pixels are drawn in the color specified by mapinfo_symbol_color.) 3 (White pixels in the image are drawn in white. Non-white pixels are drawn in the color specified by mapinfo_symbol_color) **Default:** 0

Multipoints

mapinfo_type: mapinfo_point, mapinfo_font_point, mapinfo_custom_point

MapInfo multipoint is supported as an homogeneous aggregate feature composed of points, font points or custom points.

The MapInfo multipoint uses the same attribute names control settings as the points, font points and custom point.

Polylines

mapinfo_type: mapinfo_polyline

MapInfo polyline features specify linear features defined by a sequence of x and y coordinates. Each polyline has a pen style associated with it that specifies the color, width, and pen pattern of the line. A polyline may also specify that it is a smoothed line[2], in which case MapInfo uses a curve fitting algorithm when rendering the line. If no pen style is defined, the previous style is used.

Tip

MapInfo supports a special type for two point lines. The FME transparently converts such *lines* into polylines, both as it reads and as it writes them.

The table below lists the special FME attribute names used to control the MapInfo polyline settings.

Attribute Name	Contents
`mapinfo_pen_color`	The color of the polyline. MapInfo colors are defined in relative concentrations of red, green, and blue. Each color ranges from 0 to 255, and the color value is calculated according to the formula: (red*65536) + (green*256) + blue **Range:** 0…2^24 - 1 **Default:** 0 (black)
`mapinfo_pen_pattern`	The pattern used to draw the line. See the *MapInfo Reference Manual* for a list of the available patterns. **Range:** 1…77 **Default:** 2
`mapinfo_pen_width`	The width of the line rendered for the polyline feature. This is measured as a thickness in pixels. A width of 1 is always drawn as a hairline. A width of 0 should be considered to be a line with no width, or a line with no style, or invisible, and should not normally be used. If an invisible line is necessary, it should be created by setting the pattern to 1 (None). If a hairline is desired, the pen should be created by setting the width to 1. The width can be specified as a point width, in which case this formula is used: `penwidth = (point width * 10) + 10` **Range:** 0...7 (pixel width) 11...2047 (point width) **Default:** 1
`mapinfo_smooth`	Controls whether or not the polyline will be smoothed when rendered. **Range:** true|false **Default:** false

[2] MapInfo renders smoothed polylines substantially slower than unsmoothed polylines.

Regions

mapinfo_type: mapinfo_region

MapInfo region features specify area (polygonal) features. The areas that make up a single feature may or may not be disjoint, and may contain polygons which have holes. Each region has a pen style associated with it to control the color, width, and pen pattern used when its boundary is drawn. In addition, a region may set its brush pattern, foreground, and background color to control how the area it encloses will be filled.

The following table lists the special FME attribute names used to control the MapInfo region settings.

Attribute Name	Contents
mapinfo_brush_pattern	The pattern used to fill the area the region contains. See the *MapInfo Reference Manual* for a list of the available brush patterns. **Range:** 1…71 **Default:** 2 (solid)
mapinfo_brush_foreground	The foreground color used when the region is filled. MapInfo colors are defined in relative concentrations of red, green, and blue. Each color ranges from 0 to 255, and the color value is calculated according to the formula: (red*65536) + (green*256) + blue **Range:** 0…2^24 - 1 **Default:** 0 (black)
mapinfo_brush_background	The background color used when the region is filled. (-1 specifies transparent color) **Range:** -1…2^24 - 1 **Default:** 16777215 (white)
mapinfo_brush_transparent	Controls whether or not the brush's background is transparent. **Range:** true|false **Default:** true if no brush background was specified or if set to -1; false otherwise
mapinfo_pen_color	The color of the boundary of the region. **Range:** 0…2^24 - 1 **Default:** 0 (black)
mapinfo_pen_pattern	The pattern used to draw the region's boundary. See the *MapInfo Reference Manual* for a list of the available patterns. **Range:** 1…77 **Default:** 2

Attribute Name	Contents
mapinfo_pen_width	The width of the line rendered for the region's boundary. This is measured as a thickness in pixels. A width of 1 is always drawn as a hairline. A width of 0 should be considered to be a line with no width, or a line with no style, or invisible, and should not normally be used. If an *invisible* line is necessary, it should be created by setting the pattern to 1 (None). If a hairline is desired, the pen should be created by setting the width to 1. **Range:** 0...35 **Default:** 1
mapinfo_centroid_x	The centroid x coordinate. **Range:** Any real number **Default:** 0
mapinfo_centroid_y	The centroid y coordinate. **Range:** Any real number **Default:** 0

Text

mapinfo_type: mapinfo_text

MapInfo text features are used to specify annotation information. Each text feature can have its font, color, spacing, justification, and rotation angle set independently. The following table lists the special FME attribute names used to control the MapInfo text settings.

Attribute Name	Contents
mapinfo_rotation	The rotation of the text, as measured in degrees counterclockwise from horizontal. **Range**: -360.0..360.0 **Default:** 0
mapinfo_text_fontbgcolor	The background color used when the text is drawn. **Range:** 0…2^24 - 1 **Default:** 16777215 (white)

Attribute Name	Contents
`mapinfo_text_fontfgcolor`	The foreground color used when the text is drawn. MapInfo colors are defined in relative concentrations of red, green, and blue. Each color ranges from 0 to 255, and the color value is calculated according to the formula: (red*65536) + (green*256) + blue **Range:** 0…2^24 - 1 **Default:** 0 (black)
`mapinfo_text_fontname`	The name of the font used to draw the text. The font named must be available on the destination computer system. **Range:** Any valid system font **Default:** Helve
`mapinfo_text_height`	The height of the text in ground units. **Range:** Any real number >= 0 **Default:** 10
`mapinfo_text_justification`	The justification of the text. **Range:** left | center | right **Default:** left
`mapinfo_text_spacing`	The spacing between lines of multiline text. The measure is expressed as a multiple of the text height. **Range:** 1.0 | 1.5 | 2.0 **Default:** 1.0
`mapinfo_text_linetype`	The type of line attaching the text to the anchor point. **Range:** 0 (None: do not display a line with the label.) 1 (Simple: create a callout by using a simple line that connects the label to the anchor point.) 2 (Arrow: create a callout by using an arrow and line that connects the label to anchor point.) **Default:** 0 (None)
`mapinfo_text_line_end_x`	The x position of the label line end point. The linetype needs to be 1 or 2 for the label line to be visible. **Range:** Any real number **Default:** No default

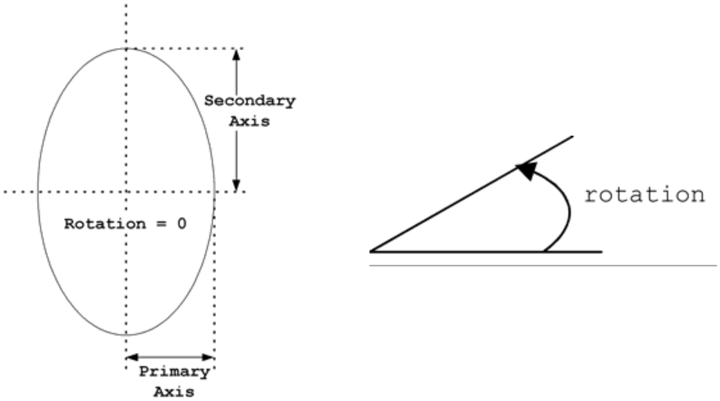
Attribute Name	Contents
`mapinfo_text_line_end_y`	The y position of the label line end point. The linetype needs to be 1 or 2 for the label line to be visible. **Range:** Any real number **Default:** No default
`mapinfo_text_fontstyle_bold`	Indicates if the text is bold or not. **Range:** true | false **Default:** false
`mapinfo_text_fontstyle_italic`	Indicates if the text is in Italics **Range:** true | false **Default:** false
`mapinfo_text_fontstyle_underline`	Indicates if the text is underlined. **Range:** true | false **Default:** false
`mapinfo_text_fontstyle_strikeout`	Indicates if the text has a line through the middle of it. **Range:** true | false **Default:** false
`mapinfo_text_fontstyle_outline`	Indicates if the text is outlined **Range:** true | false **Default:** false
`mapinfo_text_fontstyle_shadow`	Indicates if the text has a shadow. **Range:** true | false **Default:** false
`mapinfo_text_fontstyle_inverse`	Indicates if the text is shown in inverse. **Range:** true | false **Default:** false
`mapinfo_text_fontstyle_blink`	Indicates if the text is blinking. **Range:** true | false **Default:** false
`mapinfo_text_fontstyle_opaque`	Indicates if the text is opaque. **Range:** true | false **Default:** false
`mapinfo_text_fontstyle_halo`	Indicates if the text has a halo. **Range:** true | false **Default:** false

Attribute Name	Contents
mapinfo_text_fontstyle_allcaps	Indicates if the text is uppercase. **Range:** true | false **Default:** false
mapinfo_text_fontstyle_expanded	Indicates if the text is expanded. **Range:** true | false **Default:** false
mapinfo_text_string	The text to be displayed. **Range:** Any character string **Default:** No default
mapinfo_text_width	The width of the entire text string, in ground units. **Range:** Any real number >= 0 **Default:** 10

Ellipse

mapinfo_type: mapinfo_ellipse

MapInfo ellipse features are point features, and only have a single coordinate. This point serves as the centre of the ellipse. Additional attributes specify the primary axis and the secondary axis of the ellipse. MapInfo ellipses currently do not support rotation. For compatibility with other systems, the MapInfo reader always returns a rotation of 0. If a rotation is specified to the writer, the ellipse is turned into a region, vectorized, and rotated by the amount specified.



Tip

The primary ellipse axis is **not** necessarily the longest axis, but rather the one on the x axis.

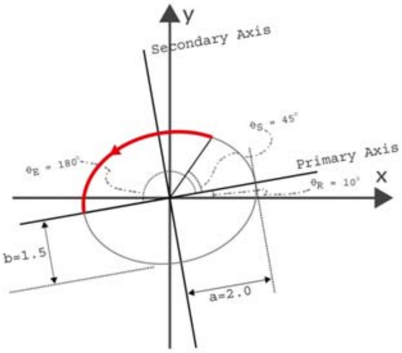
In addition to the attributes below, ellipses also make use of the brush and pen attributes as defined by `mapinfo_region`.

Attribute Name	Contents
`mapinfo_primary_axis`	The length of the semi-major axis in ground units. **Range:** Any real number > 0 **Default:** No default
`mapinfo_secondary_axis`	The length of the semi-minor axis in ground units. **Range:** Any real number > 0 **Default:** No default
`mapinfo_rotation`	The rotation of the major axis. The rotation is measured in degrees counterclockwise up from horizontal. **Range:** -360.0..360.0 **Default:** 0

Arc

`mapinfo_type:` `mapinfo_arc`

MapInfo arc features are linear features used to specify elliptical arcs. As such, the feature definition for `mapinfo_arc` is similar to the ellipse definition with two additional angles to control the portion of the ellipse boundary drawn. MapInfo arcs currently do not support rotation. For compatibility with other systems, the MapInfo reader always returns a rotation of 0. In addition, if a rotation is specified to the writer, the arc is turned into a polyline, vectorized, and rotated by the amount specified.



Tip

The function `@Arc()` can be used to convert an arc to a linestring. This is useful for storing Arcs in systems that do not support them directly.

In addition the attributes below, arcs also make use of the pen attributes as defined on `mapinfo_polyline`.

Attribute Name	Contents
`mapinfo_primary_axis`	The length of the semi-major axis in ground units. **Range:** Any real number > 0 **Default:** No default
`mapinfo_secondary_axis`	The length of the semi-minor axis in ground units. **Range:** Any real number > 0 **Default:** No default
`mapinfo_start_angle`	The start angle defines the counterclockwise distance from the primary axis to the starting point of the arc. It is measured in degrees. **Range:** 0.0..360.0 **Default:** 0
`mapinfo_sweep_angle`	The sweep angle defines the sweep of the arc from the starting point along the ellipse boundary in the counterclockwise direction. It is measured in degrees. **Range:** 0.0..360.0 **Default:** No default
`mapinfo_rotation`	The rotation of the major axis. The rotation is measured in degrees counter clockwise up from horizontal. **Range:** -360.0..360.0 **Default:** 0

Rectangle

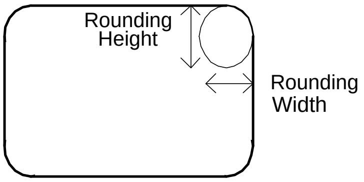
`mapinfo_type:` `mapinfo_rectangle`

MapInfo rectangle objects are represented in the FME as closed polygons. When a MapInfo rectangle is read, it is turned into a closed polygon feature. When a MapInfo rectangle is written, the minimum bounding rectangle of the feature is taken and used as the four corners of the rectangle. MapInfo rectangles take the same additional attributes as MapInfo regions to specify their brush and pen.

Rounded Rectangle

mapinfo_type: `mapinfo_rounded_rectangle`

MapInfo rounded rectangle objects are represented in the FME as closed polygons. When a MapInfo rounded rectangle is read, it is turned into a closed polygon feature and the corners are vectorized to preserve the intended shape of the rectangle. The rounding radius is also stored as an attribute. When a MapInfo rounded rectangle is written, the minimum bounding rectangle of the feature is taken and used as the four corners of the rectangle, and the rounding diameter is taken from an attribute of the feature. MapInfo rounded rectangles take the same additional attributes as MapInfo regions to specify their brush and pen.



Attribute Name	Contents
`mapinfo_rounding_width`	Contains the width, in ground units, of the ellipse used to produce the rounded corners. **Range:** Any real number > 0 **Default:** No default
`mapinfo_rounding_height`	Contains the height, in ground units, of the ellipse used to produce the rounded corners. **Range:** Any real number > 0 **Default:** value of `mapinfo_rounding_width`

Collections

mapinfo_type: `mapinfo_collection`

MapInfo collections are defined as a combination of the other MapInfo geometry types. This is represented as nonhomogeneous aggregates composed of the other geometry types.

To create MapInfo collections using FME, set the `mapinfo_type` attribute to `mapinfo_collection` on the feature destined for the MapInfo dataset. It is important that the feature to be saved as a collection is an aggregate feature.

The table below lists the special FME attribute names used to control the MapInfo collection settings:

Attribute Name	Contents
mapinfo_collection_cmp{}	This is the list attribute prefix used to store the attributes for each collection part. The suffixes are the attribute names for the control settings of the other geometric types. Range: none Default: none