

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.

# Source Code for LLUTM14\_ Latitude-Longitude <=> UTM14 Projection Program

**Gregory W Pouch  
Geohydrology Section  
Kansas Geological Survey**

This report describes the workings of projection program LLUTM14\_, which converts between Geographic Angular and Universal Transverse Mercator Zone 14 coordinates. LLUTM14 is an event-driven program written in Microsoft Visual Basic 3.0 Professional Edition.

LLUTM14 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). LLUTM14 uses a copy of GCTP obtained over the internet from the USGS at [charon.er.usgs.gov](http://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.

LLUTM14\_ is hard-wired to use only Universal Transverse Mercator and Geographic Angular coordinates.

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: LLUTM14\_.MAK**

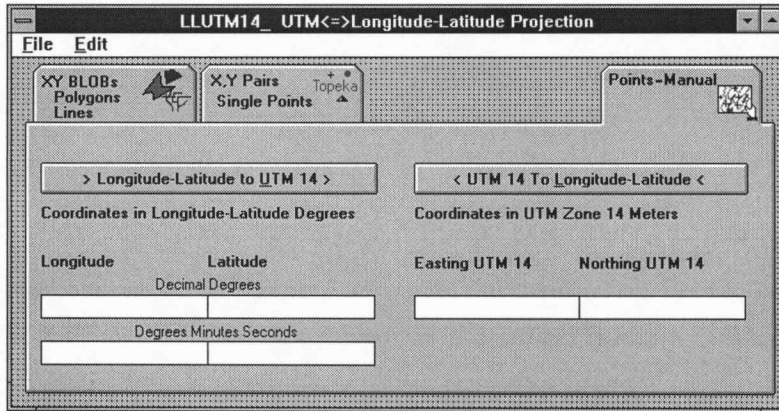
**3/31/96 2:48:13 PM**

Page #

---

Source Listing

LLUTM14_.FRM (illustration) . . . . .	1
LLUTM14_.FRM: (declarations) . . . . .	2
GCTP0001.BAS: (declarations) . . . . .	28
XYBLOB01.BAS: (declarations) . . . . .	35



```
1 | VERSION 2.00
2 | Begin Form frmLongLat_UTM14_ALL
3 |     BackColor      =    &H00C0C0C0&
4 |     Caption        =    "LLUTM14_ UTM<=>Longitude-Latitude
    Projection"
5 |     ClientHeight   =    4095
6 |     ClientLeft     =    150
7 |     ClientTop      =    2325
8 |     ClientWidth    =    8970
9 |     Height         =    4785
10 |    Icon           =    LLUTM14_.FRX:0000
11 |    Left           =    90
12 |    LinkTopic      =    "Form1"
13 |    ScaleHeight    =    273
14 |    ScaleMode      =    3 'Pixel
15 |    ScaleWidth     =    598
16 |    Top            =    1695
17 |    Width          =    9090
18 |    Begin SSRibbon cmdManual
19 |        AutoSize    =    1 'Adjust Picture Size To Button
20 |        BackColor   =    &H00C0C0C0&
21 |        BevelWidth  =    0
22 |        Height      =    720
23 |        Left        =    6825
24 |        Outline     =    0 'False
25 |        PictureDn   =    LLUTM14_.FRX:0302
26 |        PictureDnChange = 1 'Dither 'PictureUp' Bitmap
27 |        PictureUp   =    LLUTM14_.FRX:0F7C
28 |        Top         =    75
29 |        Value       =    -1 'True
30 |        Width       =    1920
31 |    End
32 |    Begin SSRibbon cmdBLOBs
33 |        AutoSize    =    1 'Adjust Picture Size To Button
34 |        BackColor   =    &H00C0C0C0&
35 |        BevelWidth  =    0
36 |        Height      =    720
37 |        Left        =    240
38 |        Outline     =    0 'False
39 |        PictureDn   =    LLUTM14_.FRX:1BF6
40 |        PictureDnChange = 1 'Dither 'PictureUp' Bitmap
41 |        PictureUp   =    LLUTM14_.FRX:2870
42 |        Top         =    75
43 |        Width       =    1920
44 |    End
45 |    Begin SSRibbon cmdPoints
46 |        AutoSize    =    1 'Adjust Picture Size To Button
47 |        BackColor   =    &H00C0C0C0&
48 |        BevelWidth  =    0
49 |        Height      =    720
50 |        Left        =    2175
51 |        Outline     =    0 'False
52 |        PictureDn   =    LLUTM14_.FRX:34EA
53 |        PictureDnChange = 1 'Dither 'PictureUp' Bitmap
54 |        PictureUp   =    LLUTM14_.FRX:4164
55 |        Top         =    75
56 |        Width       =    1920
57 |    End
```

```
58 | Begin SSPanel pnlManualEntry
59 |     BackColor          =    &H00C0C0C0&
60 |     BevelWidth         =    2
61 |     Font3D              =    0 'None
62 |     Height              =    3200
63 |     Left                =    150
64 |     Outline             =    -1 'True
65 |     TabIndex            =    48
66 |     Top                 =    750
67 |     Width               =    8715
68 | Begin TextBox txtJunk
69 |     Height              =    315
70 |     Left                =    15000
71 |     TabIndex            =    46
72 |     Text                 =    "This could be scrapped"
73 |     Top                 =    2520
74 |     Width               =    615
75 | End
76 | Begin TextBox txtLatDMS
77 |     FontBold            =    0 'False
78 |     FontItalic          =    0 'False
79 |     FontName            =    "MS Sans Serif"
80 |     FontSize            =    8.25
81 |     FontStrikethru      =    0 'False
82 |     FontUnderline       =    0 'False
83 |     Height              =    285
84 |     Left                =    2100
85 |     TabIndex            =    41
86 |     Top                 =    2580
87 |     Width               =    1950
88 | End
89 | Begin TextBox txtLongDMS
90 |     FontBold            =    0 'False
91 |     FontItalic          =    0 'False
92 |     FontName            =    "MS Sans Serif"
93 |     FontSize            =    8.25
94 |     FontStrikethru      =    0 'False
95 |     FontUnderline       =    0 'False
96 |     Height              =    285
97 |     Left                =    180
98 |     TabIndex            =    40
99 |     Top                 =    2580
100 |    Width               =    1950
101 | End
102 | Begin TextBox txtUTMNorthing
103 |     FontBold            =    0 'False
104 |     FontItalic          =    0 'False
105 |     FontName            =    "MS Sans Serif"
106 |     FontSize            =    8.25
107 |     FontStrikethru      =    0 'False
108 |     FontUnderline       =    0 'False
109 |     Height              =    285
110 |     Left                =    6420
111 |     TabIndex            =    39
112 |     Top                 =    2040
113 |     Width               =    1950
114 | End
115 | Begin TextBox txtUTMEasting
```

```
116 |         FontBold      = 0   'False
117 |         FontItalic    = 0   'False
118 |         FontName      = "MS Sans Serif"
119 |         FontSize      = 8.25
120 |         FontStrikethru = 0   'False
121 |         FontUnderline = 0   'False
122 |         Height        = 285
123 |         Left          = 4500
124 |         TabIndex      = 38
125 |         Top           = 2040
126 |         Width         = 1950
127 |     End
128 | Begin TextBox txtLatDecDeg
129 |     FontBold      = 0   'False
130 |     FontItalic    = 0   'False
131 |     FontName      = "MS Sans Serif"
132 |     FontSize      = 8.25
133 |     FontStrikethru = 0   'False
134 |     FontUnderline = 0   'False
135 |     Height        = 285
136 |     Left          = 2100
137 |     TabIndex      = 37
138 |     Top           = 2040
139 |     Width         = 1950
140 | End
141 | Begin TextBox txtLongDecDeg
142 |     FontBold      = 0   'False
143 |     FontItalic    = 0   'False
144 |     FontName      = "MS Sans Serif"
145 |     FontSize      = 8.25
146 |     FontStrikethru = 0   'False
147 |     FontUnderline = 0   'False
148 |     Height        = 285
149 |     Left          = 180
150 |     TabIndex      = 36
151 |     Top           = 2040
152 |     Width         = 1950
153 | End
154 | Begin CommandButton cmdLongLatFromUTM_MANUAL
155 |     Caption      = "< UTM 14 To &Longitude-Latitude <"
156 |     Height       = 375
157 |     Left         = 4500
158 |     TabIndex     = 35
159 |     Top          = 480
160 |     Width        = 3900
161 | End
162 | Begin CommandButton cmdUTMFromLongLat_Manual
163 |     Caption      = "> Longitude-Latitude to &UTM 14 >"
164 |     Height       = 375
165 |     Left         = 180
166 |     TabIndex     = 34
167 |     Top          = 480
168 |     Width        = 3900
169 | End
170 | Begin Label lblNorthing_Manual
171 |     BackStyle    = 0   'Transparent
172 |     Caption      = "Northing UTM 14"
173 |     Height       = 195
```

```
174 |         Left           = 6420
175 |         TabIndex      = 49
176 |         Top           = 1560
177 |         Width         = 1815
178 |     End
179 |     Begin Label lblEasting_MANUAL
180 |         BackStyle     = 0 'Transparent
181 |         Caption       = "Easting UTM 14"
182 |         Height        = 195
183 |         Left          = 4500
184 |         TabIndex      = 47
185 |         Top           = 1560
186 |         Width         = 1815
187 |     End
188 |     Begin Label lblDMS_MANUAL
189 |         BackStyle     = 0 'Transparent
190 |         Caption       = "Degrees Minutes Seconds"
191 |         FontBold      = 0 'False
192 |         FontItalic    = 0 'False
193 |         FontName      = "MS Sans Serif"
194 |         FontSize      = 8.25
195 |         FontStrikethru = 0 'False
196 |         FontUnderline = 0 'False
197 |         Height        = 195
198 |         Left          = 1260
199 |         TabIndex      = 45
200 |         Top           = 2370
201 |         Width         = 2415
202 |     End
203 |     Begin Label lblDecDeg
204 |         BackStyle     = 0 'Transparent
205 |         Caption       = "Decimal Degrees"
206 |         FontBold      = 0 'False
207 |         FontItalic    = 0 'False
208 |         FontName      = "MS Sans Serif"
209 |         FontSize      = 8.25
210 |         FontStrikethru = 0 'False
211 |         FontUnderline = 0 'False
212 |         Height        = 195
213 |         Left          = 1500
214 |         TabIndex      = 44
215 |         Top           = 1820
216 |         Width         = 1275
217 |     End
218 |     Begin Label lblLatitude_MANUAL
219 |         BackStyle     = 0 'Transparent
220 |         Caption       = "Latitude"
221 |         Height        = 195
222 |         Left          = 2100
223 |         TabIndex      = 43
224 |         Top           = 1550
225 |         Width         = 1875
226 |     End
227 |     Begin Label lblLongitude_MANUAL
228 |         BackStyle     = 0 'Transparent
229 |         Caption       = "Longitude"
230 |         Height        = 255
231 |         Left          = 180
```

```
232 |         TabIndex      = 42
233 |         Top           = 1550
234 |         Width        = 1635
235 |     End
236 |     Begin Label lblUTM14_MANUAL
237 |         BackStyle    = 0 'Transparent
238 |         Caption      = "Coordinates in UTM Zone 14 Meters"
239 |         Height       = 375
240 |         Left         = 4500
241 |         TabIndex     = 33
242 |         Top          = 980
243 |         Width        = 3975
244 |     End
245 |     Begin Label lblLongLat_MANUAL
246 |         BackStyle    = 0 'Transparent
247 |         Caption      = "Coordinates in Longitude-Latitude
248 |         Height       = 495
249 |         Left         = 180
250 |         TabIndex     = 32
251 |         Top          = 980
252 |         Width        = 3975
253 |     End
254 | End
255 | Begin SSPanel pnlSinglePoints
256 |     BackColor       = &H00C0C0C0&
257 |     BevelWidth      = 2
258 |     Font3D          = 0 'None
259 |     Height          = 3200
260 |     Left            = 150
261 |     Outline         = -1 'True
262 |     TabIndex        = 0
263 |     Top             = 750
264 |     Width           = 8715
265 |     Begin PictureBox picGray
266 |         BackColor   = &H00C0C0C0&
267 |         BorderStyle = 0 'None
268 |         Height      = 300
269 |         Index       = 5
270 |         Left        = 8120
271 |         ScaleHeight = 300
272 |         ScaleWidth  = 105
273 |         TabIndex    = 31
274 |         Top         = 1680
275 |         Width       = 100
276 |     End
277 |     Begin PictureBox picGray
278 |         BackColor   = &H00C0C0C0&
279 |         BorderStyle = 0 'None
280 |         Height      = 300
281 |         Index       = 4
282 |         Left        = 8120
283 |         ScaleHeight = 300
284 |         ScaleWidth  = 105
285 |         TabIndex    = 30
286 |         Top         = 2340
287 |         Width       = 100
288 |     End
```

```
289      Begin PictureBox picGray
290          BackColor      =    &H00C0C0C0&
291          BorderStyle    =    0  'None
292          Height         =    300
293          Index          =    3
294          Left           =    3800
295          ScaleHeight    =    300
296          ScaleWidth     =    105
297          TabIndex      =    29
298          Top            =    2340
299          Width          =    100
300      End
301      Begin PictureBox picGray
302          BackColor      =    &H00C0C0C0&
303          BorderStyle    =    0  'None
304          Height         =    300
305          Index          =    2
306          Left           =    3800
307          ScaleHeight    =    300
308          ScaleWidth     =    105
309          TabIndex      =    28
310          Top            =    1680
311          Width          =    100
312      End
313      Begin Data Dat_Points
314          Caption        =    "Move record "
315          Connect         =    ""
316          DatabaseName   =    ""
317          Exclusive      =    0  'False
318          Height         =    270
319          Left           =    9120
320          Options        =    0
321          ReadOnly       =    0  'False
322          RecordSource   =    ""
323          Top            =    1200
324          Width          =    2445
325      End
326      Begin ComboBox cmbQTLList_Points
327          Height         =    300
328          Left           =    90
329          Sorted         =    -1  'True
330          Style          =    2  'Dropdown List
331          TabIndex      =    9
332          Top            =    120
333          Width          =    8535
334      End
335      Begin ComboBox cmbXY_LongLat_Points
336          Height         =    300
337          Index          =    0
338          Left           =    180
339          TabIndex      =    8
340          Text           =    "X Longitude"
341          Top            =    1680
342          Width          =    3968
343      End
344      Begin ComboBox cmbXY_UTM14_Points
345          Height         =    300
346          Index          =    0
```

```
347 |         Left           = 4500
348 |         TabIndex      = 7
349 |         Text          = "X UTM 14"
350 |         Top           = 1680
351 |         Width         = 3968
352 |     End
353 |     Begin CommandButton cmdUTMFromLongLat_Points
354 |         Caption       = "> Longitude-Latitude to &UTM 14 >"
355 |         Height        = 375
356 |         Left          = 180
357 |         TabIndex      = 6
358 |         Top           = 480
359 |         Width         = 3900
360 |     End
361 |     Begin CommandButton cmdLatLongFromUTM14_Points
362 |         Caption       = "< UTM 14 To &Longitude-Latitude <"
363 |         Height        = 375
364 |         Left          = 4500
365 |         TabIndex      = 5
366 |         Top           = 480
367 |         Width         = 3900
368 |     End
369 |     Begin PictureBox picPercentDoneHost_Points
370 |         Height        = 255
371 |         Left          = 120
372 |         ScaleHeight   = 1
373 |         ScaleMode     = 0 'User
374 |         ScaleWidth    = 97.237
375 |         TabIndex      = 3
376 |         Top           = 2760
377 |         Width         = 8475
378 |         Begin PictureBox picPercentDone_Points
379 |             BackColor = &H00FF0000&
380 |             BorderStyle = 0 'None
381 |             Height     = 225
382 |             Left       = 0
383 |             ScaleHeight = 225
384 |             ScaleWidth = 15
385 |             TabIndex   = 4
386 |             Top        = 0
387 |             Width      = 15
388 |         End
389 |     End
390 |     Begin ComboBox cmbXY_LongLat_Points
391 |         Height      = 300
392 |         Index       = 1
393 |         Left        = 180
394 |         TabIndex    = 2
395 |         Text        = "Y Latitude"
396 |         Top         = 2340
397 |         Width       = 3968
398 |     End
399 |     Begin ComboBox cmbXY_UTM14_Points
400 |         Height      = 300
401 |         Index       = 1
402 |         Left        = 4500
403 |         TabIndex    = 1
404 |         Text        = "Y UTM 14"
```

```
405 |         Top           = 2340
406 |         Width        = 3968
407 |     End
408 |     Begin Label lblBLOBLatLong
409 |         BackStyle    = 0 'Transparent
410 |         Caption      = "Felds containing coordinates in
Longitude-Latitude Degrees"
411 |         Height       = 495
412 |         Left         = 180
413 |         TabIndex     = 15
414 |         Top          = 1000
415 |         Width        = 3968
416 |     End
417 |     Begin Label lblUTMFields
418 |         BackStyle    = 0 'Transparent
419 |         Caption      = "Fields containing coordinates in
UTM Zone 14"
420 |         Height       = 375
421 |         Left         = 4500
422 |         TabIndex     = 14
423 |         Top          = 1000
424 |         Width        = 3968
425 |     End
426 |     Begin Label lblLong_Points
427 |         BackStyle    = 0 'Transparent
428 |         Caption      = "X Longitude"
429 |         Height       = 255
430 |         Index        = 0
431 |         Left         = 180
432 |         TabIndex     = 13
433 |         Top          = 1440
434 |         Width        = 3968
435 |     End
436 |     Begin Label lblLong_Points
437 |         BackStyle    = 0 'Transparent
438 |         Caption      = "Y Latitude"
439 |         Height       = 255
440 |         Index        = 1
441 |         Left         = 180
442 |         TabIndex     = 12
443 |         Top          = 2100
444 |         Width        = 3975
445 |     End
446 |     Begin Label zUTMLabel12
447 |         BackStyle    = 0 'Transparent
448 |         Caption      = "Easting UTM 14      X"
449 |         Height       = 255
450 |         Index        = 0
451 |         Left         = 4500
452 |         TabIndex     = 11
453 |         Top          = 1440
454 |         Width        = 3975
455 |     End
456 |     Begin Label zUTMLabel12
457 |         BackStyle    = 0 'Transparent
458 |         Caption      = "Northing UTM 14    Y"
459 |         Height       = 255
460 |         Index        = 1
```

```
461         Left           = 4500
462         TabIndex        = 10
463         Top              = 2100
464         Width            = 3975
465     End
466 End
467 Begin SSPanel pnlBLOBS
468     BackColor            = &H00C0C0C0&
469     BevelWidth           = 2
470     Font3D               = 0 'None
471     Height               = 3200
472     Left                 = 150
473     Outline              = -1 'True
474     TabIndex             = 16
475     Top                  = 750
476     Width                = 8715
477     Begin PictureBox picGray
478         BackColor        = &H00C0C0C0&
479         BorderStyle       = 0 'None
480         Height           = 300
481         Index            = 1
482         Left             = 8120
483         ScaleHeight      = 300
484         ScaleWidth       = 105
485         TabIndex         = 27
486         Top              = 1680
487         Width            = 100
488     End
489     Begin PictureBox picGray
490         BackColor        = &H00C0C0C0&
491         BorderStyle       = 0 'None
492         Height           = 300
493         Index            = 0
494         Left             = 3800
495         ScaleHeight      = 300
496         ScaleWidth       = 105
497         TabIndex         = 26
498         Top              = 1680
499         Width            = 100
500     End
501     Begin Data Dat_BLOBS
502         Caption          = "Move record "
503         Connect           = ""
504         DatabaseName      = ""
505         Exclusive         = 0 'False
506         Height           = 270
507         Left              = 9120
508         Options           = 0
509         ReadOnly          = 0 'False
510         RecordSource      = ""
511         Top              = 1200
512         Width            = 2445
513     End
514     Begin ComboBox cmbQTList_BLOBS
515         Height           = 300
516         Left             = 90
517         Sorted           = -1 'True
518         Style            = 2 'Dropdown List
```

```
519 |         TabIndex      = 23
520 |         Top           = 120
521 |         Width        = 8535
522 |     End
523 |     Begin CommonDialog CMDialog1
524 |         Left         = 9120
525 |         Top          = 4320
526 |     End
527 |     Begin ComboBox cmbXY_LongLat_BLOBS
528 |         Height       = 300
529 |         Left         = 180
530 |         TabIndex     = 22
531 |         Text         = "XYBLOB Longitude Latitude"
532 |         Top          = 1680
533 |         Width        = 3968
534 |     End
535 |     Begin ComboBox cmbXY_UTM14_BLOBS
536 |         Height       = 300
537 |         Left         = 4500
538 |         TabIndex     = 21
539 |         Text         = "cmbXY_UTM14_BLOBS"
540 |         Top          = 1680
541 |         Width        = 3968
542 |     End
543 |     Begin CommandButton cmdUTMFromLongLat_BLOBS
544 |         Caption      = "> Longitude-Latitude to &UTM 14 >"
545 |         Height       = 375
546 |         Left         = 180
547 |         TabIndex     = 20
548 |         Top          = 480
549 |         Width        = 3900
550 |     End
551 |     Begin CommandButton cmdLatLongFromUTM14_BLOBS
552 |         Caption      = "< UTM 14 To &Longitude-Latitude <"
553 |         Height       = 375
554 |         Left         = 4500
555 |         TabIndex     = 19
556 |         Top          = 480
557 |         Width        = 3900
558 |     End
559 |     Begin PictureBox picPercentDoneHost
560 |         Height       = 255
561 |         Left         = 180
562 |         ScaleHeight  = 1
563 |         ScaleMode    = 0 'User
564 |         ScaleWidth   = 96.546
565 |         TabIndex     = 17
566 |         Top          = 2800
567 |         Width        = 8415
568 |     Begin PictureBox picPercentDone
569 |         BackColor    = &H00FF0000&
570 |         BorderStyle  = 0 'None
571 |         Height       = 225
572 |         Left         = 0
573 |         ScaleHeight  = 225
574 |         ScaleWidth   = 15
575 |         TabIndex     = 18
576 |         Top          = 0
```

```
577         Width           = 15
578     End
579 End
580 Begin Label lblBLOB_LatLong
581     BackStyle           = 0 'Transparent
582     Caption             = "Binary Large OBject (BLOB) field
containing coordinates in Longitude-Latitude Degrees"
583     Height              = 495
584     Left                 = 180
585     TabIndex            = 25
586     Top                  = 1200
587     Width                = 3975
588 End
589 Begin Label lblBLOBS_UTM
590     BackStyle           = 0 'Transparent
591     Caption             = "Binary Large OBject (BLOB) field
containing coordinates in UTM Zone 14 meters"
592     Height              = 375
593     Left                 = 4500
594     TabIndex            = 24
595     Top                  = 1200
596     Width                = 3975
597 End
598 End
599 Begin Menu zzzzFile
600     Caption             = "&File"
601     Begin Menu mfOpenDB
602         Caption         = "&Open Database..."
603     End
604     Begin Menu mfCloseDatabase
605         Caption         = "&Close Database"
606     End
607     Begin Menu zzzzz1
608         Caption         = "-"
609     End
610     Begin Menu mfExit
611         Caption         = "E&xit"
612     End
613 End
614 Begin Menu zzMnuEdit
615     Caption             = "&Edit"
616     Begin Menu meCopy
617         Caption         = "&Copy"
618         Shortcut        = ^{INSERT}
619     End
620     Begin Menu mePaste
621         Caption         = "&Paste"
622         Shortcut        = +{INSERT}
623     End
624 End
625 End
626 Option Explicit
627
628 Dim DBName As String' May want to change scope of this later.
629 Dim intCoordinateField As Long
630 Dim XYBlob As String
631 Dim DataPt() As RealPoint
632 Dim flgChangeRow As Integer, flgIgnoreMovement As Integer,
```

```
NoAuto As Integer
633 | Dim TabChar As String * 1, CR As String * 1, CRLF As String *
    | 2
634 |
635 | Dim nFields As Long, n As Long, nNodesAlready As Long
636 |
637 | Dim DB As Database, tblLineData As table
638 | Dim snpQTLList As Snapshot, dnsCurrentQT As Dynaset
639 |
640 | Dim flgAddingRecord As Integer
641 | Dim SepName(0 To 20) As String, SepChar(0 To 20) As String
642 | Const X_INDEX = 0
643 | Const Y_INDEX = 1
644 |
645 | Sub cmbQTLList_BLOBS_Click ()
646 |     Dim Temp As String
647 |     Temp = cmbQTLList_BLOBS
648 |     Call SetQuery_BLOBS(Temp)
649 |
650 | End Sub
651 |
652 | Sub cmbQTLList_Points_Click ()
653 |     Dim Temp As String
654 |     Temp = cmbQTLList_Points
655 |     Call SetQuery_Points(Temp)
656 |
657 | End Sub
658 |
659 | Sub cmdBLOBS_Click (Value As Integer)
660 |     If DBName = "" Then
661 |         Call GripeNoDatabase
662 |         Exit Sub
663 |     End If
664 |
665 |
666 |     Value = True
667 |     pnlBLOBS.ZOrder
668 |     cmdBLOBS.ZOrder
669 |     zzMnuEdit.Enabled = False
670 | End Sub
671 |
672 | Sub cmdLatLongFromUTM14_BLOBS_Click ()
673 |     Dim XYDelim As String, PairDelim As String, Temp As String
674 |     'First, get the xy and xypair delimiters
675 |
676 |     If cmbXY_UTM14_BLOBS.ListIndex = -1 Then
677 |         MsgBox "That field does not exist in the present
        | query/table, so you can't very well convert FROM it, now can
        | you?"
678 |         Exit Sub
679 |     End If
680 |     Dim name_LongLatTo As String, name_UTM14From As String
681 |     name_UTM14From = cmbXY_UTM14_BLOBS.Text
682 |
683 |     If cmbXY_LongLat_BLOBS.ListIndex = -1 Then
684 |         Dim strSourceTable As String
685 |         strSourceTable =
        | Dat_BLOBS.Recordset.Fields(name_UTM14From).SourceTable
```

```
686     If Trim$(cmbXY_LongLat_BLOBS.Text) = "" Then
687         name_LongLatTo = "XYBLOB_LongitudeLatitude"
688     Else
689         name_LongLatTo = Trim$(cmbXY_LongLat_BLOBS.Text)
690     End If
691     Dim nfldBLOB_LongLat As New field
692     nfldBLOB_LongLat.Name = name_LongLatTo:
nfldBLOB_LongLat.Type = DB_LONGBINARY
693
694     Temp = Dat_BLOBS.RecordSource
695     Dat_BLOBS.RecordSource = "": Dat_BLOBS.Refresh
696     'Professional Edition of VB 3.0 only allows this addition of
a field
697     DB.TableDefs(strSourceTable).Fields.Append nfldBLOB_LongLat
698     Dat_BLOBS.RecordSource = Temp: Dat_BLOBS.Refresh
699     Else
700         name_LongLatTo = cmbXY_LongLat_BLOBS.Text
701     End If
702
703
704     Dim nR As Long, nRecs As Long', Temp As String
705     Dat_BLOBS.Recordset.MoveFirst : Dat_BLOBS.Recordset.MoveLast :
Dat_BLOBS.Recordset.MoveFirst
706     nRecs = Dat_BLOBS.Recordset.RecordCount
707     For nR = 1 To nRecs
708         picPercentDone.Width = nR / nRecs * 100
709         DoEvents
710         Temp = Dat_BLOBS.Recordset(name_UTM14From)
711         Dat_BLOBS.Recordset.Edit
712         Dat_BLOBS.Recordset(name_LongLatTo) =
gctp_ProjectBLOBUTM14ToLL(Temp)
713         Dat_BLOBS.Recordset.Update
714         Dat_BLOBS.Recordset.MoveNext
715     Next nR
716
717
718
719
720
721     End Sub
722
723     Sub cmdLatLongFromUTM14_Points_Click ()
724     Dim Temp As String
725
726     'Make sure the fields the user wants to project FROM exist.
727     If (cmbXY_UTM14_Points(X_INDEX).ListIndex = -1) Or
(cmbXY_UTM14_Points(Y_INDEX).ListIndex = -1) Then
728         MsgBox "That field does not exist in the present
query/table, so you can't very well convert FROM it, now can
you?"
729         Exit Sub
730     End If
731
732     Dim name_XLong As String, name_UTM14XFrom As String
733     Dim name_YLat As String, name_UTM14YFrom As String
734     name_UTM14XFrom = cmbXY_UTM14_Points(X_INDEX).Text
735     name_UTM14YFrom = cmbXY_UTM14_Points(Y_INDEX).Text
736
```

```
737 | Dim Temp1 As Integer, Temp2 As Integer
738 | Temp1 = cmbXY_LongLat_Points(X_INDEX).ListIndex < 0
739 | Temp2 = cmbXY_LongLat_Points(Y_INDEX).ListIndex < 0
740 |
741 | If Temp1 Xor Temp2 Then
742 |     Beep
743 |     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."
744 |     Exit Sub
745 | End If
746 |
747 | If cmbXY_LongLat_Points(X_INDEX).ListIndex = -1 Then
748 |     Dim strSourceTable As String
749 |     'Put the output coordinates in the same table as the input
       | coordinates
750 |     strSourceTable =
       | Dat_Points.Recordset.Fields(name_UTM14XFrom).SourceTable
751 |     If Trim$(cmbXY_LongLat_Points(X_INDEX).Text) = "" Then
752 |         name_XLong = "X_Longitude"
753 |     Else
754 |         name_XLong = Trim$(cmbXY_LongLat_Points(X_INDEX).Text)
755 |     End If
756 |     If Trim$(cmbXY_LongLat_Points(Y_INDEX).Text) = "" Then
757 |         name_YLat = "Y_Latitude"
758 |     Else
759 |         name_YLat = Trim$(cmbXY_LongLat_Points(Y_INDEX).Text)
760 |     End If
761 |
762 |
763 |     Dim nfldX_Long As New field, nfldY_Lat As New field
764 |     nfldX_Long.Name = name_XLong: nfldX_Long.Type = DB_SINGLE
765 |     nfldY_Lat.Name = name_YLat: nfldY_Lat.Type = DB_SINGLE
766 |
767 |     Temp = Dat_Points.RecordSource
768 |     Dat_Points.RecordSource = "": Dat_Points.Refresh
769 |     'Professional Edition of VB 3.0 only allows this addition of
       | a field
770 |     DB.TableDefs(strSourceTable).Fields.Append nfldX_Long
771 |     DB.TableDefs(strSourceTable).Fields.Append nfldY_Lat
772 |     Dat_Points.RecordSource = Temp: Dat_Points.Refresh
773 |     Else
774 |         name_XLong = cmbXY_LongLat_Points(X_INDEX).Text
775 |         name_YLat = cmbXY_LongLat_Points(Y_INDEX).Text
776 |     End If
777 |
778 |
779 | Dim nR As Long, nRecs As Long', Temp As String
780 | Dat_Points.Recordset.MoveFirst : Dat_Points.Recordset.MoveLast :
       | Dat_Points.Recordset.MoveFirst
781 | nRecs = Dat_Points.Recordset.RecordCount
782 | Dim jZone14 As Long
783 | jZone14 = 14
784 | Dim dblLLOut As typDblePoint, DblUTMin As typDblePoint
785 | For nR = 1 To nRecs
786 |     picPercentDone_Points.Width = nR / nRecs * 100
787 |     DoEvents
788 |
789 |     DblUTMin.X = Dat_Points.Recordset(name_UTM14XFrom)
```

```
790     DblUTMIn.Y = Dat_Points.Recordset(name_UTM14YFrom)
791     Call gctp_DConvertUTMToDecDeg(DblUTMIn, dblLLOut, jZone14)
792     Dat_Points.Recordset.Edit
793     Dat_Points.Recordset(name_XLong) = dblLLOut.X
794     Dat_Points.Recordset(name_YLat) = dblLLOut.Y
795     Dat_Points.Recordset.Update
796     Dat_Points.Recordset.MoveNext
797 Next nR
798
799
800
801
802
803 End Sub
804
805 Sub cmdLongLatFromUTM_Manual_Click ()
806     Dim dblIn As typDblePoint, dblOut As typDblePoint
807     dblIn.X = Val(txtUTMEasting.Text)
808     dblIn.Y = Val(txtUTMNorthing.Text)
809     Call gctp_DConvertUTMToDecDeg(dblIn, dblOut, 14&)
810     txtLongDecDeg = Format$(dblOut.X, "#.000000")
811     txtLatDecDeg = Format$(dblOut.Y, "#.000000")
812     txtLongDMS = DecDegToDMS(dblOut.X)
813     txtLatDMS = DecDegToDMS(dblOut.Y)
814
815 End Sub
816
817 Sub cmdMANUAL_Click (Value As Integer)
818     Value = True
819     pnlManualEntry.ZOrder
820     cmdManual.ZOrder
821     zzMnuEdit.Enabled = True
822 End Sub
823
824 Sub cmdPoints_Click (Value As Integer)
825     If DBName = "" Then
826         Call GripeNoDatabase
827         Exit Sub
828     End If
829
830
831     Value = True
832     zzMnuEdit.Enabled = False
833     pnlSinglePoints.ZOrder
834     cmdPoints.ZOrder
835 End Sub
836
837 Sub cmdUTMFromLongLat_BLOBS_Click ()
838     Dim XYDelim As String, PairDelim As String
839     'First, get the xy and xypair delimiters
840
841     If cmbXY_LongLat_BLOBS.ListIndex = -1 Then
842         MsgBox "That field does not exist in the present
query/table, so you can't very well convert FROM it, now can
you?"
843         Exit Sub
844     End If
845     Dim name_BLOB_LongLat As String, name_BLOB_UTM14 As String
```

```
846 name_BLOB_LongLat = cmbXY_LongLat_BLOBS.Text
847
848 If cmbXY_UTM14_BLOBS.ListIndex = -1 Then
849     Dim strSourceTable As String
850     strSourceTable =
851     Dat_BLOBS.Recordset.Fields(name_BLOB_LongLat).SourceTable
852     If Trim$(cmbXY_UTM14_BLOBS.Text) = "" Then
853         name_BLOB_UTM14 = "XYname_BLOB_UTM14"
854     Else
855         name_BLOB_UTM14 = Trim$(cmbXY_UTM14_BLOBS.Text)
856     End If
857     Dim nfldBLOB_UTM14 As New field
858     nfldBLOB_UTM14.Name = name_BLOB_UTM14: nfldBLOB_UTM14.Type =
859     DB_LONGBINARY
860     Dim Temp As String
861     Temp = Dat_BLOBS.RecordSource
862     Dat_BLOBS.RecordSource = "": Dat_BLOBS.Refresh
863     'Professional Edition of VB 3.0 only allows this addition of
864     a field
865     DB.TableDefs(strSourceTable).Fields.Append nfldBLOB_UTM14
866     Dat_BLOBS.RecordSource = Temp: Dat_BLOBS.Refresh
867     Else
868     name_BLOB_UTM14 = cmbXY_UTM14_BLOBS.Text
869 End If
870
871 'CopyXYFromBLOBToMemo name_BLOB_LongLat, name_BLOB_UTM14,
872 XYDelim, PairDelim, Me.Dat_BLOBS.Recordset
873 Dim nR As Long, nRecs As Long', Temp As String
874 Dat_BLOBS.Recordset.MoveFirst : Dat_BLOBS.Recordset.MoveLast :
875 Dat_BLOBS.Recordset.MoveFirst
876 nRecs = Dat_BLOBS.Recordset.RecordCount
877 For nR = 1 To nRecs
878     picPercentDone.Width = nR / nRecs * 100
879     DoEvents
880     Temp = Dat_BLOBS.Recordset(name_BLOB_LongLat)
881     Dat_BLOBS.Recordset.Edit
882     Dat_BLOBS.Recordset(name_BLOB_UTM14) =
883     gctp_ProjectBLOBLToUTM14(Temp)
884     Dat_BLOBS.Recordset.Update
885     Dat_BLOBS.Recordset.MoveNext
886 Next nR
887 End Sub
888
889 Sub cmdUTMFromLongLat_Manual_Click ()
890     Dim dblIn As typDblePoint, dblOut As typDblePoint
891     dblIn.X = -Abs(Val(txtLongDecDeg.Text))
892     'This assure a negative value, and since we're hardwiring
893     for UTM 14, this makes sense
894     dblIn.Y = Val(txtLatDecDeg.Text)
895     Call gctp_DConvertDecDegToUTM(dblIn, dblOut, 14&)
896     txtUTMEasting = Format$(dblOut.X, "0")
897     txtUTMNorthing = Format$(dblOut.Y, "0")
898 End Sub
899
```

```
897 | Sub cmdUTMFromLongLat_Points_Click ()
898 | Dim Temp As String
899 |
900 | 'Make sure the fields the user wants to project FROM exist.
901 | If (cmbXY_LongLat_Points(X_INDEX).ListIndex = -1) Or
    | (cmbXY_LongLat_Points(Y_INDEX).ListIndex = -1) Then
902 |     MsgBox "That field does not exist in the present
    | query/table, so you can't very well convert FROM it, now can
    | you?"
903 |     Exit Sub
904 | End If
905 |
906 | Dim name_UTMX_Out As String, name_XLongFrom As String
907 | Dim name_UTMY_Out As String, name_YLatFrom As String
908 | name_XLongFrom = cmbXY_LongLat_Points(X_INDEX).Text
909 | name_YLatFrom = cmbXY_LongLat_Points(Y_INDEX).Text
910 |
911 | Dim Temp1 As Integer, Temp2 As Integer
912 | Temp1 = cmbXY_UTM14_Points(X_INDEX).ListIndex < 0
913 | Temp2 = cmbXY_UTM14_Points(Y_INDEX).ListIndex < 0
914 |
915 | If Temp1 Xor Temp2 Then
916 |     Beep
917 |     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."
918 |     Exit Sub
919 | End If
920 |
921 | If cmbXY_UTM14_Points(X_INDEX).ListIndex = -1 Then
922 |     Dim strSourceTable As String
923 |     'Put the output coordinates in the same table as the input
    | coordinates
924 |     strSourceTable =
    | Dat_Points.Recordset.Fields(name_XLongFrom).SourceTable
925 |     If Trim$(cmbXY_UTM14_Points(X_INDEX).Text) = "" Then
926 |         name_UTMX_Out = "Easting_UTM14"
927 |     Else
928 |         name_UTMX_Out = Trim$(cmbXY_UTM14_Points(X_INDEX).Text)
929 |     End If
930 |     If Trim$(cmbXY_UTM14_Points(Y_INDEX).Text) = "" Then
931 |         name_UTMY_Out = "Northing_UTM14"
932 |     Else
933 |         name_UTMY_Out = Trim$(cmbXY_UTM14_Points(Y_INDEX).Text)
934 |     End If
935 |
936 |
937 |     Dim nfldX_UTM14 As New field, nfldY_UTM14 As New field
938 |     nfldX_UTM14.Name = name_UTMX_Out: nfldX_UTM14.Type =
    | DB_SINGLE
939 |     nfldY_UTM14.Name = name_UTMY_Out: nfldY_UTM14.Type =
    | DB_SINGLE
940 |
941 |     Temp = Dat_Points.RecordSource
942 |     Dat_Points.RecordSource = "": Dat_Points.Refresh
943 |     'Professional Edition of VB 3.0 only allows this addition of
    | a field
944 |     DB.TableDefs(strSourceTable).Fields.Append nfldX_UTM14
945 |     DB.TableDefs(strSourceTable).Fields.Append nfldY_UTM14
```

```
946     Dat_Points.RecordSource = Temp: Dat_Points.Refresh
947     Else
948         name_UTMX_Out = cmbXY_UTM14_Points(X_INDEX).Text
949         name_UTMY_Out = cmbXY_UTM14_Points(Y_INDEX).Text
950 End If
951
952
953 Dim nR As Long, nRecs As Long', Temp As String
954 Dat_Points.Recordset.MoveFirst : Dat_Points.Recordset.MoveLast :
Dat_Points.Recordset.MoveFirst
955 nRecs = Dat_Points.Recordset.RecordCount
956 Dim dblLLIn As typDblePoint, DblUTMOut As typDblePoint
957 For nR = 1 To nRecs
958     picPercentDone_Points.Width = nR / nRecs * 100
959     DoEvents
960
961     dblLLIn.X = Dat_Points.Recordset(name_XLongFrom)
962     dblLLIn.Y = Dat_Points.Recordset(name_YLatFrom)
963     Call gctp_DConvertDecDegToUTM(dblLLIn, DblUTMOut, 14&)
964     Dat_Points.Recordset.Edit
965     Dat_Points.Recordset(name_UTMX_Out) = DblUTMOut.X
966     Dat_Points.Recordset(name_UTMY_Out) = DblUTMOut.Y
967     Dat_Points.Recordset.Update
968     Dat_Points.Recordset.MoveNext
969 Next nR
970
971 End Sub
972
973 Sub Form_Load ()
974     CR = Chr$(13)
975     TabChar = Chr$(9)
976     CRLF = Chr$(13) & Chr$(10)
977     ReDim DataPts(1 To 1)
978
979     'Setup the proper grid widths in the coordinate grids
980     '   grdLongLat.ColWidth(0) = 1950: grdLongLat.ColWidth(1) =
1950
981     '   grdUTM14.ColWidth(0) = 1950: grdUTM14.ColWidth(1) = 1950
982
983
984 End Sub
985
986 Sub GripeNoDatabase ()
987     Call cmdMANUAL_Click(0)
988     cmdManual = True
989     Beep
990     MsgBox "You cannot use this program to project coordinates
stored in a database until you have opened a database."
991     SendKeys "%F"
992     SendKeys "o"
993 End Sub
994
995 Sub ListQrysTblsInComboBox (DBIn As Database, ComboOut As
ComboBox)
996     ' this subroutine lists all queries and tables in a database
997     '   in a combobox, for later user selection.
998     ComboOut.Clear
999     Dim DB_SYSTEMOBJECT As Long
```

```
1000 |         DB_SYSTEMOBJECT = &H80000002
1001 |
1002 |         Dim snpQTLList As Snapshot 'declared locally so it goes away
1003 |         Set snpQTLList = DBIn.ListTables()
1004 |
1005 |         Dim QTName As String
1006 |         snpQTLList.MoveFirst
1007 |         Do Until snpQTLList.EOF 'For each Table and query
1008 |             If Not ((snpQTLList("Attributes") And DB_SYSTEMOBJECT) <>
1009 | 0) And (snpQTLList("Attributes") < 6) Then
1010 |                 'Exclude system objects from list of tables and
1011 |                 queries presented to user
1012 |                 QTName = snpQTLList("Name")
1013 |                 ComboOut.AddItem QTName
1014 |             End If
1015 |             snpQTLList.MoveNext
1016 |         Loop
1017 |         snpQTLList.Close
1018 |         Set snpQTLList = Nothing
1019 | End Sub
1020 |
1021 | Sub meCopy_Click ()
1022 |     clipboard.Clear
1023 |     Dim strClipOut As String
1024 |
1025 |     If (activecontrol Is txtLongDecDeg) Or (activecontrol Is
1026 | txtLatDecDeg) Then
1027 |         strClipOut = txtLongDecDeg & TabChar & txtLatDecDeg & CRLF
1028 |     End If
1029 |
1030 |     If (activecontrol Is txtLongDMS) Or (activecontrol Is txtLatDMS)
1031 | Then
1032 |         strClipOut = txtLongDMS & TabChar & txtLatDMS & CRLF
1033 |     End If
1034 |
1035 |     If (activecontrol Is txtUTMEasting) Or (activecontrol Is
1036 | txtUTMNorthing) Then
1037 |         strClipOut = txtUTMEasting & TabChar & txtUTMNorthing & CRLF
1038 |     End If
1039 |
1040 |     clipboard.SetText strClipOut
1041 |
1042 | End Sub
1043 |
1044 | Sub mePaste_Click ()
1045 |     If TypeOf activecontrol Is TextBox Then
1046 |         'do nothing
1047 |     Else
1048 |         Exit Sub
1049 |     End If
1050 |
1051 | End Sub
1052 |
1053 | Dim strClipIn As String, str1 As String, str2 As String, T1 As
1054 | Long, T2 As Long
1055 |
1056 | strClipIn = clipboard.GetText()
1057 | T1 = InStr(strClipIn, TabChar)
1058 | If T1 = 0 Then
1059 |     activecontrol = strClipIn
```

```
1052 |         Exit Sub
1053 |     End If
1054 |     str1 = Left$(strClipIn, T1 - 1)
1055 |     T2 = InStr(strClipIn, CRLF)
1056 |     If T2 = 0 Then T2 = InStr(strClipIn, Chr$(10))
1057 |     If T2 = 0 Then T2 = InStr(strClipIn, Chr$(13))
1058 |     If T2 = 0 Then T2 = Len(strClipIn) + 1
1059 |     str2 = Mid$(strClipIn, T1 + 1, T2 - T1 - 1)
1060 |
1061 | If (activecontrol Is txtLongDecDeg) Or (activecontrol Is
    txtLatDecDeg) Then
1062 |     txtLongDecDeg = str1
1063 |     txtLatDecDeg = str2
1064 | End If
1065 |
1066 | If (activecontrol Is txtLongDMS) Or (activecontrol Is txtLatDMS)
    Then
1067 |     txtLongDMS = str1
1068 |     txtLatDMS = str2
1069 | End If
1070 |
1071 | If (activecontrol Is txtUTMEasting) Or (activecontrol Is
    txtUTMNorthing) Then
1072 |     txtUTMEasting = str1
1073 |     txtUTMNorthing = str2
1074 | End If
1075 |
1076 |
1077 | End Sub
1078 |
1079 | Sub mfCloseDatabase_Click ()
1080 |     On Error Resume Next
1081 |     DB.Close
1082 |     DBName = ""
1083 |     Me.Caption = "LLUTM14_ UTM<=>Longitude-Latitude Projection "
    & "NO DATABASE OPEN"
1084 |     Set DB = Nothing
1085 |     Dat_Points.Recordset.Close
1086 |     Dat_BLOBS.Recordset.Close
1087 |     cmbQTList_BLOBS.Clear
1088 |     cmbQTList_Points.Clear
1089 |     cmbXY_LongLat_BLOBS.Clear
1090 |     cmbXY_LongLat_Points(X_INDEX).Clear :
    cmbXY_LongLat_Points(Y_INDEX).Clear
1091 |     cmbXY_UTM14_BLOBS.Clear
1092 |     cmbXY_UTM14_Points(X_INDEX).Clear :
    cmbXY_UTM14_Points(Y_INDEX).Clear
1093 |
1094 |
1095 |
1096 | End Sub
1097 |
1098 | Sub mfExit_Click ()
1099 |     End
1100 | End Sub
1101 |
1102 | Sub mfOpenDB_Click ()
1103 | On Error GoTo ERRmfOpenDB
```

```
1104 | CMDialog1.DialogTitle = "Database to Open"
1105 | CMDialog1.Filter = "MS Access Databases|*.mdb"
1106 | CMDialog1.Flags = OFN_FILEMUSTEXIST
1107 | CMDialog1.Action = 1
1108 | CMDialog1.CancelError = True
1109 | Dim Temp As String
1110 |
1111 | Temp = CMDialog1.FileName
1112 | If Len(Temp) = 0 Then 'the user did not enter a file name
1113 |     Else
1114 |         DBName = Temp
1115 |         ' Open an MSAccess database for non-exclusive, read-write
access.
1116 |         Set DB = OpenDatabase(DBName, False, False, "") 'Open the
desired database
1117 |         Dat_Points.DatabaseName = DBName
1118 |         Me.Caption = "LLUTM14_ UTM<=>Longitude-Latitude Projection
" & DBName
1119 |         Dat_BLOBS.DatabaseName = DBName
1120 |         Call ListQrysTblsInComboBox(DB, cmbQTList_BLOBS)
1121 |         Call ListQrysTblsInComboBox(DB, cmbQTList_Points)
1122 |         If cmdPoints.Value = True Then
1123 |             cmbQTList_Points.SetFocus
1124 |             SendKeys "{DOWN}"
1125 |         End If
1126 |         If cmdBLOBS.Value = True Then
1127 |             cmbQTList_BLOBS.SetFocus
1128 |             SendKeys "{DOWN}"
1129 |         End If
1130 |
1131 | End If
1132 | Exit Sub
1133 | ERRmfOpenDB:
1134 |     Select Case Err
1135 |         Case Else
1136 |             Exit Sub
1137 |             Resume
1138 |     End Select
1139 | End Sub
1140 |
1141 | Sub SetQuery_BLOBS (strTblQrySQL As String)
1142 | 'This subroutine sets the data control to the desired Table,
Query, or SQL string,
1143 | ' then sets the labels for the attribute grid.
1144 | On Error GoTo ERRSetQuery_BLOBS
1145 | Set dnsCurrentQT = DB.CreateDynaset(strTblQrySQL)
1146 | nFields = dnsCurrentQT.Fields.Count
1147 | dnsCurrentQT.Close
1148 | Set dnsCurrentQT = Nothing
1149 | Dat_BLOBS.RecordSource = strTblQrySQL
1150 |     'nFields = Dat_BLOBS.Recordset.Fields.Count
1151 | Dat_BLOBS.Refresh
1152 | intCoordinateField = 0
1153 | Dim fieldName As String
1154 | nFields = Dat_BLOBS.Recordset.Fields.Count
1155 |
1156 | cmbXY_LongLat_BLOBS.Clear : cmbXY_UTM14_BLOBS.Clear
1157 | For n = 0 To nFields - 1
```

```
1158 |         If (Dat_BLOBS.Recordset.Fields(n).Type = DB_LONGBINARY) And
      |         (InStr(UCCase(Dat_BLOBS.Recordset.Fields(n).Name), "LAT") <> 0)
      |     Then ' OLE field with Longitude Latitude coordinates
1159 |         '
1160 |             cmbXY_LongLat_BLOBS.AddItem
      |         Dat_BLOBS.Recordset.Fields(n).Name
1161 |         End If
1162 |         If (Dat_BLOBS.Recordset.Fields(n).Type = DB_LONGBINARY) And
      |         (InStr(UCCase(Dat_BLOBS.Recordset.Fields(n).Name), "UTM") <> 0)
      |     Then '
1163 |         '
1164 |             cmbXY_UTM14_BLOBS.AddItem
      |         Dat_BLOBS.Recordset.Fields(n).Name
1165 |         End If
1166 |
1167 |     Next n
1168 |     If cmbXY_UTM14_BLOBS.ListCount = 0 And
      |     cmbXY_LongLat_BLOBS.ListCount = 0 Then
1169 |         Beep
1170 |         MsgBox "That query or table does not contain a BLOB field by a
      |         name this program recognizes. Please choose another table/query
      |         or modify field names to include UTM or LatLong as appropriate."
1171 |         Exit Sub
1172 |     End If
1173 |
1174 |     If cmbXY_LongLat_BLOBS.ListCount > 0 Then
1175 |         cmbXY_LongLat_BLOBS.ListIndex = 0
1176 |     Else
1177 |         cmbXY_LongLat_BLOBS.Text = "XYBLOB_LongitudeLatitude"
1178 |     End If
1179 |
1180 |     If cmbXY_UTM14_BLOBS.ListCount > 0 Then
1181 |         cmbXY_UTM14_BLOBS.ListIndex = 0
1182 |     Else
1183 |         cmbXY_UTM14_BLOBS.Text = "XYBLOB_UTM14"
1184 |     End If
1185 |
1186 |     Exit Sub
1187 |
1188 | ERRSetQuery_BLOBS:
1189 | If Err = 3061 Then
1190 |     MsgBox "You tried selecting a parameter query, and this
      |     application doesn't do those. Sorry!"
1191 |     Err = 0
1192 |     Exit Sub
1193 | End If
1194 |
1195 | MsgBox "Error number " & Err & " occurred in SetQuery_BLOBS." &
      | CRLF & Error
1196 | Err = 0
1197 | Exit Sub
1198 | Resume
1199 | 'Dat_BLOBS.Recordset.MoveFirst
1200 | 'Eliminated the above line so empty tables wouldn't
1201 | ' cause it to freak out.
1202 | End Sub
1203 |
1204 | Sub SetQuery_Points (strTblQrySQL As String)
```

```
1205 | 'This subroutine sets the data control to the desired TABLE,  
      | QUERY, or SQL string,  
1206 | ' then sets the labels for the attribute grid.  
1207 | On Error GoTo ERRsetQuery  
1208 | Set dnsCurrentQT = DB.CreateDynaset(strTblQrySQL)  
1209 | nFields = dnsCurrentQT.Fields.Count  
1210 | dnsCurrentQT.Close  
1211 | Set dnsCurrentQT = Nothing  
1212 | Dat_Points.RecordSource = strTblQrySQL  
1213 | 'nFields = Dat_Points.Recordset.Fields.Count  
1214 | Dat_Points.Refresh  
1215 | intCoordinateField = 0  
1216 | Dim fieldName As String, uFieldName As String  
1217 | nFields = Dat_Points.Recordset.Fields.Count  
1218 | Dim intHasX As Integer, intHasY As Integer, intHasUTM As  
      | Integer, intHasLL As Integer  
1219 | cmbXY_LongLat_Points(X_INDEX).Clear :  
      | cmbXY_LongLat_Points(Y_INDEX).Clear  
1220 | cmbXY_UTM14_Points(X_INDEX).Clear :  
      | cmbXY_UTM14_Points(Y_INDEX).Clear  
  
1221 |  
1222 | For n = 0 To nFields - 1  
1223 |     If ((Dat_Points.Recordset.Fields(n).Type = DB_SINGLE) Or  
      | (Dat_Points.Recordset.Fields(n).Type = DB_DOUBLE)) Then '   
      | field might contain coordinates  
1224 |         intHasX = False: intHasY = False: intHasUTM = False:  
      | intHasLL = False  
1225 |         fieldName = Dat_Points.Recordset.Fields(n).Name:  
      | uFieldName = UCase$(fieldName)  
  
1226 |  
1227 |             intHasX = intHasX Or InStr(uFieldName, "X_") <> 0  
1228 |             intHasX = intHasX Or InStr(uFieldName, "EAST") <> 0  
1229 |  
1230 |             intHasY = intHasY Or InStr(uFieldName, "Y_") <> 0  
1231 |             intHasY = intHasY Or InStr(uFieldName, "NORT") <> 0  
1232 |  
1233 |  
1234 |             intHasUTM = InStr(uFieldName, "UTM") <> 0  
1235 |  
1236 |             intHasLL = intHasLL Or InStr(uFieldName, "ITUDE") <> 0  
1237 |             intHasLL = intHasLL Or InStr(uFieldName, "LL") <> 0  
1238 |             intHasLL = intHasLL Or InStr(uFieldName, "GEO") <> 0  
1239 |             intHasLL = intHasLL Or InStr(uFieldName, "LON") <> 0  
1240 |             intHasLL = intHasLL Or InStr(uFieldName, "LAT") <> 0  
1241 |  
1242 |             If (intHasX And intHasUTM) Then  
      | cmbXY_UTM14_Points(X_INDEX).AddItem fieldName  
1243 |             If (intHasY And intHasUTM) Then  
      | cmbXY_UTM14_Points(Y_INDEX).AddItem fieldName  
  
1244 |  
1245 |             If (intHasX And intHasLL) Then  
      | cmbXY_LongLat_Points(X_INDEX).AddItem fieldName  
1246 |             If (intHasY And intHasLL) Then  
      | cmbXY_LongLat_Points(Y_INDEX).AddItem fieldName  
  
1247 |  
1248 |         End If  
1249 |  
1250 | Next n
```

```
1251 |
1252 |
1253 | If cmbXY_UTM14_Points(X_INDEX).ListCount = 0 And
      | cmbXY_LongLat_Points(X_INDEX).ListCount = 0 Then
1254 |     Beep
1255 |     MsgBox "That query or table does not contain any suitable
      | coordinate fields. Please choose another table/query or modify
      | the names to include UTM or LatLong as appropriate."
1256 |     Exit Sub
1257 | End If
1258 |
1259 | If cmbXY_LongLat_Points(X_INDEX).ListCount > 0 Then
1260 |     cmbXY_LongLat_Points(X_INDEX).ListIndex = 0
1261 |     Else
1262 |         cmbXY_LongLat_Points(X_INDEX).Text = "X_Longitude"
1263 |     End If
1264 | If cmbXY_LongLat_Points(Y_INDEX).ListCount > 0 Then
1265 |     cmbXY_LongLat_Points(Y_INDEX).ListIndex = 0
1266 |     Else
1267 |         cmbXY_LongLat_Points(Y_INDEX).Text = "Y_Latitude"
1268 |     End If
1269 |
1270 |
1271 | If cmbXY_UTM14_Points(X_INDEX).ListCount > 0 Then
1272 |     cmbXY_UTM14_Points(X_INDEX).ListIndex = 0
1273 |     Else
1274 |         cmbXY_UTM14_Points(X_INDEX).Text = "X_Easting_UTM14"
1275 |     End If
1276 | If cmbXY_UTM14_Points(Y_INDEX).ListCount > 0 Then
1277 |     cmbXY_UTM14_Points(Y_INDEX).ListIndex = 0
1278 |     Else
1279 |         cmbXY_UTM14_Points(Y_INDEX).Text = "Y_Northing_UTM14"
1280 |     End If
1281 |
1282 |
1283 | Exit Sub
1284 |
1285 | ERRsetQuery:
1286 | If Err = 3061 Then
1287 |     MsgBox "You tried selecting a parameter query, and this
      | application doesn't do those. Sorry!"
1288 |     Err = 0
1289 |     Exit Sub
1290 | End If
1291 |
1292 | MsgBox "Error number " & Err & " occurred in SetQuery." & CRLF &
      | Error
1293 | Err = 0
1294 | Exit Sub
1295 | Resume
1296 | 'Dat_Points.Recordset.MoveFirst
1297 | 'Eliminated the above line so empty tables wouldn't
1298 | ' cause it to freak out.
1299 | End Sub
1300 |
1301 | Sub txtJunk_GotFocus ()
1302 |     cmdUTMFromLongLat_Manual.SetFocus
1303 | End Sub
```

```
1304 |
1305 | Sub txtLatDecDeg_KeyPress (KeyAscii As Integer)
1306 | Select Case KeyAscii
1307 |     Case 13 'Return
1308 |         'Get the previous version in decimal degrees, and use it
           | to update the DMS version
1309 |         txtLatDMS = DecDegToDMS(Val(txtLatDecDeg.Text))
1310 |         KeyAscii = 0
1311 |     Case 27 'Escape
1312 |         'Convert to decimal degrees
1313 |         txtLatDecDeg = Format(DMSToDecDeg(txtLatDMS.Text))
1314 |         KeyAscii = 0
1315 |     Case Else
1316 | End Select
1317 | End Sub
1318 |
1319 | Sub txtLatDMS_KeyPress (KeyAscii As Integer)
1320 | Select Case KeyAscii
1321 |     Case 13 'Return
1322 |         'Convert to decimal degrees
1323 |         txtLatDecDeg = Format(DMSToDecDeg(txtLatDMS.Text))
1324 |         KeyAscii = 0
1325 |     Case 27 'Escape
1326 |         'Get the previous version in decimal degrees, and use it
           | to
1327 |         ' update the DMS version
1328 |         txtLatDMS = DecDegToDMS(Val(txtLatDecDeg.Text))
1329 |         KeyAscii = 0
1330 |     Case Else
1331 | End Select
1332 |
1333 | End Sub
1334 |
1335 | Sub txtLongDecDeg_KeyPress (KeyAscii As Integer)
1336 | Select Case KeyAscii
1337 |     Case 13 'Return
1338 |         'Get the previous version in decimal degrees, and use it
           | to
1339 |         ' update the DMS version
1340 |         txtLongDMS = DecDegToDMS(Val(txtLongDecDeg.Text))
1341 |         KeyAscii = 0
1342 |     Case 27 'Escape
1343 |         'Convert to decimal degrees
1344 |         txtLongDecDeg = Format(DMSToDecDeg(txtLongDMS.Text))
1345 |         KeyAscii = 0
1346 |     Case Else
1347 | End Select
1348 |
1349 | End Sub
1350 |
1351 | Sub txtLongDMS_KeyPress (KeyAscii As Integer)
1352 | Select Case KeyAscii
1353 |     Case 13 'Return
1354 |         'Convert to decimal degrees
1355 |         txtLongDecDeg = Format(DMSToDecDeg(txtLongDMS.Text))
1356 |         KeyAscii = 0
1357 |     Case 27 'Escape
1358 |         'Get the previous version in decimal degrees, and use it
```

```
1359 |         to
1360 |             ' update the DMS version
1361 |             txtLongDMS = DecDegToDMS(Val(txtLongDecDeg.Text))
1362 |             KeyAscii = 0
1363 |         Case Else
1364 |     End Select
1365 | End Sub
```

```
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
```

---

```
265 | '*          = 0 , SUCCESSFUL TRANSFORMATION.  
      | GCTP0248  
266 | '*          = I , UNSUCCESSFUL TRANSFORMATION. ERROR CODE  
      | = I.      GCTP0249  
267 | '*  
      | GCTP0250  
268 | '          IMPLICIT REAL*8 (A-H,O-Z)  
269 |  
270 | End Sub  
271 |
```

```
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)
```

```
59 |     LSet RlPt = S8
60 |     DataPts(n) = RlPt
61 |     NPos = NPos + 8
62 | Next n
63 | End Sub
64 |
65 | Sub BLOB2XYPairsInSnglArray (BLOBin As String, sngXYData() As
    | Single)
66 | ' Take XY Chains stored in a WHEAT format XYBLOB and put them
    | into
67 | ' a one-based, two-dimensional array.
68 | ' sngXYData() must be a dimensionable array
69 | ' Point nPt will be X=sngXYData(nPt,1), Y=sngXYData(nPt,1)
70 |
71 |
72 | Dim n As Long, nPoints As Long, NPos As Long
73 | nPoints = Len(BLOBin) / 8 '8 Bytes to a RealPoint
74 |
75 | ReDim sngXYData(1 To nPoints, 1 To 2)
76 | NPos = 1 'NPos will keep track of where in the BLOB we are
77 | For n = 1 To nPoints
78 |     S8.f = Mid$(BLOBin, NPos, 8)
79 |     LSet RlPt = S8
80 |     sngXYData(n, 1) = RlPt.X
81 |     sngXYData(n, 2) = RlPt.Y
82 |     NPos = NPos + 8
83 | Next n
84 |
85 | End Sub
86 |
87 | Function RealPoints2BLOB (DataPts() As RealPoint) As String
88 | Dim n As Long
89 | Dim BigTemp As String
90 | BigTemp = ""
91 | For n = LBound(DataPts) To UBound(DataPts)
92 |     RlPt = DataPts(n)
93 |     LSet S8 = RlPt
94 |     BigTemp = BigTemp & S8.f
95 | Next n
96 | RealPoints2BLOB = BigTemp
97 | End Function
98 |
99 | Function XYPairsInSnglArray2BLOB (sngXYData() As Single) As
    | String
100 | ' Take XY Chains stored in a two-dimensional real*4 array and
    | put them
101 | ' into aWHEAT format XYBLOB
102 | ' Point nPt will be X=sngXYData(nPt,X_INDEX),
    | Y=sngXYData(nPt,Y_INDEX)
103 | Dim X_INDEX As Long, Y_INDEX As Long, nPt As Long
104 | Dim nFirst As Long, nLast As Long
105 | nFirst = LBound(sngXYData, 1): nLast = UBound(sngXYData, 1)
106 | X_INDEX = LBound(sngXYData, 2): Y_INDEX = UBound(sngXYData, 2)
107 | If Y_INDEX - X_INDEX > 1 Then Exit Function
108 | Dim BigTemp As String
109 | BigTemp = ""
110 | For nPt = nFirst To nLast
111 |     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 |
```

## **WHEAT OPEN-FILE REPORTS**

**dir of9512**

**dir of9511**

**dir of9515**

**dir of9514**

**dir of9612**

**dir of9510**

**dir of9513**