

Intergraph MGE Reader/Writer

FORMAT NOTES:

- This format is not supported by FME Base Edition.
- The **Intergraph MGE Reader/Writer** is nearly identical to the (Bentley) **MicroStation GeoGraphics Reader/Writer**. The only difference is that by default, the <ReaderKeyword> for the GeoGraphics reader is *GG*. **This chapter contains information that is applicable to both formats.**

The Intergraph Modular GIS Environment (MGE) Reader and Writer modules provide the FME with the ability to read and write design files and their associated databases.

Overview

MGE uses standard MicroStation *elements* to represent the graphical portions (geometry) of the geographic map objects. These elements are identical to the design file elements used by the Design File Reader/Writer. Any geometry enhancements made to the Design File Reader/Writer benefit the MGE Reader/Writer.

The non-graphical aspects of a map object are defined by linking an MGE *feature* to elements in a database. Features are linked to elements in the database via pairs of entity and `mslink` numbers. Feature definitions are stored in a special table within the project's database. Each row of the feature table defines a feature, providing a name, feature code, feature type, display attributes (to be applied to elements linked to that feature), and possibly a pointer to a table of non-graphical attributes to be associated with each instance of the feature.

Tip: A *feature* in Intergraph MGE is closer in nature to a *feature type* in FME terminology. In this chapter, we refer to FME's features using the term "FME features" to differentiate it from MGE's features.

The following table describes the general database tables used. Other tables not listed are non-graphical user attribute tables. The `mscatalog` and feature tables are mandatory for the database used.

Table Name	Description
<code>mscatalog</code>	The key table to determine where there are more attributes to the feature in the other tables.
<code>feature</code>	This table holds all the feature types and the general feature information.
<code>maps</code>	The table contains the corresponding map information associated with the feature.
<code>category</code>	This table holds information about the categories associated with the feature.
<code>ugfeature</code> , <code>ugcategory</code> , <code>ugjoin_cat</code> , <code>ugmap</code> , <code>ugcommand</code> , <code>ugtable_cat</code>	Extended tables of the Intergraph Design format table that hold additional information for MGE.

The MGE Reader and Writer use symbolic names for the design element types rather than numeric values. This greatly simplifies element type specification. The following table maps the design element type number to its corresponding FME feature `igds_type` value that is used by the MGE Reader and Writer. Subsequent sections describe the handling of each of these element types in detail.

IGDS Element Type	FME <code>igds_type</code>
2	<code>igds_cell</code>
3	<code>idgs_point</code>
4,12	<code>idgs_line</code>
6,14	<code>idgs_shape</code>
7	<code>idgs_text_node</code>
11,12	<code>idgs_curve</code>
15	<code>idgs_ellipse</code>
16	<code>idgs_arc</code>
17	<code>idgs_text</code>
7,17	<code>idgs_multi_text</code>
2,6,14	<code>idgs_solid</code>

MGE Quick Facts

Format Type Identifier	MGE
Reader/Writer	Both
Licensing Level	Professional
Dependencies	None
Dataset Type	File
Feature Type	GeoGraphics Feature Name
Typical File Extensions	.dgn, .cad
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
Enhanced Geometry	Yes
Geometry Type	igds_type

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		

Reader Overview

The MGE Reader extracts all elements from the design files, one at a time, via the help of the Design File Reader. For each element, it finds every attached feature and passes each on to the rest of the FME for processing, along with any corresponding non-graphical attributes. Complex elements are extracted as single FME features. If the element has any attribute linkages attached to it, these are read in and added as attributes to the FME feature created.

When the MGE Reader encounters an element type it does not recognize, it is processed as an “unlinked” element type.

Reader Directives

The following directives are processed by the MGE reader. The suffixes shown will be prefixed by the current <ReaderKeyword> in the mapping file. By default, the <ReaderKeyword> for the MGE reader is MGE. Please see the *Bentley MicroStation Design Reader/Writer* on page 271 for other keyword suffixes.

DATASET

Required/Optional: *Required*

Contains the directory name of the input MGE files.

Workbench Parameter: [<WorkbenchParameter>](#)

SERVER_TYPE

Required/Optional: *Required*

Contains the server type for the input data.

Workbench Parameter: [<WorkbenchParameter>](#)

SERVER_NAME

Required/Optional: *Required*

Contains the server name for the input data.

Workbench Parameter: [<WorkbenchParameter>](#)

USER_NAME

Required/Optional: *Optional*

Contains the user name for the database. May be required by the database being used.

Workbench Parameter: [<WorkbenchParameter>](#)

PASSWORD

Required/Optional: *Optional*

Contains the password for the database. May be required by the database being used.

Workbench Parameter: [<WorkbenchParameter>](#)

DATABASE_NAME

Required/Optional: *Optional*

Contains the database name. May be required by the database being used.

Workbench Parameter: [<WorkbenchParameter>](#)

DEF

Required/Optional: *Required*

The `<ReaderKeyword>_DEF` must be used to define a feature before elements of that feature may be read. It allows complete specification of a MGE feature. The definition of a feature consists of:

- A name for the feature.
- A set of database attributes associated with each element attached to the feature. All database attributes of a given feature are stored in a single table; elements attached to that feature contain linkages into this attribute table.
- A set of properties which apply to all map objects belonging to that feature. (These properties, described in greater detail below, define the graphic specifications applied to elements attached to the feature, as well as specifying the feature's feature code, category, and other feature-specific attributes.)

The syntax for the definition line is:

```
<ReaderKeyword>_DEF "<featureName>" \
  [<attrName> <attrType>]* \
  [<featPropertyName> "<featPropertyValue>"]*
```

The feature name must follow the conventions for MGE feature names. The letter case of alphabetic characters in the names is insignificant, as all names are converted to lowercase for internal use by the MGE Reader.

There may be zero or more attributes associated with a feature. Like feature names, the case of attribute names is insignificant. (Attribute names are conventionally specified in lowercase to more easily distinguish them from the type properties.) The following table shows the supported attribute types:

Field Type	Description
<code>char (<width>)</code>	Character fields store fixed length strings. The <code>width</code> parameter controls the maximum of characters that can be stored by the field. No padding is required for strings which are shorter than this width.
<code>date</code>	Date fields store dates as character strings with the format <code>YYYYMMDD</code> .
<code>number (<width>, <decimals>)</code>	Number fields store single and double precision floating point values. The <code>width</code> parameter is the total number of characters allocated to the field, including the decimal point. The <code>decimals</code> parameter controls the precision of the data, and is the number of digits to the right of the decimal.
<code>logical</code>	Logical fields store TRUE/FALSE data. Data read or written from/to such fields must always have a value of either <code>true</code> or <code>false</code> .

The MGE reader only requires the specification of those attributes which are of interest to the translation process; there is no need to mention any attributes which are not used. Further, every attribute which is specified in the mapping file is verified against those associated with the feature being defined, to ensure that it exists and that its type is compatible.

There are a number of properties that may be specified to define the feature of interest. Every specified property must agree with the existing MGE feature, or an error will result. No property specifications are required when reading from a MGE project; however, in order for feature attributes not belonging to the design file to be written, a `GG_ATTR_TABLE` needs to be defined (the default table name is the feature name).

UNITS

Required/Optional: *Optional*

The `<ReaderKeyword>_UNITS` directive controls the conversion between UORs in the design file and FME coordinates. There are three possibilities, outlined in the table below. If no `UNITS` directive is specified, then `GG_MASTER_UNITS` is the default.

GG_UNITS Value	Description
<code>GG_MASTER_UNITS</code>	The UORs read from the design file will be converted into master units , according to the conversion factor defined in MicroStation's terminal control block (TCB), before being stored in an FME feature. This is the default.
<code>GG_SUB_UNITS</code>	The UORs read from the design file will be converted into subunits , according to the conversion factor defined in MicroStation's TCB, before being stored in an FME feature.
<code>GG_UORS</code>	The UORs read from the design file will be stored directly in an FME feature, with no conversion.

Workbench Parameter: [<WorkbenchParameter>](#)

FEATURE_TABLE_NAME

The `<ReaderKeyword>_FEATURE_TABLE_NAME` directive identifies the name of the feature table. Use this directive only if you have a feature table with the name other than `feature`.

Workbench Parameter: [<WorkbenchParameter>](#)

CATEGORY_TABLE_NAME

The `<ReaderKeyword>_CATEGORY_TABLE_NAME` directive identifies the name of the feature table. Use this directive only if you have a category table with the name other than `category`.

Workbench Parameter: [<WorkbenchParameter>](#)

Writer Overview

The MGE Writer writes all elements to the current master file. It extracts the conversion parameters required to translate coordinates from feature coordinate units to internal Units of Resolution (UORs). It also determines the dimension of the master file.

When writing to GG/MGE, one of the Data Source types from ODBC, MDB or ORACLE has to be selected. Against the Database Server Name, an empty database has to

specified which the writer uses to write all the tables like mscatalog, feature, etc. For instance, with Access (MDB) as the data source, an empty database *.mdb file would suffice. When an existing database is used, new records are appended to the existing tables. The username and password are optional and may or may not apply in every case.

A cell library file may optionally be used by the MGE Writer. Cell libraries contain named symbol definitions which can be used to depict point features. If a cell library is specified, the MGE Writer reads in all the cell definitions for later when cell features are output. The MGE Writer can use either 2 or 3 dimension cell libraries, and will automatically convert the cell definitions to be of the correct dimension for output.

The MGE Writer then outputs each FME feature it is given. Most often, a single FME feature corresponds to a single design element. However, some of the IGDS element types cause several elements to be output as a complex unit (with the complex bit turned on). This occurs when a multi-line text object, a cell, or a closed shape or linear feature with more than 101 coordinates is output. The MGE Writer hides all of the details of complex element output.

Tip: Since coordinates in design files are ultimately stored as integer UORs, it is possible for precision to be lost or overflow to occur when they are output. Care must be taken to ensure that the conversion parameters in the seed file preserve the data precision and range.

Writer Directives

By default, the <WriterKeyword> for the MGE Writer is GG, so a typical mapping file fragment configuring the MGE Writer would be:

```
GG_CELL_LIBRARY /usr/data/dgn/cartog.cel
```

The <WriterKeyword>_UNITS directive is interpreted the same as the MGE Reader's <ReaderKeyword>_UNITS directive. Likewise, the structure of the <WriterKeyword>_DEF directive is the same as that of <ReaderKeyword>_DEF. However, if the definition line is defining a new feature (that is, one that does not yet exist in the MGE project's database), all non-null feature parameters must be specified, and all user attributes for the feature must be included in the definition.

The directives listed below are prefixed by the current <WriterKeyword> in a mapping file.

Please see the Design File Reader/Writer for other keyword suffixes.

WAREHOUSE_VERSION

Required/Optional: *Optional*

Range: 4, 5 or 6

Default: 5

Workbench Parameter: <WorkbenchParameter>

SERVER_TYPE

Required/Optional: *Required*

Contains the server type for the output data.

[Workbench Parameter: <WorkbenchParameter>](#)

SERVER_NAME

Required/Optional: *Required*

Contains the server name for the output data.

[Workbench Parameter: <WorkbenchParameter>](#)

USER_NAME

Required/Optional: *Optional*

Contains the user name for the database. May be required by the database being used.

[Workbench Parameter: <WorkbenchParameter>](#)

PASSWORD

Required/Optional: *Optional*

Contains the password for the database. May be required by the database being used.

[Workbench Parameter: <WorkbenchParameter>](#)

DATABASE_NAME

Required/Optional: *Optional*

Contains the database name. May be required by the database being used.

[Workbench Parameter: <WorkbenchParameter>](#)

UNITS

Required/Optional: *Optional*

Specifies how FME feature coordinates will be interpreted and converted into UORs. See the documentation under the MGE Reader for details.

[Workbench Parameter: <WorkbenchParameter>](#)

IMMEDIATE_WRITE

Required/Optional: *Optional*

Specifies if the database is written immediately when needed (yes) or not (no).

[Workbench Parameter: <WorkbenchParameter>](#)

TRANSACTION_INTERVAL

Required/Optional: *Optional*

The number of features that are to be in a single transaction before the FME performs a commit operation when writing to the database.

[Workbench Parameter: <WorkbenchParameter>](#)

DEF**Required/Optional:** *Required*

Defines an MGE feature. Each feature must be defined before it can be written. The definition specifies the characteristics which make up the MGE feature. Additionally, it specifies the non-graphical attributes which will appear in the correlation section for the feature. There may be many DEF lines, one for each file to be read. See section DEF on page 1234 for more details.

The following table summarizes the supported feature properties:

- Entries under the **Workbench Parameter Name** column are the keyword descriptions as seen in Workbench.
- Entries under the **Mapping File Property Name** column correspond to the actual DEF line parameters used in the mapping file.

Note: Regardless of type, all feature properties' values are specified in quotation marks.

Workbench Property Name	Mapping File Property Name	Description	Type	Required/Optional
Feature Code	GG_FEAT_CODE	A set of dot-separated integers which define numerically the feature hierarchy structure.	char (10)	Optional
Category Number	GG_CATEGORY	The name of the category containing the feature; this must be one of the categories defined on the MGE project.	char (32)	Optional
User Attribute Table Name	GG_ATTR_TABLE	The name of the database table defining the non-graphical attributes for the feature. If there are no such features, this value should be the null string (" "). This is required in order to write out the attributes that are not part of the design file. The default is the feature name.	char (32)	Required
Element Type	GG_ELEMENT_TYPE	The type of elements tagged with this feature.	integer	Optional

Workbench Property Name	Mapping File Property Name	Description	Type	Required/Optional
Element Lock	GG_ELOCK	The strength of enforcement of the above element type (0 => interest - attach to any element type; 1 => similar - may attach only to "comparable" element type; 2 => exact - element type must exactly match GG_ELEMENT_TYPE).	char(12)	Optional
Geometry Type	GG_FEAT_TYPE	The geometry type of the feature type. (-=> undefined - default; 1 => point; 2 => line; 3 => area boundary; 4 => area centroid; 5 => label).	integer	Optional
Feature Level	GG_LEVEL	The level number applied to elements attached to this feature.	integer	Optional
Feature Style or Line Code	GG_STYLE	The style or line code applied to elements attached to this feature.	integer	Optional
Feature Weight	GG_WEIGHT	The weight applied to elements attached to this feature.	integer	Optional
Feature Color	GG_COLOR	The color applied to elements attached to this feature.	integer	Optional
Feature Angle	GG_ANGLE	Angle at which feature is set.	float	Optional
Feature Height	GG_HEIGHT	Height of feature, applied to attached text and node elements.	float	Optional
Feature Width	GG_WIDTH	Width of feature, applied to attached text and node elements.	float	Optional
Line Spacing	GG_LINE_SPACING	Line spacing of text nodes attached to feature.	float	Optional
Line Length	GG_LINE_LENGTH	Line length of text nodes attached to feature.	integer	Optional
Feature Font	GG_FONT	Font used for attached text and node elements.	integer	Optional
Feature Symbol	GG_SYMBOL	Feature symbol used to form symbol text.	char(1)	Optional

Workbench Property Name	Mapping File Property Name	Description	Type	Required/Optional
Feature Justification	GG_JUSTIFICATION	Direction at which feature text is justified.	integer	Optional
Active Stream Delta	GG_STREAM_DELTA	Active stream delta.	float	Optional
Active Stream Tolerance	GG_STREAM_TOL	Active stream tolerance.	float	Optional
Feature Snap Type	GG_SNAP_TYPE	Type of feature snap.	integer	Optional
Feature Snap Tolerance	GG_SNAP_TOL	Feature snap tolerance.	integer	Optional
Database Linkage Mode	GG_NEW_DUP	Database linkage mode (-1 => NO_LINK; 0=> NEW_LINK; 1=>DUP_LINK).	integer	Optional
Feature Class	GG_CLASS	Feature class.	integer	Optional
Feature Priority	GG_FEAT_PRIO	Priority of feature relative to other features in design file.	integer	Optional
Database Links to Info-mode	GG_INFO_MODE	Sets database links to info-mode.	integer	Optional
Displayable Attribute Type	GG_DAS_TYPE	Displayable attribute type.	integer	Optional
Display Priority	GG_DISPLAY_PRIO	Priority of display for elements with multiple feature tags.	float	Optional

FEATURE_TABLE_NAME

Required/Optional: *Optional*

Specifies the name of the feature table to be written. This defaults to the name `feature`.

Workbench Parameter: [<WorkbenchParameter>](#)

CATEGORY_TABLE_NAME

Required/Optional: *Optional*

Specifies the name of the category table to be written. This defaults to the name `category`.

Workbench Parameter: [<WorkbenchParameter>](#)

LINKAGE_TYPE

Required/Optional: *Optional*

Specifies the type of database linkages that will be attached to features written to MGE layers. The value is a character string. If this directive is not specified, the MGE writer defaults to creating database linkages of type "dmr:s".

Workbench Parameter: [<WorkbenchParameter>](#)

MANGLE_DBCS_TEXT

Required/Optional: *Optional*

Controls whether or not double-byte-character set text is mangled when text strings are written. Microstation uses special header bytes to single DBCS text. If this directive is set to `YES` in the mapping file, then these special bytes will be output when a DBCS text string is encountered. The default value is `NO`. Note that the IGDS reader automatically de-mangles DBCS text.

Workbench Parameter: [<WorkbenchParameter>](#)

COMPRESS_AT_END

Required/Optional: *Optional*

Tells the writer to compact the .mdb Access database file. This compresses the file size after all the writing is done.

Range: `YES` | `NO`

Default: `NO`

Workbench Parameter: [<WorkbenchParameter>](#)

SPLIT_BIG_DGN7_FILES

Required/Optional: *Optional*

Note: This directive applies to the V7 writer only.

Allows user to split Version 7 DGN files bigger than 32 MB. Note that this directive can be manually set to `no` in the mapping file.

Range: `YES` | `NO`

Default: `YES`

Workbench Parameter: [<WorkbenchParameter>](#)

COMPRESS_AT_END

This statement instructs the FME to compact the database after all writing has been done. This makes use of the existing MDB database option to compact. The compact operation compresses the output database to a small size on disk.

Example:

```
COMPRESS_AT_END Yes
```

Workbench Parameter: [<WorkbenchParameter>](#)

MDB_VERSION

This statement instructs FME to set the version of the output Microsoft Access file version. Access file versions 97 and 2000 are the supported types. By default, an Access 2000 file is created.

Example:

```
MDB_VERSION 97
```

Workbench Parameter: [<WorkbenchParameter>](#)

Feature Representation

In addition to the generic FME feature attributes that FME Workbench adds to all features (see *About Feature Attributes* on page 7), this format adds the format-specific attributes described in this section.

Each design element may be attached to one or more MGE features. The FME feature consists of a design element, a single MGE feature, and the attributes from the row of the feature's attribute table which corresponds to the element. Special FME feature attributes are used to hold design element parameters. The MGE Writer will use these attribute values as it fills in an element structure during output. The MGE Reader will set these attributes in the FME feature it creates for each element it reads.

Tip: By using a common value for graphic group value, several otherwise separate elements may be tied together into a *logical* super-element for later processing by application programs.

The FME considers the MGE feature name to be the *FME feature type* of an element in a MGE design file. Each MGE element, regardless of its geometry type, shares a number of other parameters, as described in the following table. Please see the Design File Reader/Writer feature representation for the parameters specific descriptions to each of the supported element types.

