

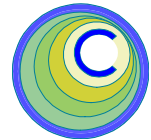
# The CEDRA Corporation's COMMAND OF THE MONTH

A monthly information bulletin

December 2006

FEATURED COMMAND

Creating Polylines by  
Connecting Points



## Application Description

Many times users will have the need to create lines or polylines by connecting points. These points can be in the form of vertex coordinates or actual point features.

For example, a GPS recorder can be used to create an ASCII based file of coordinates representing a single or multiple paths of some sort. The user would then like to import these coordinates and generate a polyline or series of polylines, without having to manually connect the points.

Another example is, rather than having an ASCII based file, a shapefile or a CAD drawing is available containing various point features. The task then is to create lines by connecting those points sharing a common attribute value.

These are typical examples of where lines need to be created using point information.

## The CEDRA Solution

To address the application of importing an ASCII based point file to mass create features, the [Import Points] command, within the {CEDRA Commands} combo-box can be employed.

To address the application of connecting existing point features which share a common attribute value, the [Connect Points] command within the {CEDRA-Polygon-Tools} toolbar can be used. Note that this functionality is only available for ArcGIS® users.



Figure 1 - CEDRA-AVcad-Menus Toolbar for ArcGIS Users

To utilize the [Import Points] command, ArcGIS® users should activate the CEDRA-AVcad-Menus toolbar to display the toolbar shown in Figure 1. Shown in Figure 2 is the {CEDRA Commands} combo-box. The [Import Points] command appears towards the bottom of the drop-down list.

Shown in Figure 3 is the {CEDRA-Polygon-Tools} toolbar. The [Connect Points] command appears towards the middle of the drop-down list.

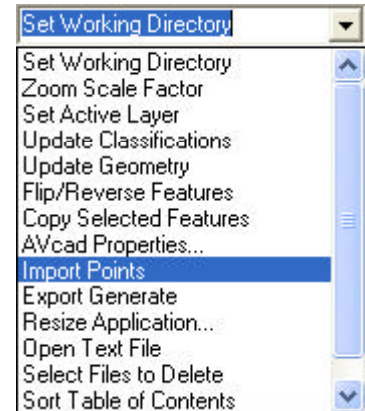


Figure 2  
{CEDRA Commands} Combo-Box

### Command Of The Month bulletin

*This month's issue discusses how polylines can be created by connecting points in a variety of formats.*

## Import Points - Overview

The intent of the [Import Points] command is to provide the user the ability to import an ASCII based file in a variety of file formats to create point, polyline, polygon and annotation features.

The command presents the user with a number of supported file formats, from which, the user selects the desired format. Depending upon the format, the command generates the appropriate types of features, storing the features in either a shapefile or in a personal

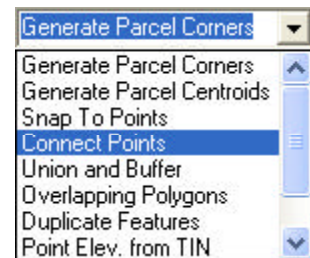
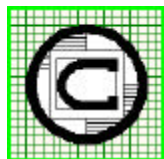


Figure 3  
{CEDRA-Polygon-Tools} Toolbar

geodatabase (PGD) depending upon which file format is selected by the user.

## Import Points - Operation

Note that the discussion provided below is intended for ArcGIS users. The operation is very similar for ArcView® GIS users with only some minor variations.



**The CEDRA Corporation**

**Total CADD for Engineers™  
Bridging Engineering and GIS™**

Phone: 585-232-6998  
Fax: 585-262-2042

151 Sully's Trail - Suite 6  
Pittsford, New York 14534

E-mail: [cedra@cedra.com](mailto:cedra@cedra.com)  
URL: <http://www.cedra.com>

Copyright © 2006 The CEDRA Corporation. The CEDRA logos, CEDRA-AVseries, CEDRA-AVcad, CEDRA-AVcogo, CEDRA-AVparcel, CEDRA-AVland, CEDRA-AVsand, CEDRA-AVwater, CEDRA-DataEditor and CEDRA-DxfExport are trademarks of The CEDRA Corporation. ArcView® GIS and ArcGIS are registered trademarks of Environmental Systems Research Institute, Inc. All World Rights Reserved.

1. Prior to invoking the [Import Points] command, the user should set the current working directory using the [Set Working Directory] command, see Figure 2.

All new data created by the [Import Points] command is stored in the current working directory. Note that it is not necessary for the ASCII based file to reside in the current working directory.

Once the working directory has been set and the [Import Points] command activated, perform the following :

2. The file specification dialog box is initially displayed, see Figure 4, at which point, click the Browse button to display the conventional file navigation dialog box.

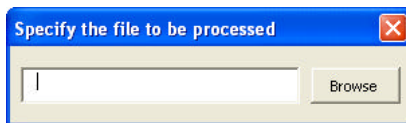


Figure 4  
File Specification Dialog Box

3. As can be seen from Figure 5, the user is able to navigate to the appropriate folder and select the desired file to be imported. Once the appropriate file has been selected, click at the:

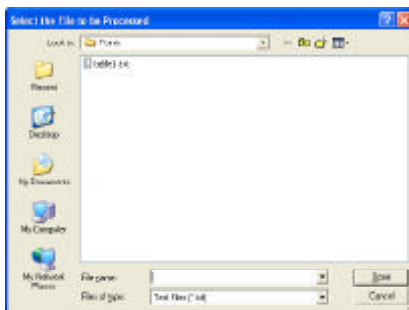


Figure 5  
File Navigation Dialog Box

- *Cancel* button to terminate the command, or the
- *Open* button to continue with the processing.

4. The [Import Points] multi-input dialog box of Figure 6 now appears.

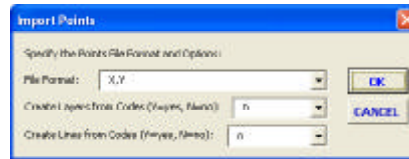


Figure 6  
File Format Specification Dialog Box

Upon inspection of the multi-input dialog box note the following:

- The first data line item enables the user to specify the file format that the ASCII based file adheres to. The user simply scrolls down in the list, see Figure 7, and selects the appropriate file format.

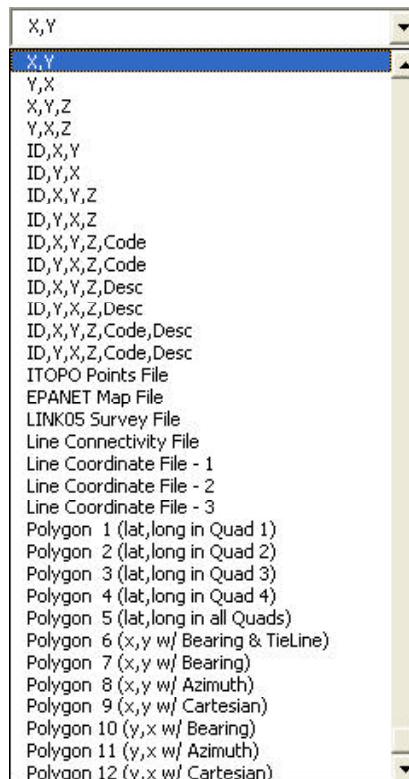


Figure 7  
Supported File Formats

- The second data line item enables the user to specify whether or not separate layers are to be generated for those formats which contain the Code parameter, such as the ID,X,Y,Z,Code file format.

Selecting the *y* option indicates that each unique value of the Code parameter is to appear in its own layer

or theme, rather than all of the points appearing in the same layer.

- The third data line item enables the user to specify whether or not two-point line features are to be generated for those formats which contain the Code parameter, such as the ID,X,Y,Z,Code file format.

Selecting the *y* option indicates that line features are to be created connecting common values of the Code parameter, in the order in which they appear in the ASCII based file. In so doing, the user is able to mass generate line features using the Code parameter values in the file.

Note that in connecting points with common values of the Code parameter, the configuration file *ptcode.lis* is taken into consideration. This file is located in the CEDRA distribution directory and is discussed later on in this bulletin.

Once the appropriate information has been specified in Figure 6, click at the:

- *Cancel* button to terminate the command, or the
- *OK* button to continue with the processing.

If the *OK* button is selected and depending upon the file format which was selected, the user may or may not be asked for additional information.

The file formats presented in Figure 7 are grouped in terms of functionality provided and are discussed below.

## L Point File Formats

Shown in Figure 8 are the various Point File Formats.

Following the selection of the *OK* button in Step 3, the command will begin to process the file.

Note that the responses given by the user for the second and third data line items of Figure 6, will effect how the

```
X,Y
Y,X
X,Y,Z
Y,X,Z
ID,X,Y
ID,Y,X
ID,X,Y,Z
ID,Y,X,Z
ID,X,Y,Z,Code
ID,Y,X,Z,Code
ID,X,Y,Z,Desc
ID,Y,X,Z,Desc
ID,X,Y,Z,Code,Desc
ID,Y,X,Z,Code,Desc
```

Figure 8  
Point File Formats

command continues with its processing for the file formats containing the Code parameter. As such, there are five possible scenarios or conditions.

A. *Formats w/o the Code parameter*

Figure 9 contains a sample ID,Y,X,Z file which does not contain the Code parameter and which uses the comma character to delineate the various items.

```
1,9745.353,42689.798,997.43
2,9745.399,42687.915,997.65
3,9801.666,42696.258,995.27
4,9801.868,42730.070,995.09
5,9801.007,42736.335,995.88
6,0056.142,42741.436,990.97
7,0320.714,42817.423,974.09
8,0299.676,42843.865,973.61
9,0295.811,42847.701,975.31
```

Figure 9  
Sample ID,Y,X,Z Points File

The point features that are created are stored in a layer (theme) whose name will match the root name of the file being processed. For example, if the user selects a file called L\_0.txt, the points that are created will be stored in a theme called L\_0, provided that L\_0 exists and is a point theme.

If L\_0 does not exist or it does exist but is not a point theme, the user will be prompted to specify the name of the shapefile or personal geodatabase in which the points are to be stored, see Figure 10.

Note that if a personal geodatabase is specified, the user must set the New

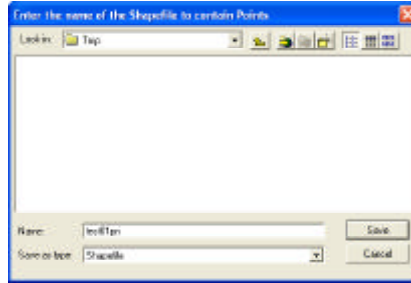


Figure 10  
Shapefile / PGD Dialog Box

Layer's False X, False Y and False XY values, prior to invoking the [Import Points] command, using the [Set Active Layer] command, otherwise, a coordinate out of bounds error message may appear.

- Once the name of the shapefile or personal geodatabase has been specified, click at the:
  - Cancel button to terminate the command, or the
  - Save button to create the new shapefile or personal geodatabase.

Note, if a personal geodatabase is to be specified, the user should set the *Save as type:* parameter, in Figure 10, to Personal GeoDatabase and enter in the data field to the right of the *Name:* label, the name of the layer and the PGD, for example: *L\_Open L\_0.mdb*. The points which are created will then appear in the *L\_Open* layer. At least one space or blank character must separate the name of the layer and the name of the PGD.

- As the command reads the data in the ASCII based file, a progress bar display, in the status bar area, will appear indicating the percent

```
1007,249745.353,2142689.798,1097.435,WV11,Valve and box
1008,249745.399,2142687.915,1097.650,FH11,Fire hydrant
1009,249801.666,2142696.258,1095.278,C411,Curb inlet
1010,249801.868,2142730.070,1095.097,C211,Curb inlet
1011,249801.007,2142736.335,1095.882,SM11,Manhole
1012,250056.142,2142741.436,1090.970,SM11,Manhole
1013,250320.714,2142817.423,1074.097,C411,Curb inlet
1014,250299.676,2142843.865,1073.619,C211,Curb inlet
1015,250295.811,2142847.701,1075.319,SM11,Manhole
```

Figure 12 - Sample ID,Y,X,Z,Code,Desc Points File

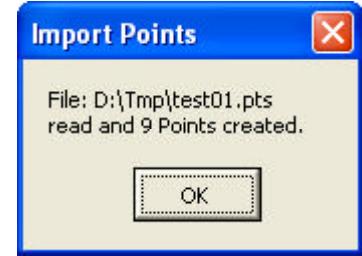


Figure 11  
Processing Results Message Box

completion of the processing. Upon completion, an information message box similar to that shown in Figure 11 will be displayed, click at the:

- OK button to acknowledge the message and continue with any other desired processing.

B. *Formats with the Code parameter and Create Layers from Codes = n and Create Lines from Codes = n*

Shown in Figure 12 is a sample ID,Y,X,Z,Code,Desc file which does contain the Code parameter.

See the *Formats w/o the Code parameter* section above. The processing under this scenario is the same as that of Scenario A.

Note, when processing files with the Code parameter, the command utilizes the configuration file, *ptcode.lis*, which resides in the CEDRA distribution folder, typically the c:\cedra\fontswin folder.

The *ptcode.lis* file enables the user to map a specific point symbol to a specific Code. In so doing, the user is able to denote which Code values actually pertain to points that represent a symbol

(tree, hydrant, manhole, etc.), rather than connectivity. That is, Codes which appear in the *ptcode.lis* file are ignored when generating lines based upon a common Code value.

C. *Formats with the Code parameter and Create Layers from Codes = y and Create Lines from Codes = n*

The point features which are created under this condition are stored in layers whose names correspond to the unique values of the Code parameter. These layers, in turn, are stored as datasets within a personal geodatabase whose name matches the root name of the file being processed. For example, if the user selects a file called *L\_0.txt*, the points created are stored in a personal geodatabase called *L\_0.mdb*. Within the personal geodatabase will be the datasets corresponding to the various unique values of the Code parameter.

Should a value for the Code parameter contain invalid characters, such as a blank or space character or the following characters: \$, #, >, -, ., the invalid character will be replaced with the underscore, \_ character.

If a point should not be assigned a Code value, the point will be stored in a layer called *NoCodes*, indicating that no code was assigned to the point.

Furthermore, under this scenario the command does not prompt the user for any further information. The command will create the personal geodatabase and datasets without any user interaction.

5. As the points are read in, the command will display a progress bar indicating the percent of completion. Upon completion of its processing, an information message box indicating the number of points which have been read in, see Figure 11 will be displayed, click at the:
  - *OK* button to acknowledge the message and continue with any other desired processing.

D. *Formats with the Code parameter and Create Layers from Codes = y and Create Lines from Codes = y*

This scenario operates as Scenario C with the exception that an additional dataset is created containing the line features which are established as a result of the processing.

Under this scenario, lines are created by connecting point features which share a common Code parameter value. The connectivity sequence matches the order of the points in the ASCII based file.

Note that Codes which appear in the *ptcode.lis* file will not be used in generating line features. Codes appearing in the *ptcode.lis* file are treated as points, which denote symbology (trees, hydrants, manholes, etc.), and are not to be used in determining connectivity.

The name of the layer containing the line features will match that of the personal geodatabase with the extension characters, *ln*, appended to the name. For example, if the name of the personal geodatabase is called *L\_0.mdb*, the name of the line layer will be *L\_0ln*. Likewise, a dataset called *L\_0ln* would appear in the personal geodatabase. The *ln* extension is used to denote a line dataset or layer.

If the line layer does not exist in the map, the command will create it without any further user interaction.

5. As the points are read in, the command will display a progress bar indicating the percent of completion. Upon completion of its processing, an information message box indicating the number of points which have been read in, see Figure 11 will be displayed, click at the:
  - *OK* button to acknowledge the message and continue with any other desired processing.

Shown in Figure 13 is an example of lines, which were automatically created by connecting points based upon a common Code value.

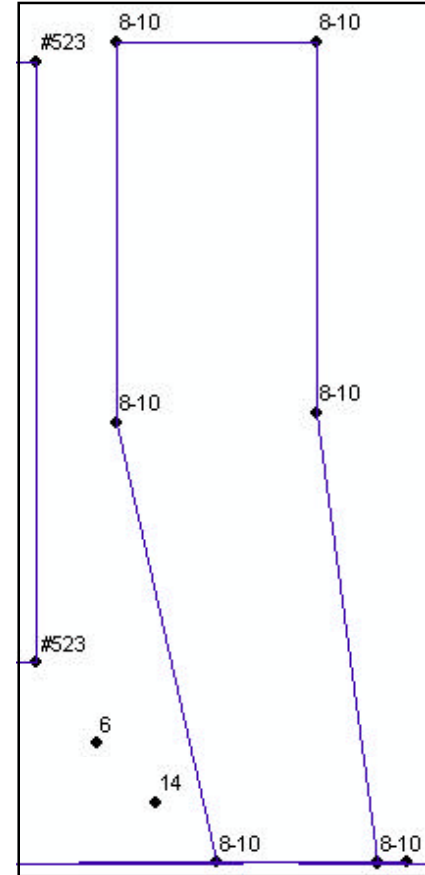


Figure 13  
Connecting Common Code Values

E. *Formats with the Code parameter and Create Layers from Codes = n and Create Lines from Codes = y*

This scenario operates as Scenario A with the exception that line features are created connecting points which share a common Code parameter value. The lines will be created once all of the points have been processed.

Note that Codes which appear in the *ptcode.lis* file will not be used in generating line features. Codes appearing in the *ptcode.lis* file are treated as points, which denote symbology (trees, hydrants, manholes, etc.), and are not to be used in determining connectivity.

5. The process for storing the lines is identical to that as described under Scenario D. That is the lines are stored in a personal geodatabase which is created by the command without user interaction and in a

215	5307.486530	5764.997409	0.000000	* 2SBB 2 S.B.B.
216	5343.858904	5829.441574	0.000000	* 2SBB 2 S.B.B.
217	5306.412762	5850.578897	0.000000	* 2SBB 2 S.B.B.
208	5027.566649	5271.950239	0.000000	* #219 BLDG #219
209	5028.686029	5303.424248	0.000000	* #219 BLDG #219
251	4960.308708	5188.067792	0.000000	* US56 HEDGE
263	4950.614511	5218.069413	0.000000	* #216 BLDG #216
264	4952.823374	5246.658147	0.000000	* #216 BLDG #216
23	4997.000000	5017.000000	0.000000	* US25 CATCH BASIN
24	4990.500000	5033.000000	0.000000	* US25 CATCH BASIN
25	4962.500000	5017.000000	0.000000	* US25 CATCH BASIN
253	4990.803190	5211.563273	0.000000	* US25 CATCH BASIN
265	4991.934934	5359.290208	0.000000	* US25 CATCH BASIN
268	4969.093895	5373.412278	0.000000	* US25 CATCH BASIN
280	5090.003014	5562.581983	0.000000	* US25 CB
287	5177.759991	5714.206375	0.000000	* US25 CB
203	5009.004000	5163.000000	0.000000	* #215 BLDG #215
204	5009.438531	5191.699582	0.000000	* #215 BLDG #215
205	5059.415559	5190.057430	0.000000	* #215 BLDG #215
206	5059.000000	5163.000000	0.000000	* #215 BLDG #215
207	5009.000000	5163.000000	0.000000	* #215 BLDG #215
96	5271.870127	5885.005375	0.000000	* WK05 LEFT SIDEWALK
97	5312.806598	5861.897857	0.000000	* WK05 LEFT SIDEWALK
98	5315.752509	5867.124867	0.000000	* WK05 LEFT SIDEWALK

Figure 14 - Sample ITOPO Points File

- layer which carries the *ln* extension.
6. Upon completion of its processing, an information message box indicating the number of points which have been read in, see Figure 11 will be displayed, click at the:
    - *OK* button to acknowledge the message and continue with any other desired processing.

Note the following regarding the point file formats shown in Figure 8.

- a. In order to define a valid point a data line must appear in the ASCII based file containing the appropriate number of items. Depending upon the file format selected by the user, the number of items will range from 2 to 6.
- b. A single comma, space or Tab character can be used to delineate or separate the various items on a data line.

- c. Blank lines, as well as, data lines beginning with the characters, /\*, or ; are ignored (columns 1 and 2, respectively).
- d. The parameters shown in Figure 8 represent the following point data:  
 X : x coordinate (easting)  
 Y : y coordinate (northing)  
 Z : z coordinate or elevation  
 ID : point number  
 CODE : point code  
 DESC : point description
- e. The parameters shown in Figure 8 will be stored in the following attributes (fields), if present:  
 X : X (easting)  
 Y : Y (northing)  
 Z : Z  
 ID : PNT (long integer)  
 CODE : PTCODE (maximum of 8 characters)  
 DESC : PTDESC (maximum of 40 characters)
- f. Figure 12 contains a sample file that follows the ID,Y,X,Z,Code,Desc file format with the comma character used as the delineator for the vari-

ous data line items. Processing this sample file would result in 9 point features being created.

### II. ITOPO Points File Format

Shown in Figure 14 is a sample ITOPO Points File. The syntax or format of this file is very similar to the ID,Y,X,Z,Code,Desc format with the exception that the asterisk character, \*, is used to delineate the point code from the elevation value and the space character is used to delineate all other items on a data line. The processing of this file is the same as that described for the Point File formats which include the Code parameter, that is, Scenarios B, C, D and E.

### III. EPANET Map File Format

The EPANET Map File is used to import an EPANET water distribution model. The March 2006 issue of Command of the Month describes the file format and the process of import an EPANET water model. Please refer to this issue for more information.

**IV. LINK05 Survey File Format**

Following the selection of the *OK* button in Step 3, the command will begin to process the file.

Note that the responses given by the user for the second and third data line items of Figure 6, will have no effect on how the command continues its processing for this specific file format.

The features that are created as a result of processing the LINK05 Survey File will be stored in a personal geodatabase whose name will match the root name of the file that was processed. For example if the name of the LINK05 Survey File was *L\_0.txt*, a personal geodatabase called *L\_0.mdb* would be created within the current working directory. Within the personal geodatabase would be the various datasets representing the layers that were established during the processing.

As the features are being created, the command will create the appropriate layers in the Table of Contents and display a progress bar display in the status bar area indicating the percent of completion.

Upon completion a message indicating the number of features which were created and similar to that shown below will appear in the status bar area.

1243 Points, 0 Lines, 0 Curves and 0 Text Strings created...

```

200,185512.898000,2166259.911000,1240.233000,07006XX
201,185429.281000,2166223.043000,1240.008000,07016XX
202,185459.571000,2168843.032000,1225.365000,07017XX
203,185482.964000,2171475.294000,1225.893000,07016XX
204,185518.005000,2174082.085000,1209.915000,07017XX
205,185551.887000,2176710.340000,1208.741000,07016XX
206,185428.569336,2166123.155850,1242.348839,0624024X
207,185415.271549,2166123.509550,1242.150187,R103224X
208,185404.028423,2166122.509400,1240.723463,06240XX
209,185396.054987,2166121.829320,1241.498682,S122134004
210,185440.843519,2166123.299350,1242.276456,L103224X
    
```

Figure 15 - Sample LINK05 Survey File

*LINK05 Survey File Description*

The LINK05 Survey File utilizes a sophisticated coding scheme in conjunction with a number of configuration files to mass generate point symbology and line connectivities. Shown in Figure 15 is a sample LINK05 Survey File.

The parameters shown in Figure 15 represent the following point data:

- ID : point number
- Y : y coordinate (northing)
- X : x coordinate (easting)
- Z : z coordinate or elevation
- CODE : survey code

The survey code, identified above, is comprised of five user defined sub-codes forming a single data field and which are not separated by spaces or other characters. The five sub-codes, and their required number of characters are:

- Concurrence 1 : alpha or 0
- Line 1 : numeric
- Description 3 : numeric
- Material 2 : numeric or X
- Size 3 : numeric or X

Thus, the survey code can range in length from 7 to 10 characters.

The use of these codes is as follows:

*Concurrence Code* This is an alphabetic (A-Z or a-z) character denoting that all points identified by this char-

acter are to be connected by lines and/or curves to form one continuous string of elements. Note that:

- An upper case character is different from its lower case counterpart. Thus M and m will indicate two distinct element strings.
- Spot point observations that are not to be associated with any line or curve bear the concurrence code of zero (0).
- Once a concurrence code has begun and closed (see Line Codes 1 and 5 below), it may be reused to designate another element string.

*Line Code* This code may be one of the following numbers denoting the following:

- 1 The beginning or start of a line.
- 2 A point on line (POL) to be joined to its preceding point of the same concurrence code by a straight line.
- 3 A point on curve (POC). If there is only one POC between two POL's, a curve will be drawn. If there are two or more POC's between two POL's, then a series of straight lines will be drawn between the POC's and the two POL's. If there is only one POC, and the program detects two straight lines before and ahead of the curve, then the curve will be drawn tangent to the two lines. The two adjacent POL's will not be erased.
- 4 The beginning point or a point on a spur line.
- 5 Ending point of an element string.
- 6 Spot point not connected to any line or curve.
- 7 Quality control point such as survey monument, or closure point.
- 8 Ending point or termination of a spur line. The next segment in the line will begin at the start point of the spur line.
- 9 Not used.
- 0 Close to the point of beginning of an element string bearing a unique concurrence code.

*Description Code* This is a user defined code between 000 and 999 denoting the type of the observed point and the layer into which the point and

any associated line and/or curve element string will be placed. The configuration files, *pcodes.txt* and *layers.txt*, contain the data which describes the point.

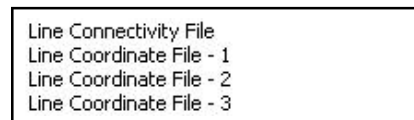
**Material Code** This is a user defined code between 00 and 99 denoting the material of the observed point or element string to be generated. If a material code is not deemed applicable to the observation, a single X can be entered as the code. The configuration file, *pcodes.txt*, contains the information describing the material code.

**Size Code** This is a user defined code between 0 and 999 denoting the size of the observation in feet (meters) or inches (mm). If a size code is not deemed applicable to the observation, a single X can be entered as the code. The configuration file, *symfac.txt*, contains the information describing the point size.

The configuration files which are used in processing a LINK05 Survey File are called *pcodes.txt*, *layers.txt*, *pscale.txt*, *symfac.txt* and *codpar.lis*. These files are located in the CEDRA distribution directory, typically c:\cedra\fontswin, and can be customized by the user as desired.

**V. Line File Formats**

Shown in Figure 16 are the various Line File Formats.



**Figure 16**  
Line File Formats

Following the selection of the OK button in Step 3, the command will begin to process the file.

Note that the responses given by the user for the second and third data line items of Figure 6, will have no effect on how the command continues its processing for these file formats.

The features which are created are stored in a layer (theme) whose name will match the root name of the file being processed. For example, if the user selects a file called L\_0.txt, the features that are created will be stored in a theme called L\_0, provided that L\_0 exists and is of the same feature type being processed.

If the layer does not exist or is not of the proper feature type, a personal geodatabase will be created using the root name of the ASCII based file as the name of the PGD. Continuing with the example presented above, a PGD called L\_0.mdb would be created if the appropriate layer (L\_0 or L\_0pn) did not exist.

The name of the layer containing the line features will match the root name of the ASCII based file with the extension characters, *ln*, appended to the name. For example, if the name of the file is L\_0.txt, the name of the line layer will be L\_0ln. Likewise, a dataset called L\_0ln would appear in the L\_0.mdb personal geodatabase. The *ln* extension is used to denote a line dataset or layer. If a point layer is to be created, the *pn* extension will be used. Missing layers are created without user input.

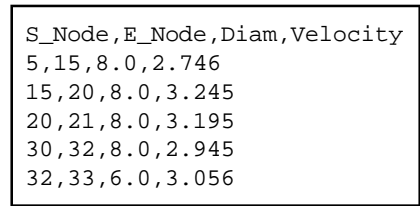
Note that if a personal geodatabase is to be created, the user must set the New Layer's False X, False Y and False XY values, prior to invoking the [Import Points] command, using the [Set Active Layer] command, otherwise, a coordinate out of bounds error message may appear.

Depending upon the specific Line File Format selected, the command operates differently. As such, there are four possible scenarios or conditions.

**A. Line Connectivity File**

Shown in Figure 17 is a sample Line Connectivity File. Note that the very first line in the file contains the names of the attributes (fields) which the enclosed data pertains to.

Using this file format it is possible to have the command generate two point



**Figure 17**  
Sample Line Connectivity File

lines based upon the connectivity of node numbers. Node numbers may be numeric or alphanumeric. Under this mode of operation a Point layer (theme), containing an attribute where the referenced node numbers are stored, must have been loaded, in the current view, prior to invoking the [Import Points] command. The ASCII based file that is processed by this command will contain references to the node numbers that are stored, under a user-specified attribute, in the Point layer (theme).

Using the root name of the file, selected by the user, the command will scan the view's Table of Contents for a visible Point layer (theme) of the same name. For example, if the user selects a file called L\_0pn.txt, the command will search for a Point layer (theme) called L\_0pn.shp. For ArcGIS users, the command will search for L\_0pn or L\_0. If no theme is found, a warning message is displayed and the command terminates. A point theme with this same root name must be present in the current view for the command to operate under this file type. It is this layer (theme) that must contain the node numbers that are referenced in the selected file.

5. If a corresponding Point layer (theme) is found, the user must specify which attribute in the Point layer (theme) contains the *node numbers* that are referenced in the Line Connectivity File. In addition, the user must specify which attributes in the Line Connectivity File correspond to the *Line Start* and *End Node* Numbers. A multi-input dialog box similar to that shown in Figure 18 will appear.

Once the appropriate information has been specified, click at the:

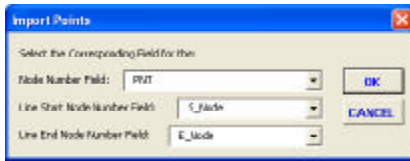


Figure 18  
Corresponding Fields or Attributes

- *Cancel* button to terminate the command, or the
  - *OK* button to continue with the processing.
6. At this point, the command will query the user as to if the attributes in the Line Connectivity File are to be included in the layer containing the line features which are created. A query dialog box similar to that shown in Figure 19 will be displayed, click at the:



Figure 19  
Query to Include Attributes

- *Cancel* button to terminate the command,
  - *No* button to not include the attributes, or the
  - *Yes* button to indicate that the attributes are to be included.
7. If the attributes are to be included, the attributes will be added, if they do not exist in the Line Layer, as text or character fields. A query dialog box similar to that shown in Figure 20 will be displayed prompting the user to enter the desired field width for the specific attribute that is missing. Once the desired value has been specified, click at the:

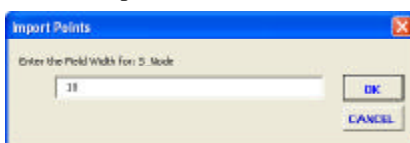


Figure 20  
Attribute or Field Width Specification

- *Cancel* button to terminate the command, or the
  - *OK* button to continue with the processing.
- Note, this query is repeated for every attribute that is missing from the Line Layer.

8. As the command reads the data in the ASCII based file, a progress bar display, in the status bar area, will appear indicating the percent completion of the processing. Upon completion, an information message box similar to that shown in Figure 21 will be displayed, click at the:

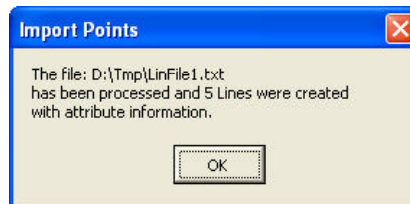


Figure 21  
Processing Results Message Box

- *OK* button to acknowledge the message and continue with any other desired processing.

#### B. Line Coordinate File - 1

Shown in Figure 22(a) is a sample Line Coordinate File - 1 file. Note that the very first line in the file contains the names of the attributes (fields) which the enclosed data pertains to. Figure 22(b) contains the image which the ASCII file represents.

```
Line_ID,Xcoor,Ycoor,Elev
1,100.0,100.0,111.0
1,200.0,100.0,111.0
2,100.0,100.0,222.0
2,100.0,200.0,222.0
2,200.0,200.0,222.0
3,500.0,500.0,333.0
3,500.0,100.0,333.0
```

Figure 22(a)  
Sample Line Coordinate File - 1

Using this file format it is possible to have the command generate polylines

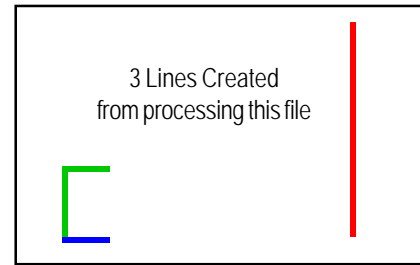


Figure 22(b)  
Line Coordinate File - 1 Image

based upon the coordinates of the vertices that comprise a line. Following this syntax, the file will contain a data line for every vertex comprising a line. All data lines associated with a single line must carry the same line identification number. It is the line identification number that indicates which vertices form a line. The vertices are connected in the same order as they appear in the file. The line identification number may be numeric or alphanumeric and the attributes within the ASCII based file can appear in any order. That is, the line identification number does not need to be the first attribute.

5. At this point, the user must specify which attributes in the Line Coordinate File represent the *line identification*, *x coordinate* and *y coordinate* parameters. A multi-input dialog box similar to that shown in Figure 23 will appear.

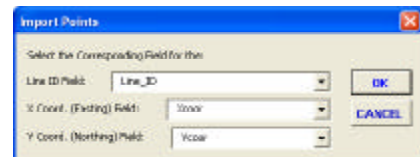


Figure 23  
Corresponding Fields or Attributes

Once the appropriate information has been specified, click at the:

- *Cancel* button to terminate the command, or the
  - *OK* button to continue with the processing.
6. At this point, the command will query the user as to if the attributes in the Line Coordinate File are to be included in the layer containing the line features which are created. A



query dialog box similar to that shown in Figure 19 will be displayed, click at the:

- *Cancel* button to terminate the command,
- *No* button to not include the attributes, or the
- *Yes* button to indicate that the attributes are to be included.

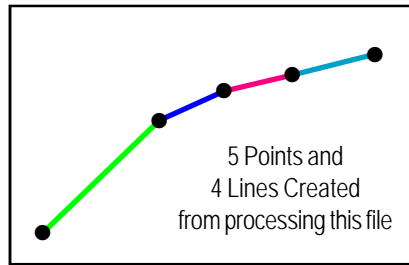


Figure 24(b)

Line Coordinate File - 2 Image

7. If the attributes are to be included, the attributes will be added, if they do not exist in the Line Layer, as text or character fields. A query dialog box similar to that shown in Figure 20 will be displayed prompting the user to enter the desired field width for the specific attribute that is missing. Once the desired value has been specified, click at the:

- *Cancel* button to terminate the command, or the
- *OK* button to continue with the processing.

Note this query is repeated for every attribute that is missing from the Line Layer.

8. As the command reads the data in the ASCII based file, a progress bar display, in the status bar area, will appear indicating the percent completion of the processing. Upon completion, an information message box similar to that shown in Figure 21 will be displayed, click at the:

- *OK* button to acknowledge the message and continue with any other desired processing.

C. Line Coordinate File - 2

Shown in Figure 24(a) is a sample Line Coordinate File - 2 file. Note that the very first line in the file contains the names of the attributes (fields) which

```
MH_Up,MH_Down,Length,Grade,Northing,Easting,Invert
3851R, 3945R, 38.32, 0.961, 5268401.110, 323850.670, 86.880
3945R, 3976R, 35.24, 0.823, 5268502.730, 323944.520, 85.500
3976R, 3987R, 11.80, 0.678, 5268518.080, 323976.250, 85.080
3987R, 4046R, 64.93, 1.555, 5268522.720, 323987.100, 84.980
4046R, 4108R, 68.32, 1.903, 5268549.880, 324046.080, 83.900
```

Figure 24(a) - Sample Line Coordinate File - 2

the enclosed data pertains to. Figure 24(b) contains the image which the ASCII file represents.

Using this file format it is possible to have the command generate two point lines based upon the coordinates of two points which comprise the line. Following this syntax, the file contains a data line for every line to be defined. On a data line, the start node number of the line appears along with its coordinates as well as the end node number of the line. Two passes are made in processing the file, the first is to read and store all of the start node numbers and their coordinates, while the second is to create the lines using the coordinates contained on a data line, as well as, the coordinates saved during the first pass. In addition to the lines that are created, the command will generate points for each start node number that is referenced in the file. The start and end node numbers may be numeric or alphanumeric.

5. At this point, the user must specify which attribute, in the file being processed, corresponds to the *Start Node Number*, *End Node Number*, *X coordinate*, *Y coordinate* and *Z coordinate* or elevation. A multi-input dialog box similar to that shown in Figure 25 will appear. Once the appropriate information has been specified, click at the:

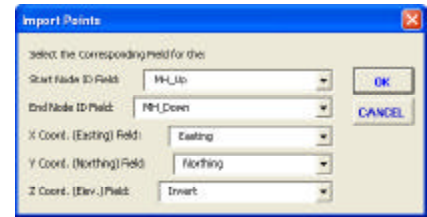


Figure 25

Corresponding Fields or Attributes

- *Cancel* button to terminate the command, or the
- *OK* button to continue with the processing.

6. At this point, the command will query the user as to if the attributes in the Line Connectivity File are to be included in the layer containing the line features which are created. A query dialog box similar to that shown in Figure 19 will be displayed, click at the:

- *Cancel* button to terminate the command,
- *No* button to not include the attributes, or the
- *Yes* button to indicate that the attributes are to be included.

7. If the attributes are to be included, the attributes will be added, if they do not exist in the Line Layer, as text or character fields. A query dialog box similar to that shown in Figure 20 will be displayed prompting the user to enter the desired field width for the specific attribute that is missing. Once the desired value has been specified, click at the:

- *Cancel* button to terminate the command, or the
- *OK* button to continue with the processing.

Note this query is repeated for every attribute that is missing from the Line Layer.

8. As the command reads the data in the ASCII based file, a progress bar display, in the status bar area, will appear indicating the percent completion of the processing. Upon completion, an information message box similar to that shown in

Figure 21 will be displayed, click at the:

- **OK** button to acknowledge the message and continue with any other desired processing.

#### D. Line Coordinate File - 3

Shown in Figure 26(a) is a sample Line Coordinate File - 3 file. Note that the very first line in the file contains the names of the attributes (fields) which the enclosed data pertains to. Figure 26(b) contains the image which the ASCII file represents.

```
Xcoord, Ycoord, Elev
100.0, 100.0, 111.0
200.0, 100.0, 111.0
300.0, 100.0, 222.0
400.0, 200.0, 222.0
500.0, 200.0, 222.0
600.0, 500.0, 333.0
700.0, 100.0, 333.0
```

Figure 26(a)  
Sample Line Coordinate File - 3

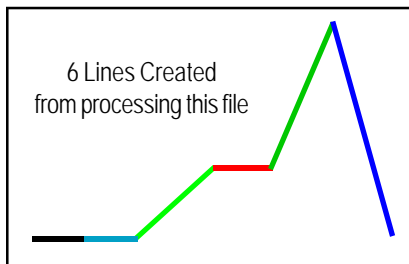


Figure 26(b)  
Line Coordinate File - 3 Image

Using this file format it is possible to have the command generate two point lines based upon the coordinates of two points which comprise a line. Following this syntax, the file contains a sequential list of points with each point appearing on an individual data line. This command processes the sequential list of points generating a line once two points have been found.

As such, data lines 1 and 2 form the first line. Data lines 2 and 3 form the second, data lines 3 and 4 the third and so forth. As such, the end point of the previous line acts as the start point of the subse-

quent line. In essence, the file contains point data representing a string, but is displayed as a series of individual lines.

5. At this point, the user must specify which attribute, in the file being processed, corresponds to the *X coordinate* and *Y coordinate*. A multi-input dialog box similar to that shown in Figure 27 will appear. Once the appropriate information has been specified, click at the:

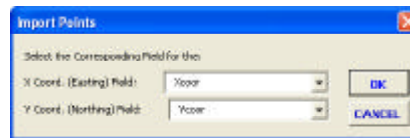


Figure 27  
Corresponding Fields or Attributes

- **Cancel** button to terminate the command, or the
  - **OK** button to continue with the processing.
6. At this point, the command will query the user as to if the attributes in the Line Coordinate File are to be included in the layer containing the line features which are created. A query dialog box similar to that shown in Figure 19 will be displayed, click at the:
    - **Cancel** button to terminate the command,
    - **No** button to not include the attributes, or the
    - **Yes** button to indicate that the attributes are to be included.

7. If the attributes are to be included, the attributes will be added, if they do not exist in the Line Layer, as text or character fields. A query dialog box similar to that shown in Figure 20 will be displayed prompting the user to enter the desired field width for the specific attribute that is missing. Once the desired value has been specified, click at the:

- **Cancel** button to terminate the command, or the
- **OK** button to continue with the processing.

Note this query is repeated for ev-

ery attribute that is missing from the Line Layer.

8. As the command reads the data in the ASCII based file, a progress bar display, in the status bar area, will appear indicating the percent completion of the processing. Upon completion, an information message box similar to that shown in Figure 21 will be displayed, click at the:

- **OK** button to acknowledge the message and continue with any other desired processing.

## VI. Polygon File Formats

Shown in Figure 28 are the various Polygon File Formats.

```
Polygon 1 (lat, long in Quad 1)
Polygon 2 (lat, long in Quad 2)
Polygon 3 (lat, long in Quad 3)
Polygon 4 (lat, long in Quad 4)
Polygon 5 (lat, long in all Quads)
Polygon 6 (x, y w/ Bearing & TieLine)
Polygon 7 (x, y w/ Bearing)
Polygon 8 (x, y w/ Azimuth)
Polygon 9 (x, y w/ Cartesian)
Polygon 10 (y, x w/ Bearing)
Polygon 11 (y, x w/ Azimuth)
Polygon 12 (y, x w/ Cartesian)
```

Figure 28  
Polygon File Formats

Following the selection of the **OK** button in Step 3, the command will begin to process the file.

Note that the responses given by the user for the second and third data line items of Figure 6, will have no effect on how the command continues its processing for these file formats.

The features which are created are stored in a layer (theme) whose name will match the root name of the file being processed. For example, if the user selects a file called L\_0.txt, the features that are created will be stored in a theme called L\_0, provided that L\_0 exists and is of the same feature type being stored.

If L\_0 does not exist or it does exist but is not of the proper feature type, the user will be prompted to specify the name of

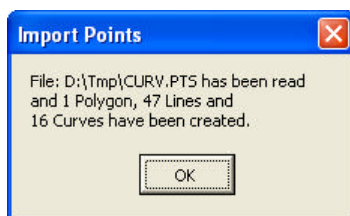
the shapefile or personal geodatabase in which the features are to be stored, see Figure 10.

Note that if a personal geodatabase is specified, the user must set the New Layer's False X, False Y and False XY values, prior to invoking the [Import Points] command, using the [Set Active Layer] command, otherwise, a coordinate out of bounds error message may appear.

5. Once the name of the shapefile or personal geodatabase has been specified, click at the:
  - *Cancel* button to terminate the command, or the
  - *Save* button to create the new shapefile or personal geodatabase.

Note that this prompt will be repeated for each feature type to be created, points, lines, curves and/or polygons depending upon the specific polygon file format that was selected.

6. As the command reads the data in the ASCII based file, a progress bar display, in the status bar area, will appear indicating the percent completion of the processing. Upon completion, an information message box similar to that shown in Figure 29 will appear, click at the:



**Figure 29**  
Processing Results Message Box

- *OK* button to acknowledge the message and continue with any other desired processing.

### *Polygon File Format Description*

Since a polygon can be comprised of many sides, each of the polygon formats, shown above, expects to read several lines of data. These data lines can be thought of in terms of types Line 1 and Line 2. Each polygon defined in

the file will have only one type Line 1 data line, while the remaining data lines comprising the polygon will be of type Line 2.

The top five polygon formats process ASCII files that contain latitudes and longitudes, while the remaining formats process ASCII files containing coordinate values. When using the top five formats, the user should ensure that the appropriate projection has been applied to the View. In so doing, the latitude and longitude values can be converted into coordinates.

Polygon formats 6 through 12 allow the user to include a radius {Rad} argument, if appropriate. The {Rad} argument, which is optional, indicates that a curve, rather than a straight-line, side is to be defined. A positive radius denotes a curve progressing in a Clockwise direction, while a negative radius denotes a Counterclockwise direction. Curves are defined by specifying a chord length, chord direction and a radius.

In addition, the Polygon File Formats can be enhanced for greater functionality by the insertion of special keycodes into the file. These keycodes enable the user to:

- (a) specify whether line and curve features are to be created rather than polygon features,
- (b) define tangent curves and
- (c) specify the starting point implicitly rather than explicitly.

When line and curve features are to be created, rather than polygon features, the user is defining a string. A string may or may not be closed, that is, the start and end points of the string may or may not be the same point. Specifically, the keycodes are comprised of:

NOPOLY,  
POLY,  
CHORD,  
ANGLE,  
ARC,  
START and  
END.

The NOPOLY keycode informs the command that polygon features are not to be created but rather line and curve features are to be created. Once the NOPOLY keycode has been specified it stays in effect until the POLY command appears.

The POLY command indicates that polygon features are to be created. As such, it is possible for the user to establish a file that creates both polygon as well as line and curve features. If a string is to be created, the NOPOLY keycode must appear in the file prior to the definition of the string(s). Note that if only strings are to be created, the NOPOLY keycode needs to appear only once.

The START keycode enables the user to specify the starting point of a polygon or a string to be the start point of a previously defined polygon or string.

Likewise, the END keycode enables the user to specify the starting point of a polygon or string to be the end point of a previously defined polygon or string.

The CHORD, ANGLE, and ARC keycodes enable the user to define tangent curves. The default curve definition consists of specifying a chord direction, chord length and radius. The CHORD, ANGLE and ARC keycodes provide the user three additional means of defining a curve. The curves that are created using these keycodes are tangent to the previous course in the polygon or string.

The PolyID value entered on the Line 1 data line type of a polygon definition will be stored with the lines, curves and polygons that are created under the PROCNO attribute (field). If the line, curve or polygon layer (theme) does not contain the PROCNO attribute, it will be created for the user automatically. The PROCNO attribute is defined as STRING type with a maximum length of 8 characters.

Presented below are detail descriptions of each of the various Polygon File Formats.

*Polygon 1 (lat, long in Quad 1)*

Line 1: PolyID Lat. Long. Nlines  
 Line 2: Distance Bearing Direction  
 Line 2 appears Nlines times

Where for Line 1:

**PolyID** is an 8 character alphanumeric string.  
**Lat.** or Latitude is expressed in one of the following formats:  
 DDMMSSS  
 DDDMMSSS  
 DDDDMMSSS  
**Long.** or Longitude is expressed in one of the following formats:  
 DDMMSSS  
 DDDMMSSS  
 DDDDMMSSS  
**Nlines** denotes the number of data lines that will appear below Line 1 to describe the polygon. Note that comment lines are not included in this value.

The pound sign (#), comma (,) or a blank character ( ) may be used to delineate the four Line 1 data items. The latitude and longitude values are comprised of the following:

**DD** denotes degrees,  
**MM** minutes and  
**SSS** seconds to a tenth. No decimal point is specified in the seconds value. For example, 123 will denote a seconds value of 12.3, while 100 denotes a seconds value of 10.0. The degrees component of the latitude or longitude may be two or three digits and may or may not contain a sign (+ or -).

For Line 2:

**Distance** represents the length of a polygon side and appears in columns 1 through 5 of the data line. No implied decimal point is recognized. If desired, the user must explicitly enter the decimal point when specifying a distance. If no decimal point is speci-

fied and the distance does not occupy all five characters, the distance must be right justified. For example, if the distance is 159, the 1 should appear in column 3 with columns 4 and 5 containing 5 and 9 respectively.

**Bearing** is one of the following: NE, SE, SW, NW, N, S, E, W.  
**Direction** is the value of the bearing in one of the following formats  
 DDMM  
 DDDMMSSS

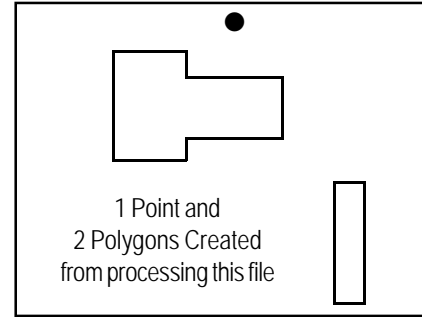
If the Bearing is N, S, E or W, the Direction item is omitted since the bearing, in this mode, will denote a due north, south, east or west direction, respectively.

No delineating character is used to separate the items on Line 2 but rather each item appears immediately after the previous item.

A sample file containing two polygons (with the # character as the delineator) is shown below, the result in Figure 30.

```
87831527#+1605565#4323253# 9
1800SW5100153
600S
2400E
1800S
2400W
700S
1800W
3100N
1800E
92832022#+1632250#4311452# 5
7250SE3652
10000S
700W
10000N
700E
```

The latitude and longitude entered on Line 1 establishes a point from which the polygon will be referenced to. That is, the starting point of the polygon is not the point that is defined on Line 1 but rather is the point that is computed by proceeding from the Line 1 point, the distance and bearing that appears on the first of the Line 2 data lines. The subsequent vertices of the polygon are



**Figure 30**  
**Sample Polygon File Format 1 Result**

computed by traversing the distances and bearings that appear on the remaining Line 2 data lines from the previously computed point.

All latitudes and longitudes will be considered as positive in sign. Any sign that appears in the input file for the latitude or longitude will be ignored.

The result of selecting this option will be the creation of not only polygon features but point features as well. The point features that are created correspond to the points that are defined on the Line 1 data lines.

The PolyID value entered on Line 1 will be stored with the polygon under the PROCNO attribute and with the point under the PTCODE attribute. If the polygon theme does not contain the PROCNO attribute, it will be created for the user automatically. The PROCNO attribute is defined as STRING type with a maximum length of 8 characters.

*Polygon 2 (lat, long in Quad 2)*

Same as Polygon 1 (lat, long in Quad 1) except that all latitudes will be considered positive (+) in sign and all longitudes will be considered negative (-) in sign.

*Polygon 3 (lat, long in Quad 3)*

Same as Polygon 1 (lat, long in Quad 1) except that all latitudes will be considered negative (-) in sign and all longitudes will be considered negative (-) in sign.

*Polygon 4 (lat, long in Quad 4)*

Same as Polygon 1 (lat, long in Quad 1) except that all latitudes will be considered negative (-) in sign and all longitudes will be considered positive (+) in sign.

*Polygon 5 (lat, long in all Quads)*

Same as Polygon 1 (lat, long in Quad 1) except that the signs of the latitudes and longitudes that appear in the input file will be maintained.

*Polygon 6 (x, y w/ Bearing & TieLine)*

Line 1: PolyID X Y Nlines  
 Line 2: Distance Bearing Direction {Rad}  
 Line 2 appears Nlines times

Essentially the same as Polygon 5 (lat, long in all Quads) except that, rather than specifying latitude and longitude (in that order), the user enters longitude and then latitude. Note that the longitude corresponds to the x coordinate or easting and that the latitude corresponds to the y coordinate or northing.

In addition, the same delineator character that was used on the Line 1 data line must be used to separate the Line 2 data items.

Furthermore, the five (5) character limit on the Distance item is not enforced under this option. The {Rad} argument denotes an optional radius value. If no radius is to be specified, the {Rad} argument should be omitted.

When entering the Bearing and Direction values, and if no radius value is specified, a bearing can appear in one of the following six formats:

- SW 23 15 00
- S 23 15 00 W
- 23 15 00 SW
- SW 23 15
- 3 23 15 0
- 323.15

However, if a radius value is to be entered, the Bearing must be one of the

following: NE, SE, SW, NW, while the Direction must contain Degrees, Minutes and Seconds values. Specifically, of the bearing formats shown above, only the example SW 23 15 00 could contain a radius value following the seconds value. Note that a positive radius value denotes a curve progressing in a clockwise direction, while a negative radius indicates a counterclockwise direction.

An example of a polygon containing both line and curve sides is shown below, the result in Figure 31.

```
87831527 5000.000 6000.00 10
586.777 SW 88 8 41.7
402.976 NW 34 3 58.3
348.473 NW 5 12 41.3 361.045
363.515 NE 23 38 35.7
407.475 NE 74 27 0.8
282.804 SE 78 4 28.4 306.486
270.802 SE 50 35 57.6
171.794 SE 20 29 13.3
505.864 SW 8 37 47.6 519.802
268.998 SW 37 44 48.5
```

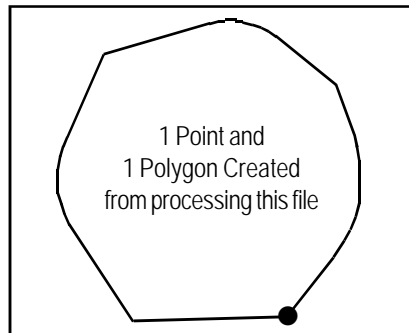


Figure 31  
 Sample Polygon File Format 6 Result

*Polygon 7 (x, y w/ Bearing)*

Line 1: PolyID X Y Nlines  
 Line 2: Distance Bearing Direction {Rad}  
 Line 2 appears Nlines times

Same as Polygon 6 (x, y w/ Bearing & TieLine) except that the polygon will begin at the point defined on the Line 1 data line. That is, there is no reference point. The first point in the polygon will be the point that is defined on the Line 1 data line.

A sample file containing one polygon (the space character is the delineator) is shown below, the result in Figure 32.

```
87831527 5000.000 6000.00 8
600.0 S
2400.0 E
1800.0 S
2400.0 W
700.0 S
1800.0 W
3100.0 N
1800.0 E
```

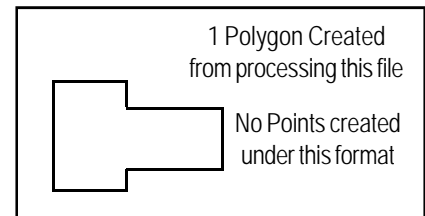


Figure 32  
 Sample Polygon File Format 7 Result

*Polygon 8 (x, y w/ Azimuth)*

Line 1: PolyID X Y Nlines  
 Line 2: Distance Azimuth {Rad}  
 Line 2 appears Nlines times

Same as Polygon 7 (x, y w/ Bearing) except rather than entering a Bearing and a Direction, the user enters an azimuth. The same delineator character used on the Line 1 data items must be used to separate the items on the Line 2 data line.

If a radius value is to be entered, the azimuth must be comprised of three values: Degrees, Minutes and Seconds values, following which, the radius value appears. Thus in this mode of operation, the Line 2 data line would be comprised of five values (one for the distance, three for the azimuth and one for the radius).

*Polygon 9 (x, y w/ Cartesian)*

Line 1: PolyID X Y Nlines  
 Line 2: Distance Cartesian {Rad}  
 Line 2 appears Nlines times

Same as Polygon 8 (x, y w/ Azimuth) except rather than entering an azimuth,

a Cartesian angle in decimal degrees is specified. A zero degree Cartesian angle is along the x axis, 90 degrees along the positive y axes, 180 degrees is due west (negative x axes) and 270 degrees is due south (negative y axes).

If a radius value is to be entered, the Cartesian angle is specified in decimal degrees, following which, the radius value appears. Thus in this mode of operation, the Line 2 data line would be comprised of three values (one for the distance, one for the Cartesian angle and one for the radius).

A sample file containing one polygon (the space character is the delineator) is shown below, the result in Figure 33. Note that this polygon does not have any curved sides.

```
87831527 5000.000 6000.00 8
 600 270.0
2400 0
1800 270
2400 180
 700 270
1800 180
3100 90
1800 0
```

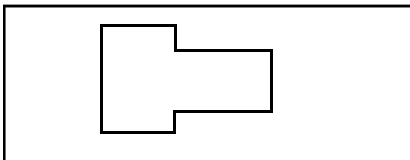


Figure 33

Sample Polygon File Format 9 Result

Polygon 10 (y, x w/ Bearing)

Line 1: PolyID Y X Nlines  
 Line 2: Distance Bearing Direction {Rad}  
 Line 2 appears Nlines times

Same as Polygon 7 (x, y w/ Bearing) except rather than entering x and y coordinates (in that order), the user enters y and then x.

Polygon 11 (y, x w/ Azimuth)

Line 1: PolyID Y X Nlines

Line 2: Distance Azimuth {Rad}  
 Line 2 appears Nlines times

Same as Polygon 8 (x, y w/ Azimuth) except rather than entering x and y coordinates (in that order), the user enters y and then x.

Polygon 12 (y, x w/ Cartesian)

Line 1: PolyID Y X Nlines  
 Line 2: Distance Cartesian {Rad}  
 Line 2 appears Nlines times

Same as Polygon 9 (x, y w/ Cartesian) except rather than entering x and y coordinates (in that order), the user enters y and then x.

Presented in Figure 34 is a sample file following the Polygon 6 File Format using the special keycodes which are available for all of the Polygon File Formats. Figure 35 contains the resulting image.

```
/*
/* defined polygon
/* or string.
/* CHORD - tangent curve
/* given chord
/* length and a
/* radius.
/* ANGLE - tangent curve
/* given central
/* angle specified
/* in Degrees,
/* Minutes, Seconds
/* and a radius.
/* ARC - tangent curve
/* given arc length
/* and a radius.
/*
/* Command denoting not to
/* create polygons
/*
NO POLY
/*
/* Figure 1 (which
/* happens to be closed)
/* with 10 vertices
/*
87831528 4000.0 4000.0 10
586.77753 SW 88 8 41.7
402.97627 NW 34 3 58.3
348.47370 NW 5 12 41.3
361.045
363.51546 NE 23 38 35.7
407.47558 NE 74 27 0.8
/*
/* Note comments can appear
/* anywhere in the file
/*
282.80400 SE 78 4 28.4
306.486
270.80277 SE 50 35 57.6
171.79402 SE 20 29 13.3
505.86450 SW 8 37 47.6
519.802
268.99867 SW 37 44 48.5
/*
/* Figure 2 defined in
/* a Clockwise direction
/*
87831529 5000.0 5000.0 5
200 NE 00 0 0.0
200 NE 90 0 0.0
/*
/* Curve defined by chord
/* length, chord direction
/* and a radius. This is the
/* default curve definition
/* (no keycode required)
/*
141.4214 SE 45 0 0 100.0
-100 NE 0
```

Figure 34 - Sample Polygon 6 File Format with Keycodes

```

-300 NE 90 /* /*
/* 87831534 5300 5900 5 /* Figure 11
/* Figure 3 defined above -300 NE 90 /*
/* but in a Counterclockwise 50 NE 0 87831539 5800 4700 5
/* direction ARC 78.540 50 300 NE 90
/* 250 NE 90 50 NE 0
87831530 5000.0 5500.0 5 -100 NE 0 ANGLE 90 0 0 -50
300 NE 90 /* -250 NE 90
100 NE 0 /* Figure 7 -100 NE 0
/* /* String with tangent curve /*
/* Negative radius denotes /* defined by chord length /* Figure 12
/* Counterclockwise direction /* and radius /* String with Reverse Curves
/* Positive radius denotes /*
/* Clockwise direction 87831535 5700 5900 5 87831533 END 87831532 12
/* -300 NE 90 250 NE 90
141.4214 NW 45 0 0 -100.0 50 NE 0 ANGLE 90 0 0 50
-200 NE 90 0 0.0 CHORD 70.7108 50 ANGLE 90 0 0 -50
-200 NE 00 0 0.0 250 NE 90 100 NE 90
/* -100 NE 0 -100 NE 0
/* Figure 4 /* -200 NE 90
/* Single line starting at /* Figure 8 50 NE 0
/* known coordinates /* String with tangent curve ANGLE 90 0 0 -50
/* /* defined by central angle ANGLE 90 0 0 50
87831531 5500.0 5500.0 1 /* and radius -100 NE 90
300 NE 90 /* ANGLE 90 0 0 50
/* 87831536 6100 5900 5 /*
/* Single line starting at -300 NE 90 /* End of the sample File
/* the end point of the 50 NE 0
/* string defined above /*
/* /* Central angle specified
87831532 END 87831531 1 /* in degrees, minutes and
-300 NE 00 /* seconds
/* /*
/* Command denoting create ANGLE 90 0 0 50
/* polygons not line and 250 NE 90
/* curve features -100 NE 0
/* /*
POLY /* Same three strings above
/* /* but defined in a
/* Figure 5 /* Counterclockwise direction
/* Polygon starting at the /*
/* start point of a string /* Figure 9
/* /*
87831533 START 87831532 5 87831537 5000 4700 5
100 NE 00 300 NE 90
300 NE 90 50 NE 0
100 NE 00 ARC 78.540 -50
-300 NE 90 -250 NE 90
-100 NE 00 -100 NE 0
/* /*
/* Command denoting do not /* Figure 10
/* create polygons /*
/* 87831538 5400 4700 5
NO POLY 300 NE 90
/* 50 NE 0
/* Figure 6 CHORD 70.7108 -50
/* String with tangent curve -250 NE 90
/* defined by arc length and -100 NE 0
/* radius
    
```

**Connect Points - Overview**

The intent of the [Connect Points] command is to create line features by connecting point features based upon a sequential attribute value and/or a common attribute value.

To use this command the user activates or selects the appropriate point layer in the Table of Contents prior to activating this command. Upon activation, the command displays a multi-input dialog box from which the user specifies:

- (a) the sequential attribute,
- (b) the common attribute,
- (c) the Z or elevation attribute (optional),
- (d) whether the selected point features are to be processed or if all of the point features in the active layer are to be processed and
- (e) the type of connectivity to be used (Sequential or Common and Sequential).

The lines which are created will be stored in a layer whose name will match the name of the current active layer.

Figure 34 - Sample Polygon 6 File Format (continued)

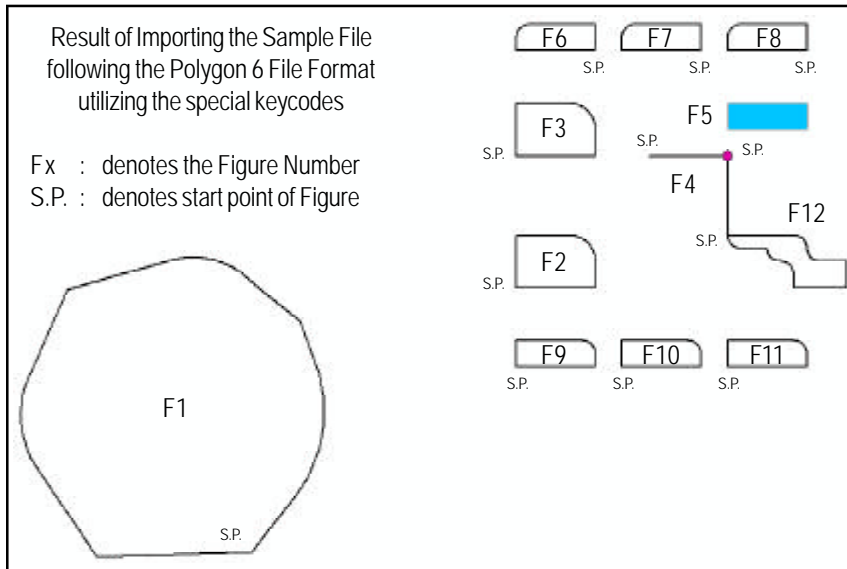


Figure 35  
Creating Lines, Curves and Polygons from an ASCII based File

Note that if this layer does not exist or if it does but is not of Line type, the *ln* extension will be appended to the name.

For example, if the name of the current active layer is L\_0 and the layer L\_0 does not exist in the Table of Contents, the lines will be stored in a layer called L\_0ln. In addition, if the L\_0ln layer does not exist, a PGD will be created called L\_0.mdb and a line feature dataset called L\_0ln will be created within the PGD.

Note that if a personal geodatabase is to be created, the user must set the New Layer's False X, False Y and False XY values, prior to invoking the [Import Points] command, using the [Set Active Layer] command, otherwise, a coordinate out of bounds error message may appear.

**Connect Points - Operation**

Once the command has been activated, the following steps can be performed:

- 1a. If the command does not find an active point layer in the Table of Contents, click at the *OK* button of Figure 36 to acknowledge the error message. Activate a point layer in the Table of Contents and reactivate the command.

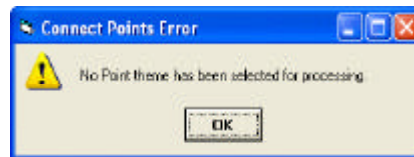


Figure 36  
Connect Points Error Message

- 1b. If the command finds more than one active point layer in the Table of Contents, click at the *OK* button of Figure 37 to acknowledge the error message. Activate a single point layer in the Table of Contents and reactivate the command.

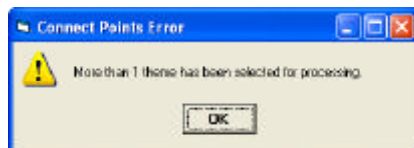


Figure 37  
Connect Points Error Message

- 2. Once a single active point layer has been found, the multi-input dialog box of Figure 38 will appear.

Figure 39 identifies the choices the user has in specifying which points within the active point layer are to be processed.



Figure 38  
Connect Points Dialog Box



Figure 39  
Point Processing Options



Figure 40  
Connectivity Processing Options

Figure 40 lists the choices the user has in specifying how the connectivity is to be performed.

A Sequential connectivity does not take into account the common attribute, while a Common and Sequential connectivity does, see Figure 13.

Once the appropriate information has been specified, click at the:

- *Cancel* button to terminate the command, or the
- *OK* button to continue with the processing.

**Summary**

The [Import Points] command provides a great deal of functionality in importing ASCII based point data with the ability to generate point, line, curve, polygon and annotation features. The [Connect Points] command is an excellent tool for generating lines from existing points using one or two attributes which are associated with the points. Readers should feel free to suggest additional formats which may be of interest.

*If you have a request for Command Of The Month, feel free to phone, fax or e-mail your request to The CEDRA Corporation.*