GXF

Grid eXchange File

Revision 3.0
May 2, 2001

SEG Gravity and Magnetic Committee
2000 Data Standards Sub-Committee

Dale Bird (dale@birdgeo.com)
Chuck Campbell (campbell@neosoft.com)
Richard Hansen (rohansen@prj.com)
David Pratt (dave@encom.com.au)
Ian MacLeod (ian.macleod@geosoft.com)
Alan Reid (alan@reid-geophys.co.uk)

CONTENTS

1. INTRODUCTION

GXF (**G**rid e**X**change **F**ile) is a standard ASCII file format for exchanging gridded data among different software systems. Software that supports the GXF standard will be able to import properly formatted GXF files and export grids in GXF format.

GXF Revision 1 was drafted by Ian MacLeod at the request of the Canadian Exploration Geophysical Society (KEGS) of Toronto. It found significant usage in the mining and environment sectors. Revision 2.00 was undertaken in conjunction with the Australian Society of Exploration Geophysicists (ASEG) who adopted it as their standard. The main advance of Revision 2 over Revision 1 was the addition of data compression through the use of base-90 numbers and simple repeat value compression. These extensions made GXF more practical for exchanging large gridded data sets. GXF Revision 2 was adopted by the Gravity/Magnetics Committee of the Society of Exploration Geophysicists (SEG) in November 1997, with the urging that the standard be further extended to describe the projected or Geographic Coordinate System of a grid.

This document defines GXF-3, which implements the exchange of 2-D coordinate system information. Coordinate systems in GXF-3 are described using names from the European Petroleum Survey Group (EPSG) geodetic data tables. Petrotechnical Open Software Corporation (POSC) coordinate systems are also based on EPSG naming conventions, which make GXF-3 POSC compatible. Where differences exist between EPSG and POSC, both EPSG and POSC usage is permitted in GXF-3. This allows GXF-3 files to conform to the most recently published EPSG tables and POSC.

Although GXF-3 uses EPSG names, all coordinate system parameters are also explicitly specified as part of the coordinate system definition. Where a particular parameter or name is not defined by EPSG, any appropriate name can be used and such names must begin with the "*" character. This is an important feature of GXF because it allows GXF to support grids that use coordinate systems that are either not defined in EPSG or known to the GXF writer at the time the GXF is created. It is not uncommon for exploration data to use ad-hoc coordinate systems, and support for such systems has been a requirement in the design of GXF-3.

Wherever EPSG names are used, the parameter specifications defined by EPSG tables can be used if supported by the GXF reader, and the parameters provided in the GXF can be ignored. GXF readers that do not have access to EPSG tables will be able to use the parameters provided in GXF. This allows a GXF reader to resolve the map projection of any coordinate system as may be required or supported by the reader application.

The full definition of EPSG parameter values is beyond the scope of this document. Developers of GXF readers and writers are referred to the EPSG and POSC references noted at the end of this document.

2. GRID DESCRIPTION

A grid is a rectangular array of points at which single data values define a two dimensional function. Grid point locations are related to a Grid Coordinate System, which is a right-handed Cartesian system with X and Y axis defined by the horizontal bottom and left sides of a grid array (see Figure 1a.). The grid point at the bottom left corner of the array is the origin of the Grid Coordinate System.

Note: Grid points should not be confused with *grid cells*, which is a term also commonly used to describe the nodes of a grid. Grid cells are ambiguous because they either define an area enclosed by four grid points, or they define a rectangular area of the same size as the grid point/row separation and centered on each grid point. GXF describes the nodes of a grid as *grid points*, which means that each grid value represents a single point location in the grid coordinate system.

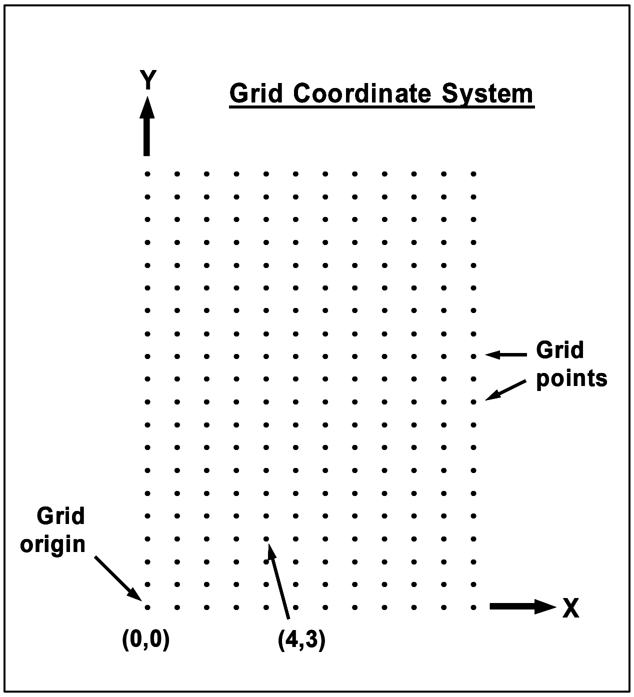


Figure 1a

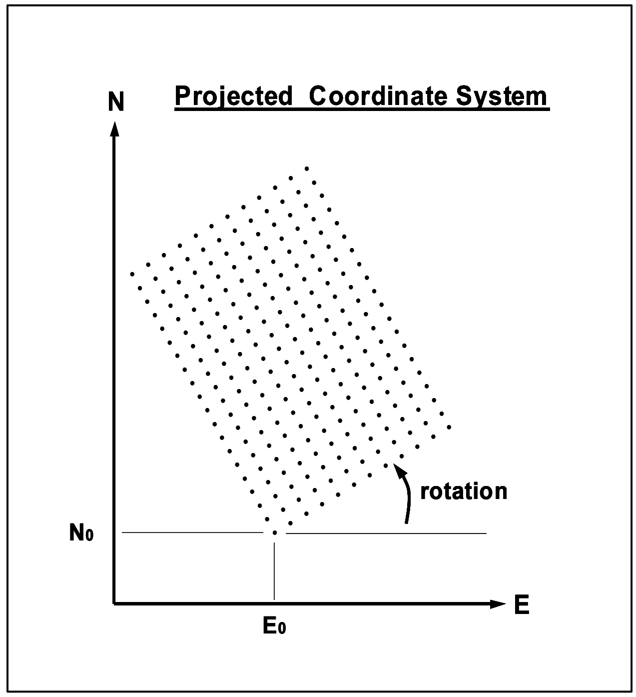


Figure 1b

Grid Coordinate System coordinates may be related to Eastings and Northings (E,N) in a Projected Coordinate System through a plane translation and rotation as shown in Figure 1b. Within the Projected Coordinate System, grid points are separated by a constant distance in the grid X and Y direction (for example, grid points in the grid X direction may be separated by 25 metres and grid points in the grid Y direction may be separated by 40 metres). The origin of the Grid Coordinate System is located at point (**E_0,N_0**) in the Projected Coordinate System, and the Grid Coordinate System can be rotated with respect to the Projected Coordinate System.

The Grid Coordinate System may alternately represent a Geographic Coordinate System, in which case the grid X and Y coordinates are degrees of Longitude and Latitude respectively on a particular geodetic datum. A Grid Coordinate System based on a Geographic Coordinate System may not be rotated.

GXF revision 3 has introduced labeled data objects #UNIT_LENGTH, #MAP_PROJECTION, and #MAP_DATUM_TRANSFORM, which are used to define the length units and Projected Coordinate System or Geographic Coordinate System. The #TRANSFORM definition has also been extended in GXF-3 to allow units of the grid values to be identified.

3. GXF FORMAT

A GXF file is an ASCII file made up of a number of labeled data objects and comments. Each labeled data object has a label line followed by one or more data lines. A label line is identified by a '**#**' character in the first column followed immediately by an upper-case label. The data associated with that label are found on one or more lines that follow the label.

Any lines that are not part of a labeled data object are ignored and can be used to place comments within a GXF file. Programs that read GXF files ignore such comment lines while searching for the next GXF data object.

All lines in a GXF file must be less than or equal to 80 characters in length. If the last non-white space character on a line is a '\', the next line is assumed to be a continuation of the current line (except for #GRID data lines). Spaces and tab characters are white space characters. Note that continuation lines are a feature of GXF revision 3, and they should only be required for defining projection parameters that require more than 80 characters. They are not required and cannot be used for writing grid data to the GXF (in the #GRID object). This allows GXF readers written for GXF revision 2 and earlier to read GXF revision 3 files.

Any name or string parameters that are part of a data object and which themselves contain spaces, must be enclosed in double quotes.

Parameters on data lines may be separated by a space or comma.

For example, the following file satisfies this format:

```
This is a very small 4 x 5 point grid.

#POINTS
5
#ROWS
4

These lines are skipped because they are not part of a data object.

#GRID
   135.28   122.21   119.64   163.25   199.15
   145.38   132.45   120.32   121.41   205.18
   140.13   151.48   132.91   119.12   219.67
   132.67   150.56   140.45   102.89   218.41
```

This file contains two comment lines at the beginning, followed by two labeled data objects, another comment and finally a **#GRID** data object. The **#GRID** object is always the last object and indicates the beginning of the gridded data. This example contains the minimum information required to define a grid, and other grid information, such as location, storage sense, sampling separations etc, assume default values. The next section documents all data objects defined in this revision. The only required data objects are **#POINTS**, **#ROWS** and **#GRID**. All other GXF objects will assume default values as defined in the next section. However, a recommended minimum set of defined data objects would also include **#PTSEPARATION**, **#RWSEPARATION**, **#XORIGIN** and **#YORIGIN**, since these objects serve to define the minimum amount of information needed to locate the grid in an assumed coordinate system.

We recommend that comments, which more fully document the grid data, be placed at the beginning of a GXF file. The type of information to include in a comment section will depend on the intended data application.

4. GXF OBJECT DEFINITIONS

#DUMMY

The grid must be rectangular (every row must have the same number of points). The dummy value defined by this object is used to define blank areas of the grid. Any grids that include blank areas must define a dummy value. The only exceptions are compressed grids, which have a pre-defined dummy value.

Default: no dummy value.

Example:

```
#DUMMY
-99999.0
```

This defines "-99999.0" to be used to represent dummy values in the grid.

It is important to specify the dummy value exactly as it will appear in the **#GRID** object. GXF readers could read the value '-99999' differently than '-99999.0', and would not be able to properly identify blank areas.

#GRID

The grid data is listed point-by-point and row-by-row. If the **#GTYPE** object is defined and non-zero, the data is written in base-90 compressed format, otherwise, the data are written as normal base-10 numbers, with each number separated by one or more spaces. The **#GRID** object and data is always the last object in a GXF file.

The first data point is at the location indicated by **#SENSE**, and is followed by successive points in that row of grid points (either horizontal or vertical), then the points in the next grid row, and so on. The points in a grid row can follow on to the next GXF line, although each new grid row must start on a new GXF line. A GXF reading program can expect **#ROWS** of **#POINTS** for a total of **#ROWS** times **#POINTS** data values.

Note that each GXF line must have no more than 80 characters, and each GXF line should only have as many data values as will fit on that line. Continuation characters are not needed and cannot be used for the grid data lines.

Default: none, must be included as the last object in a GXF file.

Example:

A 5 by 4 point grid in uncompressed format:

```
#GRID
 135.28 122.21 119.64 163.25 199.15
 145.38 132.45 120.32 121.41 205.18
 140.13 151.48 132.91 119.12 219.67
 132.67 150.56 140.45 102.89 218.41
```

or:

```
#GRID
 135.28 122.21 119.64
 163.25 199.15
 145.38 132.45 120.32
 121.41 205.18
 140.13 151.48 132.91
 119.12 219.67
 132.67 150.56 140.45
 102.89 218.41
```

Compression

If the **#GTYPE** object has been defined (and is > 0), each grid point is expressed as a base-90 integer of a fixed length specified by **#GTYPE**. If the **#GTYPE** object is not present before the **#GRID** object, the grid data is uncompressed. Although a compressed GXF file is less readable, the advantage of a smaller file can be important for large grids.

Using the base-90 numbering system, a grid data value (Z) is first converted to a whole number (positive integer) using the SCALE and OFFSET defined by the **#TRANSFORM** object:

I90 = (Z - OFFSET) / SCALE

where

I90	whole number used to represent grid values in the GXF file.
Z	real grid value
OFFSET	specified by **#TRANSFORM**
SCALE	specified by **#TRANSFORM**

The offset and scale are chosen to transform the original grid value to the range 0 to **(90^P-1)**, where **P** is the number of base-90 digits defined by **#GTYPE**.

Base-90 digits use ASCII characters in the range 37 to 126 ("%" to "~"), with the most significant character first (just as base-10 numbers use ASCII characters 48 to 57, which are characters "0" to "9"). Each grid value occupies a fixed number of characters specified by the **#GTYPE** object and there are no spaces between values. For example, assuming 3 digit base-90 numbers (**#GTYPE** set to 3), the base-90 number "**%%%**" (ASCII codes 37 37 37) is a base-10 number 0, base-90 number "**%%&**" (ASCII 37 37 38) is a base-10 number 1, and the base-90 number "**~~~**" (ASCII 126 126 126) is a base-10 number 728,999 (90^3 - 1).

Grid dummy values are always expressed as "**!**" characters (ASCII 33), so a 3 digit dummy value would be "**!!!**".

Series of consecutive grid values can be further compressed using a repeat code. The repeat character is **"** (ASCII 34) and must be duplicated **#GTYPE** times, to be followed by the number of times to repeat the following base-90 number. For example, assuming a **#GTYPE** value of 3, a repeat count has the following format:

"""NNNxxx

where	**"""**	indicates the start of a repeat sequence
	NNN	is a base-90 integer that specifies the number of times to repeat the following value
	xxx	is the base-90 grid value to be repeated, or "**!!!**" if the value is a dummy

The 3 and 2 digit repeat sequences to represent 10 dummy values in a row would be:

3-digit:	**"""%%/!!!**
2-digit:	**""%/!!**

Any lines within a compressed data object that begin with a "**$**" character in column 1 are skipped and can be used to add comments to the compressed data. Comments might be used to indicate the grid row number, for example.

Just as with uncompressed data, values in a grid row can follow to the next data line and new grid rows must start on a new line. Each row of the GXF file must have no more than 80 characters. If more than one line is required for a grid row, the line break must be at an integer multiple of **#GTYPE**. Three item repeat codes may also be split between lines.

Example:

Following is the same 5 by 4 point grid as in the previous example, but in compressed format using a precision of 3 characters per grid value. Note that **#GTYPE** is required, and **#TRANSFORM** would be required if the original data were not whole numbers in the range 0 to 728,999.

```
#TRANSFORM
0.005,-3.835000000000000
#GTYPE
 3
#GRID
(L2(/.()H)0@*&,
(bZ(Er(*v(-B*3P
(Vx(p2(Ft((:*Sb
(FD(n.(W^'^4*Pt
```

Additional compressed examples are shown in the **EXAMPLES** section.

#GTYPE

This object is used to specify the number of digits to use for base-90 compression of the grid data. If not specified, or if 0 digits are specified, base-90 compression of the data is not used. Refer to the description of base-90 compression under the **#GRID** label definition. The precision obtained by different numbers of characters (calculated as 90 raised to the power of the **#GTYPE**) is as follows:

#GTYPE = 1	1 in 90, sufficient for 4-bit data
#GTYPE = 2	1 in 8,100, sufficient for 8-bit data
#GTYPE = 3	1 in 729,000, sufficient for 16-bit data and most 32-bit data
#GTYPE = 4	1 in 65,610,000, sufficient for almost all 32-bit data
#GTYPE = 5	1 in 5,904,900,000, sufficient for most data applications

Default: If **#GTYPE** is not present, or the number of digits is set to 0, grid compression is not used. If present, a precision must be defined (typically between 1 and 5, with 4 recommended).

#MAP_PROJECTION

This defines the name and parameters of the Projected Coordinate System or Geographic Coordinate System to which the grid is related, if known. The design of this object is based on the coordinate system model described by EPSG and POSC, which distinguish between a *Geographic Coordinate System* and a *Projected Coordinate System*.

Geographic Coordinate System

A *Geographic Coordinate System* is a 2- dimensional coordinate system that uses latitude and longitude coordinates on a particular geodetic datum. It requires the identification of a geodetic datum, which includes the datum name, an ellipsoid definition and prime meridian. Common Geographic Coordinate Systems of the world are listed in the code range 4000 to 4999 in the “Coordinate system” table of the EPSG Geodesy Parameters (version 5.1). (Refer to the EPSG and POSC information sources noted in the References for further information.) EPSG Geographic Coordinate System names use the geodetic datum abbreviation or, if no abbreviation exists, the geodetic datum name.

Note that Geographic Coordinate Systems do not define geodetic datum transformation parameters to convert between systems. Coordinates may be converted between datums using a number of different methods, the most common of which are the Position Vector 7-parameter geocentric tranformation (Bursa-Wolf or Helmert transform), 3-parameter transformation (Molodenski) and the use of direct look-up tables (such as NADCON in the continental U.S., and NTv2 in Canada). If desired, a preferred 7-parameter or 3-parameter geocentric transformation may be defined separately using the #GEODETIC_TRANSFORMATION object. Defining a geocentric transformation allows a GXF reader application to use the GXF grid in different geodetic datums.

Projected Coordinate System

A *Projected Coordinate System* is a Geographic Coordinate System together with a map projection system that is used to transform (latitude, longitude) coordinates of the Geographic Coordinate System to projected coordinates (Easting, Northing). POSC and EPSG Projected Coordinate System names are composed by concatenating the geodetic datum abbreviation (if one exists) or name (if no abbreviation) and the map projection name or abbreviation separated by " / " (space forward-slash space) characters.

For example, the Projected Coordinate System "NAD83 / UTM zone 17N" defines both the Geographic Coordinate System, "NAD 83", and the projection system, "UTM zone 17N", which is a Transverse Mercator projection with standard defined transformation parameters and length units. Common Projected Coordinate Systems of the world are listed in the code range 2000 to 3999 and 20000 to 32766 in the "Coordinate system" table of the EPSG Geodesy Parameters (version 5.1).

EPSG and POSC names are case sensitive. Any name may be used if the EPSG or POSC name is not known, and the name must begin with a "*" character. For example, the South American Magnetic Mapping Project spherical coordinate system is not defined in EPSG (as of the date of this document), so it is commonly named "*SAMMP".

Two data lines are required to define a Geographic Coordinate System and three data lines are required to define a Projected Coordinate System (***Note:*** Names that include spaces must be enclosed by double quotes):

"coordinate system"
"datum", semi-major axis, inverse flattening OR eccentricity, prime meridian
"projection method", parameters

"coordinate system"	The unique key name of the coordinate system. This is the "COORD_SYS_EPSG_NAME" field value in the "Coordinate System" table of EPSG. Projected Coordinate System names are composed by concatenating the geodetic datum abbreviation or name (if no abbreviation exists) and the map projection name or abbreviation separated by " / " (space forward-slash space) characters. Geographic coordinate systems names use the geodetic datum abbreviation or, if no abbreviation exists, the geodetic datum name.
"datum"	The name of the geodetic datum, which will be the "COORD_SYS_EPSG_NAME" field value in the "Coordinate System" table of EPSG for EPSG codes in the range 4000 to 4999. Note that for Geographic

Coordinate Systems, the "*coordinate system*" and "*datum*" names will be the same.

There are times where only the ellipsoid is known (not the datum). This is an ambiguous coordinate system, and a GXF writer should attempt to determine a correct datum if possible. However, such coordinate systems can be described in EPSG and GXF by using a system from the code range 4001 to 4030 in the EPSG "Coordinate System" table. For example, a coordinate system based on the Clarke 1880 ellipsoid would be named "Unknown datum based upon the Clarke 1880 ellipsoid".

semi-major axis The ellipsoid semi-major axis in metres. This should be the value in the SEMI_MAJOR_AXIS field of the "ellipsoid" table in EPSG, converted to metres.

inverse flattening OR eccentricity
The ellipsoid inverse-flattening or the eccentricity. Values greater than 1.0 are assumed to be inverse-flattening (INV_FLATTENING from the "ellipsoid" table in EPSG). Values less than or equal to 1.0 are assumed to be eccentricity. Eccentricity and inverse flattening are related as follows:

flattening = 1 / inverse flattening
= (semi-major axis – semi-minor axis) / semi-major axis

eccentricity = √(2*flattening - flattening*flattening)

Spherical ellipsoids (where the semi-major axis = semi-minor axis) will have an eccentricity of 0.0.

prime meridian The location of the prime meridian in degrees relative to Greenwich (negative in the Western hemisphere). This is the value in the GREENWICH_LONGITUDE field of the "Prime meridian" table in EPSG, converted to decimal degrees.

"projection method" The name of the projection method, which must be one of the CTRF_METHOD_EPSG_NAME field entries in the code range 9801 to 9899 from the "Transformation Method" table of EPSG. This line is not required for a Geographic Coordinate System.

User defined names cannot be used for the projection method because the parameter list cannot be defined.

parameters	A comma delimited list of values that define all the parameters required by the projection method mathematics. The list of required parameters depends on the projection type, and the requirements are specified in Table 1. Note that angular parameters must be specified in degrees, and distance units must be specified in standard metres, regardless of the natural unit of the projection system. Be careful to use the correct conversion factor to convert from natural units to metres, particularly for False Easting and False Northing in non-metric systems. Refer to the conversion factors in the "Unit of Measure" table of EPSG, or Table 2 in section 5. Name Tables.

The EPSG "Coordinate System" table contains only coordinate systems in common use. The special requirements of exploration have resulted in the creation of numerous "ad-hoc" coordinate systems to suit the needs of a particular exploration project. To account for this, GXF-3 requires the explicit definition of all numeric parameters that describe a coordinate system be. GXF readers that can determine parameters based on the EPSG (or POSC) names, or similar aliases, may ignore the GXF parameters. GXF readers that do not have pre-defined parameters may use the provided parameter values. GXF writers should ensure that the provided parameters are correct for the most recently defined EPSG coordinate system at the time the GXF is created.

User defined coordinate systems may in fact map to a known coordinate system in a GXF reader's environment. It is the responsibility of the GXF reader to determine this mapping if it is important to the reader. Further, GXF writers may create "ad-hoc" coordinate systems even if a particular coordinate system is defined in EPSG, although the use of EPSG systems is strongly encouraged. This allows GXF files to be created for coordinate systems that do not exist in the EPSG standard tables at the time the GXF is created, although the systems may be added to the tables at some time in the future.

Default: If the **`#MAP_PROJECTION`** object is not defined, the recipient of the GXF data will be left to determine the coordinate system by some other means.

Examples:

```
#MAP_PROJECTION
"NAD27 / Ohio North"
"NAD27",6378206.4,0.082271854,0
"Lambert Conic Conformal (2SP)",40.4333333333,41.7,39.6666666667,\
-82.5,609601.22,0

#MAP_PROJECTION
"NAD83"
"NAD83",6378137,298.2572221,0

#MAP_PROJECTION
"*SAMMP sphere / *SAMMP grid projection"
"*SAMMP sphere",6378249.145,0.0,0
"Mercator",0.0,0.0,1.0,0.0,0.0
```

#MAP_DATUM_TRANSFORM

A geodetic datum transformation is not part of a coordinate system definition. *(despite many coordinate conversion applications incorrectly making it so)*. However it may be useful to relate the described coordinate system datum to another datum (usually WGS 84) by describing a "preferred" datum transformation. GXF-3 supports the description of a Position Vector 7-parameter geocentric transformation, which can also be used to approximate the 3-parameter Molodenski transformation. By providing this information, a GXF grid might be directly displayed or used in any other known datum without the need for additional information.

Two data lines are required to define a geodetic transformation:

"datum transform",dX, dY, dZ, Rx, Ry, Rz, scale, "target datum"

"datum transform"	A unique name of the datum transformation, which is either user defined or is one of the names listed in the COORD_TRF_EPS_NAME field in the "Coordinate transformation" table in EPSG. User defined names must begin with the "*" character.
dX, dY, dZ	Translation vector in metres to be ***added*** to a geocentric Cartesian coordinate point in the projection to produce geocentric Cartesian coordinate in the "target datum".
Rx, Ry, Rz	Rotations in arc-seconds (degrees/3600) to be added to a geocentric Cartesian coordinate point vector in the projection to produce a geocentric Cartesian coordinate vector in the target system. The Positional Vector sign convention is such that a positive rotation about an axis

	is defined as a clockwise rotation of the position vector when viewed from the positive direction of the axis. For example, a positive rotation about the Z axis (*Rz*) will result in a larger longitude. Note that the "Positional Vector" sign convention is opposite to the "Coordinate Frame Rotation" method that is commonly used in the USA E&P industry.
scale	The scale correction to be multiplied by the geocentric Cartesian coordinate in the map projection to obtain the correct scale in WGS 84 coordinates. The *scale* is expressed in ppm so that the true scale factor is (1 + *scale*/1,000,000).
"target datum"	(optional) If the target geodetic datum is not WGS 84, the name of the target system can be placed here. Note that most software systems convert between datums by moving first to WGS 84, then to the new datum. Because of this, few applications are able to use transform parameters for reference systems other than WGS 84.

Other transform methods, such as table look-ups (NADCON, NTv1, NTv2) or polynomial corrections (ED50 in the North Sea, Madrid datum in Spain), cannot be described in GXF. However, because these transformations are specific to particular datums, software that supports such methods may choose the appropriate method based on the geodetic datum name specified in the second line of the **#MAP_PROJECTION** parameters.

Example:

```
#MAP_DATUM_TRANSFORM
"NAD27 to WGS 84 (6)",-8,159,175,0.0,0.0,0.0,1.0
```

#POINTS

The number of points in each grid row (horizontal or vertical as defined by the **#SENSE** object).

Default: no default - this object is required.

Example:

```
#POINTS
797
```

This specifies that each grid row has 797 data points.

#PTSEPARATION

The separation between points in the grid. This should be in Base Coordinate System units (ground units for map based grids), which are defined by the **#UNIT_LENGTH** object. If the grid is defined in geographic coordinates (longitude, latitude), the separation units must be decimal degrees and the **#UNIT_LENGTH** is ignored.

Default: 1.0

Example:

```
#PTSEPARATION
25

#UNIT_LENGTH
m,1.0
```

The grid points are 25 metres apart.

#ROTATION

The rotation angle of the grid. This is the counter-clockwise angle in degrees of the bottom edge of the grid with respect to the Base Coordinate System X axis. Rotation only has meaning for Base Coordinate Systems that use the same units on the X and Y axis.

Default: 0.0

Example:

```
#ROTATION
-35
```

The grid is rotated -35 degrees (35 degrees clockwise) with respect to the Base Coordinate System (this is equivalent to an angle of 325 degrees).

#ROWS

The number of rows in the grid. A grid row (or vector) is a collection of consecutive grid points that represent the grid values along a horizontal or vertical line in the grid. The complete grid is then defined by a consecutive sequence of grid rows.

Default: no default - this object is required.

Example:

```
#ROWS
1263
```

This specifies that there are 1263 rows of points in the grid.

#RWSEPARATION

The separation between rows in the grid. These should be in Base Coordinate System units (ground units for map based grids), which are defined by the **#UNIT_LENGTH** object. If the grid is defined in geographic coordinates (longitude, latitude), the separation units must be decimal degrees and the **#UNIT_LENGTH** is ignored.

Default: 1.0

Example:

```
#RWSEPARATION
25

#UNIT_LENGTH
m,1.0
```

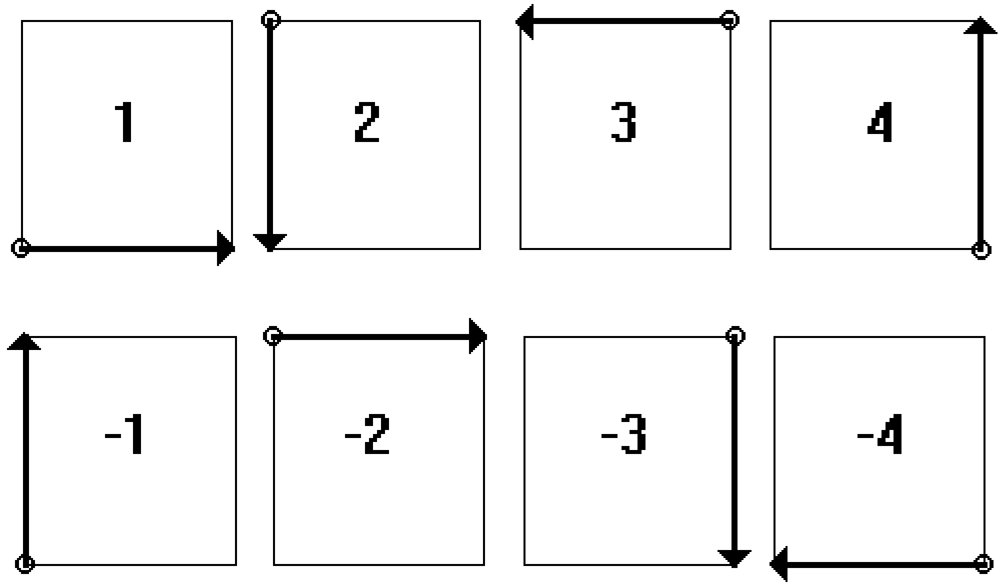
The grid rows are 25 metres apart.

#SENSE

The first point of the first row of the stored grid can be at any corner of the grid rectangle, and the grid rows can be run vertically or horizontally. The SENSE object defines this storage sense as follows:

±1	first point at bottom left of grid
±2	first point at upper left of grid
±3	first point at upper right of grid
±4	first point at bottom right of grid

A positive SENSE stores rows in a right-handed sense; a negative SENSE stores rows in a left-handed sense. This means that if you were standing at the first grid point and looking into the grid, the first grid row would extend to your right for a right handed grid (positive sense), or to your left for a left handed sense (left-handed grid):



The **SENSE** object is included to allow GXF files to be as compatible as possible with a number of existing ASCII grid formats. Almost all formats store gridded data point-by-point and row-by-row, although not always from the same grid corner. By including support for any combination of starting corner and storage sense, the effort necessary to convert existing ASCII grids to GXF files is minimized. In most cases one would only need to add the necessary header labels and specify the appropriate sense.

Default: 1 (first point at bottom left, rows left to right)

Example:

```
#SENSE
-2
```

This grid has the first point at the top left, with points in each row in order from left to right.

#TITLE

A one line descriptive title of the grid. Some grid formats include textual descriptions of the grid, and this information can be placed in a **`#TITLE`** object. Double quotes surrounding the title are optional.

Default: blank title

Example:

```
#TITLE
"Total Magnetic Field"
```

#TRANSFORM

This keyword is followed by two space or comma delimited numbers and an optional name on the same line:

scale, offset, "name"

scale,offset	values used to transform the grid data values in the **#GRID** section to the working units defined by the *unit_key* and *"description"*: working unit = (GXF_value * *scale*) + *offset*
"name"	An optional unit name description, normally the common unit abbreviation.

This transformation also applies to the **#ZMINIMUM** and **#ZMAXIMUM** values if they are specified.

The **#TRANSFORM** information is also used to transform real data to integers for base-90 compression specified by the **#GTYPE** object, in which case a *scale* and *offset* should be chosen to maximize the dynamic range of the base-90 numbers.

Default: *scale* = 1.0, offset = 0.0, *name* unknown.

Example:

```
#TRANSFORM
0.01,56000,"nT"
```

The **#GRID**, **#ZMINIMUM** and **#ZMAXIMUM** data will be multiplied by 0.01 and added to 56000 to produce units of nanotesla (nT).

#UNIT_LENGTH

A conversion factor to convert the grid length units to metres in the projection. This is defined on one data line:

"name",scale_metres

"name"	The length unit abbreviation selected from the EPSG compliant abbreviations in Table 2. If the unit abbreviation is not in the list, a user defined name may be used. User defined unit names must begin with a "*" character. Names that contain spaces must be enclosed in double quotes.

scale_metres	A multiplying factor to convert grid length units to metres.

Default: m,1.0

Example:

```
#UNIT_LENGTH
ft,0.3048
```

```
#UNIT_LENGTH
"ft US",0.3048006096012
```

For gridded data in a Geographic Coordinate System, units are assumed to be degrees of longitude and latitude in the local datum and the **#UNIT_LENGTH** object can be ignored. If a **#UNIT_LENGTH** object is created, it should be:

```
#UNIT_LENGTH
dega,1.0
```

#XORIGIN

The X location of the bottom left corner of the grid in the Base Coordinate System (**x0** in Figure 1b). These must be defined in the units defined by **#UNIT_LENGTH**, and in the coordinate space of the map projection, if defined.

Default: 0.0

Example:

```
#XORIGIN
3197250
```

The grid origin is located at X coordinate 3,197,250 in the Base Coordinate System.

#YORIGIN

The Y location of the bottom left corner of the grid in the Base Coordinate System (Refer to Figure 1b). These must be defined in the units defined by **#UNIT_LENGTH**, and in the coordinate space of the map projection, if defined.

Default: 0.0

Example:

```
#YORIGIN
52821300
```

The grid origin is located at Y coordinate 52,821,300 in the Base Coordinate System.

#ZMAXIMUM

The maximum Z data value in the grid in the units of the **`#GRID`** object.

Default: If not provided , a reading program that requires this information must read the **`#GRID`** data in order to determine the minimum and maximum values in the data.

Example:

```
#ZMAXIMUM
58650.0
```

The maximum Z value in the grid is 58,650.

#ZMINIMUM

The minimum Z data value in the grid in the units of the **`#GRID`** object.

Default: If not specified, a reading program that requires this information must read the **`#GRID`** data in order to determine the minimum and maximum values in the data.

Example:

```
#ZMINIMUM
42851.5
```

The minimum Z value in the grid is 42,851.5.

##xxxx

User defined labels. If anyone feels the need to add their own labels to the existing GXF standard, they can be added by using "**`##`**" as the prefix to the minimum four letter label. This label must not be the same as any existing defined label, otherwise it will be interpreted as the existing label. This prevents you from defining a label that may be defined in a later revision of the GXF standard. Such user-defined labels may be added to the standard at a later date if they prove to be widely used (by reducing them to a single "**`#`**"). Note that a good GXF reading program will skip and ignore any GXF labels that it does not understand, which means all "**`##`**" labels are ignored.

Example:

```
##CALIBRATE_857
1.875,2.0
```

A user-defined object is used to specify calibration coefficients for an instrument.

5. NAME TABLES

The tables in this section are used to define a common set of key parameters for projections and units used in GXF. The first column of each table is a key name, which is unique within each table. Key names that are not EPSG names start with a "*" (asterisk) character.

Table 1 Projection Transformation Methods

This table identifies all defined projection transformation methods. The parameters are listed in the order required in the **`#MAP_PROJECTION`** data object. Except for methods marked with an asterisk (*), this table was compiled using EPSG Geodetic Parameter Set version 5.1 and POSC (2.2) as data sources.

In cases where EPSG and POSC names differ, GXF allows either POSC or EPSG names. For example, EPSG "Transverse Mercator (South Orientated)" is the same as POSC "Transverse Mercator (South Oriented)" (POSC corrects the spelling of "Oriented" in this case). GXF would allow either name

Parameter Notes:

- All distance references must be specified in metres.
- All geographic references (latitudes and longitudes) are specified in degrees.
- Latitudes in the Southern hemisphere are negative.
- Longitudes in the Western hemisphere are negative.
- Longitudes are relative to the prime meridian of the datum.
- False Eastings and Northings are always specified in metres, regardless of the natural unit of the projection.

Projection method	Required parameters
Hotine Oblique Mercator	Latitude of projection center Longitude of projection center Azimuth of initial line Angle from Rectified to Skew Grid Scale factor on initial line False Easting False Northing
Laborde Oblique Mercator	Latitude of projection center Longitude of projection center Azimuth of initial line Scale factor on initial line False Easting False Northing

Projection method	**Required parameters**
Lambert Conic Conformal (1SP)	Latitude of natural origin Longitude of natural origin Scale factor at natural origin False Easting False Northing
Lambert Conic Conformal (2SP)	Latitude of false origin Longitude of false origin Latitude of first standard parallel Latitude of second standard parallel Easting at false origin Northing at false origin
Lambert Conformal (2SP Belgium)	Latitude of false origin Longitude of false origin Latitude of first standard parallel Latitude of second standard parallel Easting at false origin Northing at false origin
Mercator (1SP)	Latitude of natural origin Longitude of natural origin Scale factor at natural origin False Easting False Northing
Mercator (2SP)	Latitude of first standard parallel Longitude of natural origin False Easting False Northing
New Zealand Map Grid	Latitude of natural origin Longitude of natural origin False Easting False Northing
Oblique Stereographic	Latitude of natural origin Longitude of natural origin Scale factor at natural origin False Easting False Northing
Polar Stereographic	Latitude of natural origin Longitude of natural origin Scale factor at natural origin False Easting False Northing

Projection method	**Required parameters**
Transverse Mercator	Latitude of natural origin Longitude of natural origin Scale factor at natural origin False Easting False Northing
Transverse Mercator (South Oriented) Transverse Mercator (South Orientated)	Latitude of natural origin Longitude of natural origin Scale factor at natural origin False Easting False Northing
*Albers Equal Area	Latitude of first standard parallel Latitude of second standard parallel Latitude of false origin Longitude of false origin Easting at false origin Northing at false origin
*Equidistant Conic	Latitude of first standard parallel Latitude of second standard parallel Latitude of false origin Longitude of false origin Easting at false origin Northing at false origin
American Polyconic	Latitude of natural origin Longitude of natural origin False Easting False Northing

Table 2 Length Units

The following table is compiled from the UNIT_OF_LENGTH table in the EPSG tables. The unit names are the abbreviations defined in EPSG. This table is for convenient reference only, and the EPSG table is considered the primary reference.

Unit	Description	Factor to metres
m	metre	1
GLM	German legal metre	1.000013597
mGer	German legal metre (POSC)	1.000013597
km	kilometre	1000
ft	foot	0.3048
ftBnA	British foot (Benoit 1895 A)	0.304799733
ftBnB	British foot (Benoit 1895 B)	0.304799735
ftBr(65)	British foot (1865)	0.304800833
ftCla	Clarke's foot	0.304797265
ftGoldCoast	Gold Coast foot	0.30479971
ftInd	Indian foot	0.30479951
ftInd(37)	Indian foot (1937)	0.30479841
ftInd(62)	Indian foot (1962)	0.3047996
ftInd(75)	Indian foot (1975)	0.3047995
ftSe	British foot (Sears 1922)	0.304799472
ftUS	US survey foot	0.30480061
chBnA	British chain (Benoit 1895 A)	20.1167824
chBnB	British chain (Benoit 1895 B)	20.11678249
chCla	Clarke's chain	20.11661949
chSe	British chain (Sears 1922)	20.11676512
chUS	US survey chain	20.11684023
lkBnA	British link (Benoit 1895 A)	0.201167824
lkBnB	British link (Benoit 1895 B)	0.201167825
lkCla	Clarke's link	0.201166195
lkSe	British link (Sears 1922)	0.201167651
lkUS	US survey link	0.201168402
mi	Statute mile	1609.344
miUS	US survey mile	1609.347219
nautmi	nautical mile (POSC)	1852
NM	nautical mile	1852

Unit	Description	Factor to metres
ydBnA	British yard (Benoit 1895 A)	0.9143992
ydBnB	British yard (Benoit 1895 B)	0.914399204
ydCla	Clarke's yard	0.914391795
ydInd	Indian yard	0.914398531
ydInd(37)	Indian yard (1937)	0.91439523
ydInd(62)	Indian yard (1962)	0.9143988
ydInd(75)	Indian yard (1975)	0.9143985
ydSe	British yard (Sears 1922)	0.914398415
Bin12.5m	Bin width 12.5 metres	12.5
Bin165ftUS	Bin width 165 US survey feet	50.29210058
Bin25m	Bin width 25 metres	25
Bin3.125m	Bin width 3.125 metres	3.125
Bin330ftUS	Bin width 330 US survey feet	100.5842012
Bin37.5m	Bin width 37.5 metres	37.5
Bin6.25m	Bin width 6.25 metres	6.25
Bin82.5ftUS	Bin width 82.5 US survey feet	25.14605029
deg	degree	n/a
dega	degree (POSC)	n/a

6. EXAMPLES

The following GXF examples all store the contents of a very small 4 by 6 point grid. In order to make the examples easy to understand, the sample grid will define a linear function that increases by one at each grid node in the grid X direction, and by 10 for each grid node in the grid Y direction, as follows:

30	31	32	33	34	35
20	21	22	23	24	25
10	11	12	13	14	15
0	1	2	3	4	5

The minimum GXF file for this grid would be:

```
#POINTS
6
#ROWS
4
#GRID
      0     1     2     3     4     5
      10    11    12    13    14    15
      20    21    22    23    24    25
      30    31    32    33    34    35
```

Note that each text line of a GXF cannot exceed 80 characters. In order to support very long grid rows, rows can wrap onto the next line of the GXF file. A GXF reader will keep reading until all the points specified for a row are read. However, every new grid row must start on a new file line. The following GXF is equivalent to the previous example:

```
#POINTS
6
#ROWS
4
#GRID
      0     1     2     3
      4     5
      10    11    12    13
      14    15
      20    21    22    23
      24    25
      30    31    32    33
      34    35
```

A more conventional GXF file would also include the separation between grid points and grid rows, and the location of the grid relative to some Base Coordinate System. GXF Revision 3 provides for specifying coordinate systems, which should also be specified when known.

```
===========================================
This is a comment area which is ignored
by GXF readers.
===========================================
#POINTS
6
#ROWS
4

#PTSEPARATION
12.5
#RWSEPARATION
12.5

#XORIGIN
1750000.0
#YORIGIN
4250.0
#ROTATION
0.0

#UNIT_LENGTH
"ftUS",0.3048006096012
#MAP_PROJECTION
"NAD27 / Ohio North"
"NAD27",6378206.4,0.082271854,0
"Lambert Conic Conformal (2SP)",40.4333333333,41.7,39.6666666667,\
82.5,609601.22,0

#MAP_DATUM_TRANSFORM
"NAD27 to WGS 84 (6)",-8,159,175,0,0,0,1
#GRID
      0     1     2     3     4     5
      10    11    12    13    14    15
      20    21    22    23    24    25
      30    31    32    33    34    35
```

The next set of examples illustrates the effect of different storage senses. Note that the storage sense only effects the way in which the grid is stored in the GXF file. It does not change the grid origin or rotation angle with respect to a base coordinate system, since these are defined with respect to the bottom left corner of the grid. Examples for senses of +1, -1, +2 and -2 are shown since these are the most common formats anticipated:

```
#SENSE
1
#POINTS
6
#ROWS
4
#GRID
0     1     2     3     4     5
10    11    12    13    14    15
20    21    22    23    24    25
30    31    32    33    34    35
```

```
#SENSE
-1
#POINTS
4
#ROWS
6
#GRID
0     10    20    30
1     11    21    31
2     12    22    32
3     13    23    33
4     14    24    34
5     15    25    35
```

```
#SENSE
2
#POINTS
4
#ROWS
6
#GRID
30    20    10    0
31    21    11    1
32    22    12    2
33    23    13    3
34    24    14    4
35    25    15    5
```

```
#SENSE
-2
#POINTS
6
#ROWS
4
#GRID
30    31    32    33    34    35
20    21    22    23    24    25
10    11    12    13    14    15
0     1     2     3     4     5
```

Base-90 Compression

Uncompressed GXF data is *human* readable, and therefore has advantages as a grid exchange standard because someone can work out the contents of the grid simply by looking at the GXF listing. However, large grids can become unreasonably large in this format. For example, an airborne survey grid 1000 by 1000 points in size could be more than 15 megabytes if expressed as full precision ASCII numbers.

To deal with large grids, the GXF format supports compression using base-90 numbers in the **#GRID** object data. Base-90 digits use the ASCII character sequence 37 to 126 (**%** to **~**), which are all printable characters. The resulting grid data is no longer human readable (unless you wish to learn base-90 numbering!), but the compressed GXF file can be smaller than an original 4-byte binary grid

Compression requires that the grid values be converted to whole numbers (positive integers) with the use of the **#TRANSFORM** parameters, and that the **#GTYPE** object be present to define the number of characters to be used for each base-90 number. Three characters should be sufficient for almost all applications since it provides a precision of 1 part in 729,000.

The following three examples show the same grid in uncompressed format, then compressed format, and finally compressed format with repeat compression.

Uncompressed GXF:

```
#ROWS
8
#POINTS
10
#DUMMY
-9999
#GRID
 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999 -9999
 -9999 -9999 -9999 -9999 -9999   1.0   2.5     0  -1.0 -9999
 -9999 -9999 -9999 -9999   1.0   1.5   4.5   1.0     0 -9999
 -9999 -9999 -9999   1.5   4.8   6.2   1.1  -1.6 -9999 -9999
 -9999 -9999   4.6   9.1  11.5 -9999 -9999 -9999 -9999 -9999
 -9999 -9999   3.1   1.6     0 -9999 -9999 -9999 -9999 -9999
 -9999 -9999 -9999   0.5 -9999 -9999 -9999 -9999 -9999 -9999
```

Compressed GXF, without repeat compression:

```
#POINTS
          10
#ROWS
           8
#TRANSFORM
   5.0E-03     -118.835
#GTYPE
   3
#GRID
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!(5@(V^'y,'br!!!
!!!!!!!!!!!!(5@(@J)),(5@'y,!!!
!!!!!!!!!(@J)/h)Nr(7T'UT!!!!!!
!!!!!!)+@*5@*j^!!!!!!!!!!!!!!!
!!!!!!(c|(B^'y,!!!!!!!!!!!!!!!
!!!!!!!!!(*6!!!!!!!!!!!!!!!!!!
```

Compressed GXF, with repeat compression:

```
#POINTS
          10
#ROWS
           8
#TRANSFORM
   5.0E-03     -118.835
#GTYPE
   3
#GRID
%%/!!!
%%/!!!
%%*!!!(5@(V^'y,'br!!!
%%)!!!(5@(@J)),(5@'y,!!!
!!!!!!!!!(@J)/h)Nr(7T'UT!!!!!!
!!!!!!)+@*5@*j^%%*!!!
!!!!!!(c|(B^'y,%%*!!!
!!!!!!!!!(*6%%+!!!
```

REFERENCES

Petrotechnical Open Software Corporation (POSC, www.posc.org)
Software Integration Platform Specifications,

Version 2.2
http://www.posc.org/Epicentre.2_2/SpecViewer.html

Coordinate system information can be found in the *Subject Discussions* under the **Epicentre Logical Data Model** heading on the POSC home page (as of 1998/3/7).

European Petroleum Survey Group (EPSG)
EPSG Geodesy Parameters may be obtained from (as of November 22, 1999):
http://www.epsg.org

Snyder, John P., **Map Projections - A Working Manual**, U.S. Geological Survey professional paper 1395, U.S. Government Printing Office, 1987.