

DATUM

10xxxxxx.1

LEGAL RESTRICTIONS:

This Reusable Software Component (RSC) contains data with Unlimited Government Rights.

DESCRIPTION:

DATUM is a C language code component that provides access to datum shift parameters, relative to WGS 84, for a large collection of local datums.

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.

LEVEL OF TESTING/ACCEPTANCE:

Unit and integration testing have been performed for the functions contained in this component.

PURPOSE/INTENDED USE:

This component provides datum shifts for a large collection of local datums, WGS 72 and WGS 84. A particular datum can be accessed by using its standard 5-letter code to find its index in the datum table. The index can then be used to retrieve the datum name, type, ellipsoid code, and datum shift parameters, and to perform data shifts to or from that datum.

By sequentially retrieving all of the datum codes and/or names, a menu of the available datums can be constructed. The index values resulting from selections from this menu can then be used to access the parameters of the selected datum, or to perform datum shifts involving that datum.

This component supports both 3-parameter local datums, for which only X, Y, and Z translations relative to WGS 84 have been defined, and 7-parameter local datums, for which X,

Y, and Z rotations, and a scale factor, are also defined. It also includes entries for WGS 84 (with an index of 0), and WGS 72 (with an index of 1), but no shift parameter values are defined for these.

This component provides datum shift functions for both geocentric and geodetic coordinates. WGS 84 is used as an intermediate state when shifting from one local datum to another. When geodetic coordinates are given, Molodensky's method is normally used. Specific algorithms are used for shifts between WGS 72 and WGS 84.

This component depends on two data files, named "3_param.dat" and "7_param.dat", which contain the datum parameter values. Copies of these files must be located in the directory specified by the value of the environment variable "DATUM_DATA", if defined, or else in the current directory, whenever a program containing this component is executed.

Additional datums can be added to these files, either manually or using the Create_Datum function. However, if a large number of datums are added, the datum table array sizes in this component will have to be increased.

This component depends on two other components: the ELLIPSOID component for access to ellipsoid parameters; and the GEOCENTRIC component for conversions between geodetic and geocentric coordinates.

HARDWARE/ENVIRONMENT CONSTRAINTS:

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

FUNCTIONS:

Initialize_Datums – This function creates the datum table from two external files.

Create_Datum – This function creates a new (3-parameter) local datum with the specified code, name, and parameters, adds it to the datum table, and updates the 3-parameter datum data file.

Datum_Count – This function returns the number of datums in the table.

Datum_Index – This function returns the index of the datum with the specified code.

Datum_Code – This function returns the 5-letter code of the datum referenced by index.

Datum_Name – This function returns the name of the datum referenced by index.

Datum_Ellipsoid_Code – This function returns the 2-letter ellipsoid code for the ellipsoid associated with the datum referenced by index.

Get_Datum_Type – This function returns the type (3-parameter, 7-parameter, WGS 72, or WGS 84) of the datum referenced by index.

Datum_Seven_Parameters – This function returns the seven parameters for the datum referenced by index.

Datum_Three_Parameters – This function returns the three parameters for the datum referenced by index.

Datum_Errors – This function returns the standard errors in X, Y, and Z for the datum referenced by index.

Datum_Valid_Rectangle – This function returns the edges of the validity rectangle for the datum referenced by index.

Valid_Datum – This function checks whether or not the specified location is within the validity rectangle for the datum referenced by index.

Geocentric_Datum_Shift – This function shifts a geocentric coordinate relative to one specified datum to a geocentric coordinate relative to another specified datum.

Geocentric_Shift_To_WGS84 – This function shifts a geocentric coordinate relative to a specified datum to a geocentric coordinate relative to WGS 84.

Geocentric_Shift_From_WGS84 – This function shifts a geocentric coordinate relative to WGS 84 to a geocentric coordinate relative to a specified datum.

Geodetic_Datum_Shift – This function shifts a geodetic coordinate relative to one specified datum to a geodetic coordinate relative to another specified datum.

Geodetic_Shift_To_WGS84 – This function shifts a geodetic coordinate relative to a specified datum to a geodetic coordinate relative to WGS 84.

Geodetic_Shift_From_WGS84 – This function shifts a geodetic coordinate relative to WGS 84 to a geodetic coordinate relative to a specified datum.

Datum_Shift_Error – This function returns the 90% horizontal (circular), vertical (linear), and spherical errors for a shift from the specified source datum to the specified destination datum at the specified location.

EXAMPLE APPLICATIONS:

The following example illustrates how DATUM can be used to retrieve information about a datum.

Function Call:

```
status = Initialize_Datums()
```

Inputs:

none.

Outputs:

none.

Function Call:

```
status = Datum_Index(code, index)
```

Inputs:

code	"TOY-C"
------	---------

Outputs:

index:	201
--------	-----

Function Call:

```
status = Datum_Name(index, name)
```

Inputs:

Index:	201
--------	-----

Outputs:

name:	"TOKYO Okinawa"
-------	-----------------

Function Call:

```
status = Datum_Ellipsoid_Code(index, code)
```

Inputs:

index:	201
--------	-----

Outputs:

code: "BR"

Function Call:

```
status = Get_Datum_Type(index, type)
```

Inputs:

index: 201

Outputs:

type: Three_Param_Datum

Function Call:

```
status = Datum_Three_Parameters(index, delta_X, delta_Y, delta_Z)
```

Inputs:

index: 201

Outputs:

delta_X: -158

delta_Y: 507

delta_Z: 676

Function Call:

```
status = Geocentric_Shift_To_WGS84 (index, X, Y, Z, X_WGS84, Y_WGS84,  
Z_WGS84)
```

Inputs:

index: 201

X: 0.0

Y: 0.0

Z: 0.0

Outputs:

X_WGS84: -158.0

Y_WGS84: 507.0

Z_WGS84: 676.0