The CEDRA Corporation's

COMMAND OF THE MONTH

A monthly information bulletin   **March 2012**

FEATURED COMMAND

Importing
Survey Points

Application Description

Those who deal with surveyors know that a good deal of what they do involves field work. This field work involves the "pickup" of shots or points, which capture the position (X, Y, Z) of various types of features, such as water valves, manholes, poles, etc. This information is typically stored in the surveyor's field equipment.

Once the surveyor is back in the office, the information, which was picked up, is downloaded to the computer in the form of an ASCII based file. The question becomes how can this file be imported into the ArcGIS environment.

The CEDRA Solution

To address the application described above, the [Import Points] command within the {CEDRA-Commands ComboBox} toolbar, shown in Figure 1, can be used.

Import Points Overview

With the [Import Points] command, the user is able to import, in mass, points in a variety of formats, which are stored in an ASCII based file. Additionally, the user is able to generate line features using information stored in the ASCII based file.

To use this command, the user should:

1. Create a folder using native Windows functionality to contain the data that will be created, after which,
copy the ASCII file to be processed into this folder.

2. Invoke ArcMap.

3. Set the Map and Display units using the {View} [Data Frame Properties...] command.

4. Set the current working directory using the [Set Working Directory] command.

5. Select the [Import Points] command to process the ASCII file.

> **Command Of The Month bulletin**
>
> *This month's issue discusses how to process an ASCII based file to mass import point features and generate line connectivities.*

The result of this operation will be either a shapefile or a personal geodatabase depending upon the options selected during the operation of the [Import Points] command.

Additionally, the user has the ability to generate line features. This is accomplished by processing a point code that is assigned to a point. The user has the ability to generate lines by connecting points of common point codes.

Import Points Operation

To use this tool command, the user should:
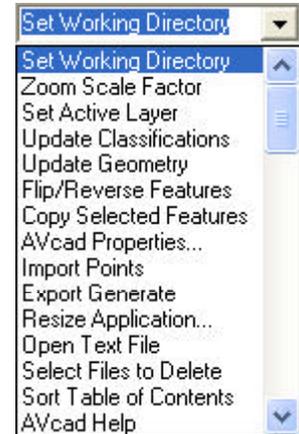
Figure 1
CEDRA Commands Combo Box

Figure 2
[Import Points] File Selection Dialog

➤ **1** **Scroll** down in the **CEDRA commands** menu combo box, and **select** the [Import Points] command (see Figure 1) to display the dialog box of Figure 2.

➤ **2** **Click** at the **Browse** button to display the conventional file navigation dialog box.

➤ **3** **Navigate** to the disk drive, directory and select the appropriate file type to be displayed followed by **selecting** the file to be imported.
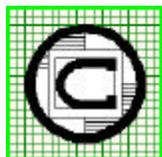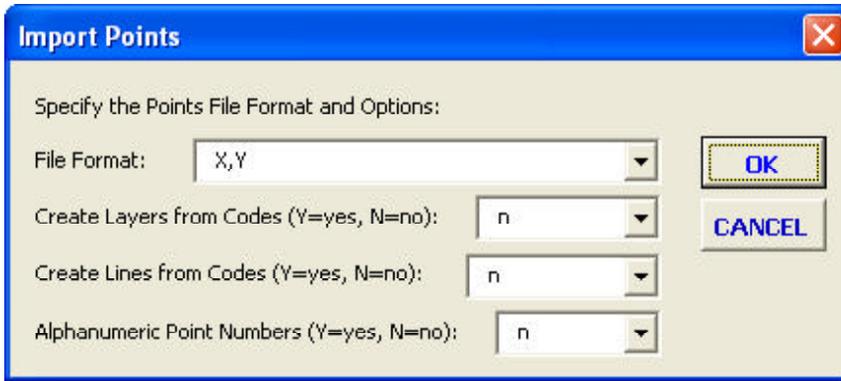
Figure 3 - [Import Points] Parameter Dialog Box
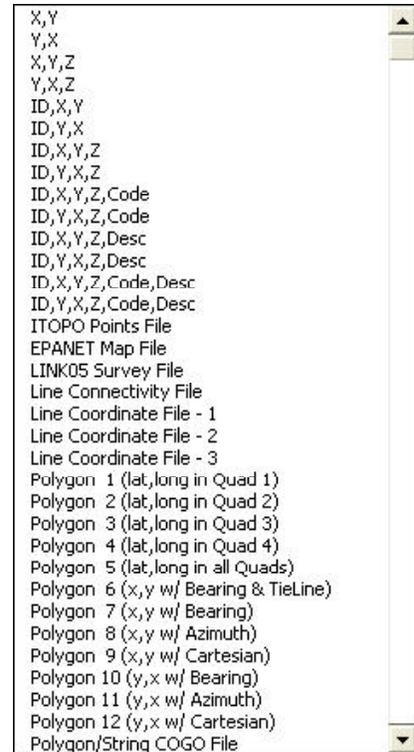
Figure 4
[Import Points] Available File Formats

➤ **4**　**Click** at the **Open** button to display the dialog box of Figure 3, which contains four choice list data fields,
or
click at the **Cancel** button to abort the command.

➤ **5**　**Scroll** down in the File Format: data field, and **select** the **file format** that represents the contents of the ASCII based file being imported, see Figure 4.

➤ **6**　**Scroll** down in the Create Layers from Codes (Y=yes, N=no): data field, and **select** the:
　• **y** option to place each feature being created, and having the same point code in a unique layer, or dataset as described later on.
　　or the
　• **n** option to place all features, regardless of the point code, in the same layer or geodatabase as described later on.

➤ **7**　**Scroll** down in the Create Lines from Codes (Y=yes, N=no): data field, and **select** the:
　• **y** option to connect with single lines points that have the same point code.
　　or the
　• **n** option to create points and no connecting lines.

➤ **8**　**Scroll** down in the Alphanumeric Point Numbers (Y=yes, N=no): data field, and **select** the:

　• **y** option to denote that the point numbers contain alphanumeric characters, such BL-4, X-45 and so forth.
　　or the
　• **n** option to denote that the point numbers are purely numeric (1, 2, 3, etc.).

➤ **9**　**Click** at the **OK** button to confirm the selections,
or
click at the **Cancel** button to abort the command.

Having clicked at the *OK* button, the program begins to read, and then process the ASCII file being imported. As the various features are being created, they are placed in the layer(s) or personal geodatabase (PGD) and dataset(s) depending on the settings specified by the user. Shown in Table 1 are the possible import scenarios.

Scenario A

Under this scenario the program checks to see if the root name of the ASCII file which is imported exists as the base name of a point layer. For example, if the ASCII file is called **L_0.txt**, the program checks to see if L_0 exists as a points layer, or not.

a. If the root name of said file (i.e. L_0) does not exist, or if it does exist, but

Data Field Parameter		Option	Scenario
	Create Layers from Codes (Y=yes, N=no):	*n*	
and	*Create Lines from Codes (Y=yes, N=no):*	*n*	*A*
	Create Layers from Codes (Y=yes, N=no):	*y*	
and	*Create Lines from Codes (Y=yes, N=no):*	*n*	*B*
	Create Layers from Codes (Y=yes, N=no):	*y*	
and	*Create Lines from Codes (Y=yes, N=no):*	*y*	*C*
	Create Layers from Codes (Y=yes, N=no):	*n*	
and	*Create Lines from Codes (Y=yes, N=no):*	*y*	*D*

Table 1 - Possible Import Scenarios
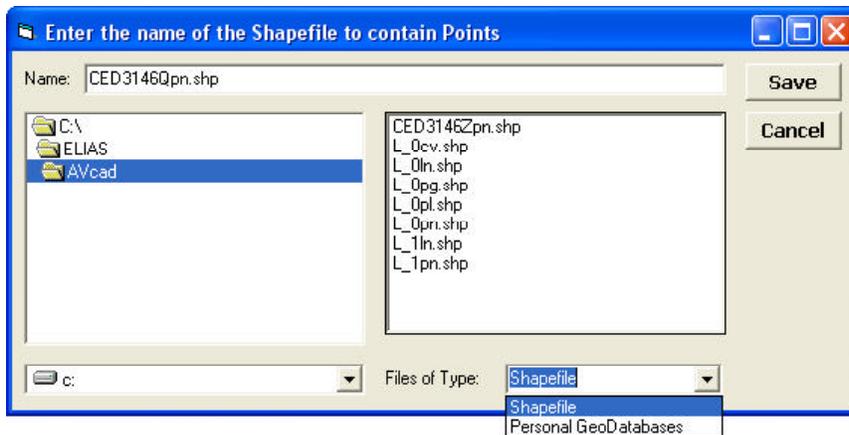
Figure 5 - Shapefile / PGD Selection

Figure 6
Processing Results

is not a points layer, the user is prompted with the dialog box of Figure 5 to specify the name of the shapefile or PGD in which the points are to be stored.

In this case the user should:

➤ **10a** **Navigate** to the appropriate directory, and **enter** in the Name: data field the desired name. If the new features are to be saved in a:
  • Shapefile, the name to be entered could be for example L_0pn.
  • PGD, the name to be entered could be for example L_0pn L_0.mdb, noting that at least one blank space must separate the layer name, L_0pn from the PGD name, L_0.mdb.

➤ **11a** **Scroll** down in the Files of Type: data field, and **select** the:
  • **Shapefile** option to create and save the features in the shapefile (see Notes below), or the
  • **Personal GeoDatabases** option to save the features in the PGD.

➤ **12a** **Click** at the **Save** button to proceed with the processing of the ASCII file, create the point features and save them in the specified:

• Layer (i.e. L_0pn, note the two letter suffix), or
• Dataset of the PGD (i.e. L_0pn in the L_0.mdb PGD),
  or
  click at the **Cancel** button to abort the command.

Having clicked at the Save button, the command begins importing the data, after which, the information box of Figure 6 is displayed informing the user of the number of features that have been created.

Note that if it is desired to save the point features which are created in a PGD, and if:
•   The ASCII file's root name does not exist as the base name of a layer, the command will determine the *False X*, *False Y* coordinates for the user. The *False XY* value that is desired can be set using the [Set Active Layer] command. A *False XY* value of at least 10000 should be used for survey grade accuracy.
•   The ASCII file's root name exists as the base name of a points layer, the user should change the name of said file, and create the PGD as stated above.

The user should now **skip** to **Step 10b**.
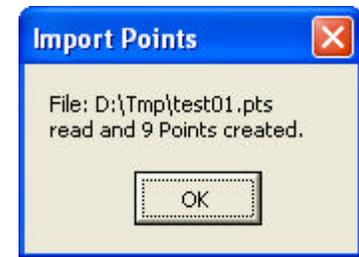
b. If the root name of said file (i.e. L_0) exists as a points layer, the program proceeds to create the points, saving them in said existing point layer, after which, the information box of Figure 6 informing the user of the number of features which have been created will be displayed.

It is important to note that:
• If the root name of the ASCII file is the same as the base name of an existing points layer, the new points that are created are appended to the file. Therefore, care should be exercised when naming the ASCII file.
• If the root name of the ASCII file is the same as the base name of an existing layer which is not a points layer, the new points that are created are saved in a new layer that bears the name of the ASCII file with the pn two letter suffix appended thereto.

➤ **10b** **Note** in the information box of Figure 6 the **number of features** that have been created (refer to the commentary below this step).

➤ **11b** **Click** at the **OK** button to acknowledge the information box and terminate the command.

Notes
•   Once the information box has been closed, it is a good practice to check the above number of features created with the number of point lines in the ASCII file that has been im-

ported. If the two numbers do not agree, then there must be some format error in said file. In this case, remove the layer, correct said file, and repeat the command.

- If for some reason the wrong file format is picked in the dialog box of Figure 3, the number of features created in the information box will be zero (0).
- Upon creating the point features, the values of the ID, X, Y, Z, Code and Desc parameters of the ASCII file are saved in the PNT, X, Y, Z, PTCODE and PTDESC attribute data fields of the point layer, respectively. Note that if any of the above mentioned parameters are not present in the file format that is selected, the command will store a blank character or zero for the attribute.
- If the command is to be repeated, it should be done so from its very beginning.

Scenario B

Under this scenario, as well as under Scenario C , the program makes use of the Code parameter of the ASCII file. Shown in Figure 7 is a sample ID,Y,X,Z,Code,Desc ASCII file.

When processing files that have the Code parameter, the command utilizes the configuration file ptcode.lis, which resides in the CEDRA distribution folder (typically in the *c:\cedra\fontswin* folder). The ptcode.lis file enables the user to map a specific point symbol to a specific Code parameter of the ASCII file being imported.

In so doing, the user is able to denote which Code values actually pertain to points that represent a symbol (tree, hydrant, manhole, etc.). The value that appears in the ptcode.lis file is stored under the MRK attribute in the points layer.

Under this Scenario B, the parameter Code is not used to create lines that connect points of the same Code.

Should the ASCII file contain an invalid character for the Code parameter, the invalid character is replaced with the underscore (_) character. The following are considered as invalid characters: dollar sign ($), pound sign (#), greater than sign (>), dash (-) and period (.). Any other keyboard characters such as the percent (%), at sign (@), or others will cause the display of an error message.

The operation of the subject command under Scenario B is the same as that of Scenario A, except for where the points being created are saved. As such, Steps 10a, 11a and 12a are bypassed, and the message information box of Figure 6 is displayed.

The point features that are created are saved in layers whose names correspond to the unique values of the Code parameter in the ASCII file. These layers, in turn, are stored as datasets within a PGD whose name is the root name of said file.

For example, if the user selects a file called **L_0.txt**, the points that are created are saved in a PGD called **L_0.mdb**. Within this PGD will be datasets corresponding to the various unique values of the file's Code parameter. This can be verified by:
- Right-clicking in the table of contents at the label of one of the layers that has been created;
- Selecting the Properties... command to display the Layer Properties dialog box
- Clicking at the Source tag to display the following information:

Data Type:	Personal Geodatabase Feature Class
Location:	c:\X\L_0, where X is the name of the current working directory, and L_0 is the root name of the ASCII file
Feature Dataset:	MH11, where MH11 is the Code parameter of the ASCII file;
Feature Class:	MH11
Feature Type:	Simple
Geometry Type:	Point

- Clicking at the *Cancel* button to close the dialog box.

If the PGD and its datasets into which the points that are created from the subject file exist, the new points are added to those already in existence.

If the PGD, or any of its datasets do not exist, they are created by the program without interaction with the user. Thus the user should have some consideration in naming the ASCII file to be imported.

Scenario C

This scenario operates in a manner similar to that of Scenario B, with the difference that in addition to point features, the command creates single line features, and saves them in their own dataset. The lines are connected based upon the point code values. The user is able to specify which point codes are to be used when generating lines. As such:

```
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 7 - Sample ID,Y,X,Z,Code,Desc Points File
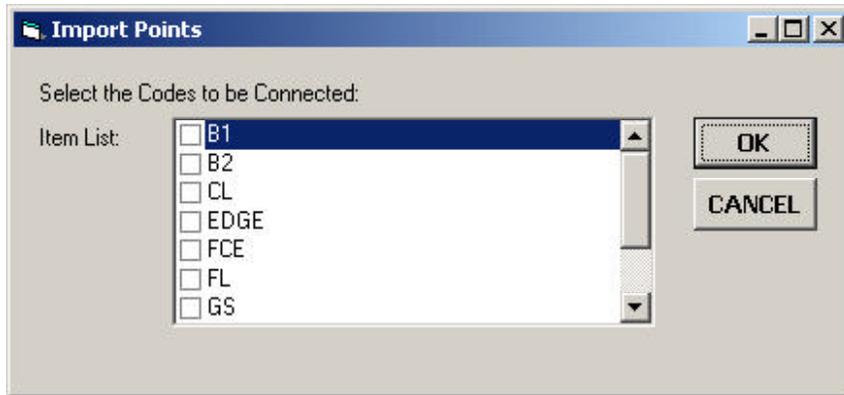
Figure 8 - Point Code Selection for Generating Lines

➤ **10c** **Select** the points codes that should be connected from a list similar to the one shown in Figure 8

➤ **11c** **Click** at the **OK** button to confirm the selection of the point codes and commence with the generation of the lines.
or
click at the **Cancel** button to skip the generation of the line features and terminate the command.

The user should note the following:

• Much like in Scenario B, the PGD created or used is that which has the root name of the ASCII file being imported. However, the name of the line layer or dataset will have the two letter suffix ln appended to the root name of said file. For example, if the name of the file being processed is **L_0.txt**, the line layer that is created will be named **L_0ln**.

• The lines that are created connect points that have the same Code value in the ASCII file. For example, if the footprints of two houses are being surveyed, and if it is desired to have these footprints shown as lines, the Code values of the points defining one house could be HOUSE1, and the points defining the other house could be HOUSE2.

• Points, which have Code values to be used to create lines, are created in addition to the lines. That is to say, if a point code is used in the establishment of a line, it does not mean that a point feature will not be created at that location.

Shown in Figure 9 is an example of lines, that have been created automatically by connecting points based upon a common Code value. The point annotations in this figure have been introduced to indicate the Code values used to create the various lines. When the lines that connect points are created, they connect points in the sequence that they appear in the ASCII file.

For example, the first point to be encountered in said file with a Code value of #523 is connected with the second point with the same Code value, the second point with the third and so on.

If a closed figure is desired, the first point should be repeated. Note that it is not necessary to group all points with the same Code parameter in sequence. They can be dispersed throughout the file, but their sequence controls the order in which the points are connected.

Note that the above said annotations are those of the ASCII file Code parameter and not those as saved in the PTCODE attribute of the point layer or dataset, in which the pound sign (#) and
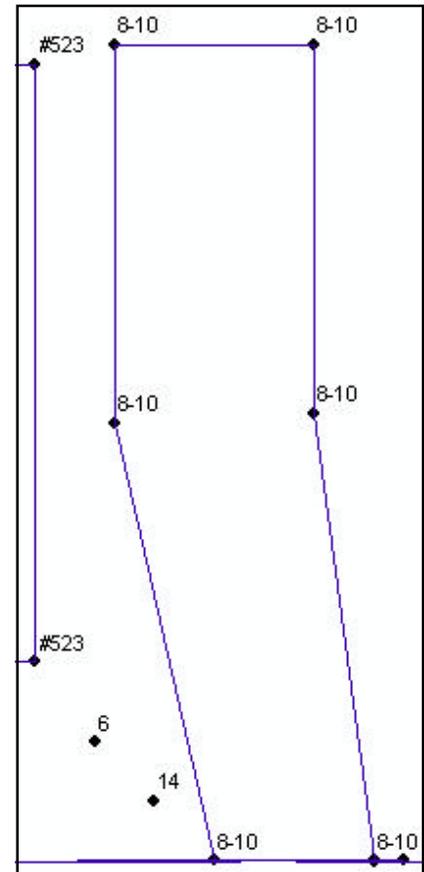
Figure 9
Connecting Common Codes

the dash (-) have been converted to an underscore (_) character.

Scenario D This scenario operates as Scenario A, with the difference that the command creates single line features, and saves them in their own layer or dataset as done under Scenario C. That is, the two letter suffix ln is appended to the root name of the file being processed. No user interaction is required when the line layer is established.

Under this scenario, there will be two layers created by the command using the root name of the file being processed and the pn and ln suffixes.

For example, if the name of the file being processed is **L_0.txt**, the layers **L_0pn** and **L_0ln** will be created by the command. The layer L_0pn will contain the point features, while L_0ln will contain the line features.

The PTCODE attribute in the point layer will contain the CODE values for the point features, while the LNCODE attribute in the line layer will contain the CODE value that was used to generate the line feature.

If the point layer does not exist in the Table of Contents, the user will need to perform steps 10a, 11a and 12a. Afterwards, the user should:

➤ **13d** **Select** the points codes that should be connected from a list similar to the one shown in Figure 8

➤ **14d** **Click** at the **OK** button to confirm the selection of the point codes and commence with the generation of the lines.
  or
  click at the **Cancel** button to skip the generation of the line features and terminate the command.

All points which are created are grouped into one layer as under Scenario A. The PTCODE and LNCODE attributes can be used to classify the layers, thereby displaying the various CODE values that were encountered during the processing.

Notes

a. The PTCODE and LNCODE attributes are of string type with a length value of 20.

b. When specifying a CODE value, the user is able to specify the ST and END keywords to indicate the start and end of a line feature, respectively. These keywords must appear after the CODE value and must have only one space or blank character separating the CODE value from the keyword. For example, the following lines of data:
  FL ST
  FL
  FL
  FL END

would result in three line features being created. The use of these keywords enable the user to not have to use unique CODE values when generating line features.

Summary

When having to import survey data, the [Import Points] command is a highly flexible and powerful command.

The [Import Points] command has undergone a few recent enhancements, which are reflected in this publication. Users with a software support agreement should check with The CEDRA Corporation on how to obtain a software update so as to be able to utilize this new functionality.

As always, users who have a need for functionality that is not presently available in CEDRA software should feel free to forward these requests to CEDRA, as well as, any other comments or suggestion you may have.

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