Using user-defined objects in the Esri Projection Engine

Document version: 13.1.0
Date: 2023-01-02

Table of Contents

Introduction

All predefined factory data in the Projection Engine (PE) is kept in various databases. The database mechanism used is configurable by a user and a user may provide his own database(s) which can replace, augment, and/or override existing databases.

The function of these databases is not to create a PE object, but to return a record that contains data and/or other factory codes used to produce a PE object. Many PE objects contain sub-objects, and thus the actual creation of such a PE object will necessitate the database lookup of multiple database records, one for each sub-object (and sub-objects can contain their own sub-objects).

An object may also contain metadata information that is not considered when comparing objects. This may consist of areas-of-interest, descriptions, versions, status, locale-specific names, etc.

The factory codes contained in a database record may contain factory codes that can be found in a totally different database. This is especially true when looking up entries in a user-supplied database, which will typically refer to objects in a previously loaded database.

By default, the "builtin" database is loaded first, and then any user-specified databases are loaded. When a lookup is done for a particular factory code, the order of search is from the last-loaded database down to the first-loaded database. Thus, each database that is loaded can override any previously loaded database. Databases can also be localized.

A database does not need to have all PE object types contained in it. In fact, typically a user-provided database will have only a few object-types in it, and will just use the "builtin" database for all other definitions (such as units, prime meridians, spheroids, etc.).

Note that internal PE objects (methods and parameters), although they are referenced by factory codes, are not kept in any database, but are created directly by the PE core routines. As such, a user cannot create his own predefined versions of these objects, but he can apply his own metadata, such as version, status, and code-changes to existing entries.

IMPORTANT - This document contains the PE database specifications for ArcGIS versions 10.4 and later. See Appendix D for the differences between the current PE database implementation and versions previous to 10.4.

PE object and record types

The following is a list of object-types that may appear in a database:

Object type	Abbreviation	Description
geogcs	gcs	Geographic coordinate reference systems
projcs	pcs	Projected coordinate reference systems
vertcs	vcs	Vertical coordinate reference systems
gxyzcs	xyz	3D Geocentric coordinate reference systems
hvcoordsys	hvc	Compound (horizontal and/or vertical) coordinate reference systems
datum	dat	Horizontal datums
vdatum	vdt	Vertical datums
geogtran	gtf	Geographic transformations
verttran	vtf	Vertical transformations
angunit	ang	Angular units of measure
linunit	lin	Linear units of measure
areaunit	are	Area units of measure
scaleunit	scl	Scale units of measure
primem	pri	Prime meridians
spheroid	sph	Spheroids (ellipsoids)
extent	ext	Areas of interest
htmethod	htm	Horizontal transformation methods
method	mth	Geographic transformation methods
vtmethod	vtm	Vertical transformation methods
parameter	par	Parameters
projection	prj	Projection methods

The following is a list of record-types used:

Record type	Abbreviation	Description
data	dat	Object data
codechange	chg	Code (WKID) changes
deprecated	dep	Deprecated codes
synonym	syn	Name synonyms
coderange	rng	Code ranges
dispname	dsp	Display name (Unicode)
description	dsc	Description (Unicode)
defstring	def	macro definitions
areainfo	inf	Coordinate-system dialog entries (Unicode)

areacode	aco	Area codes
version	ver	Version info
exception	exc	gtlist/vtlist exceptions
gcsvcs	gve	GCS-VCS equivalences
axis	axi	Axis description
ensemble	ens	Datum ensembles
epoch	epc	Dynamic frame epochs

Not all record-types are defined for each object-type. The following is a table showing all valid record-types for each object-type:

obj	dat	chg	dep	syn	rng	dsp	dsc	def	inf	aco	ver	exc	gve	axi	ens	epc
gcs	x	x	x	x	x	x	x	x	x	x	x		x	x		
pcs	x	x	x	x	x	x	x	x	x	x	x			x		
vcs	x	x	x	x	x	x	x	x	x	x	x		x	x		
xyz	x	x	x	x	x	x	x	x		x	x		x	x		
hvc	x	x	x	x	x	x	x	x		x	x					
dat	x	x	x	x	x	x	x	x			x				x	x
vdt	x	x	x	x	x	x	x	x			x				x	x
gtf	x	x	x	x	x	x	x	x		x	x	x				
vtf	x	x	x	x	x	x	x	x		x	x	x				
ang	x	x	x	x	x	x	x	x			x					
lin	x	x	x	x	x	x	x	x			x					
scl	x	x	x	x	x	x	x	x			x					
are	x	x	x	x	x	x	x	x			x					
pri	x	x	x	x	x	x	x	x			x					
sph	x	x	x	x	x	x	x	x			x					
ext	x					x										
htm		x	x	x	x	x	x	x			x					
mth		x	x	x	x	x	x	x			x					
vtm		x	x	x	x	x	x	x			x					
par		x	x	x	x	x	x	x			x					
prj		x	x	x	x	x	x	x			x					

PE database types

The term "database" here means a particular storage mechanism for the various tables described above. The following are the database types supported internally by the PE:

builtin	This database consists of all tables stored in static arrays that are linked in the library, and thus is always available.
objedit	This database consists of a directory with a separate CSV (comma-separated-value) file for each table containing user-defined entries.
xmledit	This database consists of a directory with a separate XML file for each table containing user-defined entries.

The "builtin" database contains all the above tables (except for display-name). A user-provided database need not contain all tables, only those that have user-defined entries. Normally, a user will have his database(s) loaded after the builtin database, so only additions and/or overrides need to be provided. Each database loaded takes priority over any previously-loaded database.

There is also the ability to use a user-written database by dynamically loading at run-time a DLL that implements a defined API, although that is not described in this document.

Code-ranges

The purpose of these databases is to associate a factory-code or Well-Known-ID (WKID) with the data for it. The WKIDs for each object-type fall into defined code-ranges. Originally, each code-range was unique, but then EPSG (IOGP Geodesy subcommittee) started assigning WKIDs for different object-types in overlapping code-ranges. Esri-defined code-ranges are still unique. User-defined code-ranges may overlap, but not within an object-type. WKIDs for a particular object-type are only recognized if they fall in a defined code-range for that object.

There are three authorities for predefined code-ranges, as follows:

EPSG	These are EPSG-assigned codes. They are all in the range of 1024 to 32767.
Esri	These are Esri-assigned codes. Entries are in the range of 33000 to 199999.
CUSTOM	These are user-defined entries. Entries are in the range of 200000 to 299999.

Here is a table of predefined code-ranges for each object-type:

Object	First	Last	Authority
angunit	1024	1100	EPSG
angunit	9100	9199	EPSG
angunit	109100	109199	Esri
angunit	209100	209199	CUSTOM
areaunit	109400	109499	Esri
areaunit	209400	209499	CUSTOM
datum	1024	32767	EPSG
datum	106000	106999	Esri
datum	206000	206999	CUSTOM
geogcs	1024	32767	EPSG
geogcs	37000	37999	Esri
geogcs	104000	104999	Esri
geogcs	204000	204999	CUSTOM

geogtran	1024	32767	EPSG
geogtran	108000	108899	Esri
geogtran	208000	208899	CUSTOM
gxyzcs	1024	32767	EPSG
gxyzcs	103000	103999	Esri
gxyzcs	203000	203999	CUSTOM
htmethod	119600	119699	Esri
hvcoordsys	1024	32767	EPSG
hvcoordsys	107400	107599	Esri
hvcoordsys	207400	207599	CUSTOM
linunit	1024	1100	EPSG
linunit	9000	9099	EPSG
linunit	9300	9399	EPSG
linunit	109000	109099	Esri
linunit	209000	209099	CUSTOM
method	9600	9699	EPSG
method	109600	109699	Esri
parameter	100000	100099	Esri
projcs	1024	32767	EPSG
projcs	53000	54999	Esri
projcs	65000	65199	Esri
projcs	102000	103999	Esri
projcs	202000	203999	CUSTOM
primem	8900	8999	EPSG
primem	108900	108999	Esri
primem	208900	208999	CUSTOM
projection	43000	43499	Esri
scaleunit	1024	1100	EPSG
scaleunit	9200	9299	EPSG
scaleunit	109200	109299	Esri
scaleunit	209200	209299	CUSTOM
spheroid	1024	32767	EPSG
spheroid	107000	107399	Esri

spheroid	107600	107999	Esri
spheroid	207000	207399	CUSTOM
spheroid	207600	207999	CUSTOM
vertcs	1024	32767	EPSG
vertcs	105600	105799	Esri
vertcs	115600	115999	Esri
vertcs	205600	205999	CUSTOM
vdatum	1024	32767	EPSG
vdatum	105100	105299	Esri
vdatum	205100	205299	CUSTOM
verttran	1024	32767	EPSG
verttran	110000	110099	Esri
verttran	210000	210099	CUSTOM
vtmethod	129600	129999	Esri
extent	1024	32767	EPSG
extent	180000	180999	Esri
extent	280000	280999	CUSTOM

A user may define his own custom code-range, as long as it doesn't overlap with any of the predefined code-ranges. By using user-defined WKIDs that are either in the CUSTOM range or in your own custom-defined code-ranges, you ensure that the code you are using will not override an existing code.

Database filenames

The "objedit" and "xmledit" databases consist of directories containing various text files containing user-defined table entries. The names for these files have the following syntax:

<object-type>_<record-type>[_<language>].<extension>

Where:

object-type	Either the name or the abbreviation of an object-type. For example, "geogcs" ("gcs") or "geogtran" ("gtf").
record-type	Either the name or the abbreviation of a record-type. For example, "data" ("dat") or "areainfo" ("inf"). For backwards compatibility, in an objedit database if the record-type is "data", the record-type part of the name may be omitted.
language	An optional language designation. This is for localization. For example, "de" for German or "ja" for Japanese.
extension	"txt" for objedit files. "xml" for xmledit files. For backwards compatibility, in an objedit database if the record-type is "data", the extension may be omitted.

Examples:

ang_data.txt	objedit	ANGUNIT data definitions
linunit_chg.xml	xmledit	LINUNIT code-change definitions
geogcs_inf_ja.xml	xmledit	Japanese GEOGCS areainfo table
geogtran.txt	objedit	GEOGTRAN data (legacy filename format)
projcs	objedit	PROJCS data (legacy filename format)

In a particular database, not all files need to be present, only those files that have user definitions. Any other files in the directory whose names do not match the above syntax are simply ignored.

Defining the databases to load

The databases to be loaded are specified by the PEDATABASE environment variable. This variable is always queried by the PE independently of any program using the PE. Thus, ArcGIS for Desktop, ArcGIS Pro, and ArcGIS Server will all respond to the presence of this variable. See Appendix E for details on how to set an operating system environment variable.

The PEDATABASE variable specifies a "database-environment" string. The format of this string is:

    module [@name] [(options)] [;...]

where:

module	The database-type to load (builtin, objedit, xmledit, etc.). A special value of "-" means to clear all loaded databases. Specifying "-" at the beginning of the definition-string tells the PE to not load the builtin database.
name	The name passed to the database. It is ignored for the builtin database. For the objedit and xmledit databases, it names the directory in which to find the files. This name may be a local path or a network share. The PE only needs read permission for the directory.
options	An optional comma-separated list of various options, as listed below.

Note that you normally don't include the "builtin" database in the list of databases to load. By default, it is loaded first and then any specified databases are loaded after it. Any request to load a database that is already loaded results in a no-op.

Options:

loaddata	Load all tables into memory if possible. This is always done for objedit and xmledit databases, since you don't want to search text files for a code every time you do a lookup.
loadlazy	Load data as used and cache for further use. (Not applicable for objedit and xmledit databases.)
useoldver	Report the version information of a code-changed entry using the original version. The default is to report the version of the changed-code.

dupsok	Allow user-defined databases to override existing WKID entries in previously-loaded databases.
language=aaa	Specify language to load.
objtypes=aaa	Specify object-types to include or exclude (+-). Default is to load all object-types that are present.
rectypes=aaa	Specify record-types to include or exclude (+-). Default is to load all record-types that are present.
status=aaa	Specify status to apply to all data records. Choices are: code user discontinued dead
authname=aaa	Specify default authority to apply to all data records.
authver=aaa	Specify default version to apply to all data records.
useropts=aaa	Specify user-options to use. These options are database-specific.

objedit databases:

objname	Include object-type name when writing records.
macro	Use macros instead of codes if possible when writing.
ml | multiline	Write multiple lines for data.
nosynauths	Exclude synonym records that contain authority names.

xmledit databases:

macro	Use macros instead of codes if possible when writing.
ml | multiline	Write multiple lines for data.
nosynauths	Exclude synonym records that contain authority names.

Example 1

Load an objedit database after loading the builtin database:

objedit@c:\databases\pe

Example 2

Load an objedit database after loading the builtin database, specifying a default authority of "Tin Man Oil" and a version of "12.1.2":

objedit@c:\databases\pe(authname=Tin Man Oil,authver=12.1.2)

Example 3

Load Japanese XML areainfo tables but don't load the builtin areainfo tables:

-;builtin(rectypes=-inf);xmledit@c:\databases\pe(language=ja)

The leading "-" tells the PE to not load the entire builtin database (which contains all tables, including the areainfo tables). Then we explicitly load the builtin database but without the areainfo tables, which we want to replace. Then we load our database with the replacement tables.

The reason for doing this is that the areainfo tables are different, in that they are only used by loading the entire table at one time. Thus, the user-provided tables would simply add to the builtin tables rather than overriding them, which would result in duplicate entries.

Note that we could have loaded the user database prior to the builtin database, but then any macros in the user database that referenced builtin macros would not have been recognized.

Example 4

Load only the geogtran and projcs data entries from an objedit database:

objedit@c:\databases\pe(objtypes=gtf+pcs,rectypes=dat)

PEOBJEDITHOME environment variable

An alternate (legacy) way of specifying the location of the objedit database directory is to set the value of the environment variable PEOBJEDITHOME to the absolute path of the directory. This is still supported, but its use is deprecated. The preferred way is to use the PEDATABASE variable. If both the PEOBJEDITHOME and PEDATABASE variables are present, the PEOBJEDITHOME variable is ignored.

Specifying the following:

PEOBJEDITHOME="<directory-name>"

is the same as specifying:

```
PEDATABASE="objedit@<directory-name>"
```

Record format syntax rules

Objedit text rules:

1. Whitespace outside of quotes is ignored.
2. Anything following a # is considered a comment.
3. After removing comments and extraneous whitespace, if a line is empty it is ignored.
4. A line may be continued onto another line by adding a backslash (\) at the end of it.
5. Text strings are usually enclosed in quotes, but the quotes are only necessary if the text has special characters (e.g. whitespace or a comma).
   For example, a linunit entry may be
   209001, "Egyptian Cubit", 0.447
   or
   209001, Egyptian_Cubit, 0.447
6. Each line may be optionally prefixed with the object-type.
   For example, a linunit entry may be
   209001, "Egyptian Cubit", 0.447
   or
   LINUNIT, 209001, "Egyptian Cubit", 0.447
   The object-type name is case-insensitive.

Xmledit text rules:

1. The XML file must be a valid XML file, meaning that all data entries must be contained in a document. The document-name is ignored. Here is a skeleton that is recommended be followed:
   <?xml version="1.0" encoding="UTF-8"?>
   <PE>
   ...
   </PE>
2. Keywords and object-types are all case-insensitive.
3. Comments (<!-- ... -->) are ignored.

Both objedit and xmledit:

1. Any number should be displayed in "en_US" format (i.e. a decimal point character is always used as the decimal separator character). This makes the data files more portable, since they may be used in any locale.
2. Any WKID may be described by its numeric code (e.g. 4326) or by its macro (e.g. PE_GCS_WGS_1984). Note that macros are case-insensitive. Using macros makes the data files more readable. Macros may be user-defined. See the include file pedefs.h for the list of all predefined macros.

3. The following are limits on string lengths:
    a. Object names — 79 ASCII characters
    b. Authority names — 79 ASCII characters
    c. Version strings — 23 ASCII characters
    d. UNIT display-names, plural-names, abbreviations — 79 Unicode characters
    e. AREAINFO area-names — 79 Unicode characters
    f. AREAINFO category names — 255 Unicode characters
    g. DESCRIPTION strings — 255 Unicode characters
    h. DISPNAME names — 79 Unicode characters

Record formats for data tables

The format of the data files is different for each object-type. Note that in the descriptions and examples below, extra whitespace and newlines are added for clarity.

GEOGCS – geographic coordinate reference systems

Contents:

- Geogcs-code
- Name
- Datum-code
- Primem-code
- Angunit-code
- Linunit-code (only for 3D GEOGCSs)

Objedit syntax:

```
[GEOGCS,] geogcs-code, "name", datum-code, primem-code, angunit-code [,
        linunit-code]
```

Examples:

```
3819, "GCS_HD1909", 1024, 8901, 9102

PE_GCS_HD1909, \
        "GCS_HD1909", \
        PE_D_HUNGARIAN_DATUM_1909, \
        PE_PM_GREENWICH, \
        PE_U_DEGREE
```

Xmledit syntax:

```
<GEOGCS
        code="geogcs-code"
        name="name"
        datum="datum-code"
        primem="primem-code"
        angunit="angunit-code"
        [linunit=”linunit-code”]
/>
```

Examples:

```
<GEOGCS
        code="3819"
        name="GCS_HD1909"
        datum="1024"
        primem="8901"
        angunit="9102"
/>

<GEOGCS
        code="PE_GCS_HD1909"
        name="GCS_HD1909"
        datum="PE_D_HUNGARIAN_DATUM_1909"
        primem="PE_PM_GREENWICH"
        angunit="PE_U_DEGREE"
/>
```

PROJCS – projected coordinate reference systems

Contents:

- Projcs-code
- Name
- Geogcs-code
- Projection-code
- Linunit-code
- Parameter ...
    - Parameter-code
    - Value [and optional unit code]

  Note that parameters may be specified in any order.

Objedit syntax:

```
[PROJCS,] projcs-code, "name", geogcs-code, projection-code, linunit-code
          [, parameter-code, value[:unit-code]] ...
```

Examples:

```
2000, "Anguilla_1957_British_West_Indies_Grid", \
          4600, 43006, 9001, \
          100001, 400000.0:9001, \
          100002, 0.0, \
          100010, -62.0, \
          100003, 0.9995, \
          100021, 0.0

PE_PCS_ANGUILLA_1957_BRITISH_W_INDIES, \
          "Anguilla_1957_British_West_Indies_Grid", \
          PE_GCS_ANGUILLA_1957, \
          PE_PRJ_TRANSVERSE_MERCATOR, \
          PE_U_METER, \
          PE_PAR_FALSE_EASTING, 400000.0:PE_U_METER, \
          PE_PAR_FALSE_NORTHING, 0.0, \
          PE_PAR_CENTRAL_MERIDIAN, -62.0, \
          PE_PAR_SCALE_FACTOR, 0.9995, \
          PE_PAR_LATITUDE_OF_ORIGIN, 0.0
```

Xmledit syntax:

```
<PROJCS
    code="projcs-code"
    name="name"
    geogcs="geogcs-code"
    projection="projection-code"
    linunit="linunit-code"
    >
    <PARAMETER code="parameter-code" value="value[:unit-code]" />
    ...
</PROJCS>
```

Examples:

```
<PROJCS
    code="2000"
    name="Anguilla_1957_British_West_Indies_Grid"
    geogcs="4600"
    projection="43006"
    linunit="9001"
    >
    <PARAMETER code="100001" value="400000.0:9001"/>
    <PARAMETER code="100002" value="0.0"/>
    <PARAMETER code="100010" value="-62.0"/>
    <PARAMETER code="100003" value="0.9995"/>
    <PARAMETER code="100021" value="0.0"/>
</PROJCS>

<PROJCS
    code="PE_PCS_ANGUILLA_1957_BRITISH_W_INDIES"
    name="Anguilla_1957_British_West_Indies_Grid"
    geogcs="PE_GCS_ANGUILLA_1957"
    projection="PE_PRJ_TRANSVERSE_MERCATOR"
    linunit="PE_U_METER"
    >
    <PARAMETER code="PE_PAR_FALSE_EASTING" value="400000.0"/>
    <PARAMETER code="PE_PAR_FALSE_NORTHING" value="0.0"/>
    <PARAMETER code="PE_PAR_CENTRAL_MERIDIAN" value="-62.0"/>
    <PARAMETER code="PE_PAR_SCALE_FACTOR" value="0.9995"/>
    <PARAMETER code="PE_PAR_LATITUDE_OF_ORIGIN" value="0.0"/>
</PROJCS>
```

VERTCS – vertical coordinate reference systems

Contents:

- Vertcs-code
- Name
- HVdatum-code (Datum-code or VDatum-code)
- Linunit-code
- Parameter ...
    - Parameter-code
    - Value [and optional unit code]

  Note that parameters may be specified in any order and they are all optional.

Objedit syntax:

```
[VERTCS,] vertcs-code, "name", hvdatum-code, linunit-code
        [, parameter-code, value] ...
```

Examples:

```
3855, "EGM2008_Geoid", 1027, 9001, \
        100006, 0.0, \
        100007, 1.0

PE_VCS_EGM2008_GEOID, \
        "EGM2008_Geoid", \
        PE_VERTD_EGM2008_GEOID, \
        PE_U_METER, \
        PE_PAR_VERTICAL_SHIFT, 0.0, \
        PE_PAR_DIRECTION, 1.0

3855, "EGM2008_Geoid", 1027, 9001

PE_VCS_EGM2008_GEOID, \
        "EGM2008_Geoid", \
        PE_VERTD_EGM2008_GEOID, \
        PE_U_METER
```

Xmledit syntax:

```
<VERTCS
        code="vertcs-code"
        name="name"
        hvdatum="hvdatum-code"
        linunit="linunit-code"
        >
        <PARAMETER code="parameter-code" value="value" />
        ...
</VERTCS>
```

Examples:

```
<VERTCS
        code="3855"
        name="EGM2008_Geoid"
        hvdatum="1027"
        linunit="9001"
        >
        <PARAMETER code="100006" value="0.0"/>
        <PARAMETER code="100007" value="1.0"/>
</VERTCS>

<VERTCS
        code="PE_VCS_EGM2008_GEOID"
        name="EGM2008_Geoid"
        hvdatum="PE_VERTD_EGM2008_GEOID"
        linunit="PE_U_METER"
        >
        <PARAMETER code="PE_PAR_VERTICAL_SHIFT" value="0.0"/>
        <PARAMETER code="PE_PAR_DIRECTION" value="1.0"/>
</VERTCS>
```

GXYZCS – geocentric coordinate reference systems

Contents:

- Gxyzcs-code
- Name
- Datum-code
- Primem-code
- Linunit-code

Objedit syntax:

```
[GXYZCS,] geogcs-code, "name", datum-code, primem-code, linunit-code
```

Examples:

```
3822, "TWD_1997”, 1026, 8901, 9001

PE_XYZ_TWD_1997, \
        "TWD_1997", \
        PE_D_TWD_1997, \
        PE_PM_GREENWICH, \
        PE_U_METER
```

Xmledit syntax:

```
<GEOGCS
        code="geogcs-code"
        name="name"
        datum="datum-code"
        primem="primem-code"
        linunit="linunit-code"
/>
```

Examples:

```
<GXYZCS
        code="3822"
        name="TWD_1997"
        datum="1026"
        primem="8901"
        linunit="9001"
/>
```

<GXYZCS
code="PE_XYZ_TWD_1997"
name="TWD_1997"
datum="PE_D_TWD_1997"
primem="PE_PM_GREENWICH"
linunit="PE_U_METER"
/>

HVCOORDSYS – horizontal/vertical coordinate reference systems

Contents:

- hvcoordsys-code
- Name
- Coordsys-code (geogcs-code or projcs-code) or 0
- Vertcs-code or 0

Objedit syntax:

```
[HVCOORDSYS,] coordsys-code, "name", coordsys-code, vertcs-code
```

Examples:

```
5554, "HVC_ETRS_1989_UTM_Zone_31N_and_DHHN92_Height", \
        25831, 5783

PE_HVC_ETRS_1989_UTM_31N_AND_DHHN92_HT, \
        "HVC_ETRS_1989_UTM_Zone_31N_and_DHHN92_Height", \
        PE_PCS_ETRS_1989_UTM_31N, \
        PE_VCS_DHHN92
```

Xmledit syntax:

```
<HVCOORDSYS
        code="hvcoordsys-code"
        name="name"
        coordsys="coordsys-code"
        vertcs="vertcs-code"
/>
```

Examples:

```
<HVCOORDSYS
        code="5554"
        name="HVC_ETRS_1989_UTM_Zone_31N_and_DHHN92_Height"
        coordsys="25831"
        vertcs="5783"
/>

<HVCOORDSYS
        code="PE_HVC_ETRS_1989_UTM_31N_AND_DHHN92_HT"
        name="HVC_ETRS_1989_UTM_Zone_31N_and_DHHN92_Height"
        coordsys="PE_PCS_ETRS_1989_UTM_31N"
        vertcs="PE_VCS_DHHN92"
/>
```

DATUM – horizontal datums

Contents:

- datum-code
- Name
- Spheroid-code
- [anchor]

Objedit syntax:

```
[DATUM,] datum-code-code, "name", spheroid-code [, anchor]
```

Examples:

```
1024, "D_Hungarian_Datum_1909", 7004

PE_D_HUNGARIAN_DATUM_1909, \
        "D_Hungarian_Datum_1909", \
        PE_S_BESSEL_1841

PE_D_HUNGARIAN_DATUM_1909, \
        "D_Hungarian_Datum_1909", \
        PE_S_BESSEL_1841, \
        "Fundamental point not given in information source, but presumably
Szolohegy which is origin of later HD72."
```

Xmledit syntax:

```
<DATUM
        code="datum-code"
        name="name"
        spheroid="spheroid-code"
/>
```

Examples:

```
<DATUM
	code="1024"
	name="D_Hungarian_Datum_1909"
	spheroid="7004"
/>

<DATUM
	code="PE_D_HUNGARIAN_DATUM_1909"
	name="D_Hungarian_Datum_1909"
	spheroid="PE_S_BESSEL_1841"
/>
```

VDATUM – vertical datums

Contents:

- vdatum-code
- Name
- [anchor]

Objedit syntax:

```
[VDATUM,] datum-code-code, "name" [,anchor]
```

Examples:

```
1027, "EGM2008_Geoid"

PE_VERTD_EGM2008_GEOID, "EGM2008_Geoid"

PE_VERTD_EGM2008_GEOID, "EGM2008_Geoid" , \
        "Derived through EGM2008 geoid undulation model consisting of
spherical harmonic coefficients to degree 2190 and order 2159 applied to the WGS 84
ellipsoid."
```

Xmledit syntax:

```
<VDATUM
        code="vdatum-code"
        name="name"
/>
```

Examples:

```
<VDATUM
        code="1027"
        name="EGM2008_Geoid"
/>

<VDATUM
        code="PE_VERTD_EGM2008_GEOID"
        name="EGM2008_Geoid"
/>
```

GEOGTRAN – geographic transformations

Contents:

- Geogtran-code
- Name
- Geogcs1-code
- Geogcs2-code
- Method-code
- [accuracy]
- Parameter ... (optional)
    - Parameter-code
    - Value [and optional unit] or dataset-name

  Note that parameters may be specified in any order.

Objedit syntax:

```
[GEOGTRAN,] geogtran-code, "name",
        geogcs1-code-code, geogcs2-code, method-code
        [, accuracy] [, parameter-code, value[:unit-code]] ...
```

Examples:

```
1024, "MGI_To_ETRS_1989_4", 4312, 4258, 9607, 1.0, \
        100040, 601.705, \
        100041, 84.263, \
        100042, 485.227, \
        100043, -4.7354, \
        100044, -1.3145, \
        100045, -5.393, \
        100046, -2.3887
```

```
PE_GT_MGI_TO_ETRS_1989_4, \
    "MGI_To_ETRS_1989_4", \
    PE_GCS_MGI, \
    PE_GCS_ETRS_1989, \
    PE_MTH_COORDINATE_FRAME, \
    1.0, \
    PE_PAR_X_AXIS_TRANSLATION, 601.705, \
    PE_PAR_Y_AXIS_TRANSLATION, 84.263, \
    PE_PAR_Z_AXIS_TRANSLATION, 485.227, \
    PE_PAR_X_AXIS_ROTATION, -4.7354, \
    PE_PAR_Y_AXIS_ROTATION, -1.3145, \
    PE_PAR_Z_AXIS_ROTATION, -5.393, \
    PE_PAR_SCALE_DIFFERENCE, -2.3887

PE_GT_RGNC_1991_93_TO_NEA74_NOUMEA_4_NTV2, \
    "RGNC_1991-93_To_NEA74_Noumea_4_NTv2", \
    PE_GCS_RGNC_1991_93, \
    PE_GCS_NEA74_NOUMEA, \
    PE_MTH_NTV2, \
    0.05, \
    PE_PAR_NAME_DATASET, "Dataset_france/RGNC1991_NEA74Noumea"

PE_GT_MAKASSAR_JAKARTA_TO_MAKASSAR, \
    "Makassar_Jakarta_To_Makassar", \
    PE_GCS_MAKASSAR_JAKARTA, \
    PE_GCS_MAKASSAR, \
    PE_MTH_LONGITUDE_ROTATION
```

Xmledit syntax:

```
<GEOGTRAN
    code="geogtran-code"
    name="name"
    geogcs1="geogcs1-code"
    geogcs2="geogcs2-code"
    method="method-code"
    accuracy=”value-in-meters”
    >
    <PARAMETER code="parameter-code" value="value[:unit-code]" />
    ...
</GEOGTRAN>
```

Examples:

```
<GEOGTRAN
        code="1024"
        name="MGI_To_ETRS_1989_4"
        geogcs1="4312"
        geogcs2="4258"
        method="9607"
        accuracy=1.0”
        >
        <PARAMETER code="100040" value="601.705"/>
        <PARAMETER code="100041" value="84.263"/>
        <PARAMETER code="100042" value="485.227"/>
        <PARAMETER code="100043" value="-4.7354"/>
        <PARAMETER code="100044" value="-1.3145"/>
        <PARAMETER code="100045" value="-5.393"/>
        <PARAMETER code="100046" value="-2.3887"/>
</GEOGTRAN>

<GEOGTRAN
        code="PE_GT_MGI_TO_ETRS_1989_4"
        name="MGI_To_ETRS_1989_4"
        geogcs1="PE_GCS_MGI"
        geogcs2="PE_GCS_ETRS_1989"
        method="PE_MTH_COORDINATE_FRAME"
        >
        <PARAMETER code="PE_PAR_X_AXIS_TRANSLATION"
                        value="601.705"/>
        <PARAMETER code="PE_PAR_Y_AXIS_TRANSLATION"
                        value="84.263"/>
        <PARAMETER code="PE_PAR_Z_AXIS_TRANSLATION"
                        value="485.227"/>
        <PARAMETER code="PE_PAR_X_AXIS_ROTATION" value="-4.7354"/>
        <PARAMETER code="PE_PAR_Y_AXIS_ROTATION" value="-1.3145"/>
        <PARAMETER code="PE_PAR_Z_AXIS_ROTATION" value="-5.393"/>
        <PARAMETER code="PE_PAR_SCALE_DIFFERENCE" value="2.3887"/>
</GEOGTRAN>
```

```
<GEOGTRAN
	code="PE_GT_RGNC_1991_93_TO_NEA74_NOUMEA_4_NTV2"
	name="RGNC_1991-93_To_NEA74_Noumea_4_NTv2"
	geogcs1="PE_GCS_RGNC_1991_93"
	geogcs2="PE_GCS_NEA74_NOUMEA"
	method="PE_MTH_NTV2"
	>
	<PARAMETER code="PE_PAR_NAME_DATASET"
			value="Dataset_france/RGNC1991_NEA74Noumea"/>
</GEOGTRAN>

<GEOGTRAN
	code="PE_GT_MAKASSAR_JAKARTA_TO_MAKASSAR"
	name="Makassar_Jakarta_To_Makassar"
	geogcs1="PE_GCS_MAKASSAR_JAKARTA"
	geogcs2="PE_GCS_MAKASSAR"
	method="PE_MTH_LONGITUDE_ROTATION"
/>
```

VERTTRAN – vertical transformations

Contents:

- Verttran-code
- Name
- Coordsys-code (geogcs-code or projcs-code) or 0
- Vertcs1-code
- Vertcs2-code
- VTMethod-code
- [accuracy]
- Parameter ... (optional)
    - Parameter-code
    - Value [and optional unit-code] or dataset-name

  Note that parameters may be specified in any order.

Objedit syntax:

```
[VERTTRAN,] verttran-code, "name",
        coordsys-code,
        vertcs1-code-code, vertcs2-code, vtmethod-code [,accuracy]
        [, parameter-code, value[:unit-code]] ...
```

Examples:

```
4441, "NZVD2009_To_One_Tree_Point_1", 0, 4440, 5767, 129616, 0.03, \
        100060, 0.06

PE_VT_NZVD2009_TO_ONE_TREE_POINT_1, \
        "NZVD2009_To_One_Tree_Point_1", \
        0, \
        PE_VCS_NZVD2009, \
        PE_VCS_ONE_TREE_POINT, \
        PE_VTMTH_VERTICAL_OFFSET, \
        0.03, \
        PE_PAR_VERTICAL_OFFSET, 0.06

PE_VT_NGVD29_TO_NAVD88_NAD27_1_WEST, \
        "NGVD29_To_NAVD88_NAD27_1_West", \
        PE_GCS_NAD_1927, \
        PE_VCS_NGVD_1929, \
        PE_VCS_NAVD_1988, \
        PE_VTMTH_VERTCON, \
        PE_PAR_NAME_DATASET, "Dataset_vertconw.94"
```

Xmledit syntax:

```
<VERTTRAN
        code="verttran-code"
        name="name"
        coordsys="coordsys-code"
        vertcs1="vertcs1-code"
        vertcs2="vertcs2-code"
        method="method-code"
        accuracy=”value-in-meters”
        >
        <PARAMETER code="parameter-code" value="value" />
        ...
</VERTTRAN>
```

Examples:

```
<VERTTRAN
        code="4441"
        name="NZVD2009_To_One_Tree_Point_1"
        vertcs1="4440"
        vertcs2="5767"
        vtmethod="129616"
        accuracy=”accuracy-in-meters”
        >
        <PARAMETER code="100060" value="0.06"/>
</VERTTRAN>

<VERTTRAN
        code="PE_VT_NZVD2009_TO_ONE_TREE_POINT_1"
        name="NZVD2009_To_One_Tree_Point_1"
        vertcs1="PE_VCS_NZVD2009"
        vertcs2="PE_VCS_ONE_TREE_POINT"
        vtmethod="PE_VTMTH_VERTICAL_OFFSET"
        >
        <PARAMETER code="PE_PAR_VERTICAL_OFFSET" value="0.06"/>
</VERTTRAN>
```

<VERTTRAN
code="PE_VT_NGVD29_TO_NAVD88_NAD27_1_WEST"
name="NGVD29_To_NAVD88_NAD27_1_West"
coordsys="PE_GCS_NAD_1927"
vertcs1="PE_VCS_NGVD_1929"
vertcs2="PE_VCS_NAVD_1988"
vtmethod="PE_VTMTH_VERTCON"
>
<PARAMETER code="PE_PAR_NAME_DATASET"
value="Dataset_vertconw.94"/>
</VERTTRAN>

ANGUNIT – angular units of measure

Contents:

- Angunit-code
- Name
- Conversion factor (radians per unit)
- Optional section (UTF8-encoded)
    - Display name
    - Plural name
    - Abbreviation

Objedit syntax:

```
[ANGUNIT,] angunit-code, "name", conversion-factor
        [, "display", "plural", "abbr"]
```

Examples:

```
9101, "Radian", 1.0

PE_U_DEGREE, \
        "Degree", \
        0.0174532925199433, \
        "Degree", \
        "Degrees", \
        "deg"
```

Xmledit syntax:

```
<ANGUNIT
        code="angunit-code"
        name="name"
        factor="conversion-factor"
        [ display="display-name" ]
        [ plural="plural-name" ]
        [ abbr="abbreviation" ]
/>
```

Examples:

```
<ANGUNIT code="9101" name="Radian" factor="1.0" />

<ANGUNIT
	code="PE_U_DEGREE"
	name="Degree"
	factor="0.0174532925199433"
	display="Degree"
	plural="Degrees"
	abbr="deg"
/>
```

LINUNIT – linear units of measure

Contents:

- Linunit-code
- Name
- Conversion factor (meters per unit)
- Optional section (UTF8-encoded)
    - Display name
    - Plural name
    - Abbreviation
    - Areaunit reference code or 0

Objedit syntax:

```
[LINUNIT,] linunit-code, "name", conversion-factor
    [, "display", "plural", "abbr", areaunit-reference]
```

Examples:

```
9001, "Meter", 1.0


PE_U_CENTIMETER, \
    "Centimeter", \
    0.01, \
    "Centimeter", \
    "Centimeters", \
     "cm", \
    PE_AU_SQUARE_CENTIMETER
```

Xmledit syntax:

```
<LINUNIT
    code="linunit-code"
    name="name"
    factor="conversion-factor"
    [ display="display-name" ]
    [ plural="plural-name" ]
    [ abbr="abbreviation" ]
    [ refcode="areaunit-reference" ]
/>
```

Examples:

```
<LINUNIT code="9001" name="Meter" factor="1.0" />

<LINUNIT
        code="PE_U_CENTIMETER"
        name="Centimeter"
        factor="0.01"
        display="Centimeter"
        plural="Centimeters"
        abbr="cm"
        refcode="PE_AU_SQUARE_CENTIMETER"
/>
```

AREAUNIT – area units of measure

Contents:

- Areaunit-code
- Name
- Conversion factor (square-meters per unit)
- Optional section (UTF8-encoded)
    - Display name
    - Plural name
    - Abbreviation
    - Linunit reference code or 0

Objedit syntax:

```
[AREAUNIT,] areaunit-code, "name", conversion-factor
        [, "display", "plural", "abbr", linunit-reference]
```

Examples:

```
109404, "Square_Meter", 1.0

PE_AU_SQUARE_FOOT_US, \
        "Square_Foot_US", \
        0.09290341161327487, \
        "Square US Survey Foot", \
        "Square US Survey Feet", \
        "sq ftUS", \
        PE_U_FOOT_US

PE_AU_ACRE_US, "Acre_US", 4046.87260987425, \
        "US Acre", "US Acres", "acUS", 0
```

Xmledit syntax:

```
<AREAUNIT
        code="areaunit-code"
        name="name"
        factor="conversion-factor"
        [ display="display-name" ]
        [ plural="plural-name" ]
        [ abbr="abbreviation" ]
        [ refcode="linunit-reference" ]
/>
```

Examples:

```
<AREAUNIT code="109404" name="Square_Meter" factor="1.0" />

<AREAUNIT code="PE_AU_SQUARE_FOOT_US"
    name="Square_Foot_US"
    factor="0.09290341161327487"
    display="Square US Survey Foot"
    plural="Square US Survey Feet"
    abbr="sq ftUS"
    refcode="PE_U_FOOT_US"
/>

<AREAUNIT code="PE_AU_ACRE_US" name="Acre_US"
    factor="4046.87260987425"
    display="US Acre"
    plural="US Acres"
    abbr="acUS"
    refcode="0"
/>
```

SCALEUNIT – scale units of measure

Contents:

- Scaleunit-code
- Name
- Conversion factor (to UNITY)
- Optional section (UTF8-encoded)
    - Display name
    - Plural name
    - Abbreviation

Objedit syntax:

```
[SCALEUNIT,] scaleunit-code, "name", conversion-factor
        [, "display", "plural", "abbr"]
```

Examples:

```
9202, "Parts_Per_Million", 0.000001

PE_SU_PPM, "Parts_Per_Million", 0.000001
```

Xmledit syntax:

```
<SCALEUNIT
        code="scaleunit-code"
        name="name"
        factor="conversion-factor"
        [ display="display-name" ]
        [ plural="plural-name" ]
        [ abbr="abbreviation" ]
/>
```

Examples:

```
<SCALEUNIT
        code="9202"
        name="Parts_Per_Million"
        factor="0.000001"
/>

<SCALEUNIT
        code="PE_SU_PPM"
        name="Parts_Per_Million"
        factor="0.000001"
/>
```

PRIMEM – prime meridians

Contents:

- primem-code
- Name
- Longitude (degrees +east/-west from IRM/Greenwich)
  Value must be in the range -180 to +180.
- [Angular unit]

Objedit syntax:

```
[PRIMEM,] primem-code, "name", longitude [, angular-unit]
```

Examples:

```
8901, "Greenwich", 0.0
PE_PM_GREENWICH, "Greenwich", 0.0, PE_U_ANGUNIT
PE_PM_PARIS, "Paris", 2.337229166666667
```

Xmledit syntax:

```
<PRIMEM
        code="primem-code"
        name="name"
        longitude="longitude"
        angunit=”angular-unit”
/>
```

Examples:

```
<PRIMEM code="8901" name="Greenwich" longitude="0.0"/>
<PRIMEM code="PE_PM_GREENWICH" name="Greenwich" longitude="0.0"/>
<PRIMEM code="PE_PM_PARIS" name="Paris"
                longitude="2.337229166666667"/>
```

SPHEROID – spheroids (ellipsoids)

Contents:

- spheroid-code
- Name
- Semi-major axis (in meters)
- Inverse flattening
  Value must be 0 (if a sphere) or greater than 1.
- [Linear-unit]

Objedit syntax:

[SPHEROID,] spheroid-code, "name", semi-major-axis, inv-flattening [, linear-unit]

Examples:

1024, "CGCS2000", 6378137.0, 298.257222101

PE_S_CGCS2000, "CGCS2000", 6378137.0, 298.257222101, PE_U_METER

PE_S_SPHERE, "Sphere", 6371000.0, 0.0

Xmledit syntax:

<SPHEROID
code="spheroid-code"
name="name"
axis="semi-major-axis"
flattening="inverse-flattening"
linunit=”linear-unit”
/>

Examples:

<SPHEROID
code="1024"
name="CGCS2000"
axis="6378137.0"
flattening="298.257222101"
/>

```
<SPHEROID
	code="PE_S_CGCS2000"
	name="CGCS2000"
	axis="6378137.0"
	flattening="298.257222101"
/>

<SPHEROID
	code="PE_S_SPHERE"
	name="Sphere"
	axis="6371000.0"
	flattening="0.0"
/>
```

EXTENT – extents (areas-of-use)

An EXTENT defines an area-of-interest by specifying the lower-left and upper-right latitude/longitude corners of a bounding box enclosing the area. These values are in degrees with an assumed prime meridian of Greenwich. The area-code is not a WKID; there is a separate table (the area-code table) that associates WKIDs and area-codes. Latitude values range from -90 to +90 degrees. Longitude values range from -180 to +180 degrees. Note that an extent may span the dateline, so the left longitude may be greater than the right longitude.

Most of the latitude/longitude values in both the Esri and EPSG datasets have a precision at most of .01 degrees. This corresponds to a distance (at the equator) of over 1KM, which is much greater than any datum shift. Thus, datum shifts are ignored for these values. In other words, the latitude/longitude values are considered valid in any geographic coordinate system.

NOTE: The order of latitude/longitude values in this record is different from the values in the legacy (pre-version 10.4) METADATA record. See Appendix D for details.

Contents:

- Area-code
- Area-name
- South-latitude (slat)
- North-latitude (nlat)
- Left-longitude (llon)
- Right-longitude (rlon)

Objedit syntax:

```
[EXTENT,] area-code, "name", slat, nlat, llon, rlon
```

Examples:

```
1024, "Afghanistan", 29.4, 38.48, 60.5, 74.92
```

Xmledit syntax:

```
<EXTENT
    areacode="area-code"
    name="area-name"
    slat="south-latitude"
    nlat="north-latitude"
    llon="left-longitude"
    rlon="right-longitude"
/>
```

Examples:

```
<EXTENT
    areacode="1024"
    name="Afghanistan"
    slat="29.4"
    nlat="38.48"
    llon="60.5"
    rlon="74.92"
/>
```

Record formats for non-data tables

The records for non-data types all have the same syntax for each object-type. Note that in the descriptions and examples below, extra whitespace and newlines are added for clarity.

CODECHANGE – code-changes

A CODECHANGE record describes a new code that should be used instead of a given code when doing any subsequent lookups. Code-changes do not "chain"; you cannot have a record showing "A → B" and another showing "B → C" and expect A to change to C. In such a case, the former record should be changed to "A → C".

Code changes can occur due to the following:

- There is a change in authority. For instance, Esri adds a projected CRS using an Esri-range WKID. Later, the same projected CRS is added to the EPSG geodetic registry with an EPSG-range WKID. In this case, the Esri WKID is marked as a code change to the EPSG WKID.
- There is a substantial error in an EPSG entry. In this case, EPSG will mark the entry as deprecated, and replace it with a corrected entry. The corrected entry will have a new EPSG WKID, and the old WKID is marked as a code change to the new WKID.
- Esri implemented an EPSG entry incorrectly. If it is mathematically wrong, the wrong entry is copied to an Esri-range WKID and has its name changed. The EPSG entry is corrected.

Contents:

- old-code
- new-code

Objedit syntax:

[*OBJECT-TYPE*,] old-code, new-code

Examples:

DATUM, 106001, PE_D_WGS_1966
PROJCS, 102100, 3857
102100, PE_PCS_WGS_1984_WEB_MERCATOR_AUXSPHERE

Xmledit syntax:

```
<OBJECT-TYPE
        old_code="old-code"
        new_code="new-code"
/>
```

Examples:

```
<DATUM
        old_code="106001"
        new_code="PE_D_WGS_1966"
/>

<PROJCS
        old_code="102100"
        new_code="3857"
/>

<PROJCS
        old_code="102100"
        new_code="PE_PCS_WGS_1984_WEB_MERCATOR_AUXSPHERE"
/>
```

DEPRECATED – deprecations

A DEPRECATED record describes a non-normal status associated with a WKID. A status of "discontinued" means the use of that WKID is deprecated and there is probably a better WKID to use. A status of "dead" means that WKID should not be used at all and is only kept for backward compatibility. Note that "dead" WKIDs have no macro (DEFSTRING entry) associated with them.

A status of "code-change" will override any entry in this table. For example, GEOGTRAN 8260 is a code-change to 1328, and 1328 is discontinued. Thus, 8260 will have a status of "code-change" and 1328 will have a status of "discontinued".

Contents:

- code
- status ("discontinued" or "dead")

Objedit syntax:

[*OBJECT-TYPE*,] code, "status"

Examples:

GEOGTRAN, 1086, dead
PE_GT_SAMBOJA_TO_WGS_1984, discontinued

Xmledit syntax:

<*OBJECT-TYPE*
    code="code"
    status="status"
/>

Examples:

<GEOGTRAN
    code="1086"
    status="dead"
/>

<GEOGTRAN
    code="PE_GT_SAMBOJA_TO_WGS_1984"
    status="discontinued"
/>

SYNONYM – synonyms

SYNONYM records are used in two different ways:

- The PE factory can look up a database record by name as well as by WKID. If a name-lookup is done, first a check is made to see if the name is actually a synonym, and, if so, the actual, or canonical, name is used to do the lookup.
- When processing WKT strings, all names are checked to see if they are a synonym. If the name is a synonym, the canonical name is stored in the object so comparisons will work properly. If a new WKT is produced from such an object, the original synonym is written out.

The "mistake" field indicates that there was a mistake in the original name. In such a case, the synonym-name is not written back out in a WKT string. If "mistake" is not specified, it is assumed to be false.

A record may also have an authority associated with it. A null or empty authority means that this synonym should always be checked. If an authority is specified, then that synonym will only be checked if you are processing a WKT string using that authority.

Contents:

- synonym-name
- actual-name
- [mistake T[RUE] or F[ALSE]]
- [authority]

Objedit syntax:

[*OBJECT-TYPE*,] "syn-name", "act-name" [, mistake [, "authority"]]

Examples:

GEOGCS, "AGD66", "Australian_1966", F, "EPSG"

"GCS_K0_1949", "GCS_Kerguelen_Island_1949", TRUE

Xmledit syntax:

```
<OBJECT-TYPE
    syn_name="synonym-name"
    act_name="actual-name"
    [ mistake="T[RUE]|F[ALSE]" ]
    [ authority="authority" ]
/>
```

Examples:

```
<GEOGCS
    syn_name="AGD66"
    act_name="Australian_1966"
    mistake="FALSE"
    authority="EPSG"
/>

<GEOGCS
    syn_name="GCS_K0_1949"
    act_name="GCS_Kerguelen_Island_1949"
    mistake="TRUE"
/>
```

CODERANGE – code-ranges

A CODERANGE entry describes a range of valid WKIDs for a given object-type. The "authority" field names the source for this range, and there may be an optional version associated with it. Code-ranges for a particular object-type may not overlap. User-defined code-ranges are processed as additions only to the list of code-ranges. A user entry may not override or overlap an already defined entry. See the section on Code-ranges for more details.

Contents:

- First-code
- Last-code
- Authority
- [Version]

Objedit syntax:

[*OBJECT-TYPE*,] first-code, last-code, "authority" [, "version"]

Examples:

PROJCS, 50000, 52999, "SHELL"

PROJCS, 302000, 302999, "Tin Man Oil", "12.6"

Xmledit syntax:

```
<OBJECT-TYPE
        min_code="first-code"
        max_code="last-code"
        authority="authority"
        [ version="version" ]
/>
```

Examples:

```
<PROJCS
        min_code="50000"
        max_code="52999"
        authority="SHELL"
/>
```

```
<PROJCS
	min_code="302000"
	max_code="302999"
	authority="Tin Man Oil"
	version="12.1.2"
/>
```

DISPNAME – display names

All objects have a name (which may be empty). This name is always an ASCII name, and it is used when comparing objects. An object may also have a display name, which may be a Unicode name. This display-name is defined in a DISPNAME record.

A display-name table is typically used when localizing database entries. A DISPNAME record doesn't change the object the WKID refers to, it simply adds a display-name to the object. The display-name is ignored when comparing objects.

Contents:

- code
- UTF-8-encoded display name

Objedit syntax:

[*OBJECT-TYPE*,] code, disp-name

Examples:

```
PROJCS, PE_PCS_TOKYO_JAPAN_9, "平面直角座標系 第 9 系"
```

Xmledit syntax:

```
<OBJECT-TYPE
        code="code"
        name="display-name"
/>
```

Examples:

```
<PROJCS
        code="PE_PCS_TOKYO_JAPAN_9"
        name="平面直角座標系 第 9 系"
/>
```

DESCRIPTION – descriptions

The DESCRIPTION table provides an extended description for an object in addition to its normal name. This description may be localized.

Contents:

- code
- UTF8-encoded description

Objedit syntax:

[*OBJECT-TYPE*,] code, description

Examples:

```
GEOGCS, 3819, "Hungarian Datum 1909"
PE_GCS_HD1909, "Hungarian Datum 1909"
```

Xmledit syntax:

```
<OBJECT-TYPE
        code="code"
        desc="description"
/>
```

Examples:

```
<GEOGCS
        code="3819"
        desc="Hungarian Datum 1909"
/>

<GEOGCS
        code="PE_GCS_HD1909"
        desc="Hungarian Datum 1909"
/>
```

DEFSTRING – macro definitions

The DEFSTRING table provides users with the ability to define their own macros for a particular code. This macro can then be used in the various other tables instead of the numeric value. This makes for more readable data files.

It is recommended that user-defined macros begin with a different prefix than "PE_", as that makes it easy to recognize whether a macro is a PE-defined one or a user-defined one. For example, the Tin Man Oil Company might use TMO_WELL_20 rather than PE_WELL_20 or just WELL_20.

Contents:

- code
- macro-name

Objedit syntax:

[*OBJECT-TYPE*,] code, macro-name

Examples:

GEOGCS, 3819, PE_GCS_HD1909

PROJCS, 202001, TMO_WELL_20

Xmledit syntax:

<*OBJECT-TYPE*
code="code"
macro="macro-name"
/>

Examples:

<GEOGCS
code="3819"
macro=" PE_GCS_HD1909"
/>

<PROJCS
code="202001"
macro=" TMO_WELL_20"
/>

AREAINFO – coordinate-system dialog entries

The AREAINFO tables describe the entries used to create the "Coordinate Systems" dialog in ArcGIS for Desktop and ArcGIS Pro. These tables take the place of the *.prj files in the "Coordinate Systems" folder in pre-10.0 versions of ArcMap. As such, the "category" corresponds to the folder-tree the file is in, and the "area-name" corresponds to the filename. Both of these fields are localizable. Multiple records may reference the same WKID.

Unlike all other tables in the database, which are considered indexed by WKID, these tables have no order associated with them. The intent is to show these records in some logical (usually sorted by category/area-name) order, but since the names may be localized, we have no way of knowing how to sort them.

Thus, these tables are simply read in and presented to the user in the same order they appear in the tables. Thus, any user-defined entries will appear ahead of any "builtin" entries. When localizing, the practice is to localize the entire table rather than just selected entries; otherwise you will see duplicate entries appear in the list. In this case, you would load the user-defined table and turn off the loading of the builtin areainfo table. See Example 3 in the section Defining the databases to load for an example of doing this.

The "category" field may contain sub-categories, separated by a slash (/). There is a max of eight sub-categories.

NOTE: The order of category/area-name values in this record is different from the values in the legacy (pre-version 10.4) AREAINFO record. See Appendix D for details.

Contents:

- code
- UTF8-encoded category
- UTF8-encoded area-name

Objedit syntax:

[*OBJECT-TYPE*,] code, category, area-name

Examples:

```
GEOGCS, PE_GCS_ABIDJAN_1987, \
        "Geographic Coordinate Systems/Africa", \
        "Abidjan 1987"
```

Xmledit syntax:

```
<OBJECT-TYPE
    code="code"
    category="category
    areaname="area-name"
/>
```

Examples:

```
<GEOGCS
    code="PE_GCS_ABIDJAN_1987"
    category="Geographic Coordinate Systems/Africa"
    areaname="Abidjan 1987"
/>
```

AREACODE – areacodes

The AREACODE tables associate a WKID with an area-code. Many WKIDs may reference the same area-code.

Contents:

- WKID
- area-code
- [index if more than one WKID]

Objedit syntax:

[*OBJECT-TYPE*,] WKID, area-code [, index]

Examples:

```
GEOGTRAN, PE_GT_MGI_TO_ETRS_1989_4, 1543

PE_PCS_ANGUILLA_1957_BRITISH_W_INDIES, 3214
```

Xmledit syntax:

```
<OBJECT-TYPE
        code="WKID"
        areacode="area-code"
        [ index="index" ]
/>
```

Examples:

```
<GEOGTRAN
        code="PE_GT_MGI_TO_ETRS_1989_4"
        areacode="1543"
/>

<PROJCS
        code="PE_PCS_ANGUILLA_1957_BRITISH_W_INDIES"
        areacode="3214"
/>
```

VERSION – versions

The VERSION tables provide the ability to specify particular authority/versions for each WKID. When a lookup of a WKID is made, the authority and version for that WKID is determined by the following steps:

1. Use the information in a VERSION entry for that WKID if there is one.
2. Use the default AUTHORITY and VERSION defined for a database, if specified.
3. Use the AUTHORITY and VERSION information in the CODERANGE entry for that WKID, if non-empty.
4. Use "CUSTOM" as the authority and NULL for the version.

Note: Currently, ArcGIS for Desktop and ArcGIS Pro displays only any authority name (not the version).

Contents:

- code
- authority
- [version]

Objedit syntax:

[*OBJECT-TYPE*,] code, authority [, version]

Examples:

GEOGCS, PE_GCS_HD1909, "EPSG", "6.17.1"

Xmledit syntax:

<*OBJECT-TYPE*
    code="code"
    authority="authority"
    [version="version"]
/>

Examples:

<GEOGCS
    code="PE_GCS_HD1909"
    authority="EPSG"
    version="6.17.1"
/>

EXCEPTION – gtlist/vtlist exceptions

The EXCEPTION tables provide the ability to specify particular transformations (geographic or vertical) for a given set of coordinate systems. This will override the automatic choosing of a default transformation. This table is only used if the user did not specify an extent for his data whan doing a transformation lookup.

Contents:

- code1 (GCS or VCS code)
- code2 (GCS or VCS code)
- 1st grid-based transformation code (GTF or VTF)
- 1st grid-based direction (0 = forward, 1 = inverse)
- 2nd grid-based transformation code if needed (GTF or VTF or 0)
- 2nd grid-based direction (0 = forward, 1 = inverse)
- [1st non-grid-based transformation code (GTF or VTF)]
- [1st non-grid-based direction (0 = forward, 1 = inverse)]
- [2nd non-grid-based transformationcode if needed (GTF or VTF or 0)]
- [2nd non-grid-based direction (0 = forward, 1 = inverse)]

If the non-grid-based entries are not present, they are assumed to be the same as the grid-based entries.

Objedit syntax:

```
[OBJECT-TYPE,] code1, code2,
        grid-code1, grid-dir1, grid-code2, grid-dir2,
        non-grid-code1, non-grid-dir1, non-grid-code2, non-grid-dir2
```

Examples:

```
4202, 4283, 1803, 0, 0, 0, 15979, 0, 0, 0

PE_GCS_AGD_1966, PE_GCS_GDA_1994, \
        PE_GT_AGD_1966_TO_GDA_1994_11_NTV2, 0, 0, 0, \
        PE_GT_AGD_1966_TO_GDA_1994_12, 0, 0, 0
```

Xmledit syntax:

```
<OBJECT-TYPE
        code1="code1"
        code2="code2"
        grid-code1="grid_code1"
        grid-dir1="grid-dir1"
        grid-code2="grid-code2"
        grid-dir2="grid-dir2"
        [ nong-code1="non-grid_code1" ]
        [ nong-dir1="non-grid-dir1" ]
        [ nong-code2="non-grid-code2" ]
        [ nong-dir2="non-grid-dir2" ]
/>
```

Examples:

```
<GEOGTRAN
        code1="PE_GCS_AGD_1966"
        code2="PE_GCS_GDA_1994"
        grid-code1="PE_GT_AGD_1966_TO_GDA_1994_11_NTV2"
        grid-dir1="1803"
        grid-code2="0"
        grid-dir2="0"
        nong-code1="PE_GT_AGD_1966_TO_GDA_1994_12"
        nong-dir1="15979"
        nong-code2="0"
        nong-dir2="0"
/>
```

GCSVCS – GCS-VCS equivalences

The GCSVCS table is used to correlate GCS, GCS3D, and VCS codes.

Contents:

- gcs_code
- gcs3d_code (or 0)
- xyz_code (or 0)
- vcs_code (or 0)

Objedit syntax:

```
[OBJECT-TYPE,] gcs_code, gcs3d-code, xyz_code, vcs_code
```

Examples:

```
3824, 3823, 3822, 15800

PE_GCS_TWD_1997,  \
        PE_GCS_TWD_1997_3D,   \
        PE_XYZ_TWD_1997, \
        PE_VCS_TWD_1997                  },
```

Xmledit syntax:

```
<OBJECT-TYPE
        gcs_code="gcs-code"
        g3d-code=”gcs3d-code”
        xyz-code=”xyz-code”
        vcs_code="vcs-code"  />
```

Examples:

```
<GEOGCS
        gcs_code=” PE_GCS_TWD_1997”
        g3d_code=”PE_GCS_TWD_1997_3D”
        xyz_code=”PE_XYZ_TWD_1997”
        vcs_code=”PE_VCS_TWD_1997”
/>
```

AXIS - axis codes

Rather than follow the EPSG paradigm of separate axis entries for each possible combination of coord-order, coord-type, and unit, we just have a single axis-code that describes all the axes for that coordinate system.

The PE does not directly use axis information, as all points we deal with are (lon/x, lat/y). But it can be used by higher-level routines that draw a map or read user coordinate points.

Note that the table need only contain entries that are not the default.

This is the axis-code table:

code	value	dim	cs-type	1st-coord	2nd-coord	3rd-coord	default
U	1	1	VCS	Z-UP	-	-	yes
D	2	1	VCS	Z-DN	-	-	
EN	11	2	GCS/PCS	X-EAST	Y-NORTH	-	yes
WN	12	2	GCS/PCS	X-WEST	Y-NORTH	-	
ES	13	2	GCS/PCS	X-EAST	Y-SOUTH	-	
WS	14	2	GCS/PCS	X-WEST	Y-SOUTH	-	
NE	15	2	GCS/PCS	Y-NORTH	X-EAST	-	
NW	16	2	GCS/PCS	Y-NORTH	X-WEST	-	
SE	17	2	GCS/PCS	Y-SOUTH	X-EAST	-	
SW	18	2	GCS/PCS	Y-SOUTH	X-WEST	-	
ENU	21	3	GCS/PCS	X-EAST	Y-NORTH	Z-UP	yes
WNU	22	3	GCS/PCS	X-WEST	Y-NORTH	Z-UP	
ESU	23	3	GCS/PCS	X-EAST	Y-SOUTH	Z-UP	
WSU	24	3	GCS/PCS	X-WEST	Y-SOUTH	Z-UP	
NEU	25	3	GCS/PCS	Y-NORTH	X-EAST	Z-UP	
NWU	26	3	GCS/PCS	Y-NORTH	X-WEST	Z-UP	
SEU	27	3	GCS/PCS	Y-SOUTH	X-EAST	Z-UP	
SWU	28	3	GCS/PCS	Y-SOUTH	X-WEST	Z-UP	
END	31	3	GCS/PCS	X-EAST	Y-NORTH	Z-DN	
WND	32	3	GCS/PCS	X-WEST	Y-NORTH	Z-DN	
ESD	33	3	GCS/PCS	X-EAST	Y-SOUTH	Z-DN	
WSD	34	3	GCS/PCS	X-WEST	Y-SOUTH	Z-DN	
NED	35	3	GCS/PCS	Y-NORTH	X-EAST	Z-DN	
NWD	36	3	GCS/PCS	Y-NORTH	X-WEST	Z-DN	
SED	37	3	GCS/PCS	Y-SOUTH	X-EAST	Z-DN	
SWD	38	3	GCS/PCS	Y-SOUTH	X-WEST	Z-DN	

Contents:

- cs-code
- axis-code

Objedit syntax:

```
[OBJECT-TYPE,] cs-code, axis-code
```

Examples:

```
PE_GCS_HD1909, NE
PE_PCS_S_JTSK_FERRO_KROVAK, SW
PE_VCS_BLACK_SEA_D, D
```

Xmledit syntax:

```
<OBJECT-TYPE
        cs-code="cs-code"
        axis-code="axis-code"/>
```

Examples:

```
<PROJCS
        cs-code=”PE_PCS_S_JTSK_FERRO_KROVAK”
        axis-code=”SW”
/>
```

ENSEMBLE – datum ensembles

Datum ensembles are a way of grouping similar datums and treating them as “essentially” equivalent. This is usually used when dealing with various realizations (such as the NAD83, WGS84, and ITRF series).

This mechanism is used when looking up possible transformations between two CRSs. Note that, even if such an ensemble exists, the user still has to specifically request that ensembles be taken into considering transformation paths.

Contents:

- datum-bucket-code
- datum-member-code

Objedit syntax:

[*OBJECT-TYPE*,] bucket-code, member-code

Examples:

```
PE_D_WGS_1984, PE_D_WGS_1984_G730
PE_D_WGS_1984, PE_D_WGS_1984_G873
PE_D_WGS_1984, PE_D_WGS_1984_G1150
PE_D_WGS_1984, PE_D_WGS_1984_G1674
PE_D_WGS_1984, PE_D_WGS_1984_G1762
PE_D_WGS_1984, PE_D_WGS_1984_TRANSIT
```

Xmledit syntax:

```
<DATUM
    bucket=”bucket-code"
    member="member-code"
/>
```

Examples:

```
<DATUM bucket="PE_D_WGS_1984" member="PE_D_WGS_1984_G730"/>
<DATUM bucket="PE_D_WGS_1984" member="PE_D_WGS_1984_G873"/>
<DATUM bucket="PE_D_WGS_1984" member="PE_D_WGS_1984_G1150"/>
<DATUM bucket="PE_D_WGS_1984" member="PE_D_WGS_1984_G1674"/>
<DATUM bucket="PE_D_WGS_1984" member="PE_D_WGS_1984_G1762"/>
<DATUM bucket="PE_D_WGS_1984" member="PE_D_WGS_1984_TRANSIT"/>
```

EPOCH – dynamic CRS epochs

This table is used to specify the epoch and optionally the deformation model used. This information is currently not used by the PE, but it available for users who want that information.

Contents:

- CRS-code
- Epoch (decimal date)
- Deformation-model (optional)

Objedit syntax:

[*OBJECT-TYPE*,] code, epoch [, model]

Examples:

PE_GCS_WGS_1984_G730, 1990.5, "NNR-NUVEL1"
PE_GCS_WGS_1984_G873, 1995.5, "NNR-NUVEL1A"
PE_GCS_WGS_1984_G1150, 1999.5, "NNR-NUVEL1A"
PE_GCS_WGS_1984_G1674, 2007.5, "ITRF2008-PMM"
PE_GCS_WGS_1984_G1762, 2012.5, "ITRF2008-PMM"

Xmledit syntax:

<OBJECT-TYPE
code=”code"
epoch="epoch"
[model=”model”]
/>

Examples:

<GEOGCS code=”PE_GCS_WGS_1984_G730” epoch=”1990.5”
model="NNR-NUVEL1"/>
<GEOGCS code=”PE_GCS_WGS_1984_G873” epoch=”1995.5”
model="NNR-NUVEL1A"/>
<GEOGCS code=”PE_GCS_WGS_1984_G1150” epoch=”1999.5”
model="NNR-NUVEL1A"/>
<GEOGCS code=”PE_GCS_WGS_1984_G1674” epoch=”2007.5”
model="ITRF2008-PMM"/>

Appendix A: Selected macros and well-known IDs

Linear units

Name	Macro	WKID
150 kilometer length	PE_U_KM150	109031
50 kilometer length	PE_U_KM50	109030
British Foot (1936)	PE_U_FOOT_BRITISH_1936	9095
Centimeter	PE_U_CENTIMETER	109006
Chain (Benoit 1895 A)	PE_U_CHAIN_BENOIT_A	9052
Chain (Benoit 1895 B)	PE_U_CHAIN_BENOIT_B	9062
Chain (Clarke)	PE_U_CHAIN_CLARKE	9038
Chain (Sears 1922 Truncated)	PE_U_CHAIN_SEARS_1922_TRUNC	9301
Chain (Sears)	PE_U_CHAIN_SEARS	9042
Clarke's foot	PE_U_FOOT_CLARKE	9005
Decimeter	PE_U_DECIMETER	109005
Desktop Publishing Point (1/72 of an international inch)	PE_U_DTP_POINT	109016
Fathom	PE_U_FATHOM	9014
Foot (1865)	PE_U_FOOT_1865	9070
Foot (Benoit 1895 A)	PE_U_FOOT_BENOIT_A	9051
Foot (Benoit 1895 B)	PE_U_FOOT_BENOIT_B	9061
Foot (Sears 1922 Truncated)	PE_U_FOOT_SEARS_1922_TRUNC	9300
German legal meter	PE_U_METER_GERMAN	9031
Gold Coast Foot	PE_U_FOOT_GOLD_COAST	9094
Indian foot (1937)	PE_U_FOOT_INDIAN_1937	9081
Indian foot (1962)	PE_U_FOOT_INDIAN_1962	9082
Indian foot (1975)	PE_U_FOOT_INDIAN_1975	9083
Indian geodetic foot	PE_U_FOOT_INDIAN	9080
Indian yard	PE_U_YARD_INDIAN	9084
Indian yard (1937)	PE_U_YARD_INDIAN_1937	9085
Indian yard (1962)	PE_U_YARD_INDIAN_1962	9086
Indian yard (1975)	PE_U_YARD_INDIAN_1975	9087
International Chain	PE_U_CHAIN	9097
International foot	PE_U_FOOT	9002
International inch	PE_U_INCH	109008
International Link	PE_U_LINK	9098
International meter	PE_U_METER	9001
International nautical mile	PE_U_NAUTICAL_MILE	9030
International rod	PE_U_ROD	109010

International Yard	PE_U_YARD	9096
Kilometer	PE_U_KILOMETER	9036
Link (Benoit 1895 A)	PE_U_LINK_BENOIT_A	9053
Link (Benoit 1895 B)	PE_U_LINK_BENOIT_B	9063
Link (Clarke's ratio)	PE_U_LINK_CLARKE	9039
Link (Sears 1922 Truncated)	PE_U_LINK_SEARS_1922_TRUNC	9302
Link (Sears)	PE_U_LINK_SEARS	9043
Micrometer	PE_U_MICROMETER	109017
Millimeter	PE_U_MILLIMETER	109007
Nanometer	PE_U_NANOMETER	109018
Sears' foot	PE_U_FOOT_SEARS	9041
Smoot, Height of Oliver Smoot, used to measure the Harvard Bridge	PE_U_SMOOT	109014
Statute mile	PE_U_MILE_STATUTE	9093
UK nautical mile (pre-1970)	PE_U_NAUTICAL_MILE_UK	109013
US nautical mile (pre-1954)	PE_U_NAUTICAL_MILE_US	109012
US survey chain	PE_U_CHAIN_US	9033
US survey foot	PE_U_FOOT_US	9003
US survey inch	PE_U_INCH_US	109009
US survey link	PE_U_LINK_US	9034
US survey mile	PE_U_MILE_US	9035
US survey rod	PE_U_ROD_US	109011
US survey yard	PE_U_YARD_US	109002
Vara, old Spanish unit of distance used in Texas (33 1/3 inches)	PE_U_VARA_US	109015
Yard (Benoit 1895 A)	PE_U_YARD_BENOIT_A	9050
Yard (Benoit 1895 B)	PE_U_YARD_BENOIT_B	9060
Yard (Clarke)	PE_U_YARD_CLARKE	9037
Yard (Sears 1922 Truncated)	PE_U_YARD_SEARS_1922_TRUNC	9099
Yard (Sears)	PE_U_YARD_SEARS	9040

Angular units

Name	**Macro**	**WKID**
Arc-minute	PE_U_MINUTE	9103
Arc-second	PE_U_SECOND	9104
Centesimal minute (1/100th Gon (Grad))	PE_U_MINUTE_CENTESIMAL	9112
Centesimal second (1/10000th Gon (Grad))	PE_U_SECOND_CENTESIMAL	9113
Degree	PE_U_DEGREE	9102
Gon (angle subtended by 1/400 circle)	PE_U_GON	9106
Grad (angle subtended by 1/400 circle)	PE_U_GRAD	9105

Microradian (1e-6 radian)	PE_U_MICRORADIAN	9109
Mil (angle subtended by 1/6400 circle)	PE_U_MIL_6400	9114
Milliarcsecond (1/1000 Arc-second)	PE_U_MILLIARCSECOND	1031
Radian	PE_U_RADIAN	9101

Map projections

Name	Macro	WKID
Adams_Square_II	PE_PRJ_ADAMS_SQUARE_II	43087
Aitoff	PE_PRJ_AITOFF	43043
Albers	PE_PRJ_ALBERS	43007
Aspect_Adaptive_Cylindrical	PE_PRJ_ASPECT_ADAPTIVE_CYLINDRICAL	43083
Azimuthal_Equidistant	PE_PRJ_AZIMUTHAL_EQUIDISTANT	43032
Bartholomew Times	PE_PRJ_TIMES	43048
Behrmann	PE_PRJ_BEHRMANN	43017
Berghaus Star	PE_PRJ_BERGHAUS_STAR	43060
Bonne	PE_PRJ_BONNE	43024
Cassini	PE_PRJ_CASSINI	43028
Compact_Miller	PE_PRJ_COMPACT_MILLER	43080
Craster Parabolic	PE_PRJ_CRASTER_PARABOLIC	43046
Cube	PE_PRJ_CUBE	43055
Cylindrical Equal Area	PE_PRJ_CYLINDRICAL_EQAREA	43034
Double Stereographic	PE_PRJ_DOUBLE_STEREOGRAPHIC	43038
Eckert Greifendorff	PE_PRJ_GREIFENDORFF	43073
Eckert I	PE_PRJ_ECKERT_I	43015
Eckert II	PE_PRJ_ECKERT_II	43014
Eckert III	PE_PRJ_ECKERT_III	43013
Eckert IV	PE_PRJ_ECKERT_IV	43012
Eckert V	PE_PRJ_ECKERT_V	43011
Eckert VI	PE_PRJ_ECKERT_VI	43010
Equal_Earth	PR_PRJ_EQUAL_EARTH	43085
Equidistant Conic	PE_PRJ_EQUIDISTANT_CONIC	43027
Equidistant Cylindrical	PE_PRJ_EQUIDISTANT_CYLINDRICAL	43002
Equidistant Cylindrical Ellipsoidal	PE_PRJ_EQUIDISTANT_CYLINDRICAL_ELLIPSOIDAL	43061
Flat Polar Quartic	PE_PRJ_FLAT_POLAR_QUARTIC	43045
Fuller	PE_PRJ_FULLER	43052
Gall Stereographic	PE_PRJ_GALL_STEREOGRAPHIC	43016
Gauss-Krüger	PE_PRJ_GAUSS_KRUGER	43005
Geostationary_Satellite	PE_PRJ_GEOSTATIONARY_SATELLITE	43084
Gnomonic	PE_PRJ_GNOMONIC	43047

Gnomonic Ellipsoidal	PE_PRJ_GNOMONIC_ELLIPSOIDAL	43065
Goode Homolosine	PE_PRJ_GOODE_HOMOLOSINE	43059
Hammer Aitoff	PE_PRJ_HAMMER_AITOFF	43044
Hammer Ellipsoidal	PE_PRJ_HAMMER_ELLIPSOIDAL	43071
Hotine 2 Point Center	PE_PRJ_HOTINE_TWO_POINT_CENTER	43035
Hotine 2 Point Natural Origin	PE_PRJ_HOTINE_TWO_POINT_NATORIGIN	43025
Hotine Azimuth Center	PE_PRJ_HOTINE_AZIMUTH_CENTER	43037
Hotine Azimuth Natural Origin	PE_PRJ_HOTINE_AZIMUTH_NATORIGIN	43036
IGAC Plano Cartesiano	PE_PRJ_IGAC_PLANO_CARTESIANO	43064
Krovak Oblique Lambert Conformal Conic	PE_PRJ_KROVAK	43039
Laborde Oblique Mercator	PE_PRJ_LABORDE	43063
Lambert Azimuthal Equal Area	PE_PRJ_LAMBERT_AZIMUTHAL_EQAREA	43033
Lambert Conformal Conic	PE_PRJ_LAMBERT_CONFORMAL_CONIC	43020
Local	PE_PRJ_LOCAL	43058
Loximuthal	PE_PRJ_LOXIMUTHAL	43023
Mercator	PE_PRJ_MERCATOR	43004
Mercator Variant A	PE_PRJ_MERCATOR_VARIANT_A	43069
Mercator Variant C	PE_PRJ_MERCATOR_VARIANT_C	43070
Miller Cylindrical	PE_PRJ_MILLER_CYLINDRICAL	43003
Mollweide	PE_PRJ_MOLLWEIDE	43009
Natural_Earth	PE_PRJ_NATURAL_EARTH	43077
Natural_Earth_II	PE_PRJ_NATURAL_EARTH_II	43078
New Zealand Map Grid	PE_PRJ_NEW_ZEALAND_MAP_GRID	43040
Ney Modified Conic	PE_PRJ_NEY	43062
Orthographic	PE_PRJ_ORTHOGRAPHIC	43041
Patterson	PE_PRJ_PATTERSON	43079
Plate Carrée	PE_PRJ_PLATE_CARREE	43001
Polar Stereographic Variant A	PE_PRJ_POLAR_STEREOGRAPHIC_VARIANT_A	43066
Polar Stereographic Variant B	PE_PRJ_POLAR_STEREOGRAPHIC_VARIANT_B	43067
Polar Stereographic Variant C	PE_PRJ_POLAR_STEREOGRAPHIC_VARIANT_C	43068
Polyconic	PE_PRJ_POLYCONIC	43021
Quartic Authalic	PE_PRJ_QUARTIC_AUTHALIC	43022
Quartic Authalic Ellipsoidal	PE_PRJ_QUARTIC_AUTHALIC_ELLIPSOIDAL	43072
Rectified Skew Orthomorphic - Center	PE_PRJ_RSO_CENTER	43054

Rectified Skew Orthomorphic - Natural Origin	PE_PRJ_RSO_NATORIGIN	43053
Robinson	PE_PRJ_ROBINSON	43030
Robinson from Arc/INFO Workstation	PE_PRJ_ROBINSON_AI	43057
Sinusoidal	PE_PRJ_SINUSOIDAL	43008
Stereographic	PE_PRJ_STEREOGRAPHIC	43026
Stereographic - North Pole	PE_PRJ_STEREOGRAPHIC_NORTH_POLE	43050
Stereographic - South Pole	PE_PRJ_STEREOGRAPHIC_SOUTH_POLE	43051
Transverse Mercator (Complex)	PE_PRJ_TRANSVERSE_MERCATOR_COMPLEX	43056
Transverse Mercator	PE_PRJ_TRANSVERSE_MERCATOR	43006
Transverse Mercator NGA 2014	PE_PRJ_TRANSVERSE_MERCATOR_NGA_2014	43081
Two-Point Equidistant	PE_PRJ_TWO_POINT_EQUIDISTANT	43031
Van der Grinten I	PE_PRJ_VAN_DER_GRINTEN_I	43029
Vertical Near-Side Perspective	PE_PRJ_VERTICAL_NEAR_SIDE_PERSPECTIVE	43049
Wagner_IV	PE_PRJ_WAGNER_IV	43074
Wagner_V	PE_PRJ_WAGNER_V	43075
Wagner_VII	PE_PRJ_WAGNER_VII	43076
Winkel I	PE_PRJ_WINKEL_I	43018
Winkel II	PE_PRJ_WINKEL_II	43019
Winkel Tripel	PE_PRJ_WINKEL_TRIPEL	43042
World on a Cube	PE_PRJ_CUBE	43055
	Auxiliary Spheres	
Name	**Macro**	**WKID**
Azimuthal Equidistant (Auxiliary Sphere)	PE_PRJ_AZIMUTHAL_EQUIDISTANT_AUXS	43132
Eckert IV (Auxiliary Sphere)	PE_PRJ_ECKERT_IV_AUXS	43112
Eckert VI (Auxiliary Sphere)	PE_PRJ_ECKERT_VI_AUXS	43110
Equidistant Cylindrical (Auxiliary Sphere)	PE_PRJ_EQUIDISTANT_CYLINDRICAL_AUXS	43102
Gnomonic (Auxiliary Sphere)	PE_PRJ_GNOMONIC_AUXS	43147
Lambert Azimuthal Equal Area (Auxiliary Sphere)	PE_PRJ_LAMBERT_AZIMUTHAL_EQAREA_AUXS	43133
Mercator (Auxiliary Sphere)	PE_PRJ_MERCATOR_AUXS	43104
Miller Cylindrical (Auxiliary Sphere)	PE_PRJ_MILLER_CYLINDRICAL_AUXS	43103
Mollweide (Auxiliary Sphere)	PE_PRJ_MOLLWEIDE_AUXS	43109

Orthographic (Auxiliary Sphere)	PE_PRJ_ORTHOGRAPHIC_AUXS	43141
Stereographic (Auxiliary Sphere)	PE_PRJ_STEREOGRAPHIC_AUXS	43126
Van der Grinten I (Auxiliary Sphere)	PE_PRJ_VAN_DER_GRINTEN_I_AUXS	43129

Projection Parameters

Name	Macro	WKID
X0	PE_PAR_FALSE_EASTING	100001
Y0	PE_PAR_FALSE_NORTHING	100002
K0	PE_PAR_SCALE_FACTOR	100003
ALPHA	PE_PAR_AZIMUTH	100004
HEIGHT	PE_PAR_HEIGHT	100005
LAM0	PE_PAR_CENTRAL_MERIDIAN	100010
LAM0	PE_PAR_LONGITUDE_OF_ORIGIN	100011
LAMC	PE_PAR_LONGITUDE_OF_CENTER	100012
LAM1	PE_PAR_LONGITUDE_OF_1ST	100013
LAM2	PE_PAR_LONGITUDE_OF_2ND	100014
PHI0	PE_PAR_CENTRAL_PARALLEL	100020
PHI0	PE_PAR_LATITUDE_OF_ORIGIN	100021
PHIC	PE_PAR_LATITUDE_OF_CENTER	100022
PHI1	PE_PAR_LATITUDE_OF_1ST	100023
PHI2	PE_PAR_LATITUDE_OF_2ND	100024
PHI1	PE_PAR_STANDARD_PARALLEL_1	100025
PHI2	PE_PAR_STANDARD_PARALLEL_2	100026
PHI1	PE_PAR_PSEUDO_STANDARD_PARALLEL_1	100027
AUXS	PE_PAR_AUXILIARY_SPHERE_TYPE	100035
OPTION	PE_PAR_OPTION	100036
XS	PE_PAR_X_SCALE	100037
YS	PE_PAR_Y_SCALE	100038
XYR	PE_PAR_XY_PLANE_ROTATION	100039

Vertical coordinate system parameters

Name	Macro	WKID
Z0	PE_PAR_VERTICAL_SHIFT	100006
DIR (Height +, Depth -)	PE_PAR_DIRECTION	100007

Geographic transformation methods

Name	Macro	WKID
NULL transformation	PE_MTH_NULL	109600
Units change only	PE_MTH_UNIT_CHANGE	109601
Longitude Rotation	PE_MTH_LONGITUDE_ROTATION	9601
Geocentric Translation (3-parameter)	PE_MTH_GEOCENTRIC_TRANSLATION	9603
GEOCON	PE_MTH_GEOCON	109614
Molodensky	PE_MTH_MOLODENSKY	9604
Abridged Molodensky	PE_MTH_MOLODENSKY_ABRIDGED	9605
Position Vector (7-parameter)	PE_MTH_POSITION_VECTOR	9606

Coordinate Frame (7-parameter)	PE_MTH_COORDINATE_FRAME	9607
Bursa-Wolf	PE_MTH_BURSA_WOLF	109607
NADCON	PE_MTH_NADCON	9613
NTv2	PE_MTH_NTV2	9615
Geographic 2D Offset	PE_MTH_GEOGRAPHIC_2D_OFFSET	9619
HARN (HPGN)	PE_MTH_HARN	109613
Molodensky-Badekas	PE_MTH_MOLODENSKY_BADEKAS	9636

Geographic transformation parameters

Name	Macro	WKID
DX	PE_PAR_X_AXIS_TRANSLATION	100040
DY	PE_PAR_Y_AXIS_TRANSLATION	100041
DZ	PE_PAR_Z_AXIS_TRANSLATION	100042
RX	PE_PAR_X_AXIS_ROTATION	100043
RY	PE_PAR_Y_AXIS_ROTATION	100044
RZ	PE_PAR_Z_AXIS_ROTATION	100045
DS	PE_PAR_SCALE_DIFFERENCE	100046
ND	PE_PAR_NAME_DATASET	100047
XCR	PE_PAR_X_COORD_OF_ROTATION_ORIGIN	100048
YCR	PE_PAR_Y_COORD_OF_ROTATION_ORIGIN	100049
ZCR	PE_PAR_Z_COORD_OF_ROTATION_ORIGIN	100050
DLON	PE_PAR_LONGITUDE_OFFSET	100051
DLAT	PE_PAR_LATITUDE_OFFSET	100052

Vertical transformation methods

Name	Macro	WKID
VERTCON	PE_VTMTH_VERTCON	129658
GEOID99, GEOID03, and GEOID06	PE_VTMTH_GEOID	129665
EGM96 bilinear interpolation	PE_VTMTH_EGM96	129661
EGM96 natural spline interpolation	PE_VTMTH_EGM96_NS	129761
EGM84 bilinear interpolation	PE_VTMTH_EGM84	129861
EGM84 natural spline interpolation	PE_VTMTH_EGM84_NS	129961
Vertical Offset	PE_VTMTH_VERTICAL_OFFSET	129616
Vertical Offset and Slope	PE_VTMTH_VERTICAL_OFFSET_SLOPE	129657
Generic vertical transform grid file	PE_VTMTH_VTGRIDFILE	129600

Vertical transformation parameters

Name	Macro	WKID
VO	PE_PAR_VERTICAL_OFFSET	100060
PHI0	PE_PAR_LATITUDE_OF_EVALUATION	100061
LAM0	PE_PAR_LONGITUDE_OF_EVALUATION	100062

INCN	PE_PAR_INCLINATION_NORTH	100063
INCE	PE_PAR_INCLINATION_EAST	100064
INTERP	PE_PAR_INTERPOLATION_TYPE	100065

Appendix B: Projections and their expected parameters

Projection Name		
Parameter WKID	**Parameter Macro**	**Parameter Name**
Aitoff		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Albers		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
100026	PE_PAR_STANDARD_PARALLEL_2	Standard_Parallel_2
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Aspect_Adaptive_Cylindrical		
100001	PE_PAR_FALSE_EASTING	False Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_meridian
100070	PE_PAR_ASPECT_RATIO	Aspect_Ratio
Azimuthal_Equidistant		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Azimuthal_Equidistant_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Behrmann		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Berghaus_Star		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
100039	PE_PAR_XY_PLANE_ROTATION	XY_Plane_Rotation
Bonne		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Cassini		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian

Projection Name		
Parameter WKID	**Parameter Macro**	**Parameter Name**
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Compact_Miller		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Craster_Parabolic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Cube		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100036	PE_PAR_OPTION	Option
Cylindrical_Equal_Area		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Double_Stereographic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Eckert_Greifendorff		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Eckert_I		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Eckert_II		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Eckert_III		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Eckert_IV		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Eckert_IV_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing

Projection Name		
Parameter WKID	**Parameter Macro**	**Parameter Name**
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Eckert_V		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Eckert_VI		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Eckert_VI_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Equidistant_Conic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
100026	PE_PAR_STANDARD_PARALLEL_2	Standard_Parallel_2
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Equidistant_Cylindrical		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Equidistant_Cylindrical_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Equidistant_Cylindrical_Ellipsoidal		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Flat_Polar_Quartic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Fuller		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100036	PE_PAR_OPTION	Option
Gall_Stereographic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian

Projection Name		
Parameter WKID	**Parameter Macro**	**Parameter Name**
Gauss_Kruger		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Geostationary Satellite		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100005	PE_PAR_HEIGHT	Height
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100036	PE_PAR_OPTION	Option
Gnomonic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
Gnonomic_Ellipsoidal		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
Gnomonic_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Goode_Homolosine		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100036	PE_PAR_OPTION	Option
Hammer_Aitoff		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Hammer_Ellipsoidal		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Hotine_Oblique_Mercator_Azimuth_Center		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100004	PE_PAR_AZIMUTH	Azimuth
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
Hotine_Oblique_Mercator_Azimuth_Natural_Origin		

Projection Name		
Parameter WKID	Parameter Macro	Parameter Name
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100004	PE_PAR_AZIMUTH	Azimuth
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
Hotine_Oblique_Mercator_Two_Point_Center		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100023	PE_PAR_LATITUDE_OF_1ST	Latitude_Of_1st_Point
100024	PE_PAR_LATITUDE_OF_2ND	Latitude_Of_2nd_Point
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100013	PE_PAR_LONGITUDE_OF_1ST	Longitude_Of_1st_Point
100014	PE_PAR_LONGITUDE_OF_2ND	Longitude_Of_2nd_Point
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
Hotine_Oblique_Mercator_Two_Point_Natural_Origin		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100023	PE_PAR_LATITUDE_OF_1ST	Latitude_Of_1st_Point
100024	PE_PAR_LATITUDE_OF_2ND	Latitude_Of_2nd_Point
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100013	PE_PAR_LONGITUDE_OF_1ST	Longitude_Of_1st_Point
100014	PE_PAR_LONGITUDE_OF_2ND	Longitude_Of_2nd_Point
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
IGAC_Plano_Cartesiano		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
100005	PE_PAR_HEIGHT	Height
Krovak		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100027	PE_PAR_PSEUDO_STANDARD_PARALLEL_1	Pseudo_Standard_Parallel_1
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100004	PE_PAR_AZIMUTH	Azimuth
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
100037	PE_PAR_X_SCALE	X_Scale
100038	PE_PAR_Y_SCALE	Y_Scale
100039	PE_PAR_XY_PLANE_ROTATION	XY_Plane_Rotation
Laborde_Oblique_Mercator		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100004	PE_PAR_AZIMUTH	Azimuth
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
Lambert_Azimuthal_Equal_Area		

Projection Name		
Parameter WKID	**Parameter Macro**	**Parameter Name**
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Lambert_Azimuthal_Equal_Area_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Lambert_Conformal_Conic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
100026	PE_PAR_STANDARD_PARALLEL_2	Standard_Parallel_2
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Local		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100004	PE_PAR_AZIMUTH	Azimuth
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
Loximuthal		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100020	PE_PAR_CENTRAL_PARALLEL	Central_Parallel
Mercator		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Mercator_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Mercator_Variant_A		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
Mercator_Variant_C		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian

Projection Name		
Parameter WKID	Parameter Macro	Parameter Name
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Miller_Cylindrical		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Miller_Cylindrical_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Mollweide		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Mollweide_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Natural_Earth		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Merdian
Natural_Earth_II		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
New_Zealand_Map_Grid		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100011	PE_PAR_LONGITUDE_OF_ORIGIN	Longitude_Of_Origin
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Ney_Modified_Conic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
100026	PE_PAR_STANDARD_PARALLEL_2	Standard_Parallel_2
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Orthographic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
Orthographic_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center

Projection Name		
Parameter WKID	Parameter Macro	Parameter Name
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Patterson		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Plate_Carree		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Polar_Stereographic_Variant_A		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100011	PE_PAR_LONGITUDE_OF_ORIGIN	Longitude_Of_Origin
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Polar_Stereographic_Variant_B		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100011	PE_PAR_LONGITUDE_OF_ORIGIN	Longitude_Of_Origin
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Polar_Stereographic_Variant_C		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100011	PE_PAR_LONGITUDE_OF_ORIGIN	Longitude_Of_Origin
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Polyconic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Quartic_Authalic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Quartic_Authalic_Ellipsoidal		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Rectified_Skew_Orthomorphic_Center		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100004	PE_PAR_AZIMUTH	Azimuth
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
100039	PE_PAR_XY_PLANE_ROTATION	XY_Plane_Rotation
Rectified_Skew_Orthomorphic_Natural_Origin		
100001	PE_PAR_FALSE_EASTING	False_Easting

Projection Name		
Parameter WKID	Parameter Macro	Parameter Name
100002	PE_PAR_FALSE_NORTHING	False_Northing
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100004	PE_PAR_AZIMUTH	Azimuth
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
100039	PE_PAR_XY_PLANE_ROTATION	XY_Plane_Rotation
Robinson		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Robinson_ARC_INFO		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Sinusoidal		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Stereographic		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Stereographic_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Stereographic_North_Pole		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Stereographic_South_Pole		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Times		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Transverse_Cylindrical_Equal_Area		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian

Projection Name		
Parameter WKID	Parameter Macro	Parameter Name
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Transverse_Mercator		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Transverse_Mercator_NGA_2014		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Transverse_Mercator_Complex		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100003	PE_PAR_SCALE_FACTOR	Scale_Factor
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Two_Point_Equidistant		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100023	PE_PAR_LATITUDE_OF_1ST	Latitude_Of_1st_Point
100024	PE_PAR_LATITUDE_OF_2ND	Latitude_Of_2nd_Point
100013	PE_PAR_LONGITUDE_OF_1ST	Longitude_Of_1st_Point
100014	PE_PAR_LONGITUDE_OF_2ND	Longitude_Of_2nd_Point
Van_der_Grinten_I		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
Van_der_Grinten_I_Auxiliary_Sphere		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	Auxiliary_Sphere_Type
Vertical_Near_Side_Perspective		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100012	PE_PAR_LONGITUDE_OF_CENTER	Longitude_Of_Center
100022	PE_PAR_LATITUDE_OF_CENTER	Latitude_Of_Center
100005	PE_PAR_HEIGHT	Height
Wagner_IV		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
1000021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Wagner_V		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian

Projection Name		
Parameter WKID	**Parameter Macro**	**Parameter Name**
Wagner_VII		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100021	PE_PAR_LATITUDE_OF_ORIGIN	Latitude_Of_Origin
Winkel_I		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Winkel_II		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1
Winkel_Tripel		
100001	PE_PAR_FALSE_EASTING	False_Easting
100002	PE_PAR_FALSE_NORTHING	False_Northing
100010	PE_PAR_CENTRAL_MERIDIAN	Central_Meridian
100025	PE_PAR_STANDARD_PARALLEL_1	Standard_Parallel_1

Optional parameters

In general, all the parameters for a given projection must be supplied, but some parameters are optional. They are:

WKID	Parameter Macro	Default Value
100001	PE_PAR_FALSE_EASTING	0.0
100002	PE_PAR_FALSE_NORTHING	0.0
100003	PE_PAR_SCALE_FACTOR	1.0
100035	PE_PAR_AUXILIARY_SPHERE_TYPE	0 (use semi-major-radius)

Appendix C: Transformations and their expected parameters

The Longitude_Rotation, Null, and Unit_Change methods have no parameters. File-based methods have a “Dataset Name” parameter. The base name (no extension) of the file plus a partial path is included as part of the parameter’s name.

Geographic Transformation Method		
Parameter WKID	**Parameter Macro**	**Parameter Name**
Bursa_Wolf		
100040	PE_PAR_X_AXIS_TRANSLATION	X_Axis_Translation
100041	PE_PAR_Y_AXIS_TRANSLATION	Y_Axis_Translation
100042	PE_PAR_Z_AXIS_TRANSLATION	Z_Axis_Translation
100043	PE_PAR_X_AXIS_ROTATION	X_Axis_Rotation
100044	PE_PAR_Y_AXIS_ROTATION	Y_Axis_Rotation
100045	PE_PAR_Z_AXIS_ROTATION	Z_Axis_Rotation
100046	PE_PAR_SCALE_DIFFERENCE	Scale_Difference
Coordinate_Frame		
100040	PE_PAR_X_AXIS_TRANSLATION	X_Axis_Translation
100041	PE_PAR_Y_AXIS_TRANSLATION	Y_Axis_Translation
100042	PE_PAR_Z_AXIS_TRANSLATION	Z_Axis_Translation
100043	PE_PAR_X_AXIS_ROTATION	X_Axis_Rotation
100044	PE_PAR_Y_AXIS_ROTATION	Y_Axis_Rotation
100045	PE_PAR_Z_AXIS_ROTATION	Z_Axis_Rotation
100046	PE_PAR_SCALE_DIFFERENCE	Scale_Difference
Geocentric_Translation		
100040	PE_PAR_X_AXIS_TRANSLATION	X_Axis_Translation
100041	PE_PAR_Y_AXIS_TRANSLATION	Y_Axis_Translation
100042	PE_PAR_Z_AXIS_TRANSLATION	Z_Axis_Translation
GEOCON		
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>
Geographic_2D_Offset		
100051	PE_PAR_LONGITUDE_OFFSET	Longitude_Offset
100052	PE_PAR_LATITUDE_OFFSET	Latitude_Offset
HARN		
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>
Longitude_Rotation		
Molodensky		
100040	PE_PAR_X_AXIS_TRANSLATION	X_Axis_Translation
100041	PE_PAR_Y_AXIS_TRANSLATION	Y_Axis_Translation
100042	PE_PAR_Z_AXIS_TRANSLATION	Z_Axis_Translation
Molodensky_Abridged		
100040	PE_PAR_X_AXIS_TRANSLATION	X_Axis_Translation
100041	PE_PAR_Y_AXIS_TRANSLATION	Y_Axis_Translation
100042	PE_PAR_Z_AXIS_TRANSLATION	Z_Axis_Translation
Molodensky_Badekas		
100040	PE_PAR_X_AXIS_TRANSLATION	X_Axis_Translation
100041	PE_PAR_Y_AXIS_TRANSLATION	Y_Axis_Translation
100042	PE_PAR_Z_AXIS_TRANSLATION	Z_Axis_Translation
100043	PE_PAR_X_AXIS_ROTATION	X_Axis_Rotation

Geographic Transformation Method		
Parameter WKID	**Parameter Macro**	**Parameter Name**
100044	PE_PAR_Y_AXIS_ROTATION	Y_Axis_Rotation
100045	PE_PAR_Z_AXIS_ROTATION	Z_Axis_Rotation
100046	PE_PAR_SCALE_DIFFERENCE	Scale_Difference
100048	PE_PAR_X_COORD_OF_ROTATION_ORIGIN	X_Coordinate_of_Rotation_Origin
100049	PE_PAR_Y_COORD_OF_ROTATION_ORIGIN	Y_Coordinate_of_Rotation_Origin
100050	PE_PAR_Z_COORD_OF_ROTATION_ORIGIN	Z_Coordinate_of_Rotation_Origin
NADCON		
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>
NTv2		
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>
Null		
Position_Vector		
100040	PE_PAR_X_AXIS_TRANSLATION	X_Axis_Translation
100041	PE_PAR_Y_AXIS_TRANSLATION	Y_Axis_Translation
100042	PE_PAR_Z_AXIS_TRANSLATION	Z_Axis_Translation
100043	PE_PAR_X_AXIS_ROTATION	X_Axis_Rotation
100044	PE_PAR_Y_AXIS_ROTATION	Y_Axis_Rotation
100045	PE_PAR_Z_AXIS_ROTATION	Z_Axis_Rotation
100046	PE_PAR_SCALE_DIFFERENCE	Scale_Difference
Unit_Change		

Vertical Transformation Method		
Parameter WKID	**Parameter Macro**	**Parameter Name**
EGM84		
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>
EGM84_Natural_Spline		
100047	PE_PAR_NAME_DATASET	Dataset_< base file name>
EGM96		
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>
EGM96_Natural_Spline		
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>
GEOID		
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>
VERTCON		
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>
Vertical_Offset		
100060	PE_PAR_VERTICAL_OFFSET	Vertical_Offset
Vertical_Offset_and_Slope		
100060	PE_PAR_VERTICAL_OFFSET	Vertical_Offset
100062	PE_PAR_LONGITUDE_OF_EVALUATION	Longitude_Of_Evaluation
100061	PE_PAR_LATITUDE_OF_EVALUATION	Latitude_Of_Evaluation
100063	PE_PAR_INCLINATION_NORTH	Inclination_North
100064	PE_PAR_INCLINATION_EAST	Inclination_East
VTGridFile		
100065	PE_PAR_INTERPOLATION_TYPE	Interpolation_Type
100047	PE_PAR_NAME_DATASET	Dataset_<base file name>

Appendix D: Differences in earlier database versions

The following is a history of the changes in the PE database mechanism:

- ArcGIS versions up through 9.2:
  - The only database available was the OBJEDIT database.
  - The location of the OBJEDIT directory was specified by the environment variable PEOBJEDITHOME.
  - The only files used were data files. The filename was simply "<object-type>" (e.g, "projcs").
  - Although this capability was available, it was considered something to use only in an "emergency", since the implementation slowed up all factory lookups by a factor of 4 to 10, depending on how many data files were present.

- ArcGIS version 9.3:
  - All OBJEDIT data files were read in at initialization time and tokenized. This speeded up OBJEDIT lookup considerably.
  - The data files could have the optional extension ".txt" (e.g. "projcs.txt"). This made Notepad users happy, since Notepad automatically adds that extension to a filename.

- ArcGIS version 10.0:
  - The PE switched all factory lookup mechanisms to using "database interfaces", of which OBJEDIT was one implementation.
  - The PEDATABASE environment variable was implemented to specify database-interface definitions.
  - The names of OBJEDIT files could use an object-type abbreviation as well as the object-type-name. For example, you could use either "projcs.txt" or "pcs.txt".
  - The tables available were the following:
    - Data Data definitions
    - Defstring User-defined macros
    - Dispname Localized name
    - Metadata Area-name, extent, etc.
    - Description Description (may be localized)
    - Areainfo PRJ-file replacement
    - The old "units.txt" file was broken up into "angunit.txt" and "linunit.txt". There is no backward capability to read old units files.

- ArcGIS version 10.1:

- Added the concept of record types. The following record types were defined:
  - Data dat object data (syntax object-dependent)
  - Codechange chg code-change entries (syntax same for all objects)
  - Deprecated dep WKID status entries (syntax same for all objects)
  - Synonym syn synonym entries (syntax same for all objects)
- Added the ability to write out an OBJEDIT database (previously it was read-only, and could only be created by hand using a text editor).
- Filenames changed from "<objtype>.txt" to "<objtype>_<rectype>.txt". For backwards compatibility, the "_dat" part may be omitted for data records.

- ArcGIS version 10.3:
  - Added the ability for specify record types wanted, via the "rectypes=" option.
  - Changed the "tables=" option to "objtypes=".

- ArcGIS version 10.4:
  - Added the following record-types:
    - Coderange code-range entries
    - Dispname localizable display-names
    - Description localizable descriptions
    - Defstring user-defined macros
    - Areainfo coordinate system dialog entries
    - Areacode WKID-areacode entries
    - Version Version entries
    - Exception GTLIST/VTLIST exception entries
  - Split the metadata data table into object-specific areacode tables and an extent table.
  - Split the areainfo table into object-specific areainfo tables.
  - Added the XMLEDIT database mechanism.

- ArcGIS version 10.5:
  - Added the following record-types:
    - Gcsvcs GCS-VCS equivalences

METADATA – coordinate system and transformation metadata

The METADATA data record contains both areacode and extent information, and also object-type and WKID. Currently, this information is split up between an extent entry and an object-specific areacode tables, which eliminates duplicating extent information.

This is a legacy record format used prior to version 10.4 and should not be used for new databases.

Note: The latitude/longitude values here are specified in a different order than in an extent entry and the primem and factor fields were deleted, as we now assume values are in degrees from Greenwich.

Contents:

- Object-code
- Object-type
- Name
- Left-longitude (llon)
- South-latitude (slat)
- Right-longitude (rlon)
- North-latitude (nlat)
- Prime meridian
  This is usually 0 (Greenwich)
- Angunit factor (lat/lon units to radians)
  For values in degrees this is PI/180 (0.0174532925199433)
- Accuracy (in meters)
  This is 0 for coordinate system entries
- Areacode or 0

Objedit syntax:

```
[METADATA,] object-code, object-type, llon, slat, rlon, nlat, primem,
            angunt-factor, accuracy, areacode
```

Examples:

```
METADATA, 3819, geogcs, "Hungary", \
    16.11, 45.74, 22.9, 48.58, 0.0, 0.0174532925199433, \
    0.0, 1119

METADATA, PE_GCS_SPHERE, geogcs, "World", \
    -180.0, -90.0, 180.0, 90.0, 0.0, 0.0174532925199433, 0.0, 1262

METADATA, PE_GT_MGI_TO_ETRS_1989_4, geogtran, "Austria - Styria", \
    13.58, 46.64, 16.17, 47.84, 0.0, 0.0174532925199433, 1.0, 1543
```

AREAINFO – coordinate-system dialog entries

The AREAINFO entry here has one extra field specifying the object-type, since areainfo entries for all coordinate system types are in one data table.

This is a legacy record format used prior to version 10.4 and should not be used for new databases.

Note: The area-name and category fields are in a different order than in the current areainfo entries.

Contents:

- Object-code
- Object-type
- UTF8-encoded area-name
- UTF8-encoded category

Objedit syntax:

```
[AREAINFO,] object-code, object-type, "area-name", "category"
```

Examples:

```
AREAINFO, 4143, geogcs, \
        "Abidjan 1987", "Geographic Coordinate Systems/Africa"

PE_PCS_WGS_1984_WEB_MERCATOR_AUXSPHERE, projcs, \
        "WGS 1984 Web Mercator (auxiliary sphere)", \
        "Projected Coordinate Systems/World"
```

Appendix E: Setting an environment variable

The following describes how to set an environment variable with a value. In these examples we will set the environment variable PEDATABASE with the value <database definition string>.

In Unix:

(C shell)
setenv PEDATABASE "<database definition string>"

(Korn/Bash shell)
export PEDATABASE="<database definition string>"

(Bourne shell)
PEDATABASE="<database definition string>" export PEDATABASE

In Windows:

If working at a command prompt:

set PEDATABASE=<database definition string> (No quotes!)

Otherwise, you need to set a user or system environment variable, as follows:

On Windows XP, go to Start button > Settings > Control Panel > System > Advanced tab and click the Environment Variables button. Under User variables, click the New button. The variable name is PEDATABASE and the variable value is the database definition string (without any quotes).

On Windows 7, go to Start/Windows button > Control Panel > System and Security > System > Advanced system settings > Advanced tab and click the Environment Variables button. Under User variables, click the New button. The variable name is PEDATABASE and the variable value is the database definition string (without any quotes).

Appendix F: Error reporting

The object-creation mechanism can output warning and/or error messages to a user-specified log.

The syntax of these messages is:

    PE: date time: source [level] message

For example:

```
PE: 2014-08-24 16.20.15: pe_geogcs_from_tokens [error]
        Missing prime meridian in WKT string
PE: 2015-04-27 14.14.27: WGS_1984_System_Zone_18 [warning]
        Missing parameter: False_Northing
```

The automatic logging of messages is turned on by setting the environment variable PELOGFILE. If this environment variable is not found, no automatic logging will be done. See Appendix E for details on how to set an operating system environment variable.

The values for PELOGFILE are:

<pathname>	The absolute pathname of a logfile to append messages to.
-	Output messages to stderr, if it is defined. (Note that GUI programs like ArcMap have no stderr defined.)
!	Output messages to: Unix /dev/console Windows OutputDebugStream()

Note: When looking up a WKID for a given object type (e.g.: GEOGCS 4326), it is considered an error if the WKID is out of range for that object type, but it is not an error if the WKID is simply not found.