

Adding color, markers and patterns to a map, cross section, or borehole log can make it more presentable and easier to understand. This **Technote** will discuss two methods by which you can use coded values AND save disk space. The advantages and disadvantages of these methods depend upon the complexity of the data.

Procedures:

Method 1: Using CALCULATED fields

By their nature, CALCULATED fields are equations and do not contain data. As a result, they require a negligible amount of space in the database and on the hard drive. When using CALCULATED fields as your coding method, you will automatically update your codes when you update your data.

Step 1.

Decide what criteria you would like to use to code your data. We will use two examples to show some of the more common possibilities. Using the **BOLAND DATABASE**, we will change the color and marker of drillhole collars to reflect the drill year for each hole. For color codes, when **hole_id** contains the text string "84-", we will use BLUE, for "85-" use GREEN, and for "86-"use YELLOW.

Step 2.

In the *DEFINE* program, create two INTEGER, CALCULATED fields with limits between 0 and 255. One will be called **hole_color** and the other **hole_marker**. We will place these equations in separate format files so that they can be saved and commented. This method will facilitate later updates.

File one: *HOLE_CLR.CLC*:

```

4          # Place the color blue on the stack
hole_id 84- ( ) 13 skip # If the hole contains 84-, you are finished (skip all other steps)
3          # Place the color green on the stack
hole_id 85- ( ) 7 skip  # If the hole contains 85-, you are finished (skip all other steps)
7          # Place the color yellow on the stack
hole_id 86- ( ) 1 skip  # If the hole contains 86-, you are finished (skip all other steps)
2          # If no other criteria fits, use the color red

```

Step 3.

Go to *DEFINE, Fields, Create*, enter the appropriate information, and in *Equation*, simply type: **(f,HOLE_CLR.CLC)**. Now, in any of the graphics programs where a color parameter is requested (e.g. *POSTER, Points*), you can enter the name of this CALCULATED field for the color.

Step 4.

Next, for drillhole markers, the equation in *Step 2* can be modified to get a similar result. Simply replace the color numbers with the marker numbers that you wish to use. Use the *MARKERS.MET, COLORS.MET,* and *PATTERNS.MET* files in your TBHOM directory as a guide to help you choose the attributes you desire.

Note: CALCULATED fields are limited to 32 operators and so cannot be used where the coding involves many variables. The *TBCALC* program can be used instead, but since the results must be stored in ACTUAL fields, your database size will grow. For value coding where many variables are used, refer to **Method 2**.

Method 2: Style tables.

Technote: Color, Marker, and Pattern Coding

When a field contains a finite number of constant values, and you wish to use these values as the basis for coding, an effective way to avoid duplicate records is to employ a separate “style” table and to use a JOIN table relationship. For example, you have a list of 15 rock types, where each type is represented by a specific color, pattern, and pattern scale. The following steps describe how a style table would be created:

Step 1.

You have downhole geology information for many drillholes in a field called **lithtype** in the FLAT table **GEOLOGY**. In *DEFINE*, create a new FLAT table called **GEOCODE**. This will be the style table.

Step 2.

In *Tables*, go to *Add field* and request that **lithtype** be added to the **GEOCODE** table. Next, go to the *Key field* option, and key **lithtype** in the **GEOCODE** table only. Finally, in the *Fields* option, *Create* the following three fields: **color_g** (INTEGER, ACTUAL), **patt_g** (INTEGER, ACTUAL), and **pattscale_g** (REAL, ACTUAL). For the two INTEGER fields, you can set the limits as 0 and 255, and for REAL field, you can set whatever limits you decide or no limits at all.

Step 3.

In the *Tables* option, *Create* a JOIN table called **GEOL_CODE** that joins the **GEOLOGY** table to the **GEOCODE** table. This will establish a relationship based on the common field **lithtype**.

Step 4.

Now you need to fill in the records for the **GEOCODE** table. Each record must have a unique rock type in the **lithtype** field, followed by whatever styles you want in the **color_g**, **patt_g**, and **pattscale_g** fields. Remember that because **Method 2** requires a constant set of values, it cannot be used for ranges of values (e.g. assay values). If you already have the information in a text file, you can use the *LOAD* program to enter the data, otherwise, the *TBEDIT* program will allow you to manually *Insert* records. Updates to your styles table (e.g. a new rock type) must be made manually.

Note:Make sure to have ALL rock types represented by a record in **GEOCODE**. The join will produce NULL style values if **GEOCODE** is missing a record for a certain rock type.

Your final tables and fields: **GEOLOGY** (FLAT) = **hole_id from_g to_g lithtype ...**
GEOCODE (FLAT) = ***lithtype color_g patt_g pattscale_g**
GEOL_CODE (JOIN) = **(GEOLOGY to GEOCODE)**

Now, in any of the graphics programs where color, pattern and pattern scale parameters are requested, you can enter the name of the appropriate ACTUAL fields from the **GEOCODE** style table.

See Also

[Values\(0\)](#)

[Database\(0x\)](#)

[Define\(1tb\), Tbcalc, Tbedit\(1tb\)](#)

[Poster\(1gr\)](#)

[“Technote: Check Data for Orphan Records” on May, 1995](#)