

Autodesk AutoCAD DWG/DXF Reader/Writer

FORMAT NOTES:

- This format contains Autodesk® RealDWG by Autodesk, Inc.¹

The AutoCAD® Reader and Writer modules allow the Feature Manipulation Engine (FME) to read and write files used by Autodesk® AutoCAD and compatible systems. AutoCAD drawing files consist of drawing settings and configuration, as well as a series of entities, or graphic elements, organized into layers.

Overview

FME provides broad support for many AutoCAD entity types and options and for reading and writing versions up to and including AutoCAD 2000. Reading and writing of AutoCAD version 2004 and version 2007 files is now supported for most previously supported entity types and keywords. Items excluded from supported AutoCAD 2004 reading or writing are noted in their respective sections. In addition, when AutoCAD data is output, header information may be copied from a supplied template, or prototype, file.

RealDWG



Support for the 2007 version of AutoCAD files has also been done with a new format type identifier labelled REALDWG. This change means that there are some parts of this document that apply to REALDWG specifically, and may be in duplication of pre-existing information about the handling of AutoCAD files. One significance of the REALDWG reading and writing of 2007 version AutoCAD files is that it complies with Autodesk AutoCAD TrustedDWG™ reading and writing.

This chapter assumes familiarity with AutoCAD-compatible systems and the entities (features) that are manipulated within these systems.

Note: Throughout this chapter, the AutoCAD file is referred to as a drawing file rather than a DWF file.

What is TrustedDWG?

At the option of the user, AutoCAD 2007 will notify users when the DWG file they are opening was saved using an application that was not created by an Autodesk product or RealDWG licensee.

1. Copyright © 1998-2006 Autodesk, Inc. All rights reserved.

AutoCAD DWG Quick Facts

Format Type Identifier	ACAD
Reader/Writer	Both
Dataset Type	File
Licensing Level	Base
Dependencies	None
Feature Type	Layer name
Typical File Extensions	.dwg, .dxf
Automated Translation Support	Yes
User-Defined Attributes	Yes
Coordinate System Support	No
Generic Color Support	Yes
Spatial Index	Never
Schema Required	Yes
Transaction Support	No
Geometry Type Attribute	autocad_entity
Rich Geometry	Yes
Encoding Support	Yes

Geometry Support			
Geometry	Supported?	Geometry	Supported?
aggregate	no	point	yes
circles	yes	polygon	yes
circular arc	yes	raster	no
donut polygon	yes	solid	no
elliptical arc	yes	surface	no
ellipses	yes	text	yes
line	yes	z values	yes
none	no		

AutoCAD RealDWG Quick Facts

Format Type Identifier	REALDWG
Reader/Writer	Both
Dataset Type	File
Licensing Level	Base
Dependencies	None
Feature Type	Layer name
Typical File Extensions	.dwg, .dxf
Automated Translation Support	Yes
User-Defined Attributes	Yes
Coordinate System Support	No
Generic Color Support	Yes
Spatial Index	Never
Schema Required	Yes
Transaction Support	No
Geometry Type Attribute	autocad_entity
Rich Geometry	No
Encoding Support	No

Geometry Support			
Geometry	Supported?	Geometry	Supported?
aggregate	no	point	yes
circles	yes	polygon	yes
circular arc	yes	raster	no
donut polygon	yes	solid	no
elliptical arc	yes	surface	no
ellipses	yes	text	yes
line	yes	z values	yes
none	no		

Background

There are two formats used by AutoCAD: DXF (drawing exchange format) files, which are large; and ASCII representations of the binary DWG (drawing) files. Logically, both files are identical and, therefore, the FME treats both file types in the same manner. For AutoCAD DWF reading and writing support please see the chapter on AutoCAD DWF.

AutoCAD DWG and DXF have been upgraded to take advantage of the FME rich geometry model. Advantages include the storage of linear and bulge arc segments of polylines and hatch loops, and the storage of more arcs and ellipses in more cases without being stroked to lines or polygons. Altogether, the addition of the rich geometry model support increases accuracy of geometric representation in AutoCAD-to-Au-

toCAD translations, as well as the creation and interpretation of more accurate features when translating to or from other FME formats. **(not applicable to RealDWG)**

AutoCAD files consist of sections, as follows:

1. **HEADER:** This contains settings of variables associated with the drawing.
2. **CLASSES:** This contains class definitions associated with the drawing.
3. **TABLES:** This contains a variety of tables, including:
 - **Layers:** Each layer entry contains layer definition information such as layer color, layer name, and layer linetype. The AutoCAD reader validates the layer names and may modify them to remove invalid characters.
 - **Linetypes:** Each linetype entry contains the linetype definition information such as name and alignment. The AutoCAD writer enables linetype definitions to be copied from an existing AutoCAD file, then referenced by name during the data translation.
 - **Shape Files:** Each shape file entry identifies a shape file referenced by the drawing. Shape files are used by AutoCAD as a different method for defining symbols or fonts. Note: These are similar to the TextStyles in AutoCAD 2004.

Note: AutoCAD shape files are *not* the same thing as ESRI Shapefiles. AutoCAD shape files store symbol and font definitions.

- **Applications:** Each application entry contains the name of an application referenced within the AutoCAD file.
4. **BLOCKS:** These are used to define symbols and other drawing file objects used repeatedly throughout a drawing. The AutoCAD writer enables copying of block definitions from an existing AutoCAD file, which is then referenced by name during a data translation operation.
 5. **ENTITIES:** This is the main section of a drawing file and contains the actual feature entities. Each entity contains standard information, such as its color, layer, thickness, linestyle, and geometry, as well as a number of attributes specific to its entity type. For example, a text entity has fields for font, size, and the text string in addition to the standard display attributes.

Tip: FME supports both 2D and 3D AutoCAD entities. However, many applications only support 2D DWG and DXF files. The `@Force2D` function can be used to ensure that only 2D data is written to an output DWG or DXF file.

6. **OBJECTS:** This section stores dictionaries and other helper non-entity objects.

Each entity may also have associated attribution stored within an extended entity data section. Extended entity data is fully supported by the FME.

All coordinates within a drawing file are stored as 64-bit floating point values in world coordinates. As such, there is no need to scale or otherwise alter coordinates as they are being read from or written to a drawing file.

The AutoCAD reader and writer use symbolic names for the different entity types stored within a drawing file. This simplifies feature type specification. The following table gives a brief description of each of the different AutoCAD entity types currently supported by the reader and/or writer. The entities are described in detail in subsequent sections.

AutoCAD Entity Types and Descriptions

FME autocad_entity	Description
autocad_line	Linear features stored within drawing file as a line or unclosed polyline.
autocad_point	Point features.
autocad_xline	Linear features of type xline.
autocad_ellipse	Features with an elliptical or circular representation.
autocad_shape	Features whose representation is stored in an AutoCAD shape file.
autocad_polygon	Features whose geometry is represented by a closed polyline.
autocad_face	Features represented by a 3D face object. The face object may have 3 or 4 coordinates.
autocad_arc	Features whose geometry represents a portion of a circular arc.
autocad_trace	Features with a 4 coordinate trace geometry.
autocad_solid	Features with a 3 or 4 coordinate solid geometry.
autocad_ray	Features with a linear geometry which represents a ray.
autocad_text	Text features.
autocad_spline	Spline features.
autocad_multi_text	Text features that store multiple lines of text. R14 and later only.
autocad_multi_line	A linear feature that is represented by more two or more parallel lines. Note: R14 and later only. This is supported only by the Reader.
autocad_insert	Point features that carry insert entity data.
autocad_leader	AutoCAD Leader entity representing leader lines in drawings.
autocad_hatch	Features with 2D boundary loops of various geometries that are filled with line patterns or color gradients.
autocad_attr_def	Features without geometry that contain information about AutoCAD attribute definition. Main information on features would be the Tag, Prompt and Default value for that attribute definition.

Reader Overview

The AutoCAD reader first reads the header and table information from the drawing file being processed, and caches information on blocks, shape files, layers, linetypes, and applications. These cached values are referenced by entities throughout the file and are needed when processing the entities.

The reader then extracts entities, one at a time, from the entity section of the drawing file and passes them on to the rest of the FME for processing. Complex entities such as polylines and inserts are extracted as single FME features. If the entity has attribution stored as extended entity data, then this is also read and placed in the feature.

When the AutoCAD reader encounters an entity type it does not know how to process, it simply sets the entity type of the feature and returns it. This feature is then logged by the FME correlation subsystem and the reader moves on to the next entity.

ESRI Product Coordinate System Information

To specify the FME coordinate system, the FME AutoCAD reader can recognize a coordinate system associated with AutoCAD data by ESRI products.

The AutoCAD reader will first look for the following files in the source directory:

- <filename>.prj
- esri_cad.prj

If neither of these files is present, the AutoCAD reader will try to find an ESRI_PRJ entry embedded in the file.

(This is not applicable to the RealDWG reader.)

Reader Directives

This section describes the directives that are recognized by the AutoCAD reader. Each directive is prefixed by the current <ReaderKeyword>_ when placed in a mapping file.

DATASET

Required/Optional: *Required*

The dataset into which feature data is to be read.

STORE_BULGE_INFO (only applicable with classic geometry)

Required/Optional: *Optional*

When specified, the AutoCAD Reader doesn't vectorize the Polyline and LWP Line Bulges but rather just stores the coefficients in the attribute `autocad_bulge`. In addition, when specified, this directive allows the creation of `autocad_start_width` and `autocad_end_width` attributes to represent the width properties of Polyline and LWP entities. This is generally set to `Yes` only when performing AutoCAD-to-AutoCAD translations.

Values: `YES` | `NO`

Default value: *NO*

SPLIT_BULGE_ARCS (only applicable with classic geometry)

Required/Optional: *Optional*

When specified, the AutoCAD Reader doesn't vectorize the polylines but rather returns one feature for each arc that has a bulge in it as an `autocad_arc` feature.

When features are read using rich geometry this directive will be ignored. To split rich geometry paths use the PathSplitter transformer.

Values: *YES* | *NO*

Default value: *NO*

STORE_SPLINE_DEFS

Required/Optional: *Optional*

When specified, the AutoCAD Reader, in addition to vectorizing the splines, stores the spline coefficients as attributes. See the description of Spline below for the attribute names used to store the spline definition. This is generally set to *Yes* when performing AutoCAD-to-AutoCAD translations.

Values: *YES* | *NO*

Default value: *YES*

RESOLVE_BLOCKS

Required/Optional: *Optional*

Specifies whether the reader will resolve (or explode) the block entities when processing inserts, or if it should just treat inserts as a point feature. This is generally set to *No* when performing AutoCAD-to-AutoCAD translations.

When the reader resolves blocks, it outputs a feature for each of the AutoCAD entities that are part of the block definition. The original insert is not output. This results in the full graphical representation of the block transferred through FME, but the exact insertion point of the block is lost.

Each block member feature is given the attribute `autocad_block_number` which is set to the same value for each block so that the features comprising each block may be combined in subsequent processing. Arbitrarily deep block nesting is permitted, however, the `autocad_block_number` attribute is only updated for each block at the outermost level. By default, all block members will be on the same layer as that of the original block. Setting the `USE_BLOCK_HEADER_LAYER` to *No* will cause the block members to appear on their respective layers.

If the block contains "Attribute" then each instance of "Attribute" in the block entity will be returned as "Text" entity along with a non-spatial feature containing information about that "Attribute" definition and its value for that block.

If the exact insertion point of the block is desired, then block resolution should be turned off and the insert entities for each block should be translated into point features in the output system.

Values: YES | NO

Default value: NO

DO_NOT_RESOLVE_BLOCKS

Required/Optional: *Optional*

Is only processed when RESOLVE_BLOCKS is specified. This directive is an exception list of the blocks that are not to be resolved. This is a space delimited list of the block names.

Values: *space-delimited list of block names*

Default value: *empty list*

RESOLVE_DIMENSIONS

Required/Optional: *Optional*

Specifies whether or not to resolve (explode) dimensions into their individual pieces. If the value is *yes*, then each piece of the dimension will be output as a separate feature. If the value is *no*, then an aggregate, containing all the pieces of the original dimension, will be output. This is generally set to *NO* when performing AutoCAD-to-AutoCAD translations.

Values: YES | NO

Default value: YES

PRESERVE_INSERTS

Required/Optional: *Optional*

If the value for RESOLVE_BLOCKS is *yes*, and this directive is also *yes*, then block insert points are output as point features.

Values: YES | NO

Default value: NO

CONVERT_ZERO_LENGTH_ARCS_TO_POINTS

Required/Optional: *Optional*

Specifies whether a zero length arc should be converted into a point feature (i.e., `autocad_point`). If the feature becomes a point, it will still retain all the attributes it had while it was an arc. This is generally set to `no` only when performing AutoCAD-to-AutoCAD translations.

Values: *YES | NO*

Default value: *YES*

OUTPUT_BLOCKS_AT_START

Required/Optional: *Optional*

Note: Not supported in AutoCAD 2004.

Specifies that the reader will output all the block definitions at the beginning of the translation before any other features are output. After the blocks are output, the rest of the translation is run without the blocks being resolved. When specified, this value overrides the value specified by `RESOLVE_BLOCKS`.

When set, all features that are part of a block definition have the attribute `autocad_block_definition`, with the value of the attribute being the name of the block which they are a component.

Values: *YES | NO*

Default value: *NO*

IGNORE_FROZEN_LAYERS

Required/Optional: *Optional*

Specifies whether the reader will ignore all features on the frozen layers. If set to `yes`, then features located on the frozen layers are not read from the input data set. If set to `no`, then the features are read from the frozen layer.

Values: *YES | NO*

Default value: *NO*

IGNORE_LOCKED_LAYERS

Required/Optional: *Optional*

Specifies whether the reader will ignore all features on the locked layers. If set to `yes`, then features located on the locked layers are not read from the input data set. If set to `no`, then the features are read from the locked layer.

Values: *YES | NO*

Default value: *NO*

IGNORE_HIDDEN_LAYERS

Required/Optional: *Optional*

Specifies whether the reader will ignore all features on the hidden layers. If set to *yes*, then features located on the hidden layers are not read from the input dataset. If set to *no*, then the features are read from the hidden layer. If this option is specified at the time of workspace or mapping file generation, and the schema mode is by layer, then no schema information from hidden layers will be used to generate the workspace or mapping file. This is generally set to *No* when performing AutoCAD-to-AutoCAD translations.

Values: *YES | NO*

Default value: *YES*

VISIBLE_ATTRIBUTES_AS_TEXT

Required/Optional: *Optional*

Specifies whether the reader should return visible attributes as separate text features or whether they should be returned as attributes of an insert feature. When this is *yes*, then each visible attribute is returned as a single text feature. This is generally set to *No* when performing AutoCAD-to-AutoCAD translations.

Values: *YES | NO*

Default value: *YES*

EXTENDED_ENTITY_FORMAT

Required/Optional: *Optional*

Instructs the FME to use the specified manner when decoding the extended entity data. This directive disables the automatic parsing.

Values: *ALTERNATE_NAME_VALUE | CSV(<SEPARATOR>)*

where *<SEPARATOR>* specifies the character used to delimit the attribute name from the attribute value.

When *ALTERNATE_NAME_VALUE* is specified, then it is assumed that the values stored with each feature in the extended entity portion of the feature alternate between specifying the attribute name and attribute value.

When *CSV(<SEPARATOR>)* is specified, then an attribute name value pair is specified in each extended entity value. The values are separated by *<SEPARATOR>*.

READ_PAPER_SPACE**Required/Optional:** *Optional*

Instructs the FME to also read the entities from paper space. By default, the FME only reads the entities from model space.

Values: *YES | NO***Default value:** *NO***READ_GROUPS****Required/Optional:** *Optional*

Determines whether or not AutoCAD groups will be read. By default, FME will not read groups.

Values: *YES | NO***Default value:** *NO***IGNORE_UCS****Required/Optional:** *Optional*

Instructs the FME to ignore the user defined coordinate system of the file being read. By default, the FME applies the UCS when reading the coordinate data. This is generally set to *Yes* only when performing AutoCAD-to-AutoCAD translations.

Values: *YES | NO***Default value:** *NO***USE_DXF_HEADER****Required/Optional:** *Optional*

When reading DXF files this instructs the FME Reader to use the specified dxf header file as the header for the file being read. This option is used to handle the case where organizations produce *headerless* dxf files to save storage space.

Values: *<DXF_HEADER_FILE>*

where *<DXF_HEADER_FILE>* specifies the full pathname to the dxf header file. If this is specified when reading a DWG file, then the translation is terminated with an error.

SKIP_TO_SECTION**Required/Optional:** *Optional*

When `USE_DXF_HEADERS` is specified above, this specifies how much of headerless file is to be skipped. In some cases, the headerless file has a placeholder which has to be removed before concatenating the above header file to the dataset. If not specified, then no lines are skipped.

Values: *one of the AutoCAD SECTIONs as specified in AutoCAD files*

CLASS_DEF

Required/Optional: *Optional*

Note: Class DEFs are not supported in AutoCAD 2004.

This directive defines the structure of an AutoCAD class object as defined within an AutoCAD ARX. This definition must match exactly the order in which the data is written within the ARX source file. If the definition doesn't match the layout in the object, then results are unpredictable and FME may crash. The definition of the entity's structure is not stored anywhere within the AutoCAD file, making it critical that the definition given here matches exactly the manner in which the data was stored within the file. To determine this, the source code must be inspected.

The class definition is of the following form.

```
<WriterKeyword>_CLASS_DEF <class_name> \
  [<column_name> <column_type>]+
```

where:

- `<class_name>` is the name of the class being defined. This name must match the application class name specified by the ARX that reads and writes the proxy data.
- `<column_name> <column_type>` is the definition of an attribute to be stored within the proxy data object.

The supported attribute types are described in the table below. The types specified here are dictated by the ARX extension used to read and write the data and the values must match that one.

Attribute Type	Description
char	single-character field
char(<length>)	variable-length-character field
boolean	Boolean value
double	64-bit real number
int16	16-bit integer value
int32	32-bit integer value
uint16	unsigned 16-bit integer
uint32	unsigned 32-bit integer
uint8	unsigned 8-bit integer

Attribute Type	Description
point2d	Two-dimensional coordinate. This is stored as two attributes: <column_name>.x and <column_name>.y. If the <column_name> is FME_COORD then this instructs the reader that the coordinates are to be stored as part of the feature geometry rather than as attributes. If stored as geometry, then the reader also doesn't read the graphic portion of the AutoCAD entity instead using the this field as the geometric information.
point3d	Three-dimensional coordinate. This is stored as three attributes: <column_name>.x, <column_name>.y, and <column_name>.z. If the <column_name> is FME_COORD, then this instructs the reader that the coordinates are to be stored as part of the feature geometry rather than as attributes. If stored as geometry, then the reader also doesn't read the graphic portion of the AutoCAD entity instead using the this field as the geometric information.
REPEAT	This is the type used to indicate the start and end of a repeating block. If the <column_name> is START then this marks the start of a repeating block, and if the <column_name> is END then it marks the end of the repeating block. If the <column_name> is not START or END then the definition is illegal and an error is returned to the user.
COUNT	This type is used in conjunction with REPEAT above and must immediately follow the START REPEAT column definition. The <column_name> here must match a previously defined column and is the column whose value indicates the number of times the repeating block exists.

The example below defines a proxy class called `WIRE_ENTITY`. The class has several attributes defined in the same order that they are written by the ARX. Notice the definition of three repeating blocks each delimited with `START REPEAT` and `END REPEAT`.

```
DWG_CLASS_DEF WIRE_ENTITY           \
IONID double                        \
GHOSTED boolean                     \
M_SIZE int32                        \
M_STATUS char                       \
M_SCC char(256)                     \
M_LABEL char(256)                   \
M_MATDSC char(256)                  \
NUMVERTS uint16                     \
START REPEAT                        \
NUMVERTS COUNT                      \
FME_COORD point2d                   \
coords.bulge double                 \
coords.swidht double                \
```

```

coords.ewidth double          \
END REPEAT                    \
BREAKSEGCOUNT int16         \
START REPEAT                  \
BREAKSEGCOUNT COUNT        \
BREAKSEG uint16              \
END REPEAT                    \
TEXTPTCOUNT uint16         \
START REPEAT                  \
TEXTPTCOUNT COUNT          \
TEXTPT point3d               \
END REPEAT

```

PRESERVE_COMPLEX_HATCHES

Required/Optional: *Optional*

Specifies whether or not to read hatches in a way that preserves their complex properties. If the value is `NO`, then the loops of each hatch entity will be converted to areas and aggregated together. If the value is `YES`, then the loops will be aggregated together as polygons, ordered such that any enclosing loop will be aggregated before any enclosed loop. This is generally set to `YES` when performing AutoCAD-to-AutoCAD translations.

Values: `YES` | `NO`

Default value: `NO`

READ_AS_2_5D

Required/Optional: *Optional*

Determines whether polylines should have their elevation attribute treated as a Z coordinate; when this occurs the `autocad_elevation` attribute will not be present. Applies to light-weight polylines and 2D polylines.

Note that this option should not be set when doing AutoCAD-to-AutoCAD translations as the elevations converted to Z coordinates when read in will not be converted back to elevation attributes when written out.

Values: `YES` | `NO`

Default value: `NO`

RESOLVE_ENTITY_COLOR

Required/Optional: *Optional*

Specifies whether or not to resolve the color of the entity to the color of the layer for that entity. This resolution only affects entity's who already have a color that is set to `COLOR_BYLAYER` which is indicated by a color index of 256. If the value of this directive is `yes`, the `autocad_color` attribute is set to the color index for the layer that the entity is on. If the value is `no`, then the `autocad_color` attribute will remain unresolved as a value of 256. This is generally set to `No` when performing AutoCAD-to-AutoCAD translations.

Values: YES | NO

Default value: YES

Tips for AutoCAD Reading

Tip: The AutoCAD reader automatically determines whether the file is DWG or DXF and processes it accordingly. Therefore, the same mapping file can be used to read either DXF or DWG.

Note that the AutoCAD reader directives list defaults that will produce the best generic result to any non-AutoCAD destination format, resulting in the best presentation at the cost of some of the original AutoCAD types being lost. For example, blocks and dimensions will be resolved into their component parts. If an AutoCAD-to-AutoCAD translation is desired, the following set of options should be used for best preservation of the original drawing.

```
STORE_BULGE_INFO = YES
STORE_SPLINE_DEFS = YES
RESOLVE_BLOCKS = NO
RESOLVE_DIMENSIONS = NO
VISIBLE_ATTRIBUTES_AS_TEXT = NO
IGNORE_UCS = YES
PRESERVE_COMPLEX_HATCHES = YES
IGNORE_HIDDEN_LAYERS = NO
RESOLVE_ENTIY_COLOR = NO
```

These are also noted on each individual directive description.

Writer Overview

The AutoCAD writer provides the following capabilities when writing AutoCAD files.

- **User-defined Linetypes:** New linetypes can be defined on FME mapping file lines. These linetypes can then be referenced by features being written to the AutoCAD file.
- **User-defined Layers:** Users must define the layers into which features are stored. The layers can also define the attributes to be stored within the feature.
- **Copy Block Definitions:** Often users have existing AutoCAD drawing files that contain block definitions they want the translated data to carry. Specifying the `TEMPLATEFILE` keyword in the mapping file results in block definitions being copied from the existing file to the output DWG/DXF file. These blocks can then be referred to by insert entities.
- **Copy Linetypes:** Predefined linetypes within existing DWG/DXF files are copied making them available for use by features being written to the destination file. Specifying the `TEMPLATEFILE` keyword in the mapping file results in the predefined linetypes being copied from the template file to the output drawing file. Feature entities can then refer to these linetype definitions.
- **Copy Layer Definitions:** Layer definitions within an existing DWG/DXF file identified by `TEMPLATEFILE` enable layer definitions to be copied to the destination data set and then referenced.

- **Copy Shape Header Definitions:** Shape header definitions are also copied from the file specified by the `TEMPLATEFILE` directive.
- **Automatic Block Creation:** When a feature is passed to the writer that cannot be written as a single AutoCAD entity, such as a donut polygon, the writer automatically defines an AutoCAD block and inserts entities necessary to represent the feature.
- **Flexible Attribute Support:** Attribute information is, by default, written to extended entity data for each feature written to the data set. This can be overridden, however, through the use of the `autocad_attributes` attribute being set as shown in the following table.
- **Multi-version Support:** Currently the AutoCAD DWG/DXF writer supports files that are compatible with any current AutoCAD release.

autocad_attributes value	Description
<code>extended_entity_data</code>	This results in the attribution being written to the extended entity for the feature.
<code>insert_attributes</code>	This results in the writer creating an insert entity for each feature and storing all attributes with the insert entity. The insert entity refers to a block which contains the geometry of the output feature.
<code>external_attributes</code>	This is the default value. There are no attributes written to the AutoCAD file. This is useful if the attributes are being stored in an external database or if attribute information is not wanted.

When creating AutoCAD files, the AutoCAD writer first defines the linetypes and layers defined within the FME mapping file. The writer then reads in a template file, if specified, and copies the linetypes, layer definitions, shape file header information, and block information from the template file to the output data set.

The AutoCAD writer then outputs each feature it is given to the output file in the appropriate entity type.

When writing an AutoCAD file, the format of file output is determined as follows:

- If the file name contains `.dwg` or `.DWG`, then the output data set is written in the ACAD format.
- Otherwise, if the file name contains `.dxf` or `.DXF`, then the output data set is written in DXF format.
- Otherwise, if an error exists in the mapping file, the translation is halted.

The AutoCAD writer uses the above rules to enable the same FME mapping file to be used to create both DXF and DWG output files. Users are able to specify their choice simply by changing the suffix of the output file being produced.

Writer Directives

This section describes the directives the AutoCAD writer module recognizes. Each of the directives is prefixed by the current `<WriterKeyword>_` when they are placed in a

mapping file. By default, the <WriterKeyword> for the AutoCAD writer is the same as the <WriterType>.The following directives are used by all AutoCAD.

DATASET

Required/Optional: *Required*

The dataset into which feature data is to be written.

VERSION

Required/Optional: *Required*

The version of AutoCAD file to be produced. The value corresponds with the release number of the AutoCAD file that is produced.

Values:

- *same_as_template* (not applicable to RealDWG)
This option takes the version from the template file.
- *Release12* (not applicable to RealDWG)
- *Release13* (not applicable to RealDWG)
- *Release14*
- *Release2000*
- *Release2004*
- *Release2007*

Default value: *Release2007*

Example:

The example statement below instructs the AutoCAD writer to produce a release 12 AutoCAD file:

```
DWG_VERSION Release12
```

TEMPLATEFILE

Required/Optional: *Optional*

This statement specifies the name of the existing AutoCAD DXF or DWG file that contains linetype, layer, shape header, block definitions and a codepage to be copied to the destination AutoCAD file. Some AutoCAD users also refer to this as a *prototype file*. This is an optional parameter. If the parameter is not defined, then the output file uses the linetype defined in the mapping file along with the predefined type of CONTINUOUS which is always present in an AutoCAD drawing.

Tip:

- LINETYPE definitions found in the mapping file override any linetype definitions found in the template file.
 - The template file can also be used to set the codepage of the resulting AutoCAD file.
-

The example below specifies that the file called `c:\tmp\test.dwg` contains the block, layer, shape header definitions, and linetype definitions for the output data set.

```
DWG_TEMPLATEFILE c:/tmp/test.dwg
```

Tip: Many AutoCAD users refer to the template files as *prototype* files.

AUTO_CREATE_LAYERS

Required/Optional: *Optional*

This statement tells the writer to create layers as needed. Normally, all layers must either be defined by `_DEF` lines or by the template file before they can be used. If `AUTO_CREATE_LAYERS` is specified as `YES`, then when a feature is sent to the writer with a feature type that has not previously been defined as a layer, a new layer will be created with the properties of the last `_DEF` line found in the mapping file.

This example sets the writer into a mode where it creates layers as needed. Each created layer has a color of 10 and a linetype of `CONTINUOUS`.

```
DWG_AUTO_CREATE_LAYERS yes
DWG_DEF DEFAULT
autocad_color 10
  autocad_linetype CONTINUOUS
```

Values: `YES` | `NO`

Default value: `NO`

OUTPUT_DEFINED_ATTRS_ONLY

Required/Optional: *Optional*

When this directive is set to `yes`, then only those attributes defined as part of the layer definition will be stored (see the `DEF` directive for more details).

Values: `YES` | `NO`

Default value: `YES`

USE_ATTRDEFS_FOR_INSERTS

Specifies whether the writer should use the attribute definitions that are found within blocks when placing inserts. If `no`, then all the attributes on a feature that is passed to the writer are written as insert attributes. If `yes`, then only the attributes defined within the block being placed are stored as insert attributes.

Values: `YES` | `NO`

Default value: `YES`

STRIP_HEADER_TO_SECTION

Required/Optional: *Optional*

This directive is only valid when going out to DXF and tells the FME to remove the header up to the start of the specified `SECTION`. The name of the `SECTION` can be any valid autocad section. If not specified then the file is output as before.

NEW_HEADER_CONTENT_FILE

Required/Optional: *Optional*

DEPRECATED. Please use the `TEMPLATEFILE` directive instead.

This directive is only valid with the `STRIP_HEADER_TO_SECTION` directive above and is the name of the file that contains the new header information. The contents of this file are placed at the start of the output file replacing the contents removed by `STRIP_HEADER_TO_SECTION`.

DEFAULT_ATTR_STORAGE

Required/Optional: *Optional*

This directive specifies the default manner in which attribute data will be stored. If not specified, then the default value is `extended_entity_data`. This directive changes the default value for the `autocad_attributes` feature based directive. If all attributes are to be stored in a single manner, then this directive is the easiest manner in which to do this.

Values:

- *extended_entity_data* instructs the writer to store all attribution in extended entity data as the default
- *insert_attributes* instructs the writer to store all attribution using inserts
- *external_attributes* instructs the writer to not store any attribution in the AutoCAD file

SUPPRESS_FONT_WARNINGS

Required/Optional: *Optional*

Specifies whether to suppress warnings about unknown font metrics being encountered.

Values: *YES* | *NO*

Default value: *NO*

FONT_DIRECTORY

Required/Optional: *Optional*

Specifies the directory in which all specified fonts are located. When specified, FME can calculate the font metrics, enabling it to place fonts more accurately. If not specified, then FME assumes that the full path is specified on the correlation lines via the attribute `autocad_shape_filename`. If this attribute is not set or it is not the full path of the font file, then FME will use the default font file called `default.shx`.

Values: **

SHAPE_DIRECTORY

Required/Optional: *Optional*

Specifies the directory in which to look for shapes files (*.shp) from which information about SHAPE entities will be extracted for writing.

Values: *<shape file directory path>*

DEFAULT_APPLICATION

Required/Optional: *Optional*

The application name that is used when writing extended entity data.

Default values: *ACAD*

LINETYPE

Required/Optional: *Optional*

The AutoCAD writer enables linetypes to be defined within the FME mapping file. This enables the user to control how output lines are to look in the destination data set. The linetype definition is of the following form:

```
<WriterKeyword>_LINETYPE <linetype name>           \
autocad_textpict <picture>                           \
[autocad_patternLength <pattern Length>             \
  <segment values>+                                   \
]
```

where:

- `<linetype name>` is the name used throughout the mapping file to refer to the linetype being defined by this statement.
- `<picture>` is the text or name displayed in AutoCAD when linetypes are displayed.
- `<pattern Length>` is the length of a single instance of the line.
- `<segment values>` are the length of each of the segments within the linetype segment. The segment values obey the following rules:
 - negative value – pen up length (used to create spaces of varying lengths)
 - positive value – pen down length (used to make dashes of varying lengths)
 - zero – used to create a dot

The following example creates a linetype called dash-dot that appears as “_ _ . _ _ . _ _ .” and so on when displayed on the screen.

```
DWG_LINETYPE dash-dot \
  autocad_textpict DASHDOT \
  autocad_patternLength 1.0 \
  0.5 -0.25 0 -0.25
```

CREATED_BLOCK_NAME_PREFIX

Required/Optional: *Optional*

The names of blocks created by the writer for aggregate geometry features will use the value of this keyword as a prefix if specified. If not specified the syntax will be:

```
<filename>_<writerkeyword>_FME_BLOCK_<blocknumber>
```

DEF

Required/Optional: *Optional*

The AutoCAD writer requires that every feature written to the AutoCAD file be stored within a predefined AutoCAD layer. In AutoCAD, the layers are used to store collections of logically related attributes. Within the FME, the AutoCAD layer and the type of the feature are treated synonymously as there is a one-to-one correspondence between FME feature type and AutoCAD layer.¹ The layer statement is of the following form:

```
<WriterKeyword>_DEF <layer name> \
  autocad_color <default color> \
  autocad_linetype <default linetype> \
  [autocad_layer_type frozen] \
[<attribute name> <attribute type>]
```

where:

- <layer name> is the name of the layer being defined. This is the name which is used throughout the remainder of the FME mapping files. Layer name cannot be empty. If no layer name is specified, then FME will provide a fixed name “_FME_NO_LAYER_NAME_” for such layers.
- <default color> is the color number used for all features stored within the layer unless explicitly overridden on the correlation lines below. Valid values are between 1 and 255.
- <default linetype> is the name of the linetype to use for the layer if no linetype is specified on the correlation line. The linetype specified must either be:
 - defined in the mapping file,
 - copied from a specified template file, or
 - the predefined linetype named CONTINUOUS.
- <autocad_layer_type> is the type of layer to create. Currently, only the value frozen is supported. If specified, then the created layer is frozen; otherwise, the layer is not frozen.
- <attribute name> <attribute type> is the definition of an attribute to be stored within the extended entity data of features for the layer. If no attributes are

1. Layers can also be defined through the use of a TEMPLATEFILE.

defined, then all feature attributes (except those that start with `autocad_`) are stored. The storing of attributes can be turned off by specifying a value of `external_attributes` for the `autocad_attributes` feature attribute on the correlation line. The values for `<attribute type>` are the same as those for ESRI Shapefiles.

The example below defines a layer called `boundary` in which entities are drawn using color 13 (unless otherwise specified) and a linetype called `dash-dot` (unless otherwise specified). The feature also has several attributes specified that will be written to the extended entity data of each feature within the layer.

```
DWG_DEF boundary \
  autocad_color 13 \
  autocad_linetype dash-dot \
  FEATCODE char(12) \
  PPID char(10) \
  DATECHNG date \
  SURVEYDIST number(8,2)
```

AUDIT_AND_FIX (not supported in RealDWG writer)

Required/Optional: *Optional*

This directive can be used to turn on/off internal auditing before the final drawing file is written out. By default it is set to `YES`, so auditing will be performed and any errors found will be fixed. It is recommended that you leave the auditing set to `YES`. If you set it to `NO`, it is possible that the output file may not be as per the AutoCAD file specification. As an example of how auditing fixes errors is that if there is a layer name with a space or any other invalid characters, then that layer name will be changed to something like `$DDT_AUDIT_GENERATED_(3B)`. If auditing is turned off, then the layer name will not be changed and when it is audited in AutoCAD, it will return errors such as invalid layer names.

Values: `YES` | `NO`

Default value: `YES`

COORDINATE_SYSTEM_STORAGE (not applicable to RealDWG writer)

Required/Optional: *Optional*

Note: Only supported in AutoCAD 2004.

This directive controls whether the writer will optionally store the coordinate system of its features. The coordinate system can be stored inside the output AutoCAD file as an ESRI Well Known Text (in an `ESRI_PRJ` entry in an internal dictionary in the file), according to ESRI specifications. It can also be stored externally in a companion ESRI `.prj` file that shares the output AutoCAD file's base name, but has a `.prj` extension.

Values: `NONE` | `EXTERNAL_PRJ` | `EXTERNAL_AND_INTERNAL` | `INTERNAL_WKT`

Default: `NONE`, which means projection information is not stored anywhere.

AutoCAD data files written this way with projection information will be recognized by FME and the free ArcGIS for AutoCAD application, which installs on top of the AutoCAD application.

Example:

```
DWG_COORDINATE_SYSTEM_STORAGE EXTERNAL_PRJ
```

APPEND_TO_TEMPLATEFILE

Required/Optional: *Optional*

This directive can be used to allow the file specified by the `DATASET` directive to be written as the concatenation of the full contents of the file specified by the `TEMPLATEFILE` directive with all written data. By default it is set to `NO`, so only header information but no data is used from the template file if one is specified. If this directive is set to `YES` then the full header and data information is used from the template file or nis specified.

Values: `YES` | `NO`

Default value: `NO`

Feature Representation

Special FME feature attributes are used to hold AutoCAD entity attributes. The AutoCAD writer uses these attribute values as it fills in an entity structure during output. The AutoCAD reader sets these attributes in the FME feature it creates for each entity it reads.

The FME considers the AutoCAD layer¹ to be the *FME feature type* of an AutoCAD feature. Each AutoCAD entity, regardless of its entity type, shares a number of other attributes, as described in the following table. Subsequent sections describe attributes specific to each of the supported entity types.

Attribute Name	Content
<code>autocad_layer</code>	The name of the feature's layer. This is the same value as the feature's type and is stored when reading for reasons of convenience. This value is ignored when entities are being written to a drawing file. Value: <code>char(33)</code> Default: No default
<code>autocad_layer_type</code> (not used in RealDWG)	This is used by the Reader only and indicates whether or not the feature comes from a frozen layer. Range: <code>frozen</code> <code>not_frozen</code>
<code>autocad_layer_locked</code> (only in RealDWG)	This is used by the Reader only and indicates whether or not the feature comes from a locked layer. Range: <code>yes</code> <code>no</code> Default: <code>no</code>

1. The feature layer name corresponds to be the feature type and `autocad_layer` when reading. This enables the layer name to be extracted without the need to use the `@FeatureType` function.

Attribute Name	Content
autocad_layer_hidden (only in RealDWG)	This is used by the Reader only and indicates whether or not the feature comes from a hidden layer. Range: yes no Default: yes
autocad_color	The color number of the entity. If the value is 0, then the color of the entity is that of the enclosing block; if the value is 256, then the color of the entity is that specified by the entity's layer; otherwise, the number specified determines the color of the entity. If autocad_color is not specified, then the value will be set from fme_color. If fme_color is also not specified, then it will be set to COLOR_BYLAYER. Range: 0...256 Default: 256
autocad_entity_handle	The hexadecimal unique identifier for the entity. This value is unique within each AutoCAD file. Range: Hexadecimal identifier. Default: No default
autocad_entity_visibility	This is used by the Reader only and indicates whether or not the feature is visible. Range: visible invisible.
autocad_linetype	The name of the feature's linetype. Range: char[33] Default: BYLAYER Note: When autocad_linetype is defined as BYLAYER, see autocad_resolved_linetype for the specific linetype values.
autocad_linetype_scale	The amount to scale the feature's linetype by for viewing in AutoCAD. Failure to set appropriate values for linetype may result in viewing errors such as dashed lines appearing solid. Range: 64 bit Real Default: 1.0
autocad_lineweight	The lineweight of the AutoCAD entity in 100ths of a millimeter. To set a lineweight of 0.05 mm in AutoCAD, set the attribute value to 5. Range: 0, 5, 9, 13, 15, 18, 20, 25, 30, 35, 40, 50, 53, 60, 70, 80, 90, 100, 106, 120, 140, 158, 200, 211, -1 (by layer), -2 (by block), -3 (default) Default: -3 (Default)
autocad_resolved_linetype	If the autocad_linetype has the value of BYLAYER, then this value contains the linetype assigned to the layer; otherwise, it has the same value as autocad_linetype. Range: char[33]
autocad_thickness	The thickness of the entity's lines. Range: 64 bit Real Default: 0
autocad_entity	The FME name for the type of entity this feature represents. Range: See <i>AutoCAD Entity Types and Descriptions</i> on page 109 Default: No default

Attribute Name	Content
autocad_original_entity	This attribute indicates that the entity is part of an insert entity. Basically, if this attribute exists its value will be insert. Range: See <i>AutoCAD Entity Types and Descriptions</i> on page 109 Default: insert
autocad_original_entity_type	The FME name for the original type of entity this feature represents. For example, if the autocad_entity attribute is autocad_line, this attribute will indicate what type of line, i.e. line, 2dpolyline, lwpolyline or 3dpolyline. Range: See <i>AutoCAD Entity Types and Descriptions</i> on page 109 Default: The value of the autocad_entity attribute
autocad_space	This is used by the Reader only and indicates if the entity being read came from paper space or model space. Range: model_space paper_space Default: No default
autocad_attributes	Used by the writer module only. This directs the writer on how the attributes for the feature are to be stored. If this attribute is not specified or is specified as extended_entity_data then the attribution associated with the feature is written to the extended entity portion. If the value is insert_attributes, insert entities are created for the attributes. If the value is external_attributes then the attribution is not written to extended entity data. Range: extended_entity_data insert_attributes external_attributes Default: extended_entity_data

Extended Entity Data

Each entity in an AutoCAD file may have associated extended entity data. This data is typically used by applications to store attribute information. The AutoCAD reader attempts to make extended entity data as simple to use as possible by storing it in three different formats within the FME feature object. The first two formats merely store the data as found in the drawing file in the feature, while the third format attempts to present the attribute information in a more useful manner. It is important to remember that when extended entity data is read from an AutoCAD file, all three formats are stored within a single FME feature. The format that is actually used (if any) is dependent on the configuration of the remainder of the FME mapping file.

The AutoCAD writer understands only the *interpreted format*, described in *Interpreted Format* on page 133, for extended entity data. When writing extended entity data, the FME features being output must structure their attributes in this way. That is, the attribute data is stored with each attribute being a single extended entity string in the form <attribute name> = <attribute value>. Storing the data in this manner enables the data to be easily viewed by AutoCAD and read by the FME reader module.

List Format

In this format, the data is simply stored in a list as found in the AutoCAD file. The data is stored in a single list named `extended_data_list{}`. Each value in the list is of the form `<attribute tag>: <attribute value>`. The `<attribute tag>`s supported by the FME are restricted to those given in the following table. The `<attribute tag>`s define the domain for the associated `<attribute value>`. Note that the AutoCAD codes associated with each kind of extended entity data are not stored in the FME feature.

Attribute Name	Content
<code>application_name</code>	The name of the application which the following entity data is associated. This <code>application_name</code> remains in effect until another <code>application_name</code> entry is specified. AutoCAD Code: 1001 Example: <code>application_name:ACAD</code>
<code>autocad_layer</code>	The name of the layer the extended data is associated. AutoCAD Code: 1003 Example: <code>autocad_layer:Water</code>
<code>string</code>	A character string value from 0 to 255 characters in length. AutoCAD Code: 1000 Example: <code>string:Thompson</code>
<code>Binary data</code>	A hexadecimal string from 0 to 254 characters in length. AutoCAD Code: 1004 Example: <code>binary:E3B4</code>
<code>three_reals</code>	Three 64-bit real numbers separated by commas. AutoCAD Code: 1010,1020,1030 Example: <code>three_reals:2.3,4.5,3.4</code>
<code>world_position</code>	Three real numbers which represent a world position. Each of the numbers is separated by a comma. AutoCAD Code: 1011,1021,1031 Example: <code>world_position:23.4, -123.5, 0</code>
<code>world_displacement</code>	Three real values which represent a world displacement value. Each of the values is separated by a comma. AutoCAD Code: 1012,1022,1032 Example: <code>world_displacement:1.5, 2.3, 0</code>
<code>world_direction</code>	Three real values which represent a world direction vector. Each of the values is separated by a comma. AutoCAD Code: 1013,1023,1033 Example: <code>world_direction: 30.0, -12.4, 10</code>
<code>real</code>	A 64-bit real number. AutoCAD Code: 1040 Example: <code>real:3.1415926</code>
<code>distance</code>	A 64-bit real number which represents a distance. AutoCAD Code: 1041 Example: <code>distance:4.56</code>
<code>scale</code>	A 64-bit real number which represents a scaling factor. AutoCAD Code: 1042 Example: <code>scale:34.5</code>

Attribute Name	Content
16Bit_integer	A 16-bit integer value. AutoCAD Code: 1070 Example: 16Bit_integer:245
32Bit_integer	A 32-bit integer value. AutoCAD Code: 1071 Example: 32Bit_integer:12983

For example, if the following data was stored in extended entity data:

```
1001 C_NODE
 1000 CONNOBJ_1=43F4
 1000 COUNT=3
 1000 CONNOBJ_2=43F3
 1000 CONNOBJ_3=43F2
 1005 163
 1010 45.4
 1020 -123.5
 1030 0
 1001 DPRINT
 1000 postscript
```

then the FME AutoCAD reader would store this information as a list within the FME feature:

Attribute Name	Attribute Value
extended_data_list{0}	application_name:C_NODE
extended_data_list{1}	string:CONNOBJ_1=43F4
extended_data_list{2}	string:COUNT=3
extended_data_list{3}	string:CONNOBJ_2=43F3
extended_data_list{4}	string:CONNOBJ_3=43F2
extended_data_list{5}	handle:163
extended_data_list{6}	three_reals:45.4,-123.5,0
extended_data_list{7}	application_name:DPRINT
extended_data_list{8}	string:postscript

Notice how the AutoCAD codes are converted to attribute tags when stored in the FME features.

Structure Format

In this representation of extended entity data, the fields are stored with the tags forming part of the attribute names for each of the extended entity entries. The data is stored in a single structure in the FME feature named `extended_data`. As the extended entity data within AutoCAD is grouped into sections, with each section beginning with an application group code, the `extended_data` structure itself is also divided into dif-

ferent sections with each section beginning with `extended_data{#}`. The remainder of the attribute name consists of one of the parameters:

Extended Entity Parameter	Contents
<code>application_name</code>	The name of the application which the entity data is associated. AutoCAD Code: 1001
<code>autocad_layer{#}</code>	The name of the layer the extended data is associated. AutoCAD Code: 1003
<code>string{#}</code>	A character string value from 0 to 255 characters in length. AutoCAD Code: 1000
<code>three_reals{#}.real1</code> <code>three_reals{#}.real2</code> <code>three_reals{#}.real3</code>	Three real numbers. AutoCAD Code: 1010,1020,1030
<code>world_position{#}.x</code> <code>world_position{#}.y</code> <code>world_position{#}.z</code>	Three values represent the x, y, and z components of a world_position value. AutoCAD Code: 1011, 1021, 1031
<code>world_displacement{#}.x</code> <code>world_displacement{#}.y</code> <code>world_displacement{#}.z</code>	Three values which represent a world displacement value. AutoCAD Code: 1012, 1022, 1032
<code>world_direction{#}.x</code> <code>world_direction{#}.y</code> <code>world_direction{#}.z</code>	Three real values which represent a world direction vector. AutoCAD Code: 1013,1023,1033
<code>real{#}</code>	A 64 bit real number. AutoCAD Code: 1040
<code>handle{#}</code>	AutoCAD handle value. AutoCAD Code: 1005
<code>distance{#}</code>	A 64 bit real number which represents a distance. AutoCAD Code: 1041
<code>scale{#}</code>	A 64 bit real number which represents a scaling factor. AutoCAD Code: 1042
<code>16Bit_integer{#}</code>	A 16 bit integer value. AutoCAD Code: 1070
<code>32Bit_integer{#}</code>	A 32 bit integer value. AutoCAD Code: 1071

For example, given the following extended entity data:

```
1001 C_NODE
  1000 CONNOBJ_1=43F4
  1000 COUNT=3
  1000 CONNOBJ_2=43F3
  1000 CONNOBJ_3=43F2
  1005 163
  1010 45.4
  1020 -123.5
```

```

1030 0
1001 DPRINT
1000 postscript

```

The information will be stored in the FME feature using structure notation as follows:

Attribute Name	Attribute Value
extended_data{0}.application_name	C_NODE
extended_data{0}.string{0}	CONNOBJ_1=43F4
extended_data{0}.string{1}	COUNT=3
extended_data{0}.string{2}	CONNOBJ_2=43F3
extended_data{0}.string{3}	CONNOBJ_3=43F2
extended_data{0}.three_reals{0}.real1	45.4
extended_data{0}.three_reals{0}.real2	-123.5
extended_data{0}.three_reals{0}.real3	0
extended_data{0}.handle{0}	163
extended_data{1}.application_name	DPRINT
extended_data{1}.string{0}	postscript

Notice how, in this case, the AutoCAD codes are used to form extensions for the attribute names. Also notice how the `extended_data` items are grouped in the FME feature as they are within the drawing file.

Interpreted Format

Finally, the FME AutoCAD reader module also attempts to interpret any *string* held in the extended entity data. If it is successful in interpreting any data, then it stores it as attributes within the feature. As it is reading each extended entity string entry, it attempts to determine if the value is composed of an attribute name or value pair and, if it does, it stores the information as such. For example, if the extended entity data from the previous example were read, the following interpreted values would be stored within the FME feature.

Attribute Name	Attribute Value
CONNOBJ_1	43F4
COUNT	3
CONNOBJ_2	43F3
CONNOBJ_3	43F2

The reader is able to do this by recognizing the = divider within each of the string attributes as the separator between an encoded attribute name and attribute value. The reader also recognizes a space character as a separator.

The remaining sections discuss the representation of each supported AutoCAD entity type.

Proxy Data

Proxy data is also supported within the AutoCAD reader and writer. Proxy data is yet another manner in which data is stored within AutoCAD files. This data is normally associated with ARX extensions. To the rest of the FME, the proxy objects are made to look as close as possible to regular AutoCAD data. For example, linear entity types are called `autocad_line`. Proxy features have a number of associated attributes that are not present in other entities. **Note:** Proxy data is not supported in AutoCAD 2004.

Proxy Data Attribute	Contents
<code>autocad_proxy_number</code>	A unique number that is assigned to all the components of a single object. Since a single proxy object can have a number of geometric primitives associated with it, all proxy objects are output with the same number so that, if necessary, they can be identified as belonging together by the rest of the FME processing.
<code>autocad_class_number</code>	The number given to the class of which this proxy object is an instance.
<code>autocad_class_dxfname</code>	The dxf class name of the proxy class.
<code>autocad_class_cppname</code>	The C++ class name of the proxy class.
<code>autocad_class_appname</code>	The application class name of the proxy class.
<code>autocad_class_version</code>	The class version of the proxy class.

Lines

autocad_entity: `autocad_line`

Features with `autocad_entity` set to `autocad_line` are stored in and read from drawing files in one of two ways, depending on the number of coordinates they have, and whether they store bulge arcs. Bulge arcs are limited to circular, non-closed arcs within the segments of the line. Any attempts to store elliptical or closed arcs will result in the arc being stroked into a line segment.

Number of Coordinates	AutoCAD Entity Type	Description
2	line	If the feature contained exactly two points, then an AutoCAD line entity is used to store the data.
Greater than 2	polyline	If the number of coordinates is greater than 2, then the AutoCAD polyline entity is used to store the coordinates. The polyline closed flag is set to indicate that the polyline entity is not closed.

Attribute Name	Content
autocad_bulge	Comma-separated value list of the vertex bulges. This is only useful when performing AutoCAD-to-AutoCAD translations, and is a measurement of the curvature at each vertex.
Applicable only with classic geometry.	
autocad_elevation	The elevation value stored with the line entity. This is often used to set the elevation for contour lines, as the single elevation value is applied to all the vertices.
autocad_linetype_generation	Whether the generation of the autocad linetype will restart at every vertex, or be generated continuously around the entire polyline. Not applicable for 3d polyline features. The possible values are: 0 = Restart generation at each vertex. 1 = Generate continuously around entire polyline.
autocad_width	The width of the line.
autocad_polyflag	A bit-coded flag. This attribute is only present or used on autocad_line features that contain more than two vertices. Values can be combined by using addition. The values are: 1 = The line is closed (or the line is a polygon mesh closed in the M direction). Please note that if this bit is set when reading, then the feature will be interpreted as an autocad_polygon. 128 = The linetype pattern is generated continuously around the vertices of the line.

XLines

autocad_entity: autocad_xline

Features with autocad_entity set to autocad_xline are stored in and read from drawing files as an FME feature with two coordinates representing a line. The reader and writer modules automatically convert the xline to and from its unit vector representation into a line.

There are no attributes specific to this type of entity.

Points

autocad_entity: autocad_point

Features with autocad_entity set to autocad_point are stored in and read from drawing files as a single coordinate feature.

Attribute Name	Content
autocad_ucs_xangle	The rotation angle around the z axis.

Ellipses

autocad_entity: autocad_ellipse

Ellipse features are point features used to represent both AutoCAD circle and AutoCAD ellipse entities. The point serves as the centre of the ellipse. Ellipse entities with an `autocad_primary_axis` equal to the `autocad_secondary_axis` are stored within the drawing file as a circle entity. Additional attributes specify the rotation, major axis, and minor axis of the ellipse.

Tip: The function `@Arc()` can be used to convert an ellipse to a polygon. This is useful for representing ellipses in systems that do not support them directly.

Attribute Name	Content
<code>autocad_primary_axis</code>	The length of the semi-major axis in ground units. Range: Any real number > 0 Default: No default
<code>autocad_secondary_axis</code>	The length of the semi-minor axis in ground units. Range: Any real number > 0 Default: No default
<code>autocad_rotation</code>	The rotation of the major axis. The rotation is measured in degrees counterclockwise up from horizontal. Range: -360.0..360.0 Default: 0

Polygons

autocad_entity: `autocad_polygon`

Features with `autocad_entity` set to `autocad_polygon` are stored in and read from drawing files as closed polyline entities.

Attribute Name	Content
<code>autocad_width</code>	The width of the line.

Splines

autocad_entity: `autocad_splines`

Spline features are linear or area features – depending on whether or not they are closed – and are used to represent features that have smooth curves. Each spline has a number of attributes that completely make up the spline. When `STORE_SPLINE_DEFS` is set to `yes`, the reader sets the coordinates to be either the fit points or the control points (depending on what is used to define the spline). Splines are always 3D – there is no way in AutoCAD to indicate if the feature was intended to be only 2D. If `STORE_SPLINE_DEFS` is not specified or set to `no`, then the coordinates of the spline returned by the reader are interpolated values based on the spline definition.

Tip: When you're performing an AutoCAD-to-AutoCAD translation, then you should always set `STORE_SPLINE_DEFS` to `yes` to get the best results.

AutoCAD splines have several attributes, which are returned when reading and must be specified when writing.

Attribute Name	Content
autocad_degree	The degree of the polynomial used to form the spline.
autocad_knot_tolerance	The tolerance of the spline knots.
autocad_degree	The degree of the spline.
autocad_cntl_pt_tolerance	The tolerance of the control points.
autocad_fit_tolerance	The tolerance of fit points.
autocad_knot_tolerance	The tolerance of knots.
autocad_num_cntl_pts	The number of control points.
autocad_num_fit_pts	The number of fit points.
autocad_knots	The number of knots.
autocad_flag	The flag that indicates the type of spline. It is a bit vector normally only used when going from AutoCAD to AutoCAD. 1. CLOSED 2. PERIODIC 4. RATIONAL 8. PLANAR 16. LINEAR
autocad_start_tangent_x autocad_start_tangent_y autocad_start_tangent_z	The start tangent for the spline.
autocad_end_tangent_x autocad_end_tangent_y autocad_end_tangent_z	The end tangent for the spline.
autocad_control_x autocad_control_y autocad_control_z	A comma separated list. The control point coordinates in comma separated values. If <code>STORE_SPLINE_DEFS</code> is specified, then the control points are also stored as the coordinates.
autocad_control_weights	The control point weights. A comma-separated list of the weight values for each control vertex.

When writing to splines, the spline must be specified exactly as it is returned from the reader with `STORE_SPLINE_DEFS` yes set:

1. If the spline is defined by fit points then `autocad_num_cntl_pts` must be zero and `autocad_num_fit_pts` must be the same as the number of coordinates in the feature. The coordinates of the feature are taken to be the fit points.
2. If the spline is defined by control points then the `autocad_num_fit_pts` must be zero and `autocad_num_cntl_pts` must be the same as the number of coordinates in the feature. The coordinates of the feature are taken to be the control points.

Shapes

autocad_entity: autocad_shape

Features with `autocad_entity` set to `autocad_shape` are point features that identify where to place an AutoCAD shape object. The reader and writer modules process all attributes needed to fully specify the shape object reference. Depending on the output file (`.dxf` or `.dwg`), different information will be required to write shape entity. If a template file is specified using the `TEMPLATEFILE` keyword, then information about shape entity is extracted from the template file, which requires access to the shape file.

When writing to a DXF file, a shape name and a shape file name is all that is required. The presence of a shape file during translation is not required since there is no information lookup.

When writing to a DWG file and a shape name is given, then a lookup is performed to determine the shape index or shape number from the shape file, which is what DWG stores. For the lookup to be successful, the writer needs access to the shape file, and the specified shape name should be in the shape file. Similarly when writing to DWG, a shape number and shape file is all that is required. If there is no shape index or shape number, then the writer has to perform a lookup from the shape name to the shape index, and for this, access to the shape file is required.

When specifying a shape file, either the full path or just the filename can be specified. If only the shape filename is given, the writer will first look for that shape file in the directory specified by `SHAPE_DIRECTORY` keyword, and if not found then it will look in the directory where the output dataset is being written.

Tip: When an AutoCAD file is output, any shape files it references must be shipped together with the file.

Attribute Name	Contents
<code>autocad_scale</code>	The scale of the shape object for this point. Range: Any real number. Default: 1
<code>autocad_shape_index</code>	This identifies the index of the particular shape within the shape file. A single shape file may contain many different shapes. Range: Any real number > 0 Default: No default
<code>autocad_rotation</code>	The rotation of the shape for this entity. Range: -360.0..360.0 Default: 0
<code>autocad_width_factor</code>	The width factor for the shape. Range: Any real number > 0 Default: 0
<code>autocad_oblique</code>	The oblique angle of the shape. Range: -360.0 ..360.0 Default: 0

Attribute Name	Contents
autocad_big_fontname	The name of the file which contains fonts for large character sets. Range: char[65] Default: NULL
autocad_shape_name	The name of the shape which is being read or written. Range: char[33] Default: No default
autocad_shape_filename	The name of the file in which the shape is defined. Range: char[65]; Default: No default
autocad_shape_rotation	The rotation of the shape definition relative to the shape file specification. Range: Any real number Default: 0
autocad_shape_height	The height of the shape. Range: Any real number Default: 0
autocad_shape_width	The width of the shape. Range: Any real number Default: 1

Leaders

autocad_entity: autocad_leader

Features with `autocad_entity` set to `autocad_leader` are linear features that identify where to place an AutoCAD leader entity. The reader modules returns the following leader-specific attributes. This is currently not supported by the writer. **Note:** Leaders are not currently supported in AutoCAD 2004.

Attribute Name	Contents
autocad_path_type	The type of path the leader follows. The path for a leader is one of <code>autocad_straight_leader</code> in which case the leader is a straight line, or <code>autocad_spline_leader</code> in which case the leader is a spline.
autocad_arrow_head_on	This specifies if the leader line has an arrowhead on it. Range: True False Default: True
autocad_hook_line_on_xdir	This is True if the hook line is in the same direction as the x direction and False if it is not. Range: True False Default: True
autocad_has_hook_line	This is True if the leader has a hook line, and False if it does not. Range: True False Default: True

Attribute Name	Contents
autocad_anno_type	The type of annotation of the leader. Range: autocad_anno_text – annotation is mtext entity, autocad_anno_tolerance – annotation is a tolerance entity, autocad_anno_block – annotation is a block entity, and autocad_anno_none – no annotation with leader. Default: autocad_anno_none
autocad_anno_height	The height of the associated mtext entity. Range: Real64
autocad_anno_width	The width of the associated mtext entity. Range: Real64
autocad_txt_offset_x	The offset of the last leader vertex from the annotation placement point. Range: Real64
autocad_txt_offset_y	The offset of the last leader vertex from the annotation placement point. Range: Real64
autocad_txt_offset_z	The offset of the last leader vertex from the annotation placement point. Range: Real64
autocad_x_dir_x	The x component of a vector indicating the horizontal direction of the text. Range: Any real number
autocad_x_dir_y	The y component of a vector indicating the horizontal direction of the text. Range: Any real number
autocad_x_dir_z	The z component of a vector indicating the horizontal direction of the text. Range: Any real number
autocad_offset_blkinspt_x	The x component of the offset of the last leader vertex from the block reference insertion point. Range: Any real number
autocad_offset_blkinspt_y	The y component of the offset of the last leader vertex from the block reference insertion point. Range: Any real number
autocad_offset_blkinspt_z	The z component of the offset of the last leader vertex from the block reference insertion point. Range: Any real number

Faces

autocad_entity: autocad_face

Features with `autocad_entity` set to `autocad_face` are stored as AutoCAD face entities. Additional attributes are used to define the visibility of the edges of the Face entity. Within the FME, they are stored as four-sided (five vertex) polygons.

Attribute Name	Contents
<code>autocad_edge_1</code>	The visibility of the first edge of the Face. Range: visible invisible Default: visible
<code>autocad_edge_2</code>	The visibility of the second edge of the Face. Range: visible invisible Default: visible
<code>autocad_edge_3</code>	The visibility of the third edge of the Face. Range: visible invisible Default: visible
<code>autocad_edge_4</code>	The visibility of the final edge of the Face. Range: visible invisible Default: visible

Arcs

autocad_entity: `autocad_arc`

This geometry type is stored in an AutoCAD arc entity. Arc features are like ellipse features, except two additional angles control the portion of the ellipse boundary which is drawn. There are several properties of an FME arc geometry that may result in it being written as an AutoCAD entity other than an arc. If this arc geometry is circular and has a sweep angle of 360 degrees it will be stored in an AutoCAD circle entity instead of an arc entity. If this arc geometry is not circular, it will be stored in an AutoCAD ellipse entity.

Tip: The Function `@Arc()` can be used to convert an arc to a line. This is useful for representing arcs in systems that do not support them directly.

Attribute Name	Contents
<code>autocad_primary_axis</code>	The length of the semi-major axis in ground units. Currently the value of the primary axis is always equal to the value of the secondary axis as AutoCAD arcs must be circular. When writing to an AutoCAD file, only the primary axis value is used. Range: Any real number > 0 Default: No default
<code>autocad_secondary_axis</code>	The length of the semi-minor axis in ground units. Currently the value of the primary axis is always equal to the value of the secondary axis as AutoCAD arcs must be circular. When writing to an AutoCAD file, only the primary axis value is used. Range: Any real number > 0 Default: No default

Attribute Name	Contents
autocad_start_angle	Refer to the @Arc (function) in the <i>FME Functions and Factories manual</i> for a detailed definition of start_angle. Range: 0.0..360.0 Default: 0
autocad_sweep_angle	Refer to the @Arc (function) in the <i>FME Functions and Factories manual</i> for a detailed definition of sweep_angle. Range: 0.0..360.0 Default: No default
autocad_rotation	The rotation of the ellipse that defines the arc. The rotation angle specifies the angle in degrees from the horizontal axis to the primary axis in a counterclockwise direction. This value is fixed at 0 as AutoCAD doesn't support rotation of arcs at this time. Range: 0 Default: 0

Traces

autocad_entity: autocad_trace

Features with `autocad_entity` set to `autocad_trace` are stored in and read from drawing files as a 4-coordinate AutoCAD trace entity.

There are no attributes specific to this type of entity.

Solids

autocad_entity: autocad_solid

Features with `autocad_entity` set to `autocad_solid` are stored in and read from drawing files as a 3- or 4-coordinate AutoCAD solid entity.

There are no attributes specific to this type of entity.

Rays

autocad_entity: autocad_ray

Features with `autocad_entity` set to `autocad_ray` are stored in and read from drawing files as a two coordinate line. The reader and writer modules automatically convert the ray to and from its unit vector representation into a line.

There are no attributes specific to this type of entity.

Text Entities

autocad_entity: autocad_text

Features with `autocad_entity` set to `autocad_text` are stored in and read from drawing files as text entities. A text entity is represented by a single coordinate and the following attributes.

Attribute Name	Contents
<code>autocad_text_string</code>	The text string. Range: char[1024] Default: No default
<code>autocad_rotation</code>	The rotation of the text for this entity. Range: -360.0..360.0 Default: 0
<code>autocad_true_type_font</code>	The name of the TrueType font used to display the text string. This attribute is only used by the AutoCAD Writer, since single-line text entities do not have support for TrueType fonts. If this attribute is specified on an <code>autocad_text</code> feature, a multiline text entity will be created and written instead of a single-line text entity. Default: No default
<code>autocad_text_size</code>	The text height. Range: Any real number > 0 Default: 10
<code>autocad_width_factor</code>	The scaling applied in the x direction which makes the text wider or narrower. However, this doesn't affect the weight (i.e., boldness) of the text string. Range: Any real number > 0 and <= 10000 Default: 1
<code>autocad_oblique</code>	The oblique angle of the text. Range: -360.0 ..360.0 Default: 0
<code>autocad_alignment_x</code> <code>autocad_alignment_y</code> <code>autocad_alignment_z</code>	The alignment coordinate of the text. This location is used in conjunction with the justification and the feature location to place the text correctly. This is primarily of use when performing an AutoCAD-to-AutoCAD translation. In order to override the justification of source AutoCAD dataset, you need to remove these alignment attributes from the feature and then set the <code>autocad_justification</code> attribute. This is because when performing an AutoCAD-to-AutoCAD translation, these alignment attributes override the <code>autocad_justification</code> attribute. Range: any 64-bit floating point value Default: x, y, and z value of text alignment point
<code>autocad_big_fontname</code>	The name of the file which contains fonts for large character sets. Range: char[65] Default: NULL
<code>autocad_shape_name</code>	The name of the shape which contains the text font definition. Range: char[33] Default: STANDARD

Attribute Name	Contents
autocad_shape_filename	The name of the file which contains the text font ^a definition. Range: char[65]; Default: txt
autocad_shape_rotation	The angle for the text as defined in shape file. Range: Any real number Default: 0
autocad_shape_height	The height of the text as defined in shape file. Range: Any real number Default: 0
autocad_shape_width	The width of the text as defined in shape file. Range: Any real number Default: 1
autocad_generation	The generation of the text entry. Range: autocad_normal autocad_upside_down autocad_backwards autocad_upsidedown_backwards Default: autocad_normal
autocad_justification	The justification of the text relative to its insert point. Range: autocad_top_left autocad_top_center autocad_top_right autocad_top_middle autocad_top_aligned autocad_top_fit autocad_middle_left autocad_middle_center autocad_middle_right autocad_middle_middle autocad_middle_aligned autocad_middle_fit autocad_bottom_left autocad_bottom_center autocad_bottom_right autocad_bottom_middle autocad_bottom_aligned autocad_bottom_fit autocad_baseline_left autocad_baseline_center autocad_baseline_right autocad_baseline_middle autocad_baseline_aligned autocad_baseline_fit Default: autocad_baseline_left
autocad_tracking_percent	The tracking percent. Only used by the Reader. This attribute will only exist if reading a graphical text entity. Default: N/A since this is a Reader-only attribute.

Attribute Name	Contents
autocad_backwards	Indicates whether the text is backwards. Only used by the Reader. This attribute will only exist if reading a graphical text entity. Default: N/A since this is a Reader-only attribute.
autocad_upside_down	Indicates if the text is upside down. Used only by the Reader. This attribute will only exist if reading a graphical text entity. Default: N/A since this is a Reader-only attribute.
autocad_vertical	Indicates if the text is vertical. Used only by the Reader. This attribute will only exist if reading a graphical text entity. Default: N/A since this is a Reader-only attribute.
autocad_underlined	Indicates if the text is underlined. Used only by the Reader. This attribute will only exist if reading a graphical text entity. Default: N/A since this is a Reader-only attribute.
autocad_overlined	Indicates if the text is overlined. Used only by the Reader. This attribute will only exist if reading a graphical text entity. Default: N/A since this is a Reader-only attribute.

- a. AutoCAD shape files should not be confused with ESRI Shapefiles. AutoCAD shape files hold font and symbol definitions; ESRI Shapefiles hold spatial features.

Multi-Text Entities

autocad_entity: autocad_multi_text

Features with `autocad_entity` set to `autocad_multi_text` are stored in and read from drawing files as text entities. A text entity is represented by a single coordinate and the following attributes.

Attribute Name	Contents
autocad_text_string	The text string. Range: char[1024] Default: No default
autocad_rotation	The rotation of the text for this entity. Range: -360.0..360.0 Default: 0
autocad_text_size	The text height. When reading, this value is calculated using the height of the bounding box of the feature and the estimated number of lines. Range: Any real number Default: 10
autocad_mtext_string	The original formatted mtext string. Writer will use this attribute's value to set the destination mtext entity. When performing an AutoCAD-to-AutoCAD translation, this attribute will ensure that the exact formatting is carried over to the destination. Range: char[1024] Default: None

Attribute Name	Contents
autocad_mtext_text_height	<p>The starting text size of the multi-text feature. This attribute is useful mainly for AutoCAD-to-AutoCAD translations. If this attribute doesn't exist when writing, then a value is calculated for it.</p> <p>Range: Any real number Default: None</p>
autocad_true_type_font	<p>The name of the TrueType font used to display the text string. If this attribute is not specified when writing, the text will still be written, but not using a TrueType font.</p> <p>Default: No default</p>
autocad_linespace_factor	<p>The percentage of default line spacing used.</p> <p>Range: 0.25..4.0 Default: 1</p>
autocad_mtext_ref_rect_width	<p>The width of the reference rectangle in which the text is contained.</p> <p>Range: Any real number > 0</p>
autocad_attach_point	<p>The attach point for the multi-text. Use this attribute to set the justification of Multi-Text entities. Note that there is no autocad_justification attribute for this type of entity.</p> <p>Range:</p> <ul style="list-style-type: none"> autocad_top_left autocad_top_center autocad_top_right autocad_middle_left autocad_middle_center autocad_middle_right autocad_bottom_left autocad_bottom_center autocad_bottom_right
autocad_draw_direction	<p>The direction the text is drawn.</p> <p>Range: autocad_draw_left_to_right autocad_draw_right_to_left autocad_draw_top_to_bottom autocad_draw_bottom_to_top</p>
autocad_box_width	<p>The width of the box which the multi text is located.</p> <p>Range: Any real number > 0</p>
autocad_box_height	<p>The height of the box which the multi text is located.</p> <p>Range: Any real number > 0</p>
autocad_big_fontname	<p>The name of the file which contains fonts for large character sets.</p> <p>Range: char[65] Default: NULL</p>
autocad_shape_name	<p>The name of the shape which contains the text font definition.</p> <p>Range: char[33] Default: STANDARD</p>

Attribute Name	Contents
autocad_shape_filename	The name of the file which contains the text font ^a definition. Range: char[65]; Default: txt
autocad_shape_rotation	The angle for the text as defined in shape file. Range: Any real number Default: 0
autocad_shape_height	The height of the text as defined in shape file. Range: Any real number Default: 0
autocad_shape_width	The width of the text as defined in shape file. Range: Any real number Default: 1
autocad_generation	The generation of the text entry. Range: autocad_normal autocad_upside_down autocad_backwards autocad_upsidedown_backwards Default: autocad_normal
autocad_estimated_num_lines (used by the reader only)	The estimated number of lines in the multi-text feature. This is a calculated value that uses the starting height of the multi-text feature, the height of the bounding box of the feature, and the linespacing factor. Range: Any integer Default: Not applicable since this is a reader-only attribute

a. AutoCAD shape files should not be confused with ESRI Shapefiles. AutoCAD shape files hold font and symbol definitions; ESRI Shapefiles hold spatial features.

Multi-Line

autocad_entity: autocad_multi_line

Multi-line features are linear features that represent a set of parallel lines. They are not capable of representing an arbitrary set of lines such as can be done with GIS systems. When reading a multi-line feature, the FME will output an aggregate of lines thereby hiding all AutoCAD format peculiarities. This entity is only supported by the AutoCAD Reader.

The following attributes are set when reading the multi-lines.

Attribute Name	Content
autocad_scale	The scale of the multi-line. Range: Any real number > 0.t

Attribute Name	Content
autocad_justification	The justification of the multi-line object is set to one of: Range: autocad_top autocad_middle autocad_bottom
autocad_num_verts	The number of vertices in one of the parallel multi-lines. All of the multi-lines have this number of coordinates. The feature thereby has autocad_num_verts * autocad_num_lines: Range: Number of vertices in one line.
autocad_num_lines	The number of lines in the multi-line set.: Range: Number of lines.
autocad_mline_type	The type of multi-line object. Range: autocad_open autocad_closed
autocad_base_x	The x coordinate of the base point of the multi-line entity. Range: Any real number.
autocad_base_y	The y coordinate of the base point of the multi-line entity. Range: Any real number.
autocad_base_z	The z coordinate of the base point of the multi-line entity. Range: Any real number.

Inserts

autocad_entity: autocad_insert

Inserts are point features used in AutoCAD to specify block locations and associated attribution. Inserts are another way in which attribution is stored within an AutoCAD drawing file. The features returned from the AutoCAD reader encapsulate all the information from the AutoCAD insert entity and all attribute entities that are associated with the insert entity. If an attribute passed to the writer is defined by an AutoCAD Attribute Definition in the TEMPLATE file, then the placement of the attribute is taken from the TEMPLATE file unless it is overridden by the attributes shown in the table below.

If the position of the attribute is not specified in a TEMPLATE file and is not specified in the attributes below, then the attributes are placed at the insert location. Apart from the user-defined attributes specified within it, each insert entity also has the following attributes.

Attribute Name	Contents
autocad_xscale	The scale factor for the inserted block in the x direction. Range: Any real number. Default: 1
autocad_yscale	The scale factor for the inserted block in the y direction. Range: Any real number Default: 1

Attribute Name	Contents
autocad_zscale	The scale factor for the inserted block in the z direction. Range: Any real number Default: 1
autocad_rotation	The rotation of the inserted block, counterclockwise from horizontal. Range: -360.0 ..360.0 Default: 0
autocad_number_columns	The column count for the insert. Range: 0..65536 Default: 1
autocad_number_rows	The row count for the insert. Range: 0..65536 Default: 1
autocad_column_distance	The column spacing for the insert. Range: Any real number > 0 Default: 0
autocad_row_distance	The row spacing for the insert. Range: Any real number > 0 Default: 0
autocad_block_name	The name of the block entity which is to be inserted. Range: char[33] Default: FMEBLOCK<block_number> where block_number is some unique positive integer > 0.
autocad_attributes_follow	Indicates if attributes are also to be stored with the insert entity. This must be specified if feature attributes are to be written to the AutoCAD output file. Range: true false Default: true
autocad_attribute_display	Indicates if the attribute values are to be visible or invisible. Range: visible invisible Default: invisible
autocad_attr_def_tag	Indicates the tag used for "Attribute" definition. (read-only) Range: text string
autocad_attr_def_prompt	Specifies the prompt that is displayed when you insert a block containing this attribute definition. (read-only) Range: text string
autocad_attr_def_default	Specifies the default attribute value. (read-only) Range: text string
fme_attr_info{N}.field_name	This list attribute hold the name of the N th attribute. Range: text string

Attribute Name	Contents
fme_attr_info{N}.field_size	This list attribute hold the size of the N th attribute. Range: integer Default: 0
fme_attr_info{N}.field_value	This list attribute holds the value of the N th attribute. Range: text string
autocad_<attr_name>_x autocad_<attr_name>_y autocad_<attr_name>_z fme_attr_info{N}.location_x fme_attr_info{N}.location_y fme_attr_info{N}.location_z	Used when attributes are associated with the insert elements, enabling the location of the attributes to be specified for display purposes. This specifies the exact location where the attributes are to be placed. Note that the FME attributes may contain different values than the AutoCAD attributes. This is because the values of the FME attributes will be modified if the justification is not <code>baseline_left</code> in an attempt to be more useful when translating into or out of other formats that support these FME attributes. Range: any 64-bit floating point value Default: x, y, and z value of insert coordinate (for the AutoCAD attributes) 0,0,0 (for the FME attributes)
autocad_<attr_name>_alignment_x autocad_<attr_name>_alignment_y autocad_<attr_name>_alignment_z fme_attr_info{N}.align_x fme_attr_info{N}.align_y	Used when attributes are associated with the insert elements, enabling the location of the attributes to be specified for display purposes. This specifies the alignment location where the attributes are to be placed. Note that the FME attributes may contain different values than the AutoCAD attributes. This is because the values of the FME attributes will be modified if the justification is not <code>baseline_left</code> in an attempt to be more useful when translating into or out of other formats that support these FME attributes. Range: any 64-bit floating point value Default: 0,0,0 (for the AutoCAD attributes) the value of <code>fme_attr_info{N}.location_[x y z]</code> (for the FME attributes)

Attribute Name	Contents
autocad_<attr_name>_ justification	<p>The justification of the attribute relative to its insertion point (not its alignment point).</p> <p>Range:</p> <ul style="list-style-type: none">autocad_top_left autocad_top_center autocad_top_right autocad_top_middle autocad_top_aligned autocad_top_fit autocad_middle_left autocad_middle_center autocad_middle_right autocad_middle_middle autocad_middle_aligned autocad_middle_fit autocad_bottom_left autocad_bottom_center autocad_bottom_right autocad_bottom_middle autocad_bottom_aligned autocad_bottom_fit autocad_baseline_left autocad_baseline_center autocad_baseline_right autocad_baseline_middle autocad_baseline_aligned autocad_baseline_fit <p>Default: No default</p>

Attribute Name	Contents
fme_attr_info{N}.justification	<p>The justification of the attribute relative to its insertion point (not its alignment point).</p> <p>Range:</p> <ul style="list-style-type: none"> top_left top_center top_right top_middle top_aligned top_fit middle_left middle_center middle_right middle_middle middle_aligned middle_fit bottom_left bottom_center bottom_right bottom_middle bottom_aligned bottom_fit baseline_left baseline_center baseline_right baseline_middle baseline_aligned baseline_fit <p>Default: baseline_left</p>
fme_attr_info{N}.generation	<p>The generation of the Nth attribute.</p> <p>Range:</p> <ul style="list-style-type: none"> normal upside_down backwards upside_down_backwards <p>Default: normal</p>
autocad_<attr_name>_style fme_attr_info{N}.style	<p>The name of the text style for the attribute. For this attribute to be used by the Writer, a template file containing the text style must be specified.</p> <p>Default: No default</p>
fme_attr_info{N}.width_factor	<p>The scaling applied in the x direction, which makes the text wider or narrower. However, this doesn't affect the weight (i.e., boldness) of the text string.</p> <p>Range: Any real number > 0 and <= 10000</p> <p>Default: 1</p>
autocad_<attr_name>_color	<p>The color of the attribute. When reading, this value is always 256, meaning COLOR_BYLAYER.</p> <p>Default: The color of the layer on which the insert is placed (not the same as COLOR_BYLAYER).</p>

Attribute Name	Contents
fme_attrib_info{N}.color.red fme_attrib_info{N}.color.green fme_attrib_info{N}.color.blue	These list attributes hold the color of the N th attribute in RGB values, ranged between 0.0 and 1.0. Range: real number 0.0-1.0 (inclusive) Default: No default
fme_attrib_info{N}.color.source	The source of the color. If the attribute's color does not come from the block or layer, then it is given the value <code>explicit</code> . Range: <code>use_layer</code> <code>use_block</code> <code>explicit</code> Default: <code>explicit</code>
autocad_<attr_name>_rotation fme_attrib_info{N}.rotation	This specifies the rotation of the attribute, measured in degrees. Range: degree of rotation measured counter-clockwise from the horizontal. Default: 0
fme_attrib_info{N}.oblique	This specifies the obliquing angle of the attribute, which causes the text to lean to the right or left. Range: -360.0 ..360.0 Default: 0
autocad_<attr_name>_height fme_attrib_info{N}.height	This specifies the height of the attribute in ground units. Range: any 64-bit floating point value. Default: 1
autocad_<attr_name>_attribute_flag fme_attrib_info{N}.attribute_flag	A bit-coded value. Values can be combined by using addition. Possible values are: 1 = The attribute is invisible (does not appear). 2 = The attribute contains a constant value. 4 = Verification is required on input of this attribute. 8 = Attribute is preset (no prompt during insertion). If the <code>fme_attrib_info{N}.attribute_flag</code> attribute is used, the invisibility bit will get overwritten by the <code>fme_attrib_info{N}.isVisible</code> attribute. If the <code>autocad_<attr_name>_attribute_flag</code> attribute is used, the invisibility bit will get overwritten by <code>autocad_attribute_display</code> if the attributes <code>autocad_visible_attributes{}</code> and <code>autocad_invisible_attributes{}</code> are specified. Default: 0 if no attribute definitions exist for the attribute (i.e., when <code>DEFAULT_ATTR_STORAGE</code> keyword or <code>autocad_attributes</code> attribute is set to <code>insert_attributes</code>), or value from the attribute definition if attribute definitions are used.
autocad_<attr_name>_layer fme_attrib_info{N}.layer	The layer on which the attribute definition, corresponding to this attribute, was created. Default: none

Attribute Name	Contents
<code>fme_attr_info{N}.isVisible</code>	These list attributes indicate whether or not the N th attribute should be displayed. Range: TRUE FALSE Default: FALSE
<code>autocad_visible_attributes{}</code>	The list of attributes that are set to be visible. This is a list attribute.
<code>autocad_invisible_attributes{}</code>	The list of attributes that are set to be invisible. This is a list attribute.
<code>autocad_<attr_name>_x_off</code> <code>autocad_<attr_name>_y_off</code> <code>autocad_<attr_name>_z_off</code>	Used when attributes are associated with the insert elements enabling the location of the attributes to be specified for display purposes. This specifies the offset from the location of the insert. Range: any 64-bit floating point value Default: x, y and z value of insert coordinate

Dimensions

autocad_entity: autocad_dimension

Dimensions are aggregate features used in AutoCAD to specify dimensions within an AutoCAD drawing. The dimension features have the attributes described below.

Rotated (linear) dimensions:

- The first extension line is specified by `defpt2`.
- The second extension line is specified by `defpt3`.
- The dimension line is specified by `dimlinedefpt`.

Angular dimensions:

- `defpt2` and `defpt3` are the endpoints of the first extension line.
- `dimlinedefpt` and `defpt4` are the endpoints of the second extension line.
- `arcdefpt` specifies the dimension line arc.

Angular 3-point dimensions:

- `defpt4` is the vertex of the angle.
- `defpt2` is the endpoint of the first extension line.
- `defpt3` is the endpoint of the second extension line.
- `dimlinedefpt` specifies the arc for the dimension line.

Diameter dimensions:

- `defpt4` is the point selected on the circle or arc being dimensioned.
- `dimlinedefpt` is the point on the circle exactly across from the selection point.

Ordinate dimensions:

- `defpt3` is the point which was selected.
- `defpt4` is the point indicating the endpoint of the leader.

Attribute Name	Contents
autocad_actual_measurement	The scale factor for the inserted block in the x direction. Range: Any real number Default: 1
autocad_arc_defpt.x autocad_arc_defpt.y autocad_arc_defpt.z	This defines the dimension arc for an angular dimension. This is equivalent to the 16,26,36 group in DXF. Range: Any real number Default: none
autocad_arc_defpt2.x autocad_arc_defpt2.y autocad_arc_defpt2.z	Definition Point. This is equivalent to the 13,23,33 group in DXF. Range: Any real number Default: none
autocad_arc_defpt3.x autocad_arc_defpt3.y autocad_arc_defpt3.z	Definition Point. This is equivalent to the 14,24,34 group in DXF. Range: Any real number Default: none
autocad_arc_defpt4.x autocad_arc_defpt4.y autocad_arc_defpt4.z	Definition Point. This is equivalent to the 15,25,35 group in DXF. Range: Any real number Default: none
autocad_dimension_attach_point	This is the attachment point of dimension, as defined 1 = Top Left 2 = Top Center 3 = Top Right 4 = Middle Left 5 = Middle Center 6 = Middle Right 7 = Bottom Left 8 = Bottom Center 9 = Bottom Right
autocad_dim_arrowhead_pt_<number>.x autocad_dim_arrowhead_pt_<number>.y	These attributes, provided by the reader but not used by the writer, describe the x, y coordinates of the vertex, on each arrowhead, that could be called the "tip of the arrow." <number> is some number between 1 and the number of arrowheads in the dimension. If there are no arrowheads, then this attribute will not be supplied.
autocad_dimension_flag	This is the raw value from the AutoCAD file that indicates the type of dimension. See autocad_dimension_type for the decoded version of this.

Attribute Name	Contents
autocad_dimension_style_name	The name of the dimension style used. When using the AutoCAD writer, it is important that the dimension style used is defined in the template file; otherwise, no style will be set and the dimension text will not be displayed. Even if the standard dimension style is used, the template file must hold a definition for it.
autocad_dimension_type	This indicates the type of the autocad dimension. Possible values are: autocad_rotated autocad_aligned autocad_angular autocad_diameter autocad_radius autocad_angular3Pt autocad_ordinate autocad_xordinate
autocad_leader_length	This is the length of the dimension leader line.
autocad_linespace_style	This is the style of the line spacing. It is either 1 (at least) or 2 (exact).
autocad_linespace_factor	The percentage of default line spacing used. Range: 0.25..4.0 Default: 1
autocad_rotation_angle	The rotation angle of the dimension.
autocad_text_midpoint.x autocad_text_midpoint.y autocad_text_midpoint.z	The midpoint of the text.
autocad_text_rotation	The rotation of the dimension text.
autocad_text_size	The size of the text in ground units.
autocad_text_string	The dimension text value.
autocad_ucs_xangle	The angle of the ucs (user coordinate system) when the dimension was created.

Group

autocad_entity: autocad_group

Group features are features with no geometry. This feature merely identifies the feature handles that are part of the group. This entity is only supported by the AutoCAD Reader. The feature type is set to `AUTOCAD_GROUP`. The following attributes are set when reading groups.

Attribute Name	Content
autocad_group_description	The descriptive name of the group. Range: Character string.

Attribute Name	Content
<code>autocad_group_name</code>	Name of the group. Range: Character string.
<code>autocad_group_anonymous</code>	Whether or not group is anonymous Range: yes/no.
<code>autocad_group_accessible</code>	Whether or not group is accessible. Range: yes/no.
<code>autocad_group_num_entities</code>	Number of entities in the group. Range: Numeric.
<code>autocad_group_selectable</code>	Whether or not group is selectable. Range: yes/no.
<code>autocad_entity_handle{}</code>	The list attribute which contains the hexadecimal values of the entities that make up the group. Range: Hexadecimal value.

Hatches

autocad_entity: `autocad_hatch`

Hatch features represent AutoCAD hatch entities. They are composed of two dimensional boundary loops that define areas that can be filled with line patterns or color gradients. The loops of each hatch are closed, simple, continuous, and are not self-intersecting except at their endpoints.

The AutoCAD Reader creates features of various geometry depending on the geometry of the loops that compose them. Features created may be of polygon, donut or aggregate geometry, where the aggregates may contain either donuts and polygons or just ordered polygons depending on the usage of the `PRESERVE_COMPLEX_HATCHES` keyword. The AutoCAD reader will also preserve polyline bulge information.

Note: The AutoCAD Reader has the following limitations when reading hatch features: associative hatches are not preserved, one pattern is allowed per hatch, only one or two color gradients are supported, unclosed hatch boundary loops are closed, splines are not supported for hatch boundary loops, and elliptical and circular arcs in boundary loops are stroked.

The AutoCAD Writer has the following limitations when writing hatch features: hatch features must be closed area features of polygon, donut, or aggregate geometry according to how the AutoCAD Reader created them. The AutoCAD writer will try to reconstruct polyline bulge information from the feature, but closed and elliptical arcs will be stroked into line segments.

The following attributes may be set when reading hatches.

Attribute Name	Content
<code>autocad_hatch_associative</code>	The flag indicating if the hatch is associative. Range: 0 for no 1 for yes Default: 0

Attribute Name	Content
autocad_hatch_complex_mode	<p>The flag that represents whether the hatch feature was created to preserve complex hatches. This indicates how the structure of loops is created.</p> <p>Range: 0 for no 1 for yes</p> <p>Default: 0</p>
autocad_hatch_gradient_angle	<p>The angle of the gradient fill for the hatch feature in degrees.</p> <p>Range: any 64-bit floating point value.</p> <p>Default: 0</p>
autocad_hatch_gradient_color1	<p>The first color used to interpolate a two color gradient fill. It the single color used in a one color gradient fill. Specified as a character string of comma-separated red, green and blue values.</p> <p>Range: 0..255,0..255,0..255.</p> <p>Default: None.</p>
autocad_hatch_gradient_color2	<p>The second color used to interpolate a two color gradient fill. Specified as a character string of comma-separated red, green and blue values.</p> <p>Range: 0..255,0..255,0..255</p> <p>Default: None.</p>
autocad_hatch_gradient_name	<p>The name of the gradient. Must be a predefined value for predefined gradients. This is mandatory for hatches with gradients.</p> <p>Range: Curved Cylindrical Hemispherical Linear Spherical Inverted Curved Inverted Cylindrical Inverted Hemispherical Inverted Spherical</p> <p>Default: None</p>
autocad_hatch_gradient_one_color_mode	<p>The flag indicating whether only one color should be used in gradient calculation. Gradients can be two color, or one color with a luminance value set by autocad_hatch_shade_tint_value.</p> <p>Range: 0 for no 1 for yes</p> <p>Default: 0</p>
autocad_hatch_gradient_shift	<p>The interpolation value between the default and shifted values of the gradient's definition.</p> <p>Range: 0..1</p> <p>Default: None</p>
autocad_hatch_gradient_type	<p>The type of the gradient. Currently this is set to 0 for predefined gradient. In the future a value of 1 may be supported for user-defined gradients. This is mandatory for hatches with gradients.</p> <p>Range: 0 for pre-defined 1 for user-defined</p> <p>Default: None</p>

Attribute Name	Content
<p>autocad_hatch_loop{}.autocad_hatch_bulge{}</p> <p>Applicable only with classic geometry.</p>	<p>The list of bulge values for polyline bulge arcs in each of a list of hatch boundary loops. The list of bulge values parallel the vertices in each loop. A bulge value represents the tangent of 1/4 the included angle in the arc measured counterclockwise. A value of 0 represents a line, and a value of 1 represents a semicircle.</p> <p>Range: 0..1 Default: 0</p>
<p>autocad_hatch_loop{}.autocad_hatch_bulges_present</p> <p>Applicable only with classic geometry.</p>	<p>The flag indicating if polyline bulge arcs exist in each of a list of hatch boundary loops.</p> <p>Range: 0 for no 1 for yes Default: 0</p>
<p>autocad_hatch_loop{}.autocad_hatch_type</p> <p>Applicable only with classic geometry.</p>	<p>The type of the hatch loop. This is an integer representing the addition of applicable type flags.</p> <p>Range: numeric value Default: None</p>
autocad_hatch_object_type	<p>The type of the hatch. This is mandatory for hatches with gradients and patterns other than SOLID.</p> <p>Range: 0 for classic hatch 1 for color gradient Default: 0</p>
autocad_hatch_origin_point_x	<p>The x-axis coordinate of the origin of the hatch in world coordinates.</p> <p>Range: any 64-bit floating point value Default: 0</p>
autocad_hatch_origin_point_y	<p>The y-axis coordinate of the origin of the hatch in world coordinates.</p> <p>Range: any 64-bit floating point value Default: 0.</p>
autocad_hatch_pattern_angle	<p>The angle of the pattern fill for the hatch feature in degrees.</p> <p>Range: any 64-bit floating point value Default: 0</p>
autocad_hatch_pattern_double	<p>The flag indicating if the hatch pattern is doubled by adding a second set of lines at 90 degrees to the first. This is only used for user-defined patterns.</p> <p>Range: 0 for no 1 for yes Default: 0</p>
autocad_hatch_pattern_name	<p>The name of the pattern. May be a predefined value for predefined patterns, a predefined value for custom patterns, or any string for user-defined patterns.</p> <p>Range: Pre-defined or custom pattern name any string Default: SOLID</p>
autocad_hatch_pattern_scale	<p>This represents the scaled size of the pattern for predefined and custom-defined patterns.</p> <p>Range: positive floating point value > 0 Default: 1</p>

Attribute Name	Content
autocad_hatch_pattern_space	This represents the space between the parallel lines of the hatch pattern. This is only used for user-defined patterns. Range: positive floating point value > 0 Default: 1
autocad_hatch_pattern_type	The type of the hatch pattern. Custom-defined patterns are pre-created patterns that must be present in the location of the predefined patterns. This is mandatory for hatches with patterns other than SOLID. Range: 0 for user-defined 1 for pre-defined 2 for custom-defined Default: 1
autocad_hatch_pixel_size	The size of pixels for intersection and ray casting when drawing the hatch. Range: positive floating point value > 0 Default: 1
autocad_hatch_shade_tint_value	The luminance value of the hatch. If the hatch has a gradient and is using one color mode, this value is applied to the first color. Range: 0.0..1.0 Default: 0
