Source Code for LLUTMN_ Latitude-Longitude <=> Universal Transverse Mercator Projection Program

Gregory W Pouch
Geohydrology Section
Kansas Geological Survey

This report describes the workings of projection program LLUTMN_, which converts between Geographic Angular and Universal Transverse Mercator coordinates. LLUTMN is an event-driven program written in Microsoft Visual Basic 3.0 Professional Edition.

LLUTMN is essentially a graphical user-interface wrapper around the General Cartographic Transformation Package (GCTP) from the National Oceanic and Atmospheric Administration's (NOAA).

The GCTP is a Fortran library of projection routines for a wide selection of map coordinate systems. (Elassal, 1987). LLUTMN uses a copy of GCTP obtained over the internet from the USGS at charon.er.usgs.gov: this may not correspond exactly to the original GCTP, although it does seem to project coordinates properly. The "internet" copy and some documentation on how to use it are available as KGS OFR 95-15.

Although event-driven programs are easy to use, they are not easy to document, so only a brief overview will be given. The reader is advised to get the program, open it in the Visual Basic design environment, and single-step through the program to get an idea of how it works.

For interactive-entry, the program first converts the value of the "input" text boxes to numbers, maybe performs a unit conversion, calls the GCTP routine to convert to/from latitude-longitude, then displays the coordinates in the "output" text boxes. (Note that output and input designations depend on which command button the user clicked.) For database conversions, the user selects the table and fields from comboboxes, then chooses the direction of transformation. The program reads the data a record at a time, performs the conversion as above, stores the output results in fields in the same table, and continues until there are no more records left. For point data, the coordinates are stored in two fields. For line/polygon data, the coordinates are stored in a WHEAT XYChain, which is simply a binary copy of an array of RealPoints (X, Y as two 4-byte floating point numbers) stored in a binary large object field.

LLUTMN_ is hard-wired to use only Universal Transverse Mercator and Geographic Angular coordinates. The UTM zone number is controlled read from a text box each time a projection is made, thus allowing the use any UTM zone and even the questionable practice of converting coordinates from one zone to another in a two-step process of UTM A -> Lat-Long -> UTM B.

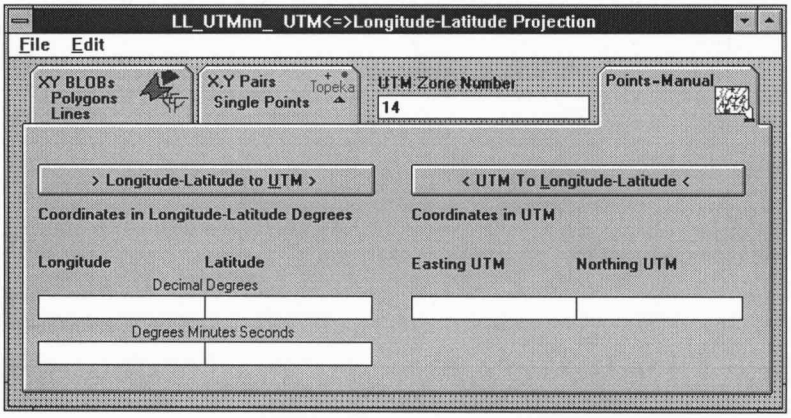
Elassal, Atef A., 1987, General Cartographic Transformation Package (GCTP), Version II, NOAA Technical Report NOS 124 CGS 9, 23 pp.

VB CROSS REFERENCE - TABLE OF CONTENTS

For file: LLUTMNN_.MAK
3/31/96 2:54:52 PM

Page #

Source Listing
    LLUTMNN_.FRM (illustration) 1
    LLUTMNN_.FRM: (declarations) 2
    GCTP0001.BAS: (declarations) 29
    XYBLOB01.BAS: (declarations) 36


LL_UTMnn_ UTM<=>Longitude-Latitude Projection
File Edit
XY BLOBs
Polygons
Lines
X,Y Pairs
Single Points
Topeka
UTM Zone Number
14
Points-Manual
> Longitude-Latitude to UTM >
< UTM To Longitude-Latitude <
Coordinates in Longitude-Latitude Degrees
Coordinates in UTM
Longitude
Latitude
Decimal Degrees
Degrees Minutes Seconds
Easting UTM
Northing UTM

```
VERSION 2.00
Begin Form frmProjection_BOTH
   BackColor       =   &H00C0C0C0&
   Caption         =   "LL_UTMnn_  UTM<=>Longitude-Latitude
Projection"
   ClientHeight    =   4095
   ClientLeft      =   255
   ClientTop       =   1545
   ClientWidth     =   8970
   Height          =   4785
   Icon            =   LLUTMNN_.FRX:0000
   Left            =   195
   LinkTopic       =   "Form1"
   ScaleHeight     =   273
   ScaleMode       =   3  'Pixel
   ScaleWidth      =   598
   Top             =   915
   Width           =   9090
   Begin TextBox txtUTMZoneNumber
      Height          =   285
      Left            =   4260
      TabIndex        =   51
      Text            =   "14"
      Top             =   420
      Width           =   2475
   End
   Begin SSRibbon cmdManual
      AutoSize        =   1  'Adjust Picture Size To Button
      BackColor       =   &H00C0C0C0&
      BevelWidth      =   0
      Height          =   720
      Left            =   6825
      Outline         =   0   'False
      PictureDn       =   LLUTMNN_.FRX:0302
      PictureDnChange =   1  'Dither 'PictureUp' Bitmap
      PictureUp       =   LLUTMNN_.FRX:0F7C
      Top             =   75
      Value           =   -1  'True
      Width           =   1920
   End
   Begin SSRibbon cmdBLOBs
      AutoSize        =   1  'Adjust Picture Size To Button
      BackColor       =   &H00C0C0C0&
      BevelWidth      =   0
      Height          =   720
      Left            =   240
      Outline         =   0   'False
      PictureDn       =   LLUTMNN_.FRX:1BF6
      PictureDnChange =   1  'Dither 'PictureUp' Bitmap
      PictureUp       =   LLUTMNN_.FRX:2870
      Top             =   75
      Width           =   1920
   End
   Begin SSRibbon cmdPoints
      AutoSize        =   1  'Adjust Picture Size To Button
      BackColor       =   &H00C0C0C0&
      BevelWidth      =   0
      Height          =   720
```

```
        Left            =   2175
        Outline         =   0   'False
        PictureDn       =   LLUTMNN_.FRX:34EA
        PictureDnChange =   1  'Dither 'PictureUp' Bitmap
        PictureUp       =   LLUTMNN_.FRX:4164
        Top             =   75
        Width           =   1920
    End
    Begin SSPanel pnlManualEntry
        BackColor       =   &H00C0C0C0&
        BevelWidth      =   2
        Font3D          =   0  'None
        Height          =   3200
        Left            =   150
        Outline         =   -1  'True
        TabIndex        =   48
        Top             =   750
        Width           =   8715
        Begin TextBox txtJunk
            Height          =   315
            Left            =   15000
            TabIndex        =   46
            Text            =   "This could be scrapped"
            Top             =   2520
            Width           =   615
        End
        Begin TextBox txtLatDMS
            FontBold        =   0   'False
            FontItalic      =   0   'False
            FontName        =   "MS Sans Serif"
            FontSize        =   8.25
            FontStrikethru  =   0   'False
            FontUnderline   =   0   'False
            Height          =   285
            Left            =   2100
            TabIndex        =   41
            Top             =   2580
            Width           =   1950
        End
        Begin TextBox txtLongDMS
            FontBold        =   0   'False
            FontItalic      =   0   'False
            FontName        =   "MS Sans Serif"
            FontSize        =   8.25
            FontStrikethru  =   0   'False
            FontUnderline   =   0   'False
            Height          =   285
            Left            =   180
            TabIndex        =   40
            Top             =   2580
            Width           =   1950
        End
        Begin TextBox txtUTMNorthing
            FontBold        =   0   'False
            FontItalic      =   0   'False
            FontName        =   "MS Sans Serif"
            FontSize        =   8.25
            FontStrikethru  =   0   'False
```

```
116|          FontUnderline   =   0   'False
117|          Height          =   285
118|          Left            =   6420
119|          TabIndex        =   39
120|          Top             =   2040
121|          Width           =   1950
122|       End
123|       Begin TextBox txtUTMEasting
124|          FontBold        =   0   'False
125|          FontItalic      =   0   'False
126|          FontName        =   "MS Sans Serif"
127|          FontSize        =   8.25
128|          FontStrikethru  =   0   'False
129|          FontUnderline   =   0   'False
130|          Height          =   285
131|          Left            =   4500
132|          TabIndex        =   38
133|          Top             =   2040
134|          Width           =   1950
135|       End
136|       Begin TextBox txtLatDecDeg
137|          FontBold        =   0   'False
138|          FontItalic      =   0   'False
139|          FontName        =   "MS Sans Serif"
140|          FontSize        =   8.25
141|          FontStrikethru  =   0   'False
142|          FontUnderline   =   0   'False
143|          Height          =   285
144|          Left            =   2100
145|          TabIndex        =   37
146|          Top             =   2040
147|          Width           =   1950
148|       End
149|       Begin TextBox txtLongDecDeg
150|          FontBold        =   0   'False
151|          FontItalic      =   0   'False
152|          FontName        =   "MS Sans Serif"
153|          FontSize        =   8.25
154|          FontStrikethru  =   0   'False
155|          FontUnderline   =   0   'False
156|          Height          =   285
157|          Left            =   180
158|          TabIndex        =   36
159|          Top             =   2040
160|          Width           =   1950
161|       End
162|       Begin CommandButton cmdLongLatFromUTM_MANUAL
163|          Caption         =   "< UTM To &Longitude-Latitude <"
164|          Height          =   375
165|          Left            =   4500
166|          TabIndex        =   35
167|          Top             =   480
168|          Width           =   3900
169|       End
170|       Begin CommandButton cmdUTMFromLongLat_Manual
171|          Caption         =   "> Longitude-Latitude to &UTM >"
172|          Height          =   375
173|          Left            =   180
```

```
174|          TabIndex        =   34
175|          Top             =   480
176|          Width           =   3900
177|       End
178|       Begin Label lblNorthing_Manual
179|          BackStyle       =   0  'Transparent
180|          Caption         =   "Northing UTM"
181|          Height          =   195
182|          Left            =   6420
183|          TabIndex        =   49
184|          Top             =   1560
185|          Width           =   1815
186|       End
187|       Begin Label lblEasting_MANUAL
188|          BackStyle       =   0  'Transparent
189|          Caption         =   "Easting UTM"
190|          Height          =   195
191|          Left            =   4500
192|          TabIndex        =   47
193|          Top             =   1560
194|          Width           =   1815
195|       End
196|       Begin Label lblDMS_MANUAL
197|          BackStyle       =   0  'Transparent
198|          Caption         =   "Degrees Minutes Seconds"
199|          FontBold        =   0   'False
200|          FontItalic      =   0   'False
201|          FontName        =   "MS Sans Serif"
202|          FontSize        =   8.25
203|          FontStrikethru  =   0   'False
204|          FontUnderline   =   0   'False
205|          Height          =   195
206|          Left            =   1260
207|          TabIndex        =   45
208|          Top             =   2370
209|          Width           =   2415
210|       End
211|       Begin Label lblDecDeg
212|          BackStyle       =   0  'Transparent
213|          Caption         =   "Decimal Degrees"
214|          FontBold        =   0   'False
215|          FontItalic      =   0   'False
216|          FontName        =   "MS Sans Serif"
217|          FontSize        =   8.25
218|          FontStrikethru  =   0   'False
219|          FontUnderline   =   0   'False
220|          Height          =   195
221|          Left            =   1500
222|          TabIndex        =   44
223|          Top             =   1820
224|          Width           =   1275
225|       End
226|       Begin Label lblLatitude_MANUAL
227|          BackStyle       =   0  'Transparent
228|          Caption         =   "Latitude"
229|          Height          =   195
230|          Left            =   2100
231|          TabIndex        =   43
```

```
232|          Top             =   1550
233|          Width           =   1875
234|       End
235|       Begin Label lblLongitude_MANUAL
236|          BackStyle       =   0  'Transparent
237|          Caption         =   "Longitude"
238|          Height          =   255
239|          Left            =   180
240|          TabIndex        =   42
241|          Top             =   1550
242|          Width           =   1635
243|       End
244|       Begin Label lblUTM14_MANUAL
245|          BackStyle       =   0  'Transparent
246|          Caption         =   "Coordinates in UTM"
247|          Height          =   375
248|          Left            =   4500
249|          TabIndex        =   33
250|          Top             =   980
251|          Width           =   3975
252|       End
253|       Begin Label lblLongLat_MANUAL
254|          BackStyle       =   0  'Transparent
255|          Caption         =   "Coordinates in Longitude-Latitude
    Degrees"
256|          Height          =   495
257|          Left            =   180
258|          TabIndex        =   32
259|          Top             =   980
260|          Width           =   3975
261|       End
262|    End
263|    Begin SSPanel pnlSinglePoints
264|       BackColor       =   &H00C0C0C0&
265|       BevelWidth      =   2
266|       Font3D          =   0  'None
267|       Height          =   3200
268|       Left            =   150
269|       Outline         =   -1  'True
270|       TabIndex        =   0
271|       Top             =   750
272|       Width           =   8715
273|       Begin PictureBox picGray
274|          BackColor       =   &H00C0C0C0&
275|          BorderStyle     =   0  'None
276|          Height          =   300
277|          Index           =   5
278|          Left            =   8120
279|          ScaleHeight     =   300
280|          ScaleWidth      =   105
281|          TabIndex        =   31
282|          Top             =   1680
283|          Width           =   100
284|       End
285|       Begin PictureBox picGray
286|          BackColor       =   &H00C0C0C0&
287|          BorderStyle     =   0  'None
288|          Height          =   300
```

```
289|          Index           =   4
290|          Left            =   8120
291|          ScaleHeight     =   300
292|          ScaleWidth      =   105
293|          TabIndex        =   30
294|          Top             =   2340
295|          Width           =   100
296|       End
297|       Begin PictureBox picGray
298|          BackColor       =   &H00C0C0C0&
299|          BorderStyle     =   0  'None
300|          Height          =   300
301|          Index           =   3
302|          Left            =   3800
303|          ScaleHeight     =   300
304|          ScaleWidth      =   105
305|          TabIndex        =   29
306|          Top             =   2340
307|          Width           =   100
308|       End
309|       Begin PictureBox picGray
310|          BackColor       =   &H00C0C0C0&
311|          BorderStyle     =   0  'None
312|          Height          =   300
313|          Index           =   2
314|          Left            =   3800
315|          ScaleHeight     =   300
316|          ScaleWidth      =   105
317|          TabIndex        =   28
318|          Top             =   1680
319|          Width           =   100
320|       End
321|       Begin Data Dat_Points
322|          Caption         =   "Move record "
323|          Connect         =   ""
324|          DatabaseName    =   ""
325|          Exclusive       =   0   'False
326|          Height          =   270
327|          Left            =   9120
328|          Options         =   0
329|          ReadOnly        =   0   'False
330|          RecordSource    =   ""
331|          Top             =   1200
332|          Width           =   2445
333|       End
334|       Begin ComboBox cmbQTList_Points
335|          Height          =   300
336|          Left            =   90
337|          Sorted          =   -1  'True
338|          Style           =   2  'Dropdown List
339|          TabIndex        =   9
340|          Top             =   120
341|          Width           =   8535
342|       End
343|       Begin ComboBox cmbXY_LongLat_Points
344|          Height          =   300
345|          Index           =   0
346|          Left            =   180
```

```
347|          TabIndex        =   8
348|          Text            =   "X Longitude"
349|          Top             =   1680
350|          Width           =   3968
351|       End
352|       Begin ComboBox cmbXY_UTM_Points
353|          Height          =   300
354|          Index           =   0
355|          Left            =   4500
356|          TabIndex        =   7
357|          Text            =   "X UTM 14"
358|          Top             =   1680
359|          Width           =   3968
360|       End
361|       Begin CommandButton cmdUTMFromLongLat_Points
362|          Caption         =   "> Longitude-Latitude to &UTM >"
363|          Height          =   375
364|          Left            =   180
365|          TabIndex        =   6
366|          Top             =   480
367|          Width           =   3900
368|       End
369|       Begin CommandButton cmdLatLongFromUTM_Points
370|          Caption         =   "< UTM To &Longitude-Latitude <"
371|          Height          =   375
372|          Left            =   4500
373|          TabIndex        =   5
374|          Top             =   480
375|          Width           =   3900
376|       End
377|       Begin PictureBox picPercentDoneHost_Points
378|          Height          =   255
379|          Left            =   120
380|          ScaleHeight     =   1
381|          ScaleMode       =   0  'User
382|          ScaleWidth      =   97.237
383|          TabIndex        =   3
384|          Top             =   2760
385|          Width           =   8475
386|          Begin PictureBox picPercentDone_Points
387|             BackColor       =   &H00FF0000&
388|             BorderStyle     =   0  'None
389|             Height          =   225
390|             Left            =   0
391|             ScaleHeight     =   225
392|             ScaleWidth      =   15
393|             TabIndex        =   4
394|             Top             =   0
395|             Width           =   15
396|          End
397|       End
398|       Begin ComboBox cmbXY_LongLat_Points
399|          Height          =   300
400|          Index           =   1
401|          Left            =   180
402|          TabIndex        =   2
403|          Text            =   "Y Latitude"
404|          Top             =   2340
```

```
405|          Width           =   3968
406|       End
407|       Begin ComboBox cmbXY_UTM_Points
408|          Height          =   300
409|          Index           =   1
410|          Left            =   4500
411|          TabIndex        =   1
412|          Text            =   "Y UTM 14"
413|          Top             =   2340
414|          Width           =   3968
415|       End
416|       Begin Label lblBLOBLatLong
417|          BackStyle       =   0  'Transparent
418|          Caption         =   "Felds containing coordinates in
    Longitude-Latitude Degrees"
419|          Height          =   495
420|          Left            =   180
421|          TabIndex        =   15
422|          Top             =   1000
423|          Width           =   3968
424|       End
425|       Begin Label lblUTMFields
426|          BackStyle       =   0  'Transparent
427|          Caption         =   "Fields containing coordinates in
    UTM"
428|          Height          =   375
429|          Left            =   4500
430|          TabIndex        =   14
431|          Top             =   1000
432|          Width           =   3968
433|       End
434|       Begin Label lblLong_Points
435|          BackStyle       =   0  'Transparent
436|          Caption         =   "X Longitude"
437|          Height          =   255
438|          Index           =   0
439|          Left            =   180
440|          TabIndex        =   13
441|          Top             =   1440
442|          Width           =   3968
443|       End
444|       Begin Label lblLong_Points
445|          BackStyle       =   0  'Transparent
446|          Caption         =   "Y Latitude"
447|          Height          =   255
448|          Index           =   1
449|          Left            =   180
450|          TabIndex        =   12
451|          Top             =   2100
452|          Width           =   3975
453|       End
454|       Begin Label zUTMLabel2
455|          BackStyle       =   0  'Transparent
456|          Caption         =   "Easting UTM 14      X"
457|          Height          =   255
458|          Index           =   0
459|          Left            =   4500
460|          TabIndex        =   11
```

```
461|          Top             =   1440
462|          Width           =   3975
463|       End
464|       Begin Label zUTMLabel2
465|          BackStyle       =   0  'Transparent
466|          Caption         =   "Northing UTM 14      Y"
467|          Height          =   255
468|          Index           =   1
469|          Left            =   4500
470|          TabIndex        =   10
471|          Top             =   2100
472|          Width           =   3975
473|       End
474|    End
475|    Begin SSPanel pnlBLOBs
476|       BackColor       =   &H00C0C0C0&
477|       BevelWidth      =   2
478|       Font3D          =   0  'None
479|       Height          =   3200
480|       Left            =   150
481|       Outline         =   -1  'True
482|       TabIndex        =   16
483|       Top             =   750
484|       Width           =   8715
485|       Begin PictureBox picGray
486|          BackColor       =   &H00C0C0C0&
487|          BorderStyle     =   0  'None
488|          Height          =   300
489|          Index           =   1
490|          Left            =   8120
491|          ScaleHeight     =   300
492|          ScaleWidth      =   105
493|          TabIndex        =   27
494|          Top             =   1680
495|          Width           =   100
496|       End
497|       Begin PictureBox picGray
498|          BackColor       =   &H00C0C0C0&
499|          BorderStyle     =   0  'None
500|          Height          =   300
501|          Index           =   0
502|          Left            =   3800
503|          ScaleHeight     =   300
504|          ScaleWidth      =   105
505|          TabIndex        =   26
506|          Top             =   1680
507|          Width           =   100
508|       End
509|       Begin Data Dat_BLOBS
510|          Caption         =   "Move record "
511|          Connect         =   ""
512|          DatabaseName    =   ""
513|          Exclusive       =   0   'False
514|          Height          =   270
515|          Left            =   9120
516|          Options         =   0
517|          ReadOnly        =   0   'False
518|          RecordSource    =   ""
```

```
519|          Top             =   1200
520|          Width           =   2445
521|       End
522|       Begin ComboBox cmbQTList_BLOBS
523|          Height          =   300
524|          Left            =   90
525|          Sorted          =   -1  'True
526|          Style           =   2  'Dropdown List
527|          TabIndex        =   23
528|          Top             =   120
529|          Width           =   8535
530|       End
531|       Begin CommonDialog CMDialog1
532|          Left            =   9120
533|          Top             =   4320
534|       End
535|       Begin ComboBox cmbXY_LongLat_BLOBS
536|          Height          =   300
537|          Left            =   180
538|          TabIndex        =   22
539|          Text            =   "XYBLOB Longitude Latitude"
540|          Top             =   1680
541|          Width           =   3968
542|       End
543|       Begin ComboBox cmbXY_UTM_BLOBS
544|          Height          =   300
545|          Left            =   4500
546|          TabIndex        =   21
547|          Text            =   "cmbXY_UTM14_BLOBS"
548|          Top             =   1680
549|          Width           =   3968
550|       End
551|       Begin CommandButton cmdUTMFromLongLat_BLOBS
552|          Caption         =   "> Longitude-Latitude to &UTM >"
553|          Height          =   375
554|          Left            =   180
555|          TabIndex        =   20
556|          Top             =   480
557|          Width           =   3900
558|       End
559|       Begin CommandButton cmdLatLongFromUTM_BLOBS
560|          Caption         =   "< UTM To &Longitude-Latitude <"
561|          Height          =   375
562|          Left            =   4500
563|          TabIndex        =   19
564|          Top             =   480
565|          Width           =   3900
566|       End
567|       Begin PictureBox picPercentDoneHost
568|          Height          =   255
569|          Left            =   180
570|          ScaleHeight     =   1
571|          ScaleMode       =   0  'User
572|          ScaleWidth      =   96.546
573|          TabIndex        =   17
574|          Top             =   2800
575|          Width           =   8415
576|          Begin PictureBox picPercentDone
```

```
577|             BackColor       =   &H00FF0000&
578|             BorderStyle     =   0  'None
579|             Height          =   225
580|             Left            =   0
581|             ScaleHeight     =   225
582|             ScaleWidth      =   15
583|             TabIndex        =   18
584|             Top             =   0
585|             Width           =   15
586|          End
587|       End
588|       Begin Label lblBLOB_LatLong
589|          BackStyle       =   0  'Transparent
590|          Caption         =   "Binary Large OBject (BLOB) field
    containing coordinates in Longitude-Latitude Degrees"
591|          Height          =   495
592|          Left            =   180
593|          TabIndex        =   25
594|          Top             =   1200
595|          Width           =   3975
596|       End
597|       Begin Label lblBLOBS_UTM
598|          BackStyle       =   0  'Transparent
599|          Caption         =   "Binary Large OBject (BLOB) field
    containing coordinates in UTM"
600|          Height          =   375
601|          Left            =   4500
602|          TabIndex        =   24
603|          Top             =   1200
604|          Width           =   3975
605|       End
606|    End
607|    Begin Label lblUTMZoneNumber
608|       BackStyle       =   0  'Transparent
609|       Caption         =   "UTM Zone Number"
610|       Height          =   255
611|       Left            =   4260
612|       TabIndex        =   50
613|       Top             =   180
614|       Width           =   2415
615|    End
616|    Begin Menu zzzzFile
617|       Caption         =   "&File"
618|       Begin Menu mfOpenDB
619|          Caption         =   "&Open Database..."
620|       End
621|       Begin Menu mfCloseDatabase
622|          Caption         =   "&Close Database"
623|       End
624|       Begin Menu zzzzz1
625|          Caption         =   "-"
626|       End
627|       Begin Menu mfExit
628|          Caption         =   "E&xit"
629|       End
630|    End
631|    Begin Menu zzMnuEdit
632|       Caption         =   "&Edit"
```

```
633|       Begin Menu meCopy
634|          Caption         =   "&Copy"
635|          Shortcut        =   ^{INSERT}
636|       End
637|       Begin Menu mePaste
638|          Caption         =   "&Paste"
639|          Shortcut        =   +{INSERT}
640|       End
641|    End
642| End
643| Option Explicit
644|
645| Dim DBName As String' May want to change scope of this later.
646| Dim intCoordinateField As Long
647| Dim XYBlob As String
648| Dim DataPt() As RealPoint
649|
650| Dim TabChar As String * 1, CR  As String * 1, CRLF  As String *
   2
651|
652| Dim nFields As Long, n As Long
653|
654| Dim DB As Database, tblLineData As table
655| Dim snpQTList As Snapshot, dnsCurrentQT As Dynaset
656|
657|
658| Dim SepName(0 To 20) As String, SepChar(0 To 20) As String
659| Const X_INDEX = 0
660| Const Y_INDEX = 1
661|
662| Dim jUTMZone As Long
663|
664| Const WHITE = 16777215
665|
666| Sub cmbQTList_BLOBS_Click ()
667|     Dim Temp As String
668|     Temp = cmbQTList_BLOBS
669|     Call SetQuery_BLOBS(Temp)
670|
671| End Sub
672|
673| Sub cmbQTList_Points_Click ()
674| Dim Temp As String
675| Temp = cmbQTList_Points
676| Call SetQuery_Points(Temp)
677|
678| End Sub
679|
680| Sub cmdBLOBs_Click (Value As Integer)
681|     If DBName = "" Then
682|         Call GripeNoDatabase
683|         Exit Sub
684|     End If
685|
686|
687|     Value = True
688|     pnlBLOBS.ZOrder
689|     cmdBLOBS.ZOrder
```

```
690|     zzMnuEdit.Enabled = False
691| End Sub
692|
693| Sub cmdLatLongFromUTM_BLOBS_Click ()
694| jUTMZone = txtUTMZoneNumber
695| If jUTMZone < 1 Or jUTMZone > 60 Then
696|     MsgBox "Please enter a valid UTM Zone Number.  The zone
    number must be between 1 and 60."
697|     Exit Sub
698|   End If
699|  Dim Temp As String
700| If cmbXY_UTM_BLOBS.ListIndex = -1 Then
701|     MsgBox "That field does not exist in the present
    query/table, so you can't very well convert FROM it, now can
    you?"
702|     Exit Sub
703| End If
704| Dim name_LongLatTo As String, name_UTMFrom As String
705| name_UTMFrom = cmbXY_UTM_BLOBS.Text
706|
707| If cmbXY_LongLat_BLOBS.ListIndex = -1 Then
708|     Dim strSourceTable As String
709|     strSourceTable =
    Dat_BLOBS.Recordset.Fields(name_UTMFrom).SourceTable
710|     If Trim$(cmbXY_LongLat_BLOBS.Text) = "" Then
711|         name_LongLatTo = "XYBLOB_LongitudeLatitude"
712|        Else
713|         name_LongLatTo = Trim$(cmbXY_LongLat_BLOBS.Text)
714|     End If
715|     Dim nfldBLOB_LongLat As New field
716|     nfldBLOB_LongLat.Name = name_LongLatTo:
    nfldBLOB_LongLat.Type = DB_LONGBINARY
717|
718|     Temp = Dat_BLOBS.RecordSource
719|     Dat_BLOBS.RecordSource = "": Dat_BLOBS.Refresh
720|     'Professional Edition of VB 3.0 only allows this addition of
    a field
721|     DB.TableDefs(strSourceTable).Fields.Append nfldBLOB_LongLat
722|     Dat_BLOBS.RecordSource = Temp: Dat_BLOBS.Refresh
723|    Else
724|     name_LongLatTo = cmbXY_LongLat_BLOBS.Text
725| End If
726|
727|
728| Dim nR As Long, nRecs As Long', Temp As String
729| Dat_BLOBS.Recordset.MoveFirst : Dat_BLOBS.Recordset.MoveLast :
    Dat_BLOBS.Recordset.MoveFirst
730| nRecs = Dat_BLOBS.Recordset.RecordCount
731| For nR = 1 To nRecs
732|     picPercentDone.Width = nR / nRecs * 100
733|     DoEvents
734|     Temp = Dat_BLOBS.Recordset(name_UTMFrom)
735|     Dat_BLOBS.Recordset.Edit
736|     Dat_BLOBS.Recordset(name_LongLatTo) =
    gctp_ProjectBLOBUTMnnToLL(Temp, jUTMZone)
737|     Dat_BLOBS.Recordset.Update
738|     Dat_BLOBS.Recordset.MoveNext
739| Next nR
```

```
740|
741|
742|
743|
744|
745| End Sub
746|
747| Sub cmdLatLongFromUTM_Points_Click ()
748|
749| jUTMZone = Val(txtUTMZoneNumber.Text)
750| If jUTMZone < 1 Or jUTMZone > 60 Then
751|     MsgBox "Please enter a valid UTM Zone Number.  The zone
    number must be between 1 and 60."
752|     Exit Sub
753|   End If
754|
755|
756| Dim Temp As String
757|
758| 'Make sure the fields the user wants to project FROM exist.
759| If (cmbXY_UTM_Points(X_INDEX).ListIndex = -1) Or
    (cmbXY_UTM_Points(Y_INDEX).ListIndex = -1) Then
760|     MsgBox "That field does not exist in the present
    query/table, so you can't very well convert FROM it, now can
    you?"
761|     Exit Sub
762| End If
763|
764| Dim name_XLong As String, name_UTMXFrom As String
765| Dim name_YLat  As String, name_UTMYFrom As String
766| name_UTMXFrom = cmbXY_UTM_Points(X_INDEX).Text
767| name_UTMYFrom = cmbXY_UTM_Points(Y_INDEX).Text
768|
769| Dim Temp1 As Integer, Temp2 As Integer
770| Temp1 = cmbXY_LongLat_Points(X_INDEX).ListIndex < 0
771| Temp2 = cmbXY_LongLat_Points(Y_INDEX).ListIndex < 0
772|
773| If Temp1 Xor Temp2 Then
774|      Beep
775|      MsgBox "Either select two NEW fields to put data into, or
    two OLD fields to put data into, but not one old and one new."
776|      Exit Sub
777| End If
778|
779| If cmbXY_LongLat_Points(X_INDEX).ListIndex = -1 Then
780|     Dim strSourceTable As String
781|     'Put the output coordinates in the same table as the input
    coordinates
782|     strSourceTable =
    Dat_Points.Recordset.Fields(name_UTMXFrom).SourceTable
783|     If Trim$(cmbXY_LongLat_Points(X_INDEX).Text) = "" Then
784|         name_XLong = "X_Longitude"
785|        Else
786|         name_XLong = Trim$(cmbXY_LongLat_Points(X_INDEX).Text)
787|     End If
788|     If Trim$(cmbXY_LongLat_Points(Y_INDEX).Text) = "" Then
789|         name_YLat = "Y_Latitude"
790|        Else
```

```
791|          name_YLat = Trim$(cmbXY_LongLat_Points(Y_INDEX).Text)
792|      End If
793|
794|
795|      Dim nfldX_Long As New field, nfldY_Lat As New field
796|      nfldX_Long.Name = name_XLong: nfldX_Long.Type = DB_SINGLE
797|      nfldY_Lat.Name = name_YLat: nfldY_Lat.Type = DB_SINGLE
798|
799|      Temp = Dat_Points.RecordSource
800|      Dat_Points.RecordSource = "": Dat_Points.Refresh
801|      'Professional Edition of VB 3.0 only allows this addition of
    a field
802|      DB.TableDefs(strSourceTable).Fields.Append nfldX_Long
803|      DB.TableDefs(strSourceTable).Fields.Append nfldY_Lat
804|      Dat_Points.RecordSource = Temp: Dat_Points.Refresh
805|     Else
806|      name_XLong = cmbXY_LongLat_Points(X_INDEX).Text
807|      name_YLat = cmbXY_LongLat_Points(Y_INDEX).Text
808| End If
809|
810|
811| Dim nR As Long, nRecs As Long', Temp As String
812| Dat_Points.Recordset.MoveFirst : Dat_Points.Recordset.MoveLast :
    Dat_Points.Recordset.MoveFirst
813| nRecs = Dat_Points.Recordset.RecordCount
814| Dim dblLLOut As typDblePoint, DblUTMIn As typDblePoint
815| For nR = 1 To nRecs
816|      picPercentDone_Points.Width = nR / nRecs * 100
817|      DoEvents
818|
819|      DblUTMIn.X = Dat_Points.Recordset(name_UTMXFrom)
820|      DblUTMIn.Y = Dat_Points.Recordset(name_UTMYFrom)
821|      Call gctp_DConvertUTMToDecDeg(DblUTMIn, dblLLOut, jUTMZone)
822|      Dat_Points.Recordset.Edit
823|      Dat_Points.Recordset(name_XLong) = dblLLOut.X
824|      Dat_Points.Recordset(name_YLat) = dblLLOut.Y
825|      Dat_Points.Recordset.Update
826|      Dat_Points.Recordset.MoveNext
827| Next nR
828|
829|
830|
831|
832|
833| End Sub
834|
835| Sub cmdLongLatFromUTM_Manual_Click ()
836| jUTMZone = Val(txtUTMZoneNumber.Text)
837| If jUTMZone < 1 Or jUTMZone > 60 Then
838|      MsgBox "Please enter a valid UTM Zone Number.  The zone
    number must be between 1 and 60."
839|      Exit Sub
840|    End If
841|
842|
843|         Dim dblIn As typDblePoint, dblOut As typDblePoint
844|         dblIn.X = Val(txtUTMEasting.Text)
845|         dblIn.Y = Val(txtUTMNorthing.Text)
```

```
846|       Call gctp_DConvertUTMToDecDeg(dblIn, dblOut, jUTMZone)
847|         txtLongDecDeg = Format$(dblOut.X, "#.000000")
848|         txtLatDecDeg = Format$(dblOut.Y, "#.000000")
849|         txtLongDMS = DecDegToDMS(dblOut.X)
850|         txtLatDMS = DecDegToDMS(dblOut.Y)
851|
852| End Sub
853|
854| Sub cmdMANUAL_Click (Value As Integer)
855|     Value = True
856|     pnlManualEntry.ZOrder
857|     cmdManual.ZOrder
858|     zzMnuEdit.Enabled = True
859| End Sub
860|
861| Sub cmdPoints_Click (Value As Integer)
862|     If DBName = "" Then
863|         Call GripeNoDatabase
864|         Exit Sub
865|     End If
866|
867|
868|    Value = True
869|    zzMnuEdit.Enabled = False
870|    pnlSinglePoints.ZOrder
871|    cmdPoints.ZOrder
872| End Sub
873|
874| Sub cmdUTMFromLongLat_BLOBS_Click ()
875|
876| jUTMZone = Val(txtUTMZoneNumber.Text)
877| If jUTMZone < 1 Or jUTMZone > 60 Then
878|     MsgBox "Please enter a valid UTM Zone Number.  The zone
    number must be between 1 and 60."
879|     Exit Sub
880|   End If
881| If cmbXY_LongLat_BLOBS.ListIndex = -1 Then
882|     MsgBox "That field does not exist in the present
    query/table, so you can't very well convert FROM it, now can
    you?"
883|     Exit Sub
884|   End If
885|
886|
887| Dim name_BLOB_LongLat As String, name_BLOB_UTM As String
888| name_BLOB_LongLat = cmbXY_LongLat_BLOBS.Text
889|
890| If cmbXY_UTM_BLOBS.ListIndex = -1 Then
891|     Dim strSourceTable As String
892|     strSourceTable =
    Dat_BLOBS.Recordset.Fields(name_BLOB_LongLat).SourceTable
893|     If Trim$(cmbXY_UTM_BLOBS.Text) = "" Then
894|         name_BLOB_UTM = "XYname_BLOB_UTM" & Format$(jUTMZone,
    "00")
895|        Else
896|         name_BLOB_UTM = Trim$(cmbXY_UTM_BLOBS.Text)
897|     End If
898|     Dim nfldBLOB_UTM As New field
```

```
899|     nfldBLOB_UTM.Name = name_BLOB_UTM: nfldBLOB_UTM.Type =
   DB_LONGBINARY
900|     Dim Temp As String
901|     Temp = Dat_BLOBS.RecordSource
902|     Dat_BLOBS.RecordSource = "": Dat_BLOBS.Refresh
903|     'Professional Edition of VB 3.0 only allows this addition of
   a field
904|     DB.TableDefs(strSourceTable).Fields.Append nfldBLOB_UTM
905|     Dat_BLOBS.RecordSource = Temp: Dat_BLOBS.Refresh
906|    Else
907|     name_BLOB_UTM = cmbXY_UTM_BLOBS.Text
908| End If
909|
910|
911|
912| 'CopyXYFromBLOBToMemo name_BLOB_LongLat, name_BLOB_UTM, XYDelim,
   PairDelim, Me.Dat_BLOBS.Recordset
913| Dim nR As Long, nRecs As Long', Temp As String
914| Dat_BLOBS.Recordset.MoveFirst : Dat_BLOBS.Recordset.MoveLast :
   Dat_BLOBS.Recordset.MoveFirst
915| nRecs = Dat_BLOBS.Recordset.RecordCount
916| For nR = 1 To nRecs
917|
918|     picPercentDone.Width = nR / nRecs * 100
919|     DoEvents
920|     Temp = Dat_BLOBS.Recordset(name_BLOB_LongLat)
921|     Dat_BLOBS.Recordset.Edit
922|     Dat_BLOBS.Recordset(name_BLOB_UTM) =
   gctp_ProjectBLOBLLToUTMnn(Temp, jUTMZone)
923|     Dat_BLOBS.Recordset.Update
924|     Dat_BLOBS.Recordset.MoveNext
925| Next nR
926|
927| End Sub
928|
929| Sub cmdUTMFromLongLat_Manual_Click ()
930| jUTMZone = Val(txtUTMZoneNumber.Text)
931| If jUTMZone < 1 Or jUTMZone > 60 Then jUTMZone = 0
932|
933|
934|
935|
936|
937|        Dim dblIn As typDblePoint, dblOut As typDblePoint
938|        dblIn.X = Val(txtLongDecDeg.Text)
939|
940|        dblIn.Y = Val(txtLatDecDeg.Text)
941|      Call gctp_DConvertDecDegToUTM(dblIn, dblOut, jUTMZone)
942|        txtUTMEasting = Format$(dblOut.X, "0")
943|        txtUTMNorthing = Format$(dblOut.Y, "0")
944|        txtUTMZoneNumber = jUTMZone
945| End Sub
946|
947| Sub cmdUTMFromLongLat_Points_Click ()
948| jUTMZone = txtUTMZoneNumber
949| If jUTMZone < 1 Or jUTMZone > 60 Then
950|     MsgBox "Please enter a valid UTM Zone Number.  The zone
   number must be between 1 and 60."
```

```
    Exit Sub
  End If



Dim Temp As String

'Make sure the fields the user wants to project FROM exist.
If (cmbXY_LongLat_Points(X_INDEX).ListIndex = -1) Or
(cmbXY_LongLat_Points(Y_INDEX).ListIndex = -1) Then
    MsgBox "That field does not exist in the present
query/table, so you can't very well convert FROM it, now can
you?"
    Exit Sub
End If

Dim name_UTMX_Out As String, name_XLongFrom As String
Dim name_UTMY_Out  As String, name_YLatFrom As String
name_XLongFrom = cmbXY_LongLat_Points(X_INDEX).Text
name_YLatFrom = cmbXY_LongLat_Points(Y_INDEX).Text

Dim Temp1 As Integer, Temp2 As Integer
Temp1 = cmbXY_UTM_Points(X_INDEX).ListIndex < 0
Temp2 = cmbXY_UTM_Points(Y_INDEX).ListIndex < 0

If Temp1 Xor Temp2 Then
     Beep
     MsgBox "Either select two NEW fields to put data into, or
two OLD fields to put data into, but not one old and one new."
     Exit Sub
End If

If cmbXY_UTM_Points(X_INDEX).ListIndex = -1 Then
    Dim strSourceTable As String
    'Put the output coordinates in the same table as the input
coordinates
    strSourceTable =
Dat_Points.Recordset.Fields(name_XLongFrom).SourceTable
    If Trim$(cmbXY_UTM_Points(X_INDEX).Text) = "" Then
        name_UTMX_Out = "Easting_UTM " & Format$(jUTMZone, "00")
       Else
        name_UTMX_Out = Trim$(cmbXY_UTM_Points(X_INDEX).Text)
    End If
    If Trim$(cmbXY_UTM_Points(Y_INDEX).Text) = "" Then
        name_UTMY_Out = "Northing_UTM" & Format$(jUTMZone, "00")
       Else
        name_UTMY_Out = Trim$(cmbXY_UTM_Points(Y_INDEX).Text)
    End If


    Dim nfldX_UTM As New field, nfldY_UTM As New field
    nfldX_UTM.Name = name_UTMX_Out: nfldX_UTM.Type = DB_SINGLE
    nfldY_UTM.Name = name_UTMY_Out: nfldY_UTM.Type = DB_SINGLE

    Temp = Dat_Points.RecordSource
    Dat_Points.RecordSource = "": Dat_Points.Refresh
    'Professional Edition of VB 3.0 only allows this addition of
a field
```

```
1002|      DB.TableDefs(strSourceTable).Fields.Append nfldX_UTM
1003|      DB.TableDefs(strSourceTable).Fields.Append nfldY_UTM
1004|      Dat_Points.RecordSource = Temp: Dat_Points.Refresh
1005|     Else
1006|      name_UTMX_Out = cmbXY_UTM_Points(X_INDEX).Text
1007|      name_UTMY_Out = cmbXY_UTM_Points(Y_INDEX).Text
1008| End If
1009|
1010|
1011| Dim nR As Long, nRecs As Long', Temp As String
1012| Dat_Points.Recordset.MoveFirst : Dat_Points.Recordset.MoveLast :
    Dat_Points.Recordset.MoveFirst
1013| nRecs = Dat_Points.Recordset.RecordCount
1014| Dim dblLLIn As typDblePoint, DblUTMOut As typDblePoint
1015| For nR = 1 To nRecs
1016|     picPercentDone_Points.Width = nR / nRecs * 100
1017|     DoEvents
1018|
1019|     dblLLIn.X = Dat_Points.Recordset(name_XLongFrom)
1020|     dblLLIn.Y = Dat_Points.Recordset(name_YLatFrom)
1021|     Call gctp_DConvertDecDegToUTM(dblLLIn, DblUTMOut, jUTMZone)
1022|     Dat_Points.Recordset.Edit
1023|     Dat_Points.Recordset(name_UTMX_Out) = DblUTMOut.X
1024|     Dat_Points.Recordset(name_UTMY_Out) = DblUTMOut.Y
1025|     Dat_Points.Recordset.Update
1026|     Dat_Points.Recordset.MoveNext
1027| Next nR
1028|
1029| End Sub
1030|
1031| Sub Form_Load ()
1032|         CR = Chr$(13)
1033|         TabChar = Chr$(9)
1034|         CRLF = Chr$(13) & Chr$(10)
1035|         ReDim DataPts(1 To 1)
1036|
1037| 'To allow the user to specify the UTM zone on the command line,
1038|  ' following bit is included
1039|   Dim Temp As String, Temp2 As String, StartAt  As Long
1040|   Temp2 = Trim$(UCase$(Command$))
1041|   StartAt = InStr(Temp2, "ZONE")
1042|   If StartAt = 0 Then
1043|     txtUTMZoneNumber = "14"
1044|    Else
1045|     Temp2 = Trim$(Mid$(Temp2, StartAt + 5))
1046|     txtUTMZoneNumber = Temp2
1047|   End If
1048|   jUTMZone = txtUTMZoneNumber
1049| End Sub
1050|
1051| Sub GripeNoDatabase ()
1052|     Call cmdMANUAL_Click(0)
1053|     cmdManual = True
1054|     Beep
1055|     MsgBox "You cannot use this program to project coordinates
    stored in a database until you have opened a database."
1056|     SendKeys "%F"
1057|     SendKeys "o"
```

```
1058| End Sub
1059|
1060| Sub ListQrysTblsInComboBox (DBIn As Database, ComboOut As
    ComboBox)
1061| ' this subroutine lists all queries and tables in a database
1062| '   in a combobox, for later user selection.
1063|     ComboOut.Clear
1064|     Dim DB_SYSTEMOBJECT As Long
1065|     DB_SYSTEMOBJECT = &H80000002
1066|
1067|     Dim snpQTList As Snapshot 'declared locally so it goes away
1068|     Set snpQTList = DBIn.ListTables()
1069|
1070|     Dim QTName As String
1071|     snpQTList.MoveFirst
1072|     Do Until snpQTList.EOF 'For each Table and query
1073|         If Not ((snpQTList("Attributes") And DB_SYSTEMOBJECT) <>
    0) And (snpQTList("Attributes") < 6) Then
1074|           'Exclude system objects from list of tables and
    queries presented to user
1075|           QTName = snpQTList("Name")
1076|           ComboOut.AddItem QTName
1077|         End If
1078|         snpQTList.MoveNext
1079|     Loop
1080|     snpQTList.Close
1081|     Set snpQTList = Nothing
1082| End Sub
1083|
1084| Sub meCopy_Click ()
1085| clipboard.Clear
1086| Dim strClipOut As String
1087|
1088| If (activecontrol Is txtLongDecDeg) Or (activecontrol Is
    txtLatDecDeg) Then
1089|    strClipOut = txtLongDecDeg & TabChar & txtLatDecDeg & CRLF
1090| End If
1091|
1092| If (activecontrol Is txtLongDMS) Or (activecontrol Is txtLatDMS)
    Then
1093|    strClipOut = txtLongDMS & TabChar & txtLatDMS & CRLF
1094| End If
1095|
1096| If (activecontrol Is txtUTMEasting) Or (activecontrol Is
    txtUTMNorthing) Then
1097|    strClipOut = txtUTMEasting & TabChar & txtUTMNorthing & CRLF
1098| End If
1099| clipboard.SetText strClipOut
1100|
1101|
1102|
1103| End Sub
1104|
1105| Sub mePaste_Click ()
1106|    If TypeOf activecontrol Is TextBox Then
1107|      'do nothing
1108|     Else
1109|       Exit Sub
```

```
1110|    End If
1111|
1112| Dim strClipIn As String, str1 As String, str2 As String, T1 As
     Long, T2 As Long
1113|    strClipIn = clipboard.GetText()
1114|    T1 = InStr(strClipIn, TabChar)
1115|    If T1 = 0 Then
1116|         activecontrol = strClipIn
1117|         Exit Sub
1118|    End If
1119|    str1 = Left$(strClipIn, T1 - 1)
1120|    T2 = InStr(strClipIn, CRLF)
1121|    If T2 = 0 Then T2 = InStr(strClipIn, Chr$(10))
1122|    If T2 = 0 Then T2 = InStr(strClipIn, Chr$(13))
1123|    If T2 = 0 Then T2 = Len(strClipIn) + 1
1124|    str2 = Mid$(strClipIn, T1 + 1, T2 - T1 - 1)
1125|
1126| If (activecontrol Is txtLongDecDeg) Or (activecontrol Is
     txtLatDecDeg) Then
1127|     txtLongDecDeg = str1
1128|     txtLatDecDeg = str2
1129| End If
1130|
1131| If (activecontrol Is txtLongDMS) Or (activecontrol Is txtLatDMS)
     Then
1132|     txtLongDMS = str1
1133|     txtLatDMS = str2
1134| End If
1135|
1136| If (activecontrol Is txtUTMEasting) Or (activecontrol Is
     txtUTMNorthing) Then
1137|    txtUTMEasting = str1
1138|    txtUTMNorthing = str2
1139| End If
1140|
1141|
1142| End Sub
1143|
1144| Sub mfCloseDatabase_Click ()
1145|  On Error Resume Next
1146|  DB.Close
1147|   DBName = ""
1148|   Me.Caption = "LLUTMnn_  UTM<=>Longitude-Latitude Projection "
     & "NO DATABASE OPEN"
1149|  Set DB = Nothing
1150|   Dat_Points.Recordset.Close
1151|   Dat_BLOBS.Recordset.Close
1152|   cmbQTList_BLOBS.Clear
1153|   cmbQTList_Points.Clear
1154|   cmbXY_LongLat_BLOBS.Clear
1155|   cmbXY_LongLat_Points(X_INDEX).Clear :
     cmbXY_LongLat_Points(Y_INDEX).Clear
1156|   cmbXY_UTM_BLOBS.Clear
1157|   cmbXY_UTM_Points(X_INDEX).Clear :
     cmbXY_UTM_Points(Y_INDEX).Clear
1158|
1159|
1160|
```

```
1161| End Sub
1162|
1163| Sub mfExit_Click ()
1164|  End
1165| End Sub
1166|
1167| Sub mfOpenDB_Click ()
1168| On Error GoTo ERRmfOpenDB
1169| CMDialog1.DialogTitle = "Database to Open"
1170| CMDialog1.Filter = "MS Access Databases|*.mdb"
1171| CMDialog1.Flags = OFN_FILEMUSTEXIST
1172| CMDialog1.Action = 1
1173| CMDialog1.CancelError = True
1174| Dim Temp As String
1175|
1176| Temp = CMDialog1.Filename
1177| If Len(Temp) = 0 Then 'the user did not enter a file name
1178|     Else
1179|      DBName = Temp
1180|      '  Open an MSAccess database for non-exclusive, read-write
     access.
1181|      Set DB = OpenDatabase(DBName, False, False, "") 'Open the
     desired database
1182|      Dat_Points.DatabaseName = DBName
1183|      Me.Caption = "LLUTMnn_  UTM<=>Longitude-Latitude Projection
     " & DBName
1184|      Dat_BLOBS.DatabaseName = DBName
1185|      Call ListQrysTblsInComboBox(DB, cmbQTList_BLOBS)
1186|      Call ListQrysTblsInComboBox(DB, cmbQTList_Points)
1187|      If cmdPoints.Value = True Then
1188|          cmbQTList_Points.SetFocus
1189|          SendKeys "%{DOWN}"
1190|      End If
1191|      If cmdBLOBS.Value = True Then
1192|          cmbQTList_BLOBS.SetFocus
1193|          SendKeys "%{DOWN}"
1194|     End If
1195|
1196| End If
1197| Exit Sub
1198| ERRmfOpenDB:
1199|  Select Case Err
1200|    Case Else
1201|    Exit Sub
1202|    Resume
1203|  End Select
1204| End Sub
1205|
1206| Sub SetQuery_BLOBS (strTblQrySQL As String)
1207| 'This subroutine sets the data control to the desired TaBLe,
     QueRY, or SQL string,
1208| ' then sets the labels for the attribute grid.
1209| On Error GoTo ERRSetQuery_BLOBS
1210| Set dnsCurrentQT = DB.CreateDynaset(strTblQrySQL)
1211| nFields = dnsCurrentQT.Fields.Count
1212| dnsCurrentQT.Close
1213| Set dnsCurrentQT = Nothing
1214| Dat_BLOBS.RecordSource = strTblQrySQL
```

```
1215|          'nFields = Dat_BLOBS.Recordset.Fields.Count
1216| Dat_BLOBS.Refresh
1217| intCoordinateField = 0
1218| Dim FieldName As String
1219| nFields = Dat_BLOBS.Recordset.Fields.Count
1220|
1221| cmbXY_LongLat_BLOBS.Clear : cmbXY_UTM_BLOBS.Clear
1222| For n = 0 To nFields - 1
1223|     If (Dat_BLOBS.Recordset.Fields(n).Type = DB_LONGBINARY) And
    (InStr(UCase(Dat_BLOBS.Recordset.Fields(n).Name), "LAT") <> 0)
    Then  ' OLE field with Longitude Latitude coordinates
1224|                     '
1225|         cmbXY_LongLat_BLOBS.AddItem
    Dat_BLOBS.Recordset.Fields(n).Name
1226|     End If
1227|     If (Dat_BLOBS.Recordset.Fields(n).Type = DB_LONGBINARY) And
    (InStr(UCase(Dat_BLOBS.Recordset.Fields(n).Name), "UTM") <> 0)
    Then '
1228|                     '
1229|         cmbXY_UTM_BLOBS.AddItem
    Dat_BLOBS.Recordset.Fields(n).Name
1230|     End If
1231|
1232| Next n
1233| If cmbXY_UTM_BLOBS.ListCount = 0 And
    cmbXY_LongLat_BLOBS.ListCount = 0 Then
1234|   Beep
1235|   MsgBox "That query or table does not contain a BLOB field with
    a name this program recognizes.  Please choose another
    table/query or modify the field names to include UTM or LatLong
    as appropriate."
1236|   Exit Sub
1237| End If
1238|
1239| If cmbXY_LongLat_BLOBS.ListCount > 0 Then
1240|     cmbXY_LongLat_BLOBS.ListIndex = 0
1241|    Else
1242|     cmbXY_LongLat_BLOBS.Text = "XYBLOB_LongitudeLatitude"
1243| End If
1244|
1245| If cmbXY_UTM_BLOBS.ListCount > 0 Then
1246|     cmbXY_UTM_BLOBS.ListIndex = 0
1247|    Else
1248|     cmbXY_UTM_BLOBS.Text = "XYBLOB_UTM" & Format$(jUTMZone,
    "00")
1249| End If
1250|
1251| Exit Sub
1252|
1253| ERRSetQuery_BLOBS:
1254| If Err = 3061 Then
1255|     MsgBox "You tried selecting a parameter query, and this
    application doesn't do those.  Sorry!"
1256|     Err = 0
1257|     Exit Sub
1258| End If
1259|
1260| MsgBox "Error number " & Err & " occurred in SetQuery_BLOBS." &
```

```
      CRLF & Error
1261| Err = 0
1262| Exit Sub
1263| Resume
1264| 'Dat_BLOBS.Recordset.MoveFirst
1265| 'Eliminated the above line so empty tables wouldn't
1266| '  cause it to freak out.
1267| End Sub
1268|
1269| Sub SetQuery_Points (strTblQrySQL As String)
1270| 'This subroutine sets the data control to the desired TaBLe,
      QueRY, or SQL string,
1271| ' then sets the labels for the attribute grid.
1272| On Error GoTo ERRSetQuery
1273| Set dnsCurrentQT = DB.CreateDynaset(strTblQrySQL)
1274| nFields = dnsCurrentQT.Fields.Count
1275| dnsCurrentQT.Close
1276| Set dnsCurrentQT = Nothing
1277| Dat_Points.RecordSource = strTblQrySQL
1278|         'nFields = Dat_Points.Recordset.Fields.Count
1279| Dat_Points.Refresh
1280| intCoordinateField = 0
1281| Dim FieldName As String, uFieldName As String
1282| nFields = Dat_Points.Recordset.Fields.Count
1283| Dim intHasX As Integer, intHasY As Integer, intHasUTM As
      Integer, intHasLL As Integer
1284| cmbXY_LongLat_Points(X_INDEX).Clear :
      cmbXY_LongLat_Points(Y_INDEX).Clear
1285| cmbXY_UTM_Points(X_INDEX).Clear :
      cmbXY_UTM_Points(Y_INDEX).Clear
1286|
1287| For n = 0 To nFields - 1
1288|     If ((Dat_Points.Recordset.Fields(n).Type = DB_SINGLE) Or
      (Dat_Points.Recordset.Fields(n).Type = DB_DOUBLE)) Then    '
      field might contain coordinates
1289|       intHasX = False: intHasY = False: intHasUTM = False:
      intHasLL = False
1290|       FieldName = Dat_Points.Recordset.Fields(n).Name:
      uFieldName = UCase$(FieldName)
1291|
1292|       intHasX = intHasX Or InStr(uFieldName, "X_") <> 0
1293|       intHasX = intHasX Or InStr(uFieldName, "EAST") <> 0
1294|
1295|       intHasY = intHasY Or InStr(uFieldName, "Y_") <> 0
1296|       intHasY = intHasY Or InStr(uFieldName, "NORT") <> 0
1297|
1298|
1299|       intHasUTM = InStr(uFieldName, "UTM") <> 0
1300|
1301|       intHasLL = intHasLL Or InStr(uFieldName, "ITUDE") <> 0
1302|       intHasLL = intHasLL Or InStr(uFieldName, "LL") <> 0
1303|       intHasLL = intHasLL Or InStr(uFieldName, "GEO") <> 0
1304|       intHasLL = intHasLL Or InStr(uFieldName, "LON") <> 0
1305|       intHasLL = intHasLL Or InStr(uFieldName, "LAT") <> 0
1306|
1307|       If (intHasX And intHasUTM) Then
      cmbXY_UTM_Points(X_INDEX).AddItem FieldName
1308|       If (intHasY And intHasUTM) Then
```

```
      cmbXY_UTM_Points(Y_INDEX).AddItem FieldName
1309|
1310|       If (intHasX And intHasLL) Then
      cmbXY_LongLat_Points(X_INDEX).AddItem FieldName
1311|       If (intHasY And intHasLL) Then
      cmbXY_LongLat_Points(Y_INDEX).AddItem FieldName
1312|
1313|     End If
1314|
1315| Next n
1316|
1317|
1318| If cmbXY_UTM_Points(X_INDEX).ListCount = 0 And
      cmbXY_LongLat_Points(X_INDEX).ListCount = 0 Then
1319|   Beep
1320|   MsgBox "That query or table does not contain any suitable
      coordinate fields.  Please choose another table/query or modify
      the field names to include UTM or LatLong as appropriate."
1321|   Exit Sub
1322| End If
1323|
1324| If cmbXY_LongLat_Points(X_INDEX).ListCount > 0 Then
1325|     cmbXY_LongLat_Points(X_INDEX).ListIndex = 0
1326|    Else
1327|     cmbXY_LongLat_Points(X_INDEX).Text = "X_Longitude"
1328| End If
1329| If cmbXY_LongLat_Points(Y_INDEX).ListCount > 0 Then
1330|     cmbXY_LongLat_Points(Y_INDEX).ListIndex = 0
1331|    Else
1332|     cmbXY_LongLat_Points(Y_INDEX).Text = "Y_Latitude"
1333| End If
1334|
1335|
1336| If cmbXY_UTM_Points(X_INDEX).ListCount > 0 Then
1337|     cmbXY_UTM_Points(X_INDEX).ListIndex = 0
1338|    Else
1339|     cmbXY_UTM_Points(X_INDEX).Text = "X_Easting_UTM" &
      Format$(jUTMZone, "00")
1340| End If
1341| If cmbXY_UTM_Points(Y_INDEX).ListCount > 0 Then
1342|     cmbXY_UTM_Points(Y_INDEX).ListIndex = 0
1343|    Else
1344|     cmbXY_UTM_Points(Y_INDEX).Text = "Y_Northing_UTM" &
      Format$(jUTMZone, "00")
1345| End If
1346|
1347|
1348| Exit Sub
1349|
1350| ERRSetQuery:
1351| If Err = 3061 Then
1352|     MsgBox "You tried selecting a parameter query, and this
      application doesn't do those.  Sorry!"
1353|     Err = 0
1354|     Exit Sub
1355| End If
1356|
1357| MsgBox "Error number " & Err & " occurred in SetQuery." & CRLF &
```

```
      Error
 1358| Err = 0
 1359| Exit Sub
 1360| Resume
 1361| 'Dat_Points.Recordset.MoveFirst
 1362| 'Eliminated the above line so empty tables wouldn't
 1363| '  cause it to freak out.
 1364| End Sub
 1365|
 1366| Sub txtJunk_GotFocus ()
 1367|    cmdUTMFromLongLat_Manual.SetFocus
 1368| End Sub
 1369|
 1370| Sub txtLatDecDeg_KeyPress (KeyAscii As Integer)
 1371| Select Case KeyAscii
 1372|     Case 13 'Return
 1373|         'Get the previous version in decimal degrees, and use it
      to update the DMS version
 1374|         txtLatDMS = DecDegToDMS(Val(txtLatDecDeg.Text))
 1375|         KeyAscii = 0
 1376|     Case 27 'Escape
 1377|        'Convert to decimal degrees
 1378|         txtLatDecDeg = Format(DMSToDecDeg(txtLatDMS.Text))
 1379|         KeyAscii = 0
 1380|     Case Else
 1381| End Select
 1382| End Sub
 1383|
 1384| Sub txtLatDMS_KeyPress (KeyAscii As Integer)
 1385| Select Case KeyAscii
 1386|     Case 13 'Return
 1387|        'Convert to decimal degrees
 1388|         txtLatDecDeg = Format(DMSToDecDeg(txtLatDMS.Text))
 1389|         KeyAscii = 0
 1390|     Case 27 'Escape
 1391|         'Get the previous version in decimal degrees, and use it
      to
 1392|         ' update the DMS version
 1393|         txtLatDMS = DecDegToDMS(Val(txtLatDecDeg.Text))
 1394|         KeyAscii = 0
 1395|     Case Else
 1396| End Select
 1397|
 1398| End Sub
 1399|
 1400| Sub txtLongDecDeg_KeyPress (KeyAscii As Integer)
 1401| Select Case KeyAscii
 1402|     Case 13 'Return
 1403|         'Get the previous version in decimal degrees, and use it
      to
 1404|         ' update the DMS version
 1405|         txtLongDMS = DecDegToDMS(Val(txtLongDecDeg.Text))
 1406|         KeyAscii = 0
 1407|     Case 27 'Escape
 1408|        'Convert to decimal degrees
 1409|         txtLongDecDeg = Format(DMSToDecDeg(txtLongDMS.Text))
 1410|         KeyAscii = 0
 1411|     Case Else
```

```
1412| End Select
1413|
1414| End Sub
1415|
1416| Sub txtLongDMS_KeyPress (KeyAscii As Integer)
1417| Select Case KeyAscii
1418|     Case 13 'Return
1419|         'Convert to decimal degrees
1420|          txtLongDecDeg = Format(DMSToDecDeg(txtLongDMS.Text))
1421|          KeyAscii = 0
1422|          'txtLongDMS.backcolor = WHITE
1423|     Case 27 'Escape
1424|          'Get the previous version in decimal degrees, and use it
    to
1425|          ' update the DMS version
1426|          txtLongDMS = DecDegToDMS(Val(txtLongDecDeg.Text))
1427|          KeyAscii = 0
1428|          'txtLongDecDeg.backcolor = WHITE
1429|     Case Else
1430| End Select
1431| End Sub
1432|
```

```
 1| Option Explicit
 2| Type typDblePoint
 3|    X As Double
 4|    Y As Double
 5| End Type
 6| Type typRealPoint
 7|    X As Single
 8|    Y As Single
 9| End Type
10|
11|
12|
13|
14| Global Const OneDegreeInRadians = .01745329251994
15| Global Const OneRadianInDegrees = 57.29577951308
16|
17|
18| 'GCTP Main routine, calls other routines as appropriate
19| Declare Sub GTRNZ0 Lib "GCTP0001.DLL" (DblCoordIn As
    typDblePoint, JCoordSysIn As Long, SetupCoordSysIn As Double,
    DblCoordOut As typDblePoint, JCoordSysOut As Long,
    SetupCoordSysOut As Double, JZero As Long, jFlag As Long)
20|
21| 'Initialization routine for UTM
22| Declare Sub IS01Z0 Lib "GCTP0001.DLL" (jZone As Long,
    SetupCoordSys As Double, JZero As Long, jFlag As Long)
23| 'Forward UTM transformation            Lat Long Radians     ->
    UTM Meters
24| Declare Sub PF01Z0 Lib "GCTP0001.DLL" (DblCoordIn As
    typDblePoint, DblCoordOut As typDblePoint, jFlag As Long)
25| 'Inverse UTM                          UTM Meters           ->
    Lat Long Radians
26| Declare Sub PI01Z0 Lib "GCTP0001.DLL" (DblCoordIn As
    typDblePoint, DblCoordOut As typDblePoint, jFlag As Long)
27|
28|
29| Global Const J0Zero = 0&
30| Dim CoordSysIn(1 To 13) As Double, CoordSysOut(1 To 13) As
    Double
31| Dim flgInitialized As Long, j0 As Long, jFlag As Long
32| Global dblPoint1 As typDblePoint, dblPoint2 As typDblePoint
33|
34|
35|
36|   'Declare Function ClipLineToPolygon Lib "WhtGeom1.DLL" (P1 As
    RealPoint, P2 As RealPoint, nPts As Long, Poly As RealPoint,
    Crossings As RealPoint) As Long
37|
38| Function DecDegToDMS (ByVal DecDegIn As Single) As String
39|      Dim strDeg As String, strMin As String, strSec As String
40|      Dim Temp1 As Single, Temp2 As Single
41|      strDeg = "": strMin = "": strSec = ""
42|      Temp1 = Fix(DecDegIn)
43|      strDeg = Format(Temp1)
44|      Temp1 = Abs(DecDegIn - Temp1)
45|
46|
47|      Temp1 = 60 * Temp1
```

```
   48|      Temp2 = Fix(Temp1)
   49|      strMin = Format$(Temp2, "00")
   50|
   51|      Temp2 = 60 * (Temp1 - Temp2)
   52|      strSec = Format$(Temp2, "00.#")
   53|
   54|      DecDegToDMS = strDeg & " " & strMin & " " & strSec
   55| End Function
   56|
   57| Function DMSToDecDeg (ByVal DMSIn As String) As Single
   58|    Dim a As Single, b As Single, c As Single
   59|     Dim str1 As String, str2 As String, str3 As String
   60|     Dim Loc1 As Integer, Loc2 As Integer, Loc3 As Integer
   61|     Dim Sign As Integer
   62|     DMSIn = Trim$(DMSIn)
   63|     Loc1 = InStr(DMSIn, " ")
   64|     If Loc1 = 0 Then  'If there are no spaces, just degrees. So
       skip out now
   65|         DMSToDecDeg = Val(DMSIn)
   66|         Exit Function
   67|     End If
   68|
   69|     str1 = Trim$(Mid$(DMSIn, 1, Loc1))'Get the degrees
   70|     a = Val(str1)
   71|     Sign = Sgn(a)
   72|
   73|     str2 = Trim$(Mid$(DMSIn, Loc1))
   74|     Loc2 = InStr(str2, " ")
   75|     If Loc2 = 0 Then   'Just degrees and minutes, so skip out
       now.
   76|        b = Val(str2) / 60#
   77|        DMSToDecDeg = a + Sign * b
   78|        Exit Function
   79|     End If
   80|      str3 = Trim$(Mid$(str2, Loc2))
   81|      str2 = Left$(str2, Loc2)
   82|
   83|     b = Val(str2) / 60#
   84|
   85|     c = Val(str3) / 3600#
   86|     DMSToDecDeg = a + Sign * b + Sign * c
   87| End Function
   88|
   89| Sub gctp_COMMENTS ()
   90|  'This module contains definitions and helper routines for
   91|  ' use with the GCTP Map Projection package.
   92|  ' Source code for GCTP comes from USGS and NOAA
   93|  '  See NOAA Technical Report NOS 124 CGS9
   94|  ' General Cartographic Transformation Package (GCTP), Version
       II
   95|  '  by Atef A. Elassal
   96| 'Slightly modified by Greg Pouch at KGS 9/27/94 and 11/1/94
   97| '   to get the SPCZ routines to read properly.
   98| ' Compiled and linked as a Windows DLL by Greg Pouch 95 03 02
   99| ' Calling routines and declare statements  95 03 02
  100|
  101|
  102| End Sub
```

```
103|
104| Sub gctp_DConvertDecDegToUTM (DblLatLongIn As typDblePoint,
   DblUTMOut As typDblePoint, jZone As Long)
105| '' Takes Double Points and converts from Lat Long in Decimal
   Degrees to UTM Meters
106|     Dim dblTemp As typDblePoint
107|     j0 = J0Zero
108|     dblTemp.X = DblLatLongIn.X * OneDegreeInRadians
109|     dblTemp.Y = DblLatLongIn.Y * OneDegreeInRadians
110|     Call IS01Z0(jZone, CoordSysIn(1), j0, jFlag)
111|     Call PF01Z0(dblTemp, DblUTMOut, jFlag)
112| End Sub
113|
114| Sub gctp_DConvertUTMToDecDeg (DblUTMIn As typDblePoint,
   DblLatLongOut As typDblePoint, jZone As Long)
115| ''Takes Double Points to convert from UTM to Lat Long.
116|     j0 = J0Zero
117|     Call IS01Z0(jZone, CoordSysIn(1), j0, jFlag)
118|     Call PI01Z0(DblUTMIn, DblLatLongOut, jFlag)
119|     DblLatLongOut.X = DblLatLongOut.X * OneRadianInDegrees
120|     DblLatLongOut.Y = DblLatLongOut.Y * OneRadianInDegrees
121| End Sub
122|
123| Function gctp_ProjectBLOBLLToUTM14 (XYBLOBIn As String) As
   String
124|     Dim nP As Long, nPoints As Long
125|     Dim dblIn As typDblePoint, dblOut As typDblePoint
126|
127|     nPoints = Len(XYBLOBIn) \ 8
128|     ReDim XYCoords(1 To 1) As RealPoint
129|     Call BLOB2RealPoints(XYBLOBIn, XYCoords())
130|
131|     For nP = 1 To nPoints
132|         dblIn.X = XYCoords(nP).X: dblIn.Y = XYCoords(nP).Y
133|         Call gctp_DConvertDecDegToUTM(dblIn, dblOut, 14&)
134|         XYCoords(nP).X = dblOut.X: XYCoords(nP).Y = dblOut.Y
135|     Next nP
136|    gctp_ProjectBLOBLLToUTM14 = RealPoints2BLOB(XYCoords())
137| End Function
138|
139| Function gctp_ProjectBLOBLLToUTMnn (XYBLOBIn As String, jZone As
   Long) As String
140|     Dim nP As Long, nPoints As Long
141|     Dim dblIn As typDblePoint, dblOut As typDblePoint
142|
143|     nPoints = Len(XYBLOBIn) \ 8
144|     ReDim XYCoords(1 To 1) As RealPoint
145|     Call BLOB2RealPoints(XYBLOBIn, XYCoords())
146|
147|     For nP = 1 To nPoints
148|         dblIn.X = XYCoords(nP).X: dblIn.Y = XYCoords(nP).Y
149|         Call gctp_DConvertDecDegToUTM(dblIn, dblOut, jZone)
150|         XYCoords(nP).X = dblOut.X: XYCoords(nP).Y = dblOut.Y
151|     Next nP
152|    gctp_ProjectBLOBLLToUTMnn = RealPoints2BLOB(XYCoords())
153|
154| End Function
155|
```

```
156| Function gctp_ProjectBLOBUTM14ToLL (XYBLOBIn As String) As
   String
157|     Dim nP As Long, nPoints As Long
158|     Dim dblIn As typDblePoint, dblOut As typDblePoint
159|
160|     nPoints = Len(XYBLOBIn) \ 8
161|     ReDim XYCoords(1 To 1) As RealPoint
162|     Call BLOB2RealPoints(XYBLOBIn, XYCoords())
163|
164|     For nP = 1 To nPoints
165|         dblIn.X = XYCoords(nP).X: dblIn.Y = XYCoords(nP).Y
166|         Call gctp_DConvertUTMToDecDeg(dblIn, dblOut, 14&)
167|         XYCoords(nP).X = dblOut.X: XYCoords(nP).Y = dblOut.Y
168|     Next nP
169|    gctp_ProjectBLOBUTM14ToLL = RealPoints2BLOB(XYCoords())
170|
171| End Function
172|
173| Function gctp_ProjectBLOBUTMnnToLL (XYBLOBIn As String, jZone As
   Long) As String
174|     Dim nP As Long, nPoints As Long
175|     Dim dblIn As typDblePoint, dblOut As typDblePoint
176|
177|     nPoints = Len(XYBLOBIn) \ 8
178|     ReDim XYCoords(1 To 1) As RealPoint
179|     Call BLOB2RealPoints(XYBLOBIn, XYCoords())
180|
181|     For nP = 1 To nPoints
182|         dblIn.X = XYCoords(nP).X: dblIn.Y = XYCoords(nP).Y
183|         Call gctp_DConvertUTMToDecDeg(dblIn, dblOut, jZone)
184|         XYCoords(nP).X = dblOut.X: XYCoords(nP).Y = dblOut.Y
185|     Next nP
186|    gctp_ProjectBLOBUTMnnToLL = RealPoints2BLOB(XYCoords())
187|
188| End Function
189|
190| Sub gctp_SConvertDecDegToUTM (sngLatLongIn As typRealPoint,
   sngUTMOut As typRealPoint, jZone As Long)
191| ''Takes Real Points to convert from Lat Long to  UTM.
192|      dblPoint1.X = sngLatLongIn.X: dblPoint1.Y = sngLatLongIn.Y
193|      Call gctp_DConvertUTMToDecDeg(dblPoint1, dblPoint2, jZone)
194|      sngUTMOut.X = dblPoint2.X: sngUTMOut.Y = dblPoint2.Y
195| End Sub
196|
197| Sub gctp_SConvertUTMToDecDeg (sngUTMIn As typRealPoint,
   sngLatLongOut As typRealPoint, jZone As Long)
198| ''Takes Real Points to convert from UTM to Lat Long.
199|      dblPoint1.X = sngUTMIn.X: dblPoint1.Y = sngUTMIn.Y
200|      Call gctp_DConvertUTMToDecDeg(dblPoint1, dblPoint2, jZone)
201|      sngLatLongOut.X = dblPoint2.X: sngLatLongOut.Y =
   dblPoint2.Y
202| End Sub
203|
204| Sub zzzGTRNZ0 ()
205| '*
   ************************************************************
   *******GCTP0197
206| '* ** NOAA/USGS GENERAL MAP PROJECTION PACKAGE ..... DR. A. A.
```

```
      ELASSAL **GCTP0198
  207| '* ** GCTP/II                  VERSION 1.0.2           SEPTEMBER
      1,1986 **GCTP0199
  208| '*
      ************************************************************
      *******GCTP0200
  209| '      SUBROUTINE GTRNZ0 (CRDIN,INDEF,TPARIN,CRDIO,IODEF,TPARIO,
      GCTP0201
  210| '     .                    IPFILE,IFLG)
      GCTP0202
  211| 'CRDIN  Input coordinates as a pair of real*8s
  212| 'INDEF  Integer*4 3-element array defining input coordianate
      system
  213| 'TPARIN  Thirteen Parameters real*8 defining input coordinate
      system
  214| 'CRDIO  Output coordinates as a pair of real*8s
  215| 'IODEF  Integer*4 3-element array defining output coordianate
      system
  216| 'TPARIO Thirteen Parameters real*8 defining input coordinate
      system
  217| 'IPFILE SET TO ZERO, pass as an integer*4
  218| 'IFLG   Return code.  Zero is good, other numbers indicate a
      problem.
  219|
  220| '*
      GCTP0203
  221| '* GENERAL PROGRAM FOR TRANSFORMATION BETWEEN VARIOUS REFERENCE
      SYSTEMS  GCTP0204
  222| '*
      GCTP0205
  223| '* INPUT
      ************************************************************
      GCTP0206
  224| '* CRDIN    : COORDINATES IN INPUT SYSTEM (2 DP WORDS ARRAY).
      GCTP0207
  225| '* INDEF(1) : CODE NUMBER OF INPUT COORDINATE SYSTEM (INTEGER).
      GCTP0208
  226| '*               =  0 , GEOGRAPHIC
      GCTP0209
  227| '*               =  1 , U T M
      GCTP0210
  228| '*               =  2 , STATE PLANE
      GCTP0211
  229| '*               =  3 , ALBERS CONICAL EQUAL-AREA
      GCTP0212
  230| '*               =  4 , LAMBERT CONFORMAL CONIC
      GCTP0213
  231| '*               =  5 , MERCATOR
      GCTP0214
  232| '*               =  6 , POLAR STEREOGRAPHIC
      GCTP0215
  233| '*               =  7 , POLYCONIC
      GCTP0216
  234| '*'               =  8 , EQUIDISTANT CONIC
      GCTP0217
  235| '*               =  9 , TRANSVERSE MERCATOR
      GCTP0218
  236| '*               = 10 , STEREOGRAPHIC
```

```
      GCTP0219
  237| '*                = 11 , LAMBERT AZIMUTHAL EQUAL-AREA
      GCTP0220
  238| '*                = 12 , AZIMUTHAL EQUIDISTANT
      GCTP0221
  239| '*                = 13 , GNOMONIC
      GCTP0222
  240| '*'                = 14 , ORTHOGRAPHIC
      GCTP0223
  241| '*                = 15 , GENERAL VERTICAL NEAR-SIDE PERSPECTIVE
      GCTP0224
  242| '*                = 16 , SINUSOIDAL
      GCTP0225
  243| '*                = 17 , EQUIRECTANGULAR
      GCTP0226
  244| '*                = 18 , MILLER CYLINDRICAL
      GCTP0227
  245| '*                = 19 , VAN DER GRINTEN I
      GCTP0228
  246| '*                = 20 , OBLIQUE MERCATOR (HOTINE)
      GCTP0229
  247| '* INDEF(2) : CODE NUMBER OF INPUT COORDINATE ZONE (INTEGER).
      GCTP0230
  248| '* TPARIN   : PARAMETERS OF INPUT REFERENCE SYSTEM (15 DP WORDS
      ARRAY).  GCTP0231
  249| '* INDEF(3) : CODE NUMBER OF UNITS OF MEASURE FOR INPUT COORDS
      (INTEGER).GCTP0232
  250| '*                = 0 , RADIANS.
      GCTP0233
  251| '*                = 1 , FEET.
      GCTP0234
  252| '*                = 2 , METERS.
      GCTP0235
  253| '*                = 3 , SECONDS OF ARC.
      GCTP0236
  254| '*                = 4 , DEGREES OF ARC.
      GCTP0237
  255| '*                = 5 , PACKED DMS.
      GCTP0238
  256| '* IODEF(1) : CODE NUMBER OF OUTPUT COORDINATE SYSTEM (INTEGER).
      GCTP0239
  257| '* IODEF(2) : CODE NUMBER OF OUTPUT COORDINATE ZONE (INTEGER).
      GCTP0240
  258| '* TPARIO   : PARAMETERS OF OUTPUT REFERENCE SYSTEM (15 DP WORDS
      ARRAY). GCTP0241
  259| '* IODEF(3) : CODE NUMBER OF UNITS OF MEASURE FOR OUTPUT COORDS
      (INTEGER)GCTP0242
  260| '* IPFILE   : LOGICAL NUMBER OF FILE FOR MESSAGES.
      GCTP0243
  261| '*
      GCTP0244
  262| '* OUTPUT
      ************************************************************
      GCTP0245
  263| '* CRDIO    : COORDINATES IN OUTPUT REFERENCE SYSTEM (2 DP WORDS
      ARRAY). GCTP0246
  264| '* IFLG     : RETURN FLAG (INTEGER).
      GCTP0247
```

```
'*              = 0 , SUCCESSFUL TRANSFORMATION.
GCTP0248
'*              = I , UNSUCCESSFUL TRANSFORMATION. ERROR CODE
= I.     GCTP0249
'*
GCTP0250
'       IMPLICIT REAL*8 (A-H,O-Z)

End Sub

```

```
 1| Option Explicit
 2| Type RealPoint
 3|     X As Single
 4|     Y As Single
 5| End Type
 6| Type String8
 7|     f As String * 8
 8| End Type
 9| Dim RlPt As RealPoint
10| Dim S8 As String8
11| 'File Open/Save Dialog Flags
12| Global Const OFN_READONLY = &H1&
13| Global Const OFN_OVERWRITEPROMPT = &H2&
14| Global Const OFN_HIDEREADONLY = &H4&
15| Global Const OFN_NOCHANGEDIR = &H8&
16| Global Const OFN_SHOWHELP = &H10&
17| Global Const OFN_NOVALIDATE = &H100&
18| Global Const OFN_ALLOWMULTISELECT = &H200&
19| Global Const OFN_EXTENSIONDIFFERENT = &H400&
20| Global Const OFN_PATHMUSTEXIST = &H800&
21| Global Const OFN_FILEMUSTEXIST = &H1000&
22| Global Const OFN_CREATEPROMPT = &H2000&
23| Global Const OFN_SHAREAWARE = &H4000&
24| Global Const OFN_NOREADONLYRETURN = &H8000&
25|
26| ' Field Data Types
27| Global Const DB_BOOLEAN = 1
28| Global Const DB_BYTE = 2
29| Global Const DB_INTEGER = 3
30| Global Const DB_LONG = 4
31| Global Const DB_CURRENCY = 5
32| Global Const DB_SINGLE = 6
33| Global Const DB_DOUBLE = 7
34| Global Const DB_DATE = 8
35| Global Const DB_TEXT = 10
36| Global Const DB_LONGBINARY = 11
37| Global Const DB_MEMO = 12
38|
39| Sub BLOB2RealPoints (BLOBin As String, DataPts() As RealPoint)
40| Dim n As Long, nPoints As Long, NPos As Long
41| nPoints = Len(BLOBin) / 8 '8 Bytes to a RealPoint
42| Dim NStart As Long
43| 'We will append datapoints to the end of the current array.
44| ' This way, we can have an unlimited number (kind of) of points
45| ' in a line.  Before starting to fill this array using
46| ' this subroutine, you should redim it to contain zero points
47| If LBound(DataPts) <> UBound(DataPts) Then
48|     NStart = UBound(DataPts) + 1
49|     ReDim Preserve DataPts(1 To UBound(DataPts) + nPoints)
50|   Else
51|     NStart = 1
52|     ReDim Preserve DataPts(1 To nPoints)
53| End If
54|
55| 'ReDim Preserve DataPts(1 To UBound(DataPts) + NPoints)
56| NPos = 1  'NPos will keep track of where in the BLOB we are
57| For n = NStart To UBound(DataPts)
58|     S8.f = Mid$(BLOBin, NPos, 8)
```

```
    LSet RlPt = S8
    DataPts(n) = RlPt
    NPos = NPos + 8
Next n
End Sub

Sub BLOB2XYPairsInSnglArray (BLOBin As String, sngXYData() As
Single)
'' Take XY Chains stored in a WHEAT format XYBLOB and put them
into
'' a one-based, two-dimensional array.
'' sngXYData() must be a dimensionable array
''  Point nPt will be X=sngXYData(nPt,1), Y=sngXYData(nPt,1)


Dim n As Long, nPoints As Long, NPos As Long
nPoints = Len(BLOBin) / 8 '8 Bytes to a RealPoint

ReDim sngXYData(1 To nPoints, 1 To 2)
NPos = 1  'NPos will keep track of where in the BLOB we are
For n = 1 To nPoints
    S8.f = Mid$(BLOBin, NPos, 8)
    LSet RlPt = S8
    sngXYData(n, 1) = RlPt.X
    sngXYData(n, 2) = RlPt.Y
    NPos = NPos + 8
Next n

End Sub

Function RealPoints2BLOB (DataPts() As RealPoint) As String
Dim n As Long
Dim BigTemp  As String
BigTemp = ""
For n = LBound(DataPts) To UBound(DataPts)
    RlPt = DataPts(n)
    LSet S8 = RlPt
    BigTemp = BigTemp & S8.f
Next n
RealPoints2BLOB = BigTemp
End Function

Function XYPairsInSnglArray2BLOB (sngXYData() As Single) As
String
'' Take XY Chains stored in a two-dimensional real*4 array and
put them
'' into aWHEAT format XYBLOB
''  Point nPt will be X=sngXYData(nPt,X_INDEX),
Y=sngXYData(nPt,Y_INDEX)
Dim X_INDEX As Long, Y_INDEX As Long, nPt As Long
Dim nFirst As Long, nLast As Long
nFirst = LBound(sngXYData, 1): nLast = UBound(sngXYData, 1)
X_INDEX = LBound(sngXYData, 2): Y_INDEX = UBound(sngXYData, 2)
If Y_INDEX - X_INDEX > 1 Then Exit Function
Dim BigTemp  As String
BigTemp = ""
For nPt = nFirst To nLast
    RlPt.X = sngXYData(nPt, X_INDEX): RlPt.Y = sngXYData(nPt,
```

```
      Y_INDEX)
  112|     LSet S8 = R1Pt
  113|     BigTemp = BigTemp & S8.f
  114| Next nPt
  115| XYPairsInSnglArray2BLOB = BigTemp
  116|
  117| End Function
  118|
```

Kansas Geological Survey
Open-file Report

Disclaimer

The Kansas Geological Survey does not guarantee this document to be free from errors or inaccuracies and disclaims any responsibility or liability for interpretations based on data used in the production of this document or decisions based thereon. This report is intended to make results of research available at the earliest possible date, but is not intended to constitute final or formal publication.