REUSE MANUAL

MGRS

10xxxxxx.1

Implementation

TABLE OF CONTENTS

SECTION 1. INTRODUCTION

1.1 PURPOSE OF THE REUSE MANUAL

This document describes the characteristics of the MGRS reusable software component and provides instructions on its installation and operation. The manual is a self-contained reference for the software engineer intending to incorporate the component in another software system. This manual was written with the assumption that the user has a basic working knowledge of C and is familiar with fundamental C concepts and terminology.

1.2 PURPOSE OF THE REUSABLE SOFTWARE COMPONENT

The purpose of MGRS is to provide a reusable software component which supports the following coordinate conversions :

- Geodetic coordinates (latitude and longitude in radians) to Military Grid Reference System (MGRS) coordinate string, and
- Military Grid Reference System (MGRS) coordinate string to Geodetic coordinates (latitude and longitude in radians).
- Universal Transverse Mercator (UTM) coordinates (zone, hemisphere, easting and northing in meters) to MGRS coordinate string.
- MGRS coordinate string to Universal Transverse Mercator (UTM) coordinates (zone, hemisphere, easting and northing in meters).
- Universal Polar Stereographic (UPS) coordinates (hemisphere, easting and northing in meters) to MGRS coordinate string.
- MGRS coordinate string to Universal Polar Stereographic (UPS) coordinates (hemisphere, easting and northing in meters).

A particular ellipsoid is specified in terms of the following parameters:

- Semi-Major Axis (a): Radius (in meters) at the equator,
- Semi-Minor Axis (b): Radius (in meters) at a pole, and
- Ellipsoid Code Standard 2-letter ellipsoid code.

1.3 GENERAL INFORMATION

1.3.1 POINT OF CONTACT

U.S. Army Topographic Engineering Center (USATEC)
Geospatial Information Division (GID)
ATTN: CETEC-GD-A (Dan Specht)
7701 Telegraph Road
Alexandria, VA 22315-3864
Dan Specht (703) 428 - 6761 Project Manager

1.3.2 CERTIFICATION LEVEL

This RSC has been certified at level 4. A level 4 component satisfies the criteria for reliability, testing, and documentation for the Army Reuse Center (ARC). The component comes with test materials and a Reuse Manual that aids in integrating the component into a software system.

1.3.3 LEGAL RESTRICTIONS

SECTION 2. INSTALLATION

The following is a list of the files which make up the MGRS component:

Source Code Files:

```
mgrs.c
```

Header Files :

```
mgrs.h
```

Data Files :

```
none
```

The compilation instructions for the MGRS component are as follows:

DOS Makefile (Uses Microsoft C):

```
cl /nologo /W3 /FR /G2 /DNDEBUG /Gs /Ox /AM /D_DOS /c mgrs.c
```

UNIX Makefile (Uses gcc compiler):

```
cc -g -O -ansi -Wall -c mgrs.c
```

The compilation order of the MGRS component relative to other components is unconstrained.

2.1 PARTIAL REUSE

The MGRS component does not allow for partial reuse.

2.2 MODIFICATIONS

The MGRS component does not permit modifications.

SECTION 3. ENVIRONMENT

This section provides details on the environment under which MGRS was developed, tested, and executed.

3.1 HARDWARE

3.1.1 DEVELOPMENT

The following is a list of hardware configurations under which MGRS was developed and tested.

- SUN SparcStation 20
- IBM compatible Pentium PC

3.1.2 TARGET

The following is a list of hardware configurations under which MGRS was executed.

- SUN SparcStation 20
- IBM compatible Pentium PC

3.2 SOFTWARE

3.2.1 OPERATING SYSTEM

The following is a list of operating systems under which MGRS was executed and tested.

- Solaris 2.5
- Windows 95

3.2.2 COMPILERS

The following is a list of compilers on which MGRS was compiled successfully.

- GCC version 2.8.1

- Microsoft Visual C++ version 6

3.3 ASSUMPTIONS AND PERFORMANCE LIMITATIONS

There are no hardware or environment constraints. There are no limitations.

This RSC is written in ANSI C.

SECTION 4. GLOBAL RSC ENVIRONMENT

4.1 TYPES

Not applicable.

4.2 CONSTANTS

The following is a list of significant visible constants declared globally in MGRS with their descriptions.

```
MGRS_NO_ERROR                   : No errors occurred in function
MGRS_LAT_ERROR                  : Latitude outside of valid range
                                    (-90 to 90 degrees)
MGRS_LON_ERROR                  : Longitude outside of valid range
                                    (-180 to 360 degrees)
MGRS_STR_ERROR                  : An MGRS string error: string too long,
                                    string too short, or badly formed
MGRS_PRECISION_ERROR            : The precision must be between 0 and 5
                                    inclusive.
MGRS_A_ERROR                    : Semi-major axis less than or equal to zero
MGRS_B_ERROR                    : Semi-minor axis less than or equal to zero
MGRS_A_LESS_B_ERROR             : Semi-major axis less than semi-minor axis
MGRS_ZONE_ERROR                 : Zone outside of valid range (1 to 60)
MGRS_HEMISPHERE_ERROR           : Invalid hemisphere (‘N’ or ‘S’)
MGRS_EASTING_ERROR              : Easting outside of valid range
MGRS_NORTHING_ERROR             : Northing outside of valid range
```

4.3 VARIABLES

Not applicable.

4.4 INCLUDE FILES

```
<ctype.h>                       : Standard C character handing library
<math.h>                        : Standard C math library
<stdio.h>                       : Standard C input/output library
<string.h>                      : Standard C string handling library
ups.h                           : Used to convert between geodetic and UPS
utm.h                           : Used to convert between geodetic and UTM
mgrs.h                          : Error codes and prototype error checking
```

4.5 DEPENDENCIES

The following is a list of the software external to the RSC and its descriptions.

UTM, since MGRS coordinates are based on Universal Transverse Mercator (UTM) coordinates in all but the polar regions.

UPS, since MGRS coordinates are based on Universal Polar Stereographic (UPS) coordinates in the polar regions.

SECTION 5. FUNCTIONS

5.1 SET_MGRS_PARAMETERS

5.1.1 DESCRIPTION

This function sets the ellipsoid parameters to the specified values.

5.1.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
long Set_MGRS_Parameters (double a,
             double b);
```

a Semi-major axis of ellipsoid in meters (input),

b Semi-minor axis of ellipsoid in meters (input).

Example:

```
status = Set_MGRS_Parameters (a, b)
```

Inputs:

a 6378137.0

b 6356752.3142

Outputs:

None.

5.1.3 DECLARATIONS

5.1.3.1 TYPES

Not applicable.

5.1.3.2 CONSTANTS

Not applicable.

5.1.3.3 VARIABLES

Not applicable.

5.1.4 DEPENDENCIES

None.

5.1.5 ERROR HANDLING

This function returns the following status codes:

```
MGRS_NO_ERROR                    : No errors occurred in function
MGRS_A_ERROR                     : Semi-major axis less than or equal to zero
MGRS_B_ERROR                     : Semi-minor axis less than or equal to zero
MGRS_A_LESS_B_ERROR              : Semi-major axis less than semi-minor axis
```

5.2 GET_MGRS_PARAMETERS

5.2.1 DESCRIPTION

This function returns the current values of the ellipsoid parameters.

5.2.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
void Get_MGRS_Parameters (double *a,
             double *b);
```

a Semi-major axis of ellipsoid in meters (output),

b Semi-minor axis of ellipsoid in meters (output).

5.2.3 DECLARATIONS

5.2.3.1 TYPES

Not applicable.

5.2.3.2 CONSTANTS

Not applicable.

5.2.3.3 VARIABLES

Not applicable.

5.2.4 DEPENDENCIES

None.

5.2.5 ERROR HANDLING

No errors are reported by this function.

5.3 CONVERT_GEODETIC_TO_MGRS

5.3.1 DESCRIPTION

This function converts Geodetic coordinates (latitude and longitude in radians) to an MGRS coordinate string with the specified level of precision.

5.3.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
long Convert_Geodetic_To_MGRS (double Latitude,
              double Longitude,
              long Precision,
              char *MGRS_String)
```

Latitude	Latitude in radians (input),
Longitude	Longitude in radians (input),
Precision	Level of precision (number of digits) for MGRS string (input),
MGRS_String	MGRS coordinate string (output).

Example:

```
status = Convert_Geodetic_To_MGRS (Latitude, Longitude, Precision,
            MGRS_String)
```

Inputs:

Latitude:	35.0
Longitude:	-75.0
Precision	5

Outputs:

MGRS_String:	18SWD0000073042

5.3.3 DECLARATIONS

5.3.3.1 TYPES

Not applicable.

5.3.3.2 CONSTANTS

Not applicable.

5.3.3.3 VARIABLES

Not applicable.

5.3.4 DEPENDENCIES

Set_UTM_Parameters, in the UTM component – used to set the parameters of the Universal Transverse Mercator (UTM) projection, and

Convert_Geodetic_To_UTM, in the UTM component – used to convert geodetic (latitude and longitude) coordinates to Universal Transverse Mercator (UTM) projection (hemisphere, zone, easting and northing) coordinates.

Set_UPS_Parameters, in the UPS component – used to set the parameters of the Universal Polar Stereographic (UPS) projection, and

Convert_Geodetic_To_UPS, in the UPS component – used to convert geodetic (latitude and longitude) coordinates to Universal Polar Stereographic (UPS) projection (hemisphere, easting and northing) coordinates.

5.3.5 ERROR HANDLING

This function returns the following status codes:

```
MGRS_NO_ERROR                   : No errors occurred in function
MGRS_LAT_ERROR                  : Latitude outside of valid range
                                    (-90 to 90 degrees)
MGRS_LON_ERROR                  : Longitude outside of valid range
                                    (-180 to 360 degrees)
MGRS_PRECISION_ERROR            : The precision must be between 0 and 5
                                    inclusive.
```

5.4 CONVERT_MGRS_TO_GEODETIC

5.4.1 DESCRIPTION

This function converts an MGRS coordinate string to Geodetic coordinates (latitude and longitude in radians).

5.4.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
long Convert_MGRS_To_Geodetic (char *MGRS_String,
             double *Latitude,
             double *Longitude);
```

MGRS_String MGRS coordinate string (input),

Latitude Latitude in radians (output),

Longitude Longitude in radians (output).

Example:

```
status = Convert_MGRS_To_Geodetic (MGRS_String, &Latitude, &Longitude)
```

Inputs:

MGRS_String: 18SWD0000073042

Outputs:

Latitude: 35.0

Longitude: -75.0

5.4.3 DECLARATIONS

5.4.3.1 TYPES

Not applicable.

5.4.3.2 CONSTANTS

Not applicable.

5.4.3.3 VARIABLES

Not applicable.

5.4.4 DEPENDENCIES

Set_UTM_Parameters, in the UTM component – used to set the parameters of the Universal Transverse Mercator (UTM) projection, and

Convert_UTM_To_Geodetic, in the UTM component – used to convert Universal Transverse Mercator (UTM) projection (hemisphere, zone, easting and northing) coordinates to geodetic (latitude and longitude) coordinates.

Set_UPS_Parameters, in the UPS component – used to set the parameters of the Universal Polar Stereographic (UPS) projection, and

Convert_UPS_To_Geodetic, in the UPS component – used to convert Universal Polar Stereographic (UPS) projection (hemisphere, easting and northing) coordinates to geodetic (latitude and longitude) coordinates.

5.4.5 ERROR HANDLING

This function returns the following status codes:

```
MGRS_NO_ERROR                    : No errors occurred in function
```

```
MGRS_STR_ERROR                    : An MGRS string error: string to long,
                                      string to short, or badly formed.
```

5.5 CONVERT_UTM_TO_MGRS

5.5.1 DESCRIPTION

This function converts Universal Transverse Mercator (UTM) coordinates (zone, hemisphere, easting and northing in meters) to an MGRS coordinate string with the specified level of precision.

5.5.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
long Convert_UTM_To_MGRS (long Zone,
             char Hemisphere,
             double Easting,
             double Northing,
             long Precision,
             char *MGRS_String)
```

Zone	UTM zone (1 to 60) (input),
Hemisphere	Northern (‘N’) or Southern (‘S’) hemisphere (input),
Easting	Easting (X) in meters (input),
Northing	Northing (Y) in meters (input).
Precision	Level of precision (number of digits) for MGRS string (input),
MGRS_String	MGRS coordinate string (output).

Example:

```
status = Convert_UTM_To_MGRS (Zone, Hemisphere, Easting, Northing, Precision,
            MGRS_String)
```

Inputs:

Inputs:

Zone:	18
Hemisphere:	‘N’

Easting: 500000

Northing: 3873042

Precision 5

Outputs:

MGRS_String: 18SWD0000073042

5.5.3 DECLARATIONS

5.5.3.1 TYPES

Not applicable.

5.5.3.2 CONSTANTS

Not applicable.

5.5.3.3 VARIABLES

Not applicable.

5.5.4 DEPENDENCIES

Set_UTM_Parameters, in the UTM component – used to set the parameters of the Universal Transverse Mercator (UTM) projection, and

Convert_Geodetic_To_UTM, in the UTM component – used to convert geodetic (latitude and longitude) coordinates to Universal Transverse Mercator (UTM) projection (hemisphere, zone, easting and northing) coordinates.

5.5.5 ERROR HANDLING

This function returns the following status codes:

```
MGRS_NO_ERROR                   : No errors occurred in function
MGRS_ZONE_ERROR                 : Zone outside of valid range
                                    (1 to 60)
MGRS_HEMISPHERE_ERROR           : Invalid hemisphere (‘N’ or ‘S’)
MGRS_EASTING_ERROR              : Easting outside of valid range
                                    (100,000 to 900,000m)
```

```
MGRS_NORTHING_ERROR              : Northing outside of valid range
                                     (0 to 10,000,000m)
MGRS_PRECISION_ERROR             : The precision must be between 0 and 5
                                     inclusive.
```

5.6 CONVERT_MGRS_TO_UTM

5.6.1 DESCRIPTION

This function converts an MGRS coordinate string to Universal Transverse Mercator (UTM) projection coordinates (zone, hemisphere, easting in meters, and northing in meters).

5.6.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
long Convert_MGRS_To_UTM (char *MGRS_String,
             long   *Zone,
             char   *Hemisphere,
             double *Easting,
             double *Northing)
```

MGRS_String	MGRS coordinate string (input),
Zone	UTM zone (1 to 60) (output),
Hemisphere	Northern (‘N’) or Southern (‘S’) hemisphere (output),
Easting	Easting (X) in meters (output),
Northing	Northing (Y) in meters (output).

Example:

```
status = Convert_MGRS_To_UTM (MGRS_String, &Zone, &Hemisphere, &Easting,
            &Northing)
```

Inputs:

MGRS_String: 18SWD0000073042

Outputs:

Zone: 18

Hemisphere: ‘N’

Easting: 500000

Northing: 3873042

5.6.3 DECLARATIONS

5.6.3.1 TYPES

Not applicable.

5.6.3.2 CONSTANTS

Not applicable.

5.6.3.3 VARIABLES

Not applicable.

5.6.4 DEPENDENCIES

None.

5.6.5 ERROR HANDLING

This function returns the following status codes:

```
MGRS_NO_ERROR                    : No errors occurred in function
MGRS_STR_ERROR                   : An MGRS string error: string to long,
                                     string to short, or badly formed.
```

5.7 CONVERT_UPS_TO_MGRS

5.7.1 DESCRIPTION

This function converts Universal Polar Stereographic (UPS) coordinates (hemisphere, easting and northing in meters) to an MGRS coordinate string with the specified level of precision.

5.7.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
long Convert_UTM_To_MGRS (char Hemisphere,
             double Easting,
             double Northing,
             long Precision,
             char *MGRS_String)
```

Hemisphere	Northern (‘N’) or Southern (‘S’) hemisphere (input),
Easting	Easting (X) in meters (input),
Northing	Northing (Y) in meters (input).
Precision	Level of precision (number of digits) for MGRS string (input),
MGRS_String	MGRS coordinate string (output).

Example:

```
status = Convert_UPS_To_MGRS (Hemisphere, Easting, Northing, Precision,
            MGRS_String)
```

Inputs:

Hemisphere:	“S”
Easting:	2222990
Northing:	1797470
Precision	5

Outputs:

MGRS_String:	BCK2299097470

5.7.3 DECLARATIONS

5.7.3.1 TYPES

Not applicable.

5.7.3.2 CONSTANTS

Not applicable.

5.7.3.3 VARIABLES

Not applicable.

5.7.4 DEPENDENCIES

None.

5.7.5 ERROR HANDLING

This function returns the following status codes:

```
MGRS_NO_ERROR                   : No errors occurred in function
MGRS_HEMISPHERE_ERROR           : Invalid hemisphere (‘N’ or ‘S’)
MGRS_EASTING_ERROR              : Easting outside of valid range
                                    (0 to 4,000,000m)
MGRS_NORTHING_ERROR             : Northing outside of valid range
                                    (0 to 4,000,000m)
MGRS_PRECISION_ERROR            : The precision must be between 0 and 5
                                    inclusive.
```

5.8 CONVERT_MGRS_TO_UPS

5.8.1 DESCRIPTION

This function converts an MGRS coordinate string to Universal Polar Stereographic (UPS) projection coordinates (hemisphere, easting in meters, and northing in meters).

5.8.2 INTERFACES AND EXAMPLES

The following is a list of the formal arguments required to use this function.

```
long Convert_MGRS_To_UPS (char *MGRS_String,
             char   *Hemisphere,
             double *Easting,
             double *Northing)
```

MGRS_String MGRS coordinate string (input),

Hemisphere Northern (‘N’) or Southern (‘S’) hemisphere (output),

Easting Easting (X) in meters (output),

Northing Northing (Y) in meters (output).

Example:

```
status = Convert_MGRS_To_UPS (MGRS_String, &Hemisphere, &Easting, &Northing);
```

Inputs:

MGRS_String: BCK2299097470

Outputs:

Hemisphere: “S”

Easting: 2222990

Northing: 1797470

5.8.3 DECLARATIONS

5.8.3.1 TYPES

Not applicable.

5.8.3.2 CONSTANTS

Not applicable.

5.8.3.3 VARIABLES

Not applicable.

5.8.4 DEPENDENCIES

None.

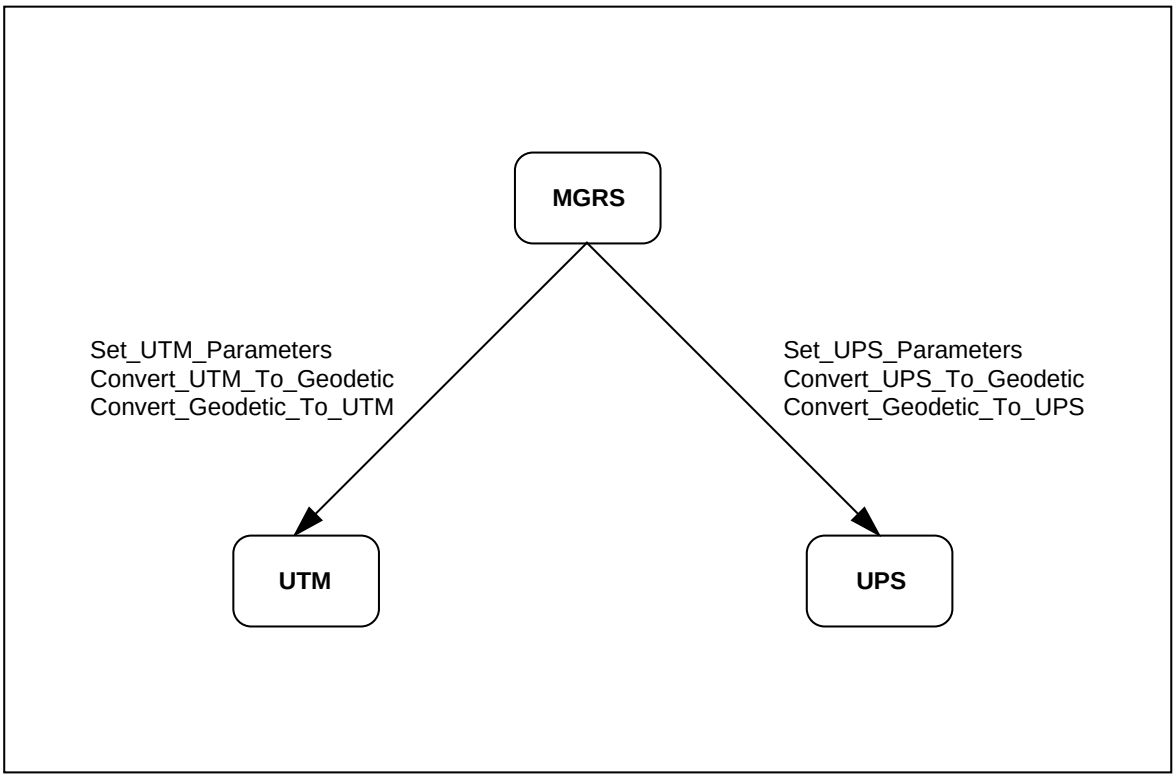
5.6.5 ERROR HANDLING

This function returns the following status codes:

```
MGRS_NO_ERROR                   : No errors occurred in function
MGRS_STR_ERROR                  : An MGRS string error: string to long,
                                    string to short, or badly formed.
```

APPENDIX A STRUCTURE/DEPENDENCY DIAGRAMS

This component consists of a single compilation unit and depends on both the UTM and UPS components.



APPENDIX B DEFINITIONS/GLOSSARY

Coordinate – Linear or angular quantities that designate the position that a point occupies in a given reference frame or system. Also used as a general term to designate the particular kind of reference frame or system, such as Cartesian coordinates or spherical coordinates.

Ellipsoid – The surface generated by an ellipse rotating about one of its axes.

Geodetic Coordinates – The quantities of latitude and longitude that define the position of a point on the surface of the earth with respect to the reference ellipsoid. Also, imprecisely called geographic coordinates.

Geodetic Latitude – The angle between the plane of the equator and the normal to the ellipsoid through the computation point. Geodetic latitude is positive north of the equator and negative south of the equator.

Geodetic Longitude – The angle between the plane of a meridian and the plane of the prime meridian. A longitude can be measured from the angle formed between the local and prime meridians at the pole of rotation of the reference ellipsoid, or by the arc along the Equator intercepted by these meridians.

MGRS Coordinates – The U.S. Military Grid Reference System (MGRS) is an alphanumeric system, based on the UTM and UPS map projections, for identifying positions. An MGRS coordinate consists of a zone designation, alphabetic 100,000-meter grid square designator, and numeric coordinates within the 100,000 meter grid square.

APPENDIX C REFERENCES

(1) Topographic Engineering Center, TEC-SR-7, **Handbook for transformation of DATUMS, PROJECTIONS, GRIDS, AND COMMON COORDINATE SYSTEMS**, January 1996.