POLYGON tables lend themselves to a number of mapping and modelling applications. For example, polygon data might include geologic unit contacts, ore body shapes, mineral claim boundaries, or township and range data. Loading polygon files into polygon tables is an easy task. This **Technote** will outline the process of how to load data into a POLYGON table.

Procedure:

Step 1.

The first step is to create a file in the correct polygon format. This can be done by digitizing the polygons through the *SDIGIT* program, or by exporting the files from another program. The file should contain the X-coordinate, Y-coordinate, and an optional Z-coordinate. After those two or three columns, you can include any additional polygon attributes that you wish. The X-, Y-, and Z-coordinates must be delimited by WHITESPACE. For further details about polygon files, consult **"Polygon Files" on page appendix-116**.

Example 1:

```
21509       44095       polygon1    123         4
21525       44122       polygon1
21509       44095       polygon1

21609       44000       polygon2    145         3
21834       44038       polygon2
21609       44000       polygon2
```

Example 2:

```
Polygon1  123  4
21509       44095
21525       44122
21509       44095

Polygon2  145 3
21609       44000
21834       44038
21609       44000
```

Example 3:

```
21509       44095       polygon1
21525       44122       polygon1
21509       44095       polygon1
21609       44000       polygon2
21834       44038       polygon2
21609       44000       polygon2
```

Step 2.

Use the *DEFINE* program to create a POLYGON table. As you define the table, **TECHBASE** will automatically create an EDGE table (_e) and a VERTEX table (_v) to go with your POLYGON table. If you already have a pre-defined VERTEX and/or EDGE table that you would rather use, you may do so by entering the name or names at the prompt. When the VERTEX table is entered, you will be asked to enter information on the X-, Y-, and Z-coordinates. If you only want a 2D POLYGON, then leave the Z-coordinate blank.

The POLYGON table does not yet contain a polygon identification field or any of the other fields you may need to code your polygons. You should create at least a polygon id field, however you may also want to consider a polygon pattern field and a polygon color field. In our example, we will create the following three fields:

Name	Type	Class	Attributes
poly_id	TEXT	ACTUAL	10 Left justified
poly_colo	INTEGER	ACTUAL	0 to 255
poly_pat	INTEGER	ACTUAL	0 to 255

Step 3.

Use *LOAD* to load the polygons into the POLYGON table. In the *Setup* menu, enter the name of the polygon file. The following formats will load the above examples into a POLYGON table:

Example 1:

null null poly_id poly_color poly_pat (poly)

Example 2:

poly_id poly_color poly_pat / (poly)

Example 3:

null null poly_id (poly)

The (poly) notation is used to inform the *LOAD* program that it must distribute the data to the POLYGON, VERTEX, and EDGE tables. To load a 3D polygon in **Examples 1** and **3**, you would need a third consecutive “null” for the Z-coordinate.

You have now populated all three tables with the data. The EDGE table contains the information about the edges or line segments joining vertices from the VERTEX table. The POLYGON table holds the attributes of the polygons, including the attributes loaded from the polygon file. The VERTEX table holds the information on each of the vertices of the polygons.

The data is now ready for use in any of the **TECHBASE** programs. For example, use the *PERSPECT* program to make a 3D graphic. Use the color and pattern fields to enhance the graphic.

For more information on creating POLYGON tables, see **“POLYGON” on page 1tb-11**. For more information about loading data, see **“Load – load data files into a database” on page 1tb-35.**

