User's Guide to WHEAT : the Windows-based Hydrogeologic Exploration and Analysis Toolkit

Greg Pouch
Kansas Geological Survey
Tuesday, December 06, 1994

Introduction

WHEAT is a set of user-friendly programs developed at the KGS for the retrieval, analysis, manipulation and display of information on natural resources, especially groundwater. It runs under the Microsoft Windows 3.1 operating system and is intended to be easy to use. The principle goals in development of WHEAT have been: ease of use; hardware independence; applicability to resource management problems; end-user extendibility; and the ability to exchange information with other analysis packages.

This document is intended for the beginning WHEAT user interested in displaying, querying, and printing maps; it provides some guidance on designing a database.

This open-file report is a "living document", in that revisions will be made to it to reflect new programs added, new chapters, or changes to old programs. New programs and changes to old ones will update and enhance older versions, so rarely will something that worked in one version not work later. It is organized into separate chapters describing major programs. This version includes E-MAP, the electronic mapper; the CONTOUR program, and the database design guide.

The programs, included on the diskettes, are distributed without charge or warranty or technical support. **WHEAT is not intended as a complete stand-alone package, and certain capabilities are better provided through commercially-available packages:** to fully use WHEAT, each site should also have a relational database management system, preferably Microsoft Access, a spreadsheet, and a presentation graphics package.

Organization of this report

Part	Title
A	User's Guide to WHEAT (this cover document)
B	How to use WHEAT Electronic Mapper (E-MAP)
C	How to Design and Build a WHEAT Database
D	How to Contour in WHEAT

DISCLAIMER

Neither the Kansas Geological Survey nor GMD 2 guarantees these programs to be free from errors or inaccuracies and disclaims responsibility for interpretations based on data processed using this software or decisions based thereon. Neither agency provides free technical support.

What is WHEAT

WHEAT is a set of user-friendly programs developed at the Kansas Geological Survey for the retrieval, analysis, manipulation and display of information on natural resources, especially groundwater. It runs under the Microsoft (MS) Windows 3.1 operating system and is intended to be easy to use.

WHEAT is modular and menu-driven so users can learn by example as much or as little of the system as necessary for their needs. WHEAT does not require the detailed understanding of the data model or extensive command vocabulary that many commercial GIS or CADs require.

WHEAT stores data in MSAccess databases: using EXPORTDB you can also export to ASCII text files. The end-user can extend and enhance WHEAT by writing programs to manipulate data or to provide additional transfer capabilities with spreadsheets, mainframe programs, and GIS systems. (See the database design guide and WHTCODE.MDB.)

Using E-MAP, the user can display tile (section), point, line, polygon, text label, and grid data; select the viewing region by zooming and panning; retrieve information on a geographic feature by double-clicking on it with the mouse; and print maps directly from E-Map or copy to graphics programs for further manipulation using the Windows Clipboard. Maps can also be exported to Windows Metafiles.

CONTOUR allows the user to contour and interpolate point data.

Two programs are included to provide limited database manipulation capabilities:

GRDEDT01 EXE is a spreadsheet-like table editor

RECEDIT1 EXE allows editing of a single record in a database

WHEAT includes a number of specialized import-export programs that are used infrequently. These include:

BLOBMEMO EXE converts Wheat internal XY-BLOBS into ASCII chains of coordinates and vice versa. Lets you import virtually any line data you as a human can read.

FXDWDTH1 EXE is used to import fixed-width files into Access.

For use with ARC/INFO imports and exports, there are:

ARCS_IN2 EXE imports ARC/INFO ungenerate format files into polygons or lines

BILDPOLY EXE converts arc-node topology into chain topology used in Wheat. Unnecessary for use after ARC/INFO 6

EXPORTDB EXE creates an ARC/INFO AML to define a table and a comma separated value file containing the data to export from an Access-format database. Used to move tabular data into ARC/INFO.

System requirements

In addition to the computer (below), each site should have database management software (that can use the ODBC interchange standard) for manipulating data in non-graphical manner, such as Microsoft Access, and a Windows-based spreadsheet such as Excel. A presentation graphics package such as CorelDraw is also useful for "prettying up" maps. (Software suggestions are the author's favorites, and do not constitute any official endorsement.)

Program and attendant libraries take about 4MB of disk space. WHEAT will run on a Windows for Workgroups network or under Windows -- no claims can be made for other networks or operating systems.

Minimum Computer

WHEAT will run on a computer that can run Microsoft Windows 3.1 programs. For PCs, at least a 386 with 2 MB RAM. You will also need 4MB available disk space for programs; and a network connection to or disk space for the database.

Desired Computer

WHEAT, and especially E-MAP, will run well on a 486DX with 8 MB RAM, a fast hard drive, and a good (256 colors or more) video card with Windows 3.1. Other configurations are possible. If you are buying a computer, I recommend (12/6/94) at least a 486DX2/66, a 500 MByte hard disk, a 16 million-color video card, and a 15" monitor. Because of the large size of the databases you will end up generating, you will want to back them up frequently, so you should probably also get a QIC-80 tape drive.

Extending WHEAT

Under certain circumstances, Kansas Geological Survey can contract to provide enhancements to WHEAT. Contact Greg Pouch at KGS.

If you wish to extend WHEAT yourself, there are several options. If you need to modify data and write your own data processing programs you can use MS Access or Visual Basic. If you want to modify the programs, you need Microsoft Visual Basic Professional 3.0 or higher. Many of the spatial query routines are in FORTRAN, so a FORTRAN 16-bit compiler that can produce Windows Dynamic Link Libraries would be needed. The cursors live in Microsoft Visual C++ resource-only Dynamic Link Library called WHTCURS2.DLL.

Acknowledgments

These programs were developed at the Kansas Geological Survey, Lawrence, Kansas during 1994 on the WHEAT (Windows-based Hydrogeologic Exploration and Analysis Toolkit) project, in cooperation with the Equus Beds Groundwater Management District (GMD #2), Halstead, Kansas.

How to use WHEAT Electronic Mapper (E-MAP)

This document explains the use of the electronic map (E-MAP) tool included in the Windows-based Hydrogeologic Exploration and Analysis Toolkit (WHEAT). It is intended as a how-to introduction to E-MAP for general audiences, and assumes you already have a database. If you do not, USE THIS MANUAL WITH THE SAMPLE DATABASE, then, using the directions in *How to Design and Build a WHEAT Database* to create a database.

Using E-MAP, a non-specialist can open, design, save and print maps, as well as perform spatial queries on points by double-clicking. Using LookupTable Editor allows the more advanced user to design a legend and apply it to a table or query. This document does not explain the use of LookupTable Editor, or other import/export tools

Greg Pouch
Kansas Geological Survey
Tuesday, December 06, 1994

How WHEAT stores data

WHEAT E-MAP uses data stored in a relational database. Specifically, E-MAP uses queries and tables stored in a Microsoft Access format database to provide both the attributes and coordinates of various geographic features.

A relational database stores all information on a topic, such as water users or schools, in tables. Tables are organized sets of information containing records, one for each individual thing in the database. A record is composed of fields, which describe the thing. Thus, a table concerning schools might have records for each school in a county, and the records might include the name (as text), date founded (as date), operating budget (as a floating point number), date closed (as date), number of buses owned (as integer number), principal's name (as text), and number of pupils (as integer number). Notice that some of the fields listed, such as date closed and number of buses, will not occur for each individual school. In these cases, a relational database stores a null, indicating that no information is available.

While a relational database *stores* information in tables, it can *retrieve* information as virtual tables called queries. A query is the answer to a question asked about the datatable, such as: "show all schools with an operating budget above $500,000". A query can return data from several related tables by joining them. For example, suppose we had another table containing information about students. It might contain student's name, age, address, and name of school attended. We could then find out who a student's principal is by designing a query which relates the school name field in the student table to the name field in the school table. Queries are generally used for combining information from several tables, selecting subsets of records, or selecting subsets of fields. By allowing data to be stored in a single location (such as the principal's name in the school field) and related at a later time, relational databases can reduce storage requirements, and allow the updating of data to be considerably easier than if each record contained all data that might ever be needed.

E-MAP uses a relational database to store information about geographic features, both their descriptive (or tabular) attributes and their locational attributes. The location(s) of feature are described as X (east-west) and Y (north-south). WHEAT does not use any particular coordinate system, except that X and Y will be plotted as equal distances on the screen and on printing, and X increases to the right and Y increases to the top.

For points (for plotting symbols or text), WHEAT stores the X and Y locations in separate numerical fields in the table. These might be named something like Northing and Easting or X and Y, although, since the fields containing coordinates are chosen by the user while running E-MAP, they can be named anything. For features with multiple locations, like the set of line segments defining a river or the boundaries of a county, WHEAT stores the data in a special type of field called a Binary Large Object (BLOB or XYBLOB). This is simply a binary representation of a set of X and Y locations, stored as a chain of points.

The attributes and locational information can all be stored in one table for each set of items, or they can be stored in separate tables and related in a query. Additionally,

plotting instructions (like color, line width, and symbol) can be stored in a table or defined as constants.

Displaying a Map with WHEAT E-MAP

Overview and Definitions

In WHEAT E-MAP, a map consists of a set of overlain themes. Each **theme** is about a particular topic (such as political units or drainage basin or water appropriations) and of a particular geometric type (points, lines, tiles, polygons, text labels, rastergrids; defined below.) Thus, a map of counties with a star at the county seat and a label for the county would consist of three themes: the location of the county seats (points); the names of the counties (text labels); and the outlines (polygon or line) of the counties.

In addition to points, lines, and polygons, WHEAT E-MAP includes three other useful geometric types. **Tiles** are rectangles with their edges oriented due north and due east (relative to the coordinate grid); under favorable conditions, tiles can be used to represent sections in the US Public Land Survey System. **Text labels** are text at a given point, such as the name of a school. **RasterGrids** consist of a value at each point on a regular grid, used for things such as elevation. They tend to be huge and unwieldy and their use is discouraged unless you have a good reason.

Two other useful concepts before beginning are the DataBounds and the ViewBounds of a map. As map themes are created, data is loaded into memory . The region occupied by the data is referred to as the **DataBounds** (the areal extent of the data). The region currently being viewed in the map window is the **ViewBounds** or view region.

Getting Started

In order to begin a session with Wheat E-MAP, you first open a database containing the geographic information. You do this by starting E-MAP, and selecting File=>Open_Database... from the menu, then choosing a database that contains the geographic information you wish to display. Once you have successfully opened a database, several dimmed menu options become undimmed and you can begin designing a map.

Loading Previously Defined Themes

If there are previously defined themes saved in the database, you can choose File=>Open_Map_Theme... to display a list of map themes that have been saved. (These map themes are defined in a table called zWHTtblThemes, to which you can save theme definitions from the File=>Save_Theme... menu item or edit by hand once you know how.) The list on the left is the available themes, the list on the right is the themes you have so far chosen to display. You add a theme to display by selecting it on the list at left, then double-clicking on it or clicking on the > command button. Double clicking a theme listed on the left removes it from the list of themes to display, as does clicking on the < button. (This should be a lot like the font manager in the Windows Control Panel.) Once the

themes you want are listed in the list on the left, click **OK** to accept, or **Cancel** to not load any themes.

Loading a New Theme

Begin by deciding what kind of theme you wish to display (point, line, polygon, or such) and selecting File=>New_Map_Theme=>Symbols... or File=> New_Map_Theme=>Polygons... or whatever you want. You then choose the table or query containing the information you want from a list of tables and queries available in the database. E-MAP checks the field names and, based on settings in your WHEAT.INI file, makes guesses about what fields might be used for what. Thus, if you're trying to draw a symbol themes, and have fields by the names of X_Lambert, Y_Lambert, SymbolCode, FontCode, AppropriationName, WellTotalDepth and AppropriationNumber, E-MAP will add X_Lambert to the list of potential fields for X, Y_Lambert to the list of potential fields for Y, SymbolCode and FontCode to the list of fields for SymbolCode and FontCode, and AppropriationName will be its guess for a field for unique name. You can enter constants in any of the fields provided except X and Y, or you can choose a field name.

If the color you want is not listed in the dropdown box, click on the label Color, and a dialogue box will appear from which you can select a color. This is the same color dialogue as most other Windows applications use. For symbol codes, click on the label Symbol, and a dialogue box that displays symbol codes and symbols will appear. The up and down arrows can be used to browse symbols in the currently selected font. Choose OK and that symbol code (the ascii character number of the symbol) will be added to the list of symbol codes as the currently selected symbol.

Note that WellTotalDepth was not added to any of the lists in the previous example except the list of all plottable fields. If you want to add a field to a list or use it for a criteria, select the field in the list of fields. You can then copy it to desired location using standard Windows keyboard cut-and-paste commands, or you can click on the button labeled Drag Field Name, move the icon that appears using the mouse, and click the left mouse button to add the field name to the whatever list you drop it on. (This is actually easier done than said.) If you drop the icon on the label next to a text box, it will overwrite that whole text box. If you drop it on the text box itself, it will paste the field name at the current insertion point. (This is mainly useful in setting criteria.)

Once you have chosen how the theme should be plotted, select OK and E-MAP will run a query based on your choices, load the data into memory and draw it. If you choose Cancel, it will not load the data. Unless you do something about it, E-MAP displays the themes in the order you added them in, so it's usually best to start with polygons and tiles, then add lines, points, and labels. You can click on **Stop Redrawing** to halt the current screen redrawing, or you can force a new redrawing by clicking on **Refresh Map**.

Sometimes, your choices lead to a query containing no records. (In the school example, you might have had a criteria with the PrincipalName="REWPOUFDS:LK", in which case you would get back an empty query). When this happens, E-MAP usually does not do anything other than display an error message. Sometimes, the query defines an

absurd area and the DataBounds and ViewBounds are reset to unreasonable values, in which case it's easiest to simply start over.

If the data of interest is spread over two or more tables, or needs to be limited more than can be done with the criteria box on the spatial query forms, use Access or some other database manager to define the query needed.

Adjusting what is displayed

By checking or unchecking a theme under the window brought up by Options=>Themes_Visible..., you can toggle whether the theme will be shown or not. This is useful if you have a theme containing lots of data that takes a long time to draw, and you just need to navigate. (Rivers, roads, and soil polygons cause this problem a lot.) By clicking on the scroll bars, you can adjust the ViewRegion. Numerous options under the View menu allow various adjustments of the view region. In the View=>Specify_View_Region window, you can enter a view region by hand, either by defining the bounding rectangle or the center point and the size of the view region. You need to hit the ENTER key after entering a number in one of the text boxes for it to take effect.

By clicking Options=>Themes_Order..., you bring up a window similar to the Load Themes and Save Themes windows, except that this will control plot order instead of what themes have been loaded or saved.

When E-MAP starts, it will automatically expand the DataRegion to include every theme that has been added. If you're looking at a county and have a statewide road dataset, this would be annoying. The solution is to load themes to define your area of interest, then make sure that Options=>Expand_Map_Area_to_Include_All_Data is unchecked. This only affects themes as they are loaded, so turning it back on later will allow each theme to expand the DataRegion as appropriate.

If re-display of the map is taking too long and you wish to adjust the view region by scrolling the scroll bars or by choosing View=>Shift_Right and View=>Zoom_In and such, unchecking Options=>Redraw_Map_Whenever_View_Shifts will cause E-MAP to not re-draw until you tell it to. If you will be looking at one area for a long time and moving back and forth between applications, you probably want to check Options=>Keep_Copy_of_Map_in_Memory to prevent long screen redraws.

If other applications are giving **Out of Memory Errors**, make sure Options=>Unload_Forms_After_Use_to_Free_Memory is checked. This keeps E-Map from using as much memory, but it does not keep track of your last choices for a theme definition so you can load a slightly modified theme definition.

Saving, Loading, Printing

Once you have designed a map theme you like, you can save its definition using menu File=>Save_Map_Theme.... You can load a previously saved theme selecting File=>Open_Map_Theme.... You can print to the default Windows printer by selecting File=>Print.... This brings up a dialogue window with blue rectangle indicating the area that will be occupied by the map; it can be changed by dragging and dropping the sizing handles. The map title can be set by typing in the text box at the top of the form. Once you have entered the title, double-click on the text box, then drag-and-drop the title onto

the page-layout box. Choosing **Printer Setup...** allows you to change the default printer and its orientation. Choose **OK** or **Cancel** to print or not print.

Spatial Querying with WHEAT E-MAP

Once you have displayed a theme with E-MAP, you may want to perform spatial queries, such as finding the river closest to a given point, or assemble a list of all points within a five-mile radius. With Wheat E-MAP, you do this by running a spatial query. To select the theme and type of spatial query, choose menu Options=>Spatial_Query... . When the dialog box is displayed, select the theme containing the data you wish to search, the type of spatial query, and click on **OK**. To turn off spatial querying, select None for the spatial query type and click on **OK**. When the map has redrawn, you can perform a spatial query by double-clicking on the map with the mouse. For more precision, you can enter the coordinates in the text box at the upper right, and click on run spatial query. Either way, E-MAP highlights the point the spatial query will be run with, and loads the data into the datagrid at the lower right corner. In addition, each time you run a spatial query, a new row (or several) are added to the datagrid in the Spatial Query Results window. You can copy these results to other Windows applications using the clipboard and standard Windows commands, or save the results to a table or a file by choosing menu options.

If you have many points on which you wish to perform spatial queries, you can store them in a table, load a theme to display them, load the other theme you want to run the spatial query with, then choose File=>Spatial_Analysis... . This will bring up a screen which will allow you to use the points from one spatial query to run a series of spatial queries on another theme. For instance, you could find the nearest river to each well by 1) loading the wells, 2) loading the rivers, choosing File=>Spatial_Analysis..., then specifying that you want to find the closest line in Lines: Rivers to each point in Symbols: Wells. The result is a new table containing the name of the point, the name of the other feature, of the original layers, a distance (if it makes sense), and a line connecting the two features (if it makes sense and was feasible).

Working with Other Applications

One of the main purposes in developing E-MAP was to allow other applications to analyze and manipulate data shared with WHEAT. A copy of the current map can be copied as a Windows Metafile to the clipboard for pasting into a graphics application like CorelDraw or PowerPoint. To do this, click on the map region once you have it looking the way you want; choose Edit=>Copy to copy the map to the clipboard; go into the other application; choose Edit=>Paste; and a copy of the map should appear. (If it is invisible or upside down, try ungrouping it. It turns out some things are not as standard as they could be.) To save the map into a Windows Metafile, choose File=>Export..., then supply a file name.

To copy textual data, like the results of a spatial query, to some other application, like a spreadsheet or word processor, run the spatial query, then choose the data you wish to copy (typically by clicking on the upper left corner of whatever grid contains the data you're interested in) then choosing Edit=>Copy or **Copy** . Go to the other application,

choose Edit=>Paste, and the data should appear laid out as a table with the columns labeled.

Adding Spatial Features with WHEAT E-MAP

In addition to displaying and querying pre-existing features, E-MAP allows you to add new features with the mouse. You do this by selecting Edit=>Add=>Lines or =>Points or =>Polygons. Then click the mouse on the desired location(s) to add (a) feature(s). If you are adding lines or polygons, using the space bar will end the current feature, and allow you to start a new one. To stop adding features, select Edit=>Add=>Stop. Remember to save your results to a file or table or copy them to some other application before quitting E-MAP. Lines and polygons, which are stored in ASCII format for readability, need to be converted to XYBLOBs before displaying in E-MAP using the included BLOBMEMO utility program, which allows conversion either way.

Keyboard Shortcuts While Viewing a Map

While the map window in E-Map has the focus, (if you don't know what that means, see the Windows manual or ask an experienced Windows user what that means and how to set the focus.) a number of keyboard shortcuts are available. The **arrow keys** can be used to move the cursor location, with plain arrow keys moving it one pixel, CONTROLled arrow keys moving it 5 pixels, and ALTed arrows moving it 20 pixels. Hitting the ESCAPE key while the map has the focus will halt a screen redraw, as will CTRL-BREAK. Hitting X will cause the current cursor location to be copied to the clipboard (as in "X" marks the spot).Hitting H will highlight the current point. As mentioned earlier, if you are adding lines or polygons using the mouse, using the space bar will end the current feature, and allow you to start a new one.

How to Design and Build a WHEAT Database

This document explains how to design a database for use with the Windows-based Hydrogeologic Exploration and Analysis Toolkit (WHEAT) and how to use the import-export tools included in WHEAT. It is intended as a planning reference in designing a database for use with WHEAT, and particularly for use with the electronic mapping application E-MAP.

This document assumes you are familiar with the use of E-MAP, the particular problems you or your client will be using WHEAT to solve, and somewhat familiar with the design of Microsoft Access relational databases. If you are unfamiliar with E-MAP, stop now and use E-MAP with the sample database; once you understand E-MAP from a user's perspective, come back here. If you are not familiar with the problems your client wishes to solve, sit down and talk. If you are totally unfamiliar with relational database design, buy a copy of Microsoft Access, and at least read the *Getting Started* manual before proceeding. It is much easier to use Wheat if a commercial quality database management package, specifically Access, is used.

Greg Pouch
Kansas Geological Survey
Tuesday, December 06, 1994

How WHEAT stores data

WHEAT uses data stored in a Microsoft Access format relational database for both the attributes and coordinates of various geographic features. The Microsoft Access database format is also the native database format for Visual Basic 3.0.

A relational database stores all information on a topic, such as water users or schools, in tables. Tables are organized sets of information containing records, one for each individual thing in the database. A record is composed of fields, which describe the thing. Thus, a table concerning schools might have records for each school in a county, and the records might include the name (as text), date founded (as date), operating budget (as a floating point number), date closed (as date), number of buses owned (as integer number), principal's name (as text), and number of pupils (as integer number). Notice that some of the fields listed, such as date closed and number of buses, will not occur for each individual school. In these cases, a relational database stores a null, indicating that no information is available.

While a relational database *stores* information in tables, it can *retrieve* information as virtual tables called queries. A query is the answer to a question asked about the datatable, such as: "show all schools with an operating budget above $500,000". A query can return data from several related tables by joining them. For example, suppose we had another table containing information about students. It might contain student's name, age, address, and name of school attended. We could then find out who a student's principal is by designing a query which relates the school name field in the student table to the name field in the school table. Queries are generally used for combining information from several tables, selecting subsets of records, or selecting subsets of fields. By allowing data to be stored in a single location (such as the principal's name in the school field) and related at a later time, relational databases can reduce storage requirements, and allow the updating of data to be considerably easier than if each record contained all data that might ever be needed.

E-MAP uses a relational database to store information about geographic features, both their descriptive (or tabular) attributes and their locational attributes. The location(s) of feature are described as X (east-west) and Y (north-south). WHEAT does not use any particular coordinate system, except that X and Y will be plotted as equal distances on the screen and on printing, and X increases to the right and Y increases to the top.

For points (for plotting symbols or text), WHEAT stores the X and Y locations in separate numerical fields in the table. These might be named something like Northing and Easting or X and Y, although, since the fields containing coordinates are chosen by the user while running E-MAP, they can be named anything. For features with multiple locations, like the set of line segments defining a river or the boundaries of a county, WHEAT stores the data in a special type of field called a Binary Large Object (BLOB or XYBLOB). This is simply a binary representation of a set of X and Y locations, stored as a chain of points.

The attributes and locational information can all be stored in one table for each set of items, or they can be stored in separate tables and related in a query. Additionally,

plotting instructions (like color, line width, and symbol) can be stored in a table or defined as constants.

Getting Data into a WHEAT Database

There are two main classes of geometric features handled by Wheat: 1) Points; and 2) Lines and Polygons. The former are geometrically described by a single coordinate pair, the latter by an ordered chain of coordinates. For points, the X and Y coordinates are stored as separate fields. For lines and polygons, the points on the line or boundary are all stored in a Binary Large Object field (referred to as an OLE field in the Access documentation and LargeBinary field in the Visual Basic documentation).

Other Windows Applications

If the data is coming from some other Windows or other PC-based application, the easiest approach is to export it as a comma-separated, quote-delimited ascii text file, and use the File_Import_Delimited option in Access to import it. There is no equivalent technique in WHEAT. For point data, simply include the coordinates as fields in the data file. A typical file might look like the following

```
'Well ID','rate','appropriation','use code 1','use cod 2','year','srate','intrs','inrs','ino','approp2','Easting UTM 14','Northing UTM 14'
'SF_033520',1000,240,'3','G','A79',0.57,1,0,1,240,520569.3,4.234481e+06
'SF_033447',335,24,'5','G','A79',0.57,1,1,1,24,537534.3,4.234521e+06
'SF_039054',1200,195,'3','G','A88',0.57,1,0,1,195,513611.8,4.234248e+06
'SF_027075',715,131,'3','G','A76',0.57,1,0,1,131,520121.3,4.234234e+06
'SF_026135',1000,183,'3','G','A76',0.57,1,0,1,183,508122.3,4.234142e+06
'SF_026134',810,183,'3','G','A76',0.57,1,0,1,183,508891.3,4.234143e+06
'SF_037847',1200,110,'3','G','A85',0.57,1,0,1,110,511288.9,4.234149e+06
'SF_039068',1200,85,'3','G','A88',0.57,1,0,1,85,511288.9,4.234149e+06
'SF_027281',900,228,'3','G','A76',0.57,1,0,1,228,512907,4.234152e+06
```

For line/polygon data, the coordinates should also be included as one field in the data file, but with some complications. The XY-chains need to be stored in a field with the coordinates in $X_1Y_1\ X_2Y_2\ \ldots\ X_NY_N$ order, with separators between the X and Y coordinates, and AFTER EACH XY PAIR, including the last one. The separators must be something other than commas, quotes and spaces. A good choice for XY separators would be a carat (^) , and a semi-colon (;) after each XY pair. This will be imported into a memo or text field in the datatable, and you will then convert it to an XY blob using BLOBMEMO.EXE (described below). An typical file might look like the following

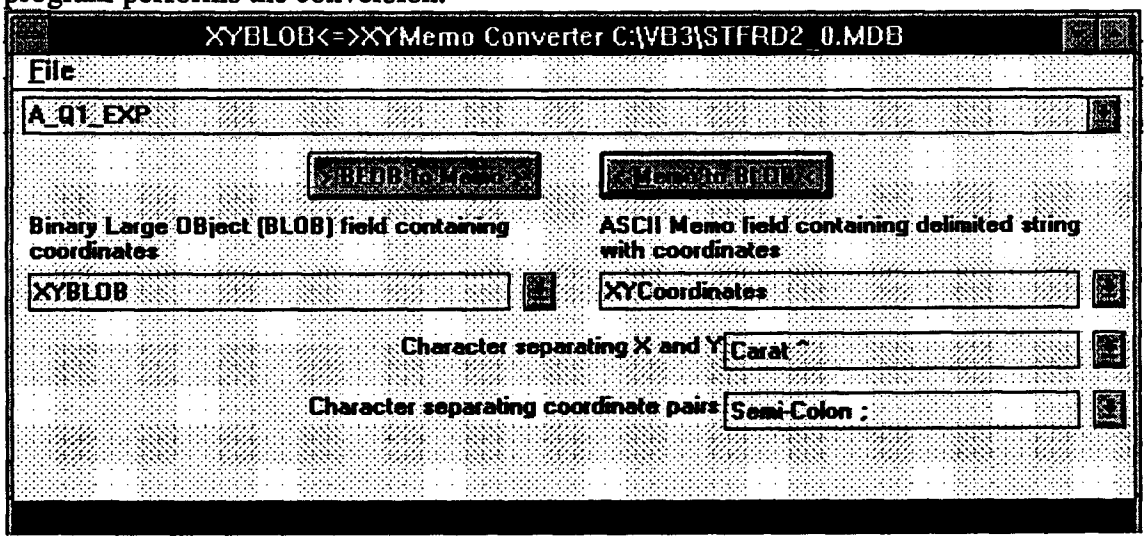
```
"CountyName","CNTYABBR","Easting UTM14","Northing UTM14","XYCoordinates"
"BARTON","BT",521460.8,4.258134e+06,"497180.3^4282925;497181.6^4282849;497184.1^4282415;497161.4^4282109;497166.
  4^4281445;497167.8^4281266;497172.8^4280577;497177.9^4279913;497179.1^4279657;497158.9^4279019;497162.8^427845
  9;497166.6^4278049;497170.4^4277514;497174.2^4276952;497151.4^4276569;497152.8^4276440;497155.3^4276083;497160.
  4^4275470;497164.1^4274961;498788.5^4282913;497180.3^4282925;"
"RICE","RC",568892.1,4.244567e+06,"545538.1^4234684;545717^4234685;546125.6^4234662;546201.8^4234663;546205.3^423
  4280;546209.6^4233745;546211.1^4233489;546249.4^4224887;546249.4^4224887;546428.4^4224913;547679.3^4224948;547
  780.9^4224948;548852.8^4224956;549414.1^4224962;550945.7^4224998;551023.2^4224999;552605.9^4224985;552682.1^422
  4985;553907.4^4225046;545514.7^4237901;545526.4^4236293;545538.1^4234684;"
```

This file has commas separating fields, semi-colons separating XY pairs, and carats separating X and Y. The Easting UTM14 and Northing UTM14 fields contain points in the middle of the county, which is useful for labeling them.

Once this file is imported into Access as a delimited text file, you would then use BLOBMEMO to produce an XYBLOB field that WHEAT can use for plotting a line/polygon.

Using BLOBMEMO

BLOBMEMO.EXE is a utility included in Wheat to allow the conversion of XY-chains representing lines or polygons from binary (BLOB) to ascii (MEMO) formats, and from ascii to binary. E-MAP uses only the binary format. Continuing the example above, suppose you have imported the county outlines into a table called A_Q1_EXP. You would start BLOBMEMO, open the database containing the data using FILE=>OPEN... , then choose the table or query containing the ascii XY-chains. Then you would make sure the desired MEMO field was selected in the list on the right, accept the default name provided or enter your choice of names in the list on the left, select the correct delimiter characters (in this case, a carat and a semi-colon), then click the **Memo_to_BLOB** and the program performs the conversion.



To convert a binary XY-chain (an XYBLOB) to an ascii XY-chain, you would select the database and file as above, select the fields or choose a new name for the output field as above, and choose the separators from the lists, then click the **BLOB_to_Memo** button, and the program converts the XYBLOB field on the left and stores the ascii output in the XYMemo field on the right.

ARC/INFO

For point data coming from ARC/INFO, just ADDXY to the PAT and UNLOAD it within TABLES or INFO. Then import it into Access using the delimited option, making sure to indicate that ' is used as the text delimiter. For line and polygon data coming from ARC/INFO, make sure the COVERAGENAME-ID is unique, UNGENERATE the coverage using the line or poly option and export the AAT and/or PAT delimited files with the UNLOAD command, then move everything to the PC. Once there, import the PAT/AAT using the same method described above. (You will probably have to edit field names to eliminate # , . , and - characters.) Then use the WHEAT application ARCS_IN2 to import the generate format files containing the arcs (and optional label points) into the

database. This will create a table with fields containing the XYBLOB and LineIDNumber. The LineIDNumber is COVERAGENAME-ID item from the AAT/PAT. To link the AAT/PAT to the geometry, create a query joining the two tables with a joining COVERAGENAME-ID to LineIDNumber. You can leave this query in place and produce maps based on the query, or you can make this a make-table query and store the results; the latter method is preferable.

ARCS_IN2 is used to import arcs from an ARC/INFO generate format file into a WHEAT format table in a database. This is done by starting ARCS_IN2, opening the database, opening the file containing the arcs, and choosing the File=>Import_Now menu item. If you have many generate files in the same directory, you can import them all at once using File=>Import_All_Files_In_Directory.., then choosing a wildcard string and the directory to search.

Fixed Width Files

Many mainframe and PC programs produce data files where the fields are not delimited by spacing or with other characters, but by position within a record. Most USGS databases produce output in this format, such as the file below. You can either use the Access File=>Import...Fixed Width menu option and type in the definition or use the Wheat program FXDWDTH1.EXE to import such files.

Example Fixed Width Data File

```
40000102000000125SC00104804AAC      SWNENES04T001SR048WS   4220.00WI      215
38083510200040001 1SC02204813BAA    NENENWS13T022SR048WS   3795.00WS       17.0
38131910200140001 1SC02104813CAA    NENESWS13T021SR048WS   3842.00WI      46.0
38120310200300001 1SC02104825BBB2    NWNWNWS25T021SR048WS   3842.00WI       44.0
38120310200300101 1SC02104825BBB2
38074210200320001 1SC02204824BBB    NWNWNWS24T022SR048WS   3786.00 U       12.0
```

Such programs almost always generate a description file such as the one below, which FXDWDTH1 takes advantage of. Simply edit the description file to look like the one below. Blank lines and anything after a ' are treated as comments. Field names and table names cannot contain blanks. The first active line contains the keyword TableName followed by the desired table name, the next active line contains the keyword Style followed by the layout keyword NameStartWidthType. All subsequent lines contain a field name, the starting column, the number of columns the data occupies, the type of data (integer, long, single, double, text, memo, date, or FORTRAN datatype names), and for text fields, the maximum length of text. See the on-line help for more details

Description File Example 1

```
'This is an example of a description file for FxdWdth. Even these lines, because they
'start with single quotes. This is example was generated by the USGS's GWSI database.

'The above line was intentionally left blank.
tablename WLSiteDescriptionColorado
style NAMESTARTWIDTHTYPE
    SiteID_StationNumber              1  15   text 16 'This originally had
'                   parentheses, which caused FxdWdth to crash. Note use
'  of single quote to start a comment.
    CountyCode                       16   3   integer
    LocalWellNumber                  19  24   text 24
    LandNetLocation                  43  23   text 23
    AltitudeLandSurface              66   8   real*4
    PrimaryUseOfSite                 74   1   character 1
```

Choosing Field Names

Any Access-allowed field name can be used by WHEAT, except that field names should not begin with ZQ. To preserve portability in case you should decide to move to another database management system, it is best to limit field names to letters and numbers and use no punctuation other than underscores (_). In particular, avoid commas, periods, parentheses, brackets, and anything that might look like a mathematical function or operator. The word "date" is reserved as an Access keyword, but "date_" works fine. Avoiding spaces in field names, and using capitalization to indicate word breaks is useful as well, as in naming a field AppropriationDate rather than Appropriation Date. It is safest if the lengths of field names are kept below 16 characters

When the user designs a theme, he/she is actually designing a Structured Query Language (SQL) query telling E-MAP which fields or constants to use for feature attributes like X and Y coordinates, symbol size, symbol code, and color. (SQL is a computer language used to handle databases. See the Access manual or any of many widely available books that discuss databases and SQL for more information.) These query statements are in text boxes on the right side of the form, and can be modified by a user interested in doing so. (If you want to use a calculated field, enclose the whole calculation in parentheses, and field names should be in square brackets.) A typical theme definition might look like

```
SELECT DISTINCTROW [County line] AS zqPolygonOutline, [PlotColor] AS zqFillColor, 0 AS
zqFillStyle, 0 AS zqOutlineColor, 2 AS zqOutlineWidth, format([CountyName]) AS
zqFeatureName  FROM [Counties]  ORDER BY [PlotColor] , 0 , 0 , 2 ;
```

The SELECT clause indicates series of field names and constants aliased to fields starting with ZQ. If you name a field zqFillColor and choose it as the fill color, it will result in circular definition problems. The FROM clause specifies that the data come from a table called Counties. The ORDER BY clause is used to speed redraws and minimize the size of graphics files- you should ignore it.

When using E-MAP, the program guesses at what fields might contain coordinates or plotting instructions based on the field name and the search strings contained in the [DefaultFields] section of the WHEAT.INI file. By carefully choosing field names, E-MAP can be very intuitive for the user.

Each of the keywords in the DefaultFields section ends with LookFor or ShowAlways. Not surprisingly, the LookFor lines contain strings that will be searched for in the list of fields and ShowAlways lines contain constants that will be included below the fields in the list. The constants and strings are separated by spaces, and each line should have one space after the last word. In the example below, any field containing the string FONT will be added to the symbol size list in the text labels and symbols dialogue windows, and the constants 10, 12, 18, 24, and 36 will always be shown.

```
[DefaultFields]
SizeLookFor=FONTSIZE HEIGHT WIDTH SIZE FONT
SizeShowAlways =10 12 18 24 36
ColorLookFor=COLOR COLOUR TINT HUE
ColorShowAlways=0<=>Black 255<=>BrightRed 65280<=>BrightGreen 16711680<=>BrightBlue 8421504<=>Gray
    16777215<=>White 65535<=>Yellow 33023<=>Orange
```

```
SymbolCodeLookFor=SYMBOL CODE
SymbolCodeShowAlways=45 51 33
XLookFor=X_ EAST WEST LONG
YLookFor=Y_ NORTH SOUTH LAT
FeatureNameLookFor=SERIAL NAME ID # NUMBER
XYLookFor=XY BLOB CHAIN LINE POLY BORDER BOUNDARY
FontCodeLookFor=FONTCODE FONT TYPE
FontCodeShowAlways=0 1 2 3
FeatureTextLookFor=TEXT LABEL NAME
FeatureTextShowAlways=Put_Your_Message_Here_In_DoubleQuotes
LineWidthLookFor=LINEWIDTH WIDTH LINE WIDE
LineWidthShowAlways=1 2 3 4 5 6
LineStyleLookFor=LINESTYLE STYLE LINE TYPE
LineStyleShowAlways=0<=>Solid 1<=>Dash 2<=>Dot 3<=Dash-Dot 4<=>Dash-Dot-Dot 5<=>Transparent 6<=>Inside_Solid
FillStyleLookFor=FILLSTYLE FILL STYLE INTERIOR INSIDE
FillStyleShowAlways=0<=>Solid 1<=>Transparent 2<=>Horizontal 3<=>Vertical 4<=>Upward_Diagonal 5<=>Downward_Diagonal
    6<=>HVCross(+) 7<=>DiagonalCross(x)
AlignmentLookFor=ALIGN ORIENT
AlignmentShowAlways=0<=>Left_Top 2<=>Right_Top 6<=>Center_Top 8<=>Left_Bottom 10<=>Right_Bottom
    14<=>Center_Bottom 24<=>Left_Baseline 26<=>Right_Baseline 30<=>Center_Baseline
```

By having X_ as the first entry on the XLookFor line and naming all fields containing X coordinates X_Center or X_Well or even just X_, the program will correctly choose those fields as the default X coordinate where an X coordinate is called for. If the field name does not appear in the proper lists, the user can still drag the field name to the list.

Spatial Querying with WHEAT E-MAP

Once you have displayed a theme with E-MAP, you may want to perform spatial queries, such as finding the river closest to a given point, or assemble a list of all points within a five-mile radius. With Wheat E-MAP, you do this by running a spatial query. To select the theme and type of spatial query, choose menu Options=>Spatial_Query... . When the dialogue box is displayed, select the theme containing the data you wish to search, the type of spatial query, and click on **OK**.

Unique Names Needed for Spatial Querying

When E-MAP loads data on a feature into a theme, it stores a feature name. When E-MAP performs a spatial query, it gets the feature name from the theme, then runs a query looking for all data in records where that feature name occurs and displays the results in the datagrid and spatial query results window. Because of this, it is important that tables contain a column of data suitable for performing this link. This should be one column with unique values, such as a serial number. THE FEATURE NAME SHOULD BE UNIQUE or you will end up returning incorrect results. A one-field primary key would be best.

One-To-Many Spatial Queries

When the user saves a theme definition, E-Map writes the SQL statement that defines the graphical data and the SQL statement that defines the tabular data to be retrieved by a spatial query in fields MapSource and DBSource in table zWHEATtblThemes. You can modify these definitions to suit your needs. For instance, if we had county outlines that we wished to display from a table called Counties, but

census data in a table called CountiesCensus which contains census data for each decade for each county (a one-to-many relate between counties and census data), and we want to be able to select a county and retrieve census data, we could add a new record to the zWHEATtblThemes table, set the MapSource field to

```
SELECT DISTINCTROW [County line] AS zqPolygonOutline, [PlotColor] AS zqFillColor, 0 AS zqFillStyle, 0 AS
  zqOutlineColor, 2 AS zqOutlineWidth, format([CountyName]) AS zqFeatureName  FROM [Counties]  ORDER BY
  [PlotColor] , 0 , 0 , 2 ;
```

and set the DBSource field to

```
SELECT DISTINCTROW *, format([CountyName]) AS zqFeatureName FROM [CountiesCensus]  ORDER BY [CensusYear];
```

and the ThemeName field to "Counties with Census Data", then loading that theme will allow us to perform the desired one-to-many spatial query. This technique is useful whenever many observations might be associated with a single geographic feature, such as water levels with a well or soil layers with a soil series polygon.

Adding Spatial Features with WHEAT E-MAP

In addition to displaying and querying pre-existing features, E-MAP allows you to add new features with the mouse. You do this by selecting Edit=>Add=>Lines or =>Points or =>Polygons. Then click the mouse on the desired location(s) to add (a) feature(s). If you are adding lines or polygons, using the space bar will end the current feature, and allow you to start a new one. To stop adding features, select Edit=>Add=>Stop. Remember to save your results to a file or table or copy them to some other application before quitting E-MAP. Lines and polygons, which are stored in ASCII format for readability, need to be converted to XYBLOBs before displaying in E-MAP using the included BLOBMEMO utility program, which allows conversion either way.

How to Contour in WHEAT

This document explains the use of the contouring tool (contour1.exe) included in the Windows-based Hydrogeologic Exploration and Analysis Toolkit (WHEAT). The contouring tool takes scattered measurements with X, Y and Z values from a table or query, grids them, then contours them. The grid is saved as a WHEAT rastergrid, and the contours are saved to a new table whose fields include the contour, its value, and a serial number. This paper is intended as a how-to introduction to contouring for geologists and assumes you already have a database. If you do not already have a database, read the E-MAP manual first, then build a database using the guidelines suggested in *How to Design and Build a Wheat Database.*

Greg Pouch
Kansas Geological Survey
Tuesday, December 06, 1994

Background and Terminology

A contour is a line that passes through a set of locations where some variable has a particular value. On a topographic map, a line connecting points where the elevation (the variable or Z-value) is 2100' (the particular value) is a contour. The Z-value can be any quantifiable variable, such as water level, bedrock elevation, temperature, or magnetic field strength intensity. By convention, X measures distance in the east-west direction and Y measures the north-south position, with X increasing to the east and Y increasing to the north. On topographic maps, the interval between contours is fixed at the contour interval; this need not be so and the contour-tool does not require it.

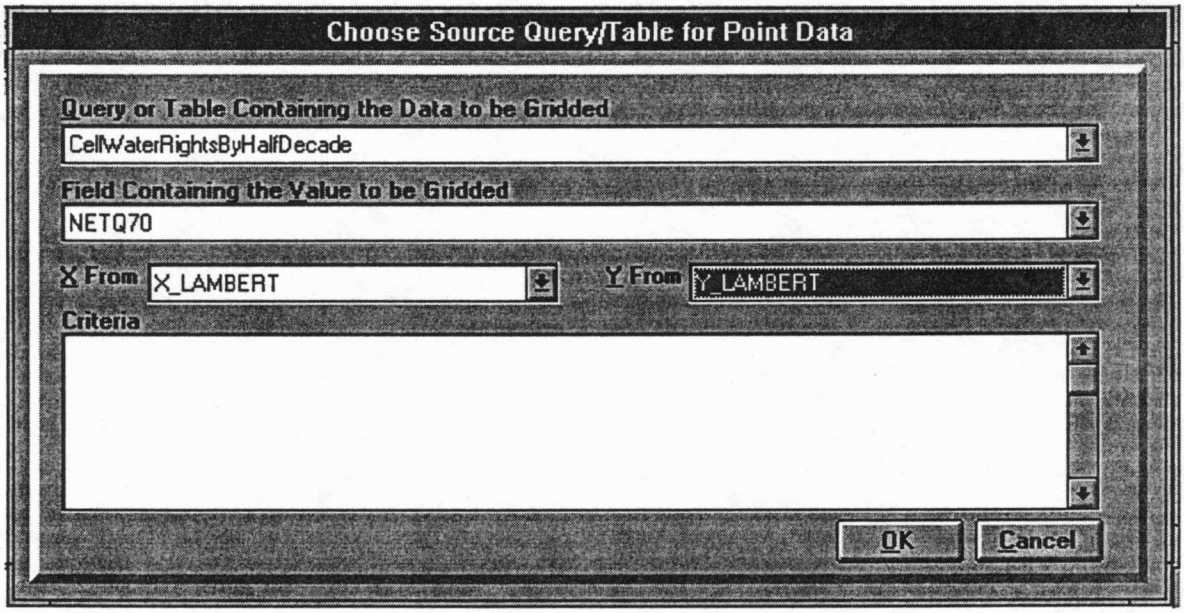
Procedure

Open the database

1) Open the database containing the data you wish to contour by choosing File=>Open_Database...

Choose Field to Contour

2) To contour scattered XYZ data already in the database (the only supported option) Choose File=>Open_Wheat_Data..=>Scattered_XYZ_from_Query/Table....That will bring up the following dialog box. Choose the name of the table of query containing the data you want to contour, the field you want to contour in the dropdown list labeled Field Containing the Value to be Gridded, and the fields containing X and Y (often X_Something and Y_Something, or Easting and Northing) in the appropriate lists. Then click on the **OK** button.



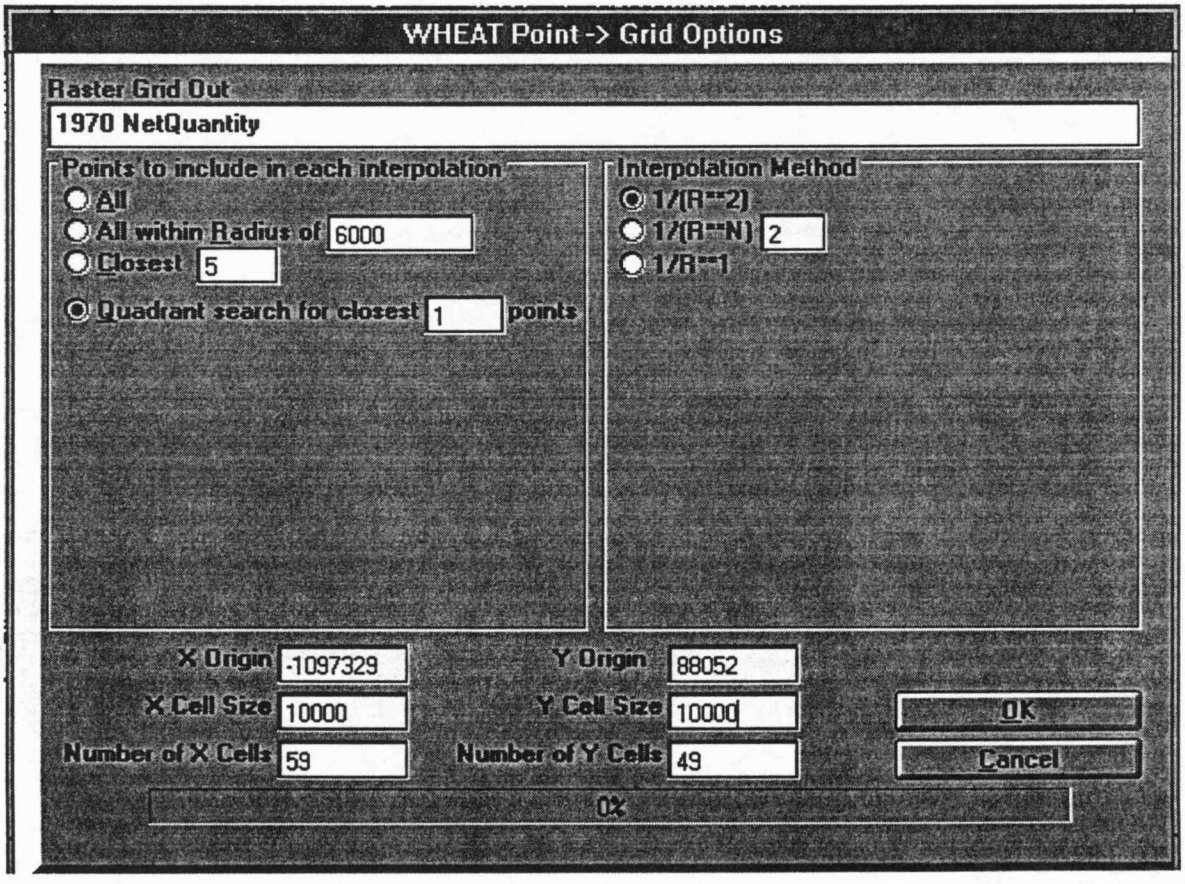
If you wish to stipulate additional selection criteria add these to the Criteria box. For example, suppose you have already chosen a query containing all water levels, Z, X and Y, and now wish to limit the contouring to only water levels whose date is between 1 August 1984 and 5 May 1985. Then you would add the text

([DateMeasurement] between #8/1/1984# and #5/5/1985#)

to the criteria box. (Read the Microsoft Access manual or any text on SQL for details of setting criteria.)

Choose Gridding Method

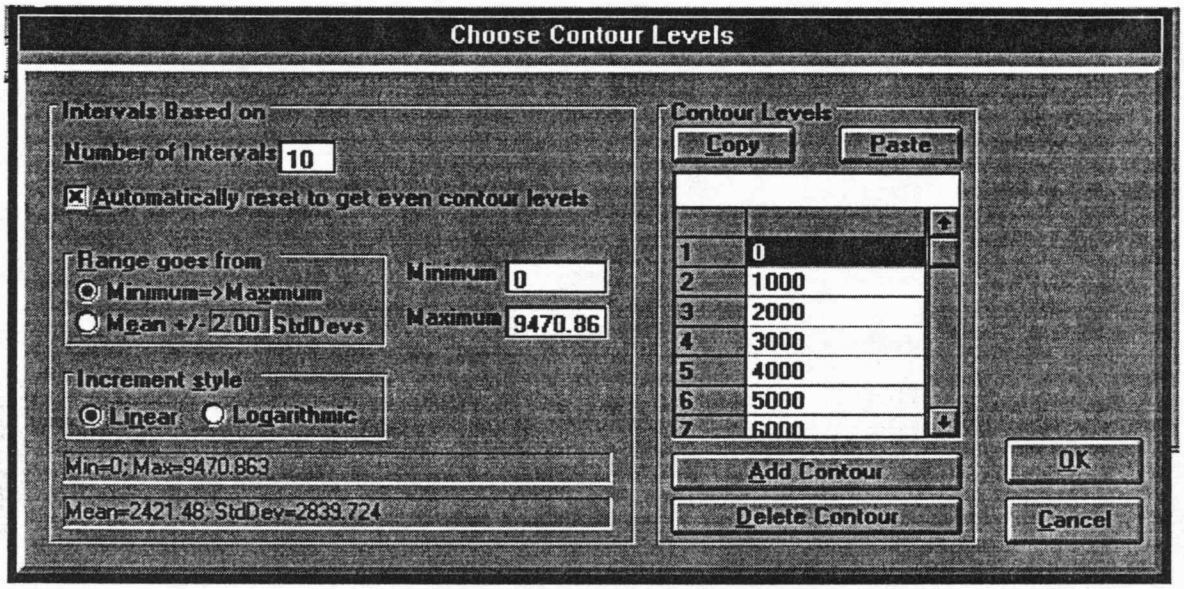
3) That will bring up the following dialog box, which asks how you want to grid the data. First, choose a convenient, descriptive name for the rastergrid you will be generating. In the example below, the gridded net quantities will be stored in a rastergrid called "1970 NetQuantity" and the contours will go into a table called "Contours from 1970 NetQuantity". Next, choose which points to include in interpolating and how to interpolate. For noisy data, I use Quadrant search for closest 2 points and $1/R^2$ interpolation; with less noise, Quadrant search for closest 1 points and $1/R^2$. In both the X and Y directions, you can choose either how many grid cells to have, or how far apart their centers are, to control the speed of execution (few cells) or the smoothness of the lines (more cells, spaced closely). To set the X cell spacing to 10000, type 10000 in the textbox labeled X Cell Size, then hit return. The Number of X Cells textbox is updated to reflect this change. If you enter the Number of X Cells, then hit return, the X Cell Size is updated. When you are done making selections, click on **OK** to accept or **Cancel** to abandon contouring.



The program then grids your data. **This often takes a long time, so this is a good time to get some coffee.**

Choose contours

4) After gridding you data, the program asks for the contour levels. You can change the number of contour levels and the limits. Usually, twenty linearly-spaced contours from the minimum to the maximum values is a good idea.



You can choose to bound your contour range with the minimum and maximum Z-values, or the mean +/- some number of standard deviations, or manually override the minimum and maximum.

You can change the number of contours: if the Automatically reset... checkbox is checked, the number of contours will be decreased to get even contour ranges.

You can also choose linear or logarithmic spacing of contours (Logarithmic contours would be most useful for chemical concentrations or other variables that have a very large range).

You can also manually add contour levels by clicking **Add Contour**, highlighting the newly-generated, blank contour level, entering the desired level in the textbox, and hitting return. You can remove a contour level by selecting the contour level, then clicking **Delete Contour.**

Finally click **OK** or **Cancel** and the program will produce contours and store them in a table which can later be viewed in E-MAP.

At this point, you are done contouring.