

USING THE CORPSCON DLL

This document describes how to use the Corpscon DLL. All relevant functions, their arguments and return values are given below. Function definitions are given in C. Users who want to use the DLL in another language (VisualBasic, for example) will need to refer to language specific documentation on how to implement and call DLL functions.

Conceptual Algorithm

Figure 1 is contains pseudo-code for an algorithm illustrating the steps required to perform conversions using the Corpscon DLL.

```
/* Set default configuration. */
corpscon_default_config()

/* Use Set functions to set data file paths. */
SetNadconPath()
SetVertconPath()
SetGeoidPath()

/* Use Set functions to systems, datums, zones, units, etc. */
SetInSystem()
SetInDatum()
...etc.
SetOutSystem()
SetOutDatum()
...etc.

/* Configure the DLL internally. */
corpscon_initialize_convert()

/* Conversion loop. */
loop

    /* Set input values. */
    SetXIn()
    SetYIn()
    SetZIn()

    /* Convert values input above. */
    corpscon_convert()

    /* Get output values. */
    GetXOut()
    GetYOut()
    GetZOut()

    /* Save or display output as needed. */
    your_code_or_function_to_do_something_with_output()

next

/* Clean up after conversions. */
corpscon_clean_up()
```

Figure 1 - Conceptual Algorithm

Function Definitions

Below is a list of all relevant functions included in the Corpscon DLL. They are listed approximately in the order that they should be used. An additional set functions used to check internal configuration settings is also included.

```
__declspec(dllexport) int __stdcall corpscon_default_config(void)
```

Description Initializes internal configuration structure. Call this function before calls to any other function.

Arguments None

Returns 1 for success; negative value for error

```
__declspec(dllexport) int __stdcall SetInSystem(int val)  
__declspec(dllexport) int __stdcall SetOutSystem(int val)
```

Description Sets input/output system (geographic, state plane, UTM, or USNG).

Arguments val: 1 - geographic
 2 - state plane
 3 - UTM
 4 - USNG

Returns 1 for success; negative value for error

```
__declspec(dllexport) int __stdcall SetInDatum(int val)  
__declspec(dllexport) int __stdcall SetOutDatum(int val)
```

Description Sets input/output horizontal datum (NAD27, NAD83, HPGN).

Arguments val: 1927 - NAD27
 1983 - NAD83
 1991 - HPGN

Returns 1 for success; negative value for error

```
__declspec(dllexport) int __stdcall SetInUnits(int val)  
__declspec(dllexport) int __stdcall SetOutUnits(int val)
```

Description Sets input/output horizontal units (U.S. Survey Feet, International Survey Feet, Meters).

Arguments val: 1 - U.S. Survey Feet
 2 - International Survey Feet
 3 - Meters

Returns 1 for success; negative value for error

```
__declspec(dllexport) int __stdcall SetInZone(int val)  
__declspec(dllexport) int __stdcall SetOutZone(int val)
```

Description Sets input/output state plane or UTM zone.

Arguments val: State Plane or UTM zone number

Returns 1 for success; negative value for error

```
__declspec(dllexport) int __stdcall SetInVDatum(int val)  
__declspec(dllexport) int __stdcall SetOutVDatum(int val)
```

Description Sets input/output vertical datum (NGVD29, NAVD88, GRS80).

Arguments val: 0 - No Vertical Conversion
 1929 - NGVD29
 1988 - NAVD88
 1980 - GRS80

Returns 1 for success; negative value for error

```
__declspec(dllexport) int __stdcall SetInVUnits(int val)  
__declspec(dllexport) int __stdcall SetOutVUnits(int val)
```

Description Sets input/output horizontal units (U.S. Survey Feet, International Survey Feet, Meters).

Arguments val: 1 - U.S. Survey Feet
 2 - International Survey Feet
 3 - Meters

Returns 1 for success; negative value for error

```
__declspec(dllexport) int __stdcall SetOutUSNGDigits(int val)
```

Description Sets number of output digits for USNG coordinates.

Arguments val: 1 - 99ZZ16
 2 - 99ZZ1267
 3 - 99ZZ123678
 4 - 99ZZ12346789
 5 - 99ZZ1234567890

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall SetNadconPath(char *path)

Description Sets the path where the Nadcon files (.las/.los) are located.

Arguments path: path name to Nadcon files

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall SetInHPGNArea(char *area)

__declspec(dllexport) int __stdcall SetOutHPGNArea(char *area)

Description Sets the input/output base file name for the files used in HPGN conversions. This area file will be appended to the Nadcon path, so HPGN files must be located in the directory given by SetNadconPath().

Arguments path: HPGN base file name (no extension)

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall SetVertconPath(char *path)

Description Sets the path where the Vertcon files (.94) are located.

Arguments path: path name to Vertcon files

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall SetUseVertconCustomAreas(int opt)

Description Specifies whether to use standard or custom Vertcon files.

Arguments opt: 0 - use standard Vertcon files
1 - use files from Vertcon custom area list file

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall SetVertconCustomAreaListFile(char *file)

Description Sets the filename for a Vertcon custom area list file.

Arguments file: Vertcon custom area file

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall SetGeoidPath(char *path)

Description Sets the path where the Geoid files (.geo/.bin) are located. Geoid90/93/96/99 files should be in this directory.

Arguments path: path name to Geoid files

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall SetGeoidCodeBase(int val)

Description Sets Geoid model for standard conversion or sets code base to use for custom Geoid conversions.

Arguments val: 1990 - Geoid90 model or Geoid90/93/96 code base
1993 - Geoid93 model or Geoid90/93/96 code base
1996 - Geoid96 model or Geoid90/93/96 code base
1999 - Geoid99 model or Geoid99/03 code base
2003 - Geoid03 model or Geoid99/03 code base

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall SetUseGeoidCustomAreas(int opt)

Description Specifies whether to use standard or custom Geoid files.

Arguments opt: 0 - use standard Geoid files
1 - use files from Geoid custom area list file

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall SetGeoidCustomAreaListFile(char *file)

Description Sets the filename for a Geoid custom area list file.

Arguments file: Geoid custom area file

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall corpscon_initialize_convert(void)

Description Initialize DLL internally for conversions. This function prepares internal functions for conversion and opens required Nadcon, Vertcon, and Geoid files for reading.

Arguments None

Returns 1 for success; negative value for error

__declspec(dllexport) int _stdcall SetXIn(double val)

Description Set X/Easting/Longitude value to be used when corpscon_convert() is called.

Arguments Val: X/Easting/Longitude value

Returns 1 for success; negative value for error

__declspec(dllexport) int _stdcall SetYIn(double val)

Description Set Y/Northing/Latitude value to be used when corpscon_convert() is called.

Arguments Val: Y/Northing/Latitude value

Returns 1 for success; negative value for error

__declspec(dllexport) int _stdcall SetZIn(double val)

Description Set Z/Elevation value to be used when corpscon_convert() is called.

Arguments Val: Z/Elevation value

Returns 1 for success; negative value for error

__declspec(dllexport) int _stdcall SetUSNGIn(char *val)

Description Set USNG input value to be used when corpscon_convert() is called. Use this function instead of SetXIn() and SetYIn() when performing USNG conversions.

Arguments val: USNG value

Returns 1 for success; negative value for error

__declspec(dllexport) int _stdcall corpscon_convert(void)

Description Perform a conversion.

Arguments None

Returns 1 for success; negative value for error

__declspec(dllexport) int _stdcall GetErrorMessage(int err_code, char *msg)

Description Retrieve error message from return value. Use this function to get a brief text description of return codes. Corpscon generally returns 1 as success. Other return values can be passed as *err_code* arguments to this function.

Arguments *err_code*: error code
msg: text string describing error (passed by reference)

Returns always returns 1

__declspec(dllexport) double _stdcall GetXOut(void)

Description Get converted X/Easting/Longitude value.

Arguments None

Returns Converted X/Easting/Longitude value

__declspec(dllexport) double _stdcall GetYOut(void)

Description Get converted Y/Northing/Latitude value.

Arguments None

Returns Converted Y/Northing/Latitude value

__declspec(dllexport) double _stdcall GetZOut(void)

Description Get converted Z/Elevation value.

Arguments none

Returns converted Z/Elevation value

__declspec(dllexport) int _stdcall GetUSNGOut(char *val)

Description Get output USNG value. Use this function instead of *GetXOut()* and *GetYOut()* when performing USNG conversions.

Arguments *val*: output USNG value (passed by reference)

Returns 1 for success; negative value for error

```
__declspec(dllexport) int _stdcall GetCorpsconValue(int code, double *val)
```

Description Get other data about conversion.

Arguments code: 5 - output scale factor
 6 - output combined factor
 7 - output convergence in decimal degrees
 10 - output latitude shift in meters
 11 - output longitude shift in meters
 105 - input scale factor
 106 - input combined factor
 107 - input convergence in decimal degrees
 110 - input latitude shift in meters
 111 - input longitude shift in meters
 204 - total latitude shift in meters
 205 - total longitude shift in meters

 val: retrieved value (passed by reference)

Returns 1 for success; negative value for error

```
__declspec(dllexport) int _stdcall corpscon_clean_up(void)
```

Description Clean up the conversion. This function closes all Nadcon, Vertcon, and Geoid files opened during conversion.

Arguments none

Returns 1 for success; negative value for error

Set Functions

These functions can be used to check the internal configuration of the DLL. They are not needed to perform conversions, but are included as tools to help perform error checking.

```
__declspec(dllexport) int __stdcall GetInSystem(void)  
__declspec(dllexport) int __stdcall GetOutSystem(void)
```

Description Retrieves internal settings for input/output system (geographic, state plane, UTM, or USNG).

Arguments none

Returns 1 - geographic
 2 - state plane
 3 - UTM
 4 - USNG

 negative value for error

```
__declspec(dllexport) int __stdcall GetInDatum(void)  
__declspec(dllexport) int __stdcall GetOutDatum(void)
```

Description Retrieves internal settings for input/output datum (NAD27, NAD83, HPGN).

Arguments none

Returns 1927 - NAD27
 1983 - NAD83
 1991 - HPGN

 negative value for error

```
__declspec(dllexport) int __stdcall GetInUnits(void)  
__declspec(dllexport) int __stdcall GetOutUnits(void)
```

Description Retrieves internal settings for input/output horizontal units (U.S. Survey Feet, International Feet, Meters).

Arguments none

Returns 1 - U.S. Survey Feet
 2 - International Feet
 3 - Meters

 negative value for error

```
__declspec(dllexport) int _stdcall GetInZone(void)  
__declspec(dllexport) int _stdcall GetOutZone(void)
```

Description Retrieves internal settings for input/output state plane or UTM zone.

Arguments none

Returns zone number; negative value for error

```
__declspec(dllexport) int _stdcall GetInVDatum(void)  
__declspec(dllexport) int _stdcall GetOutVDatum(void)
```

Description Retrieves internal settings for input/output vertical datum (NGAD29, NAVD88, GRS80).

Arguments none

Returns 0 - No vertical conversion
 1929 - NGVD29
 1988 - NAVD83
 1980 - GRS80

 negative value for error

```
__declspec(dllexport) int _stdcall GetInVUnits(void)  
__declspec(dllexport) int _stdcall GetOutVUnits(void)
```

Description Retrieves internal settings for input/output vertical units (U.S. Survey Feet, International Feet, Meters).

Arguments none

Returns 1 - U.S. Survey Feet
 2 - International Feet
 3 - Meters

 negative value for error

__declspec(dllexport) int __stdcall GetOutUSNGDigits(void)

Description Retrieves internal settings for output USNG coordinate digits.

Arguments none

Returns 1 - 99ZZ16
 2 - 99ZZ1267
 3 - 99ZZ123678
 4 - 99ZZ12346789
 5 - 99ZZ1234567890

 negative value for error

__declspec(dllexport) int __stdcall GetNadconPath(char *path)

Description Retrieves internal settings for Nadcon files path.

Arguments path: path to Nadcon files (passed by reference)

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall GetInHPGNArea(char *area)
__declspec(dllexport) int __stdcall GetOutHPGNArea(char *area)

Description Retrieves internal settings for HPGN base file names.

Arguments area: base file name of HPGN area (passed by reference)

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall GetVertconPath(char *path)

Description Retrieves internal settings for Vertcon files path.

Arguments path: path to Vertcon files (passed by reference)

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall GetVertconCustomAreaListFile(char *filename)

Description Retrieves internal settings for Vertcon custom area list file name.

Arguments filename: Vertcon custom area list file (passed by reference)

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall GetUseVertconCustomAreas(void)

Description Retrieves internal setting for use of standard or custom Vertcon custom areas.

Arguments none

Returns 0 - used standard Vertcon files
 1 - for files in Vertcon custom areas file

 negative value for error

__declspec(dllexport) int __stdcall GetGeoidCodeBase(void)

Description Retrieves internal setting Geoid model/code base.

Arguments none

Returns 1990 - Geoid90
 1993 - Geoid93
 1996 - Geoid96
 1999 - Geoid99
 2003 - Geoid03

 negative value for error

__declspec(dllexport) int __stdcall GetGeoidPath(char *path)

Description Retrieves internal settings for Geoid files path.

Arguments path: path to Geoid files (passed by reference)

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall GetGeoidCustomAreaListFile(char *filename)

Description Retrieves internal settings for Geoid custom area list file name.

Arguments filename: Geoid custom area list file (passed by reference)

Returns 1 for success; negative value for error

__declspec(dllexport) int __stdcall GetUseGeoidCustomAreas(void)

Description Retrieves internal setting for use of standard or custom Geoid custom areas.

Arguments none

Returns 0 - used standard Geoid files
 1 - for files in Geoid custom areas file

 negative value for error