

1996/18

c2

AGSO

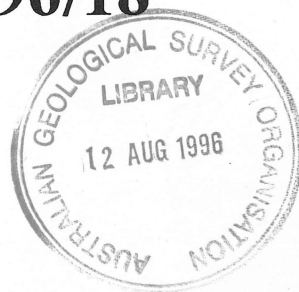
# VISUALISING WATERBORE DATA WITH ARCVIEW2

BMR PUBLICATIONS COMPACTUS  
(LENDING SECTION)

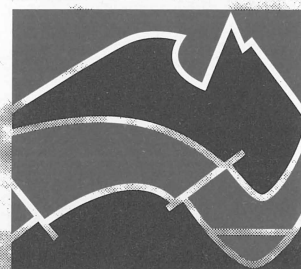
BY

**R. GALLAGHER**  
**GISOLUTIONS**

**RECORD 1996/18**



**AGSO**



**AUSTRALIAN  
GEOLOGICAL SURVEY  
ORGANISATION**

BMR comp  
1996/18  
c2

**RECORD 1996/18**

**Visualising waterbore data with ArcView2**

by

**Robyn Gallagher  
GISolutions**

**Land and Water Resources  
Australian Geological Survey Organisation**



\* R 9 6 0 1 8 0 1 \*

## DEPARTMENT OF PRIMARY INDUSTRIES AND ENERGY

Minister for Primary Industries and Energy: Hon. J. Anderson, M.P.

Minister for Resources and Energy: Senator the Hon. W.R. Parer

Secretary: Paul Barratt

## AUSTRALIAN GEOLOGICAL SURVEY ORGANISATION

Executive Director: Neil Williams

© Commonwealth of Australia 1996

ISSN: 1039-0073

ISBN: 0 642 24969 5

This work is copyright. Apart from any fair dealings for the purposes of study, research, criticism or review, as permitted under the *Copyright Act 1968*, no part may be reproduced by any process without written permission. Copyright is the responsibility of the Executive Director, Australian Geological Survey Organisation. Requests and inquiries concerning reproduction and rights should be directed to the **Principal Information Officer, Australian Geological Survey Organisation, GPO Box 378, Canberra City, ACT, 2601.**

## Contents

Introduction	1
The data model	1
Setting up the data model in ArcView2	2
The one-to-many- tools	3
General tools	8
The Avenue scripts	9
Conclusions	17



## Abstract

This report describes a method for visualising complex borehole data using a desktop GIS. The data can be stored in any tabular data base structure, and are displayed and analysed using new software modules added to the ArcView2 desktop GIS package.

The report deals with the borehole data model, which is a series of tables attached to points locating the bores, in the traditional 'one-to-many' situation. It also describes how ArcView is able to use that data model, and some analysis tools implemented in Avenue.

The package is relevant in many geological situations, such as drillholes and field sampling programmes, where many pieces of information need to be associated with each location on the ground.

## Introduction

The Western Water Study (Wiluraratja Kapi) is attempting to improve water supply to Aboriginal lands in the arid zone of central Australia. It is a cooperative program over two years between AGSO, the Northern Territory Power and Water Authority (PAWA), and the Central Land Council (CLC). A major task is to link the information systems of the three organisations in an Arc/Info geographic information system for the pilot study area and to view the data using the simpler ArcView2 system.

The NT Power and Water Authority's newly-commissioned database of waterbore information HYDSYS is the source of stratigraphic and groundwater information for the Western Water Study (Wiluraratja Kapi). The data is accessible as dBase tables. This borehole data shares characteristics associated with many other geological data sets such as mineral deposits, drillholes, and field sample sites. The spatial data are represented as points on (or below) the ground, with other tables containing additional descriptive information linked to the points by some identifying field such as a sample or hole number. This linked information is measured many times for the same point either over depth intervals, over time, or for many different elements/commodities: a situation referred to in database theory as a **one-to-many** relationship.

Because one-to-many relationships are so common in geology, this contract was an opportunity to find a methodology that would be generally useful for handling such data in ArcView. The software developed may be obtained from the project leader for the Western Water Study, Gerry Jacobson, at the Australian Geological Survey Organisation.

## The Data Model

The borehole **spatial data** must be stored as points in a form recognised by ArcView (an Arc/Info coverage, an event theme or a shapefile) then loaded into an ArcView view as a theme with an appropriate name (I use *bores* in my ArcView demonstration and in this report). The points must have at least two attributes: the identifier used to link the points to the relevant information in other tables; and an elevation value. These attributes will be visible in a table which ArcView names "Attributes of theme" (eg *Attributes of bores*).

The **other tables** must also be in a format useable by ArcView: dBase IV is the most convenient format since it is ArcView's native format; INFO is a possibility; CSV text has some reduced functionality in ArcView; Access, Oracle etc are possible, but the user must make them visible to ArcView. The tables should be added to an ArcView project, and given meaningful names (eg *borehole geology*, *borehole stratigraphy*, *borehole water quality*). Each of these tables must contain a field, such as a hole identifier, which links records in the table back to the borehole points. This field may have a different name in each table, and a different name from the field in the point attribute table.

## Setting up the Data Model in ArcView2

Information about the theme and its tables is communicated to ArcView via a **text file**, which can be created using any editor. The file contains a very simple description of the data model, and ArcView stores the information it contains for use in all of the new tools which work on one-to-many data.

The text file contains one line for the theme, and one line for each of the linked tables. There may be as many linked tables as the user wishes. The information the file stores is the following:

```
view_name, theme name, link field, elevation field, SWL field
table name1, table type 1, link field1, from field1, to field1, label field1
table name2, table type 2, link field2, from field2, to field2, label field2
.
.
table name n, table type n, link field n, from field n, to field n, label field n
```

where

view name	is the name of the view containing the theme
theme name	is the name of the theme
link field	is the field used to link the point to information in other tables
elevation field	is the field containing the elevation of the point
SWL field	is the field containing the most recent standing water level (or "none")
table name	is the name of a linked (one-to-many) table
table type	is a code for the type of the table:
	1 the table contains information which is measured over depth
	2 the table contains information which is measured over time
link field	is the field in that table containing an identifier which corresponds to the link field of the point
from field	is the name of the field in the linked table containing the starting depth or the date of measurement
to field	is the name of the field in the linked table containing the end depth or the time of measurement
label field	is any field in the linked table to be used for labelling intervals in cross-sections etc

The following is an example of such a text file but other table entries may be added:

```
View1,bores,rego_no,elev_ahd,bed_swl
borehole geology,1,reg_no,from,to,rock_type
borehole stratigraphy,1,reg_no,from,to,symbol
borehole drillers log,1,rego_no,lith_from,lith_to,lith_desc
borehole water quality,2,station,date,time,date
borehole drill,1,rego_no,drl_from,drl_to,drl_type
borehole casing,1,rego_no,cas_from,cas_to,cas_type
borehole screens,1,rego_no,scr_from,scr_to,scr_type
borehole water cuts,1,rego_no,bed_from,bed_to,bed_ph
```



An Avenue script, called *Generic.SetupLinks*, is used to read the data model file and store its contents as a list of lists within the project. The script is not attached to a button, as it only needs to be used when the data model is created or modified.

## The One-to-Many Tools

Once a data model has been stored in a project's Object Tag, it is available via a number of new tools attached to View buttons. Six of these buttons are in the Button Bar, and three are in the Tool Bar. For convenience they will all be referred to as **tool buttons**. Following ArcView convention the utilities in the Tool Bar require some user interaction with the theme in the view, whereas the utilities in the Button Bar operate on the currently selected set of features, or all features if none are selected.

Most of the utilities require the user to pick which linked table is to be used for the current operation of the tool. A list of tables, extracted from the data model, is presented when the tool button is selected with the mouse, so that the user can choose a table. The tool continues to operate on that table until a tool button is clicked again to allow the user to choose a different table. Some tools may optionally operate on the attribute table of the theme instead of on a one-to-many table. While a single tool operation is restricted to one table, sequential queries can be done on different tables.

When tools create graphics, the graphics are stored in a temporary view. The view may be saved in a project or included in a layout for printing, but disappears when it is closed.

The tools are the following:

### Information

This tool is just like the existing ArcView Information Tool, except that it also displays records from a linked table. When a user points to a theme feature with the mouse, the visible fields in the theme attribute table, and all entries in the linked table for that point, appear in an expanded Information window. The main entry appears first in the lefthand side of the window, followed by an entry for each "many" record, titled with its "link field" and "from field" entry. The user can pick any of these entries, to see it in full in the righthand window.

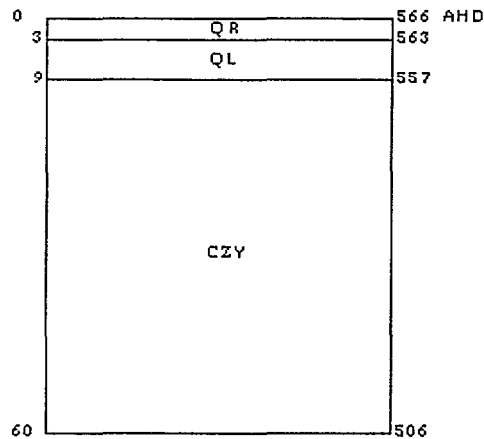
### Draw Information

This tool draws a picture of the Information for a feature, instead of listing fields from its tables. If there is no information in a linked table for a given feature, nothing is drawn.

For tables of type 1 (those containing depth intervals), this results in a simple downhole log, drawn in a temporary pop-up graphics window. The contents of the log depend on the linked table selected by the user when the tool is invoked. When a user

points to a theme feature with the mouse, the data model information is used to find the “from field” and “to field” of the linked table for drawing downhole intervals, as well as the “label field” for labelling them. The drawings are temporary - when a window’s pin is pulled it is deleted.

Here is an example of a drawing:



For tables of type 2 (those containing date and time), a time series graph is drawn in a temporary pop-up graphics window. The contents of the graph are determined when the user picks an additional numeric field to be graphed. When a user points to a theme feature with the mouse, the data model information is used to find the “date” and “time” fields of the linked table.

The time series picture is a simple bar graph showing the value of the nominated field for each different value of date+time. The drawings are temporary - when a window’s pin is pulled it is deleted. If the time series can be created just for “date”, **Charts** create much more useful documents than this simple drawing.

#### Select Many

This tool is almost identical to the standard ArcView Spatial Selection Tool, which already makes use of any linked tables. This version of the tool simply ensures that the linked table is open, with its selected records promoted to the top

#### Draw a Cross Section

This tool creates a schematic picture for a line of bores. It is not a true cross-section in that the intervals are not joined between bores. The picture is drawn in a temporary pop-up graphics window, which is deleted when the window’s pin is pulled.

The information plotted for each table is similar to that portrayed in the Draw Information tool: the data model is used to find the “from field” and “to field” of the linked table for drawing downhole intervals, as well as the “label field” for labelling



The table used for this calculation may be any of the linked tables, or the main bore attribute table. A selection must be made from the list of available tables.

If the main bore attribute table is picked, the user must simply choose a numeric field in that table (eg *bed\_swl*) for use in the calculation. If a one-to-many linked table is chosen, the user must construct a query which defines the records in that table which are to be used (eg *era* = "CAINOZOIC"). The tool finds the greatest value of the "to field" (ie the bottom of the deepest interval) for each selected bore, where a record satisfies the query, and calculates its reduced value. For intervals which are only partially-penetrated by the bore (considered as those whose "to field" is within a metre of total depth) the label includes a "less than" symbol eg <50.

The label uses the current text font, size and colour, all of which may be set or changed using the Palette.

#### Label with Thickness ♦

This tool is similar in operation to the preceding one in that it also adds a graphic label to each selected feature, or to all features if none are selected. The numeric value used for the label is the depth or thickness of some chosen feature (eg depth of SWL or the thickness of Cainozoic stratigraphy) by the bore elevation.

The table used for this calculation may be any of the linked tables, or the main bore attribute table. A selection must be made from the list of available tables.

If the main bore attribute table is picked, the user must simply choose a numeric field in that table (eg *bed\_swl*) whose value is plotted. If a one-to-many linked table is chosen, the user must construct a query which defines the records in that table which are to be used (eg *era* = "CAINOZOIC"). The tool finds the difference between the greatest value of the "to field" and the smallest value of the "from field" for each selected bore, where a record satisfies the query, and plots that thickness. For intervals which are only partially-penetrated by the bore (considered as those whose "to field" is within a metre of total depth), the label includes a "greater than" symbol eg >5.

The label uses the current text font, size and colour, all of which may be set or changed using the Palette.

#### Label with Most Recent \*<sup>1.1</sup>

This tool only operates on linked tables with a date and time field (type 2). It adds a graphic label to each selected feature, or to all features if none are selected, where the label is taken from the record in the linked table with the most recent date. If more than one record at any point has the same date, the time field determines 'most recent'.

The user must select a table from a list of all tables containing a date field, then a label field from a list of all that table's fields.

The label uses the current text font, size and colour, all of which may be set or changed using the Palette.

### Main Aquifer



This tool derives a one-to-one table of 'main aquifers' for selected bores. Because it is quite slow, it only operates if there is a selected set imposed on the theme (that selected set may be all the bores, but the user must explicitly select them).

The main aquifer of a bore is defined as the stratigraphic unit (ie the record of the stratigraphy table) which is intersected by the water bed with the lowest (non-zero) ec value. Since many bores are missing such ec values, the tool can not generally be used to assign aquifer stratigraphy, which is unavailable in HYDSYS.

The user is presented with a list of all linked tables, so that he can select the table of water beds, then with a list of the numeric fields of that table so that he can select the field containing the ec value.

A second picklist of tables is presented for the user to select the stratigraphy table.

Finally, the user must nominate the name of the (dBase) table which will be created to hold the main aquifers. By default it will be called *aquifers.dbf*. The table has fields containing the bore identifier, the "from field" and "ec field" of the water beds table, and all fields from the stratigraphy table.

For each selected bore, the entry in the water beds table with the lowest (non-zero) ec value is found: the starting depth of the interval (or the end depth if the starting depth is zero) is recorded. Then the interval in the stratigraphy table which contains that depth is found, and this record is the main aquifer.

Whenever a bore has such an entry in the stratigraphy table, a main aquifer entry is created for it in the output table.

### Super Bore Picture



This tool draws all available information for a bore. The picture and text are drawn in a temporary pop-up graphics window, which is deleted when the window's pin is pulled.

The information about the bore is accessed by finding the link and depth fields for each table from the data model stored in the project's Object Tag. If a bore has no information in any of the linked tables, the space for that information is empty in the bore picture.

From the lefthand side of the picture the following information is presented for each depth interval where it is available:

- \* a 'ruler' measuring depth downhole, with a marker for hole depth
- \* text information about rock type, description and colour from the geology table

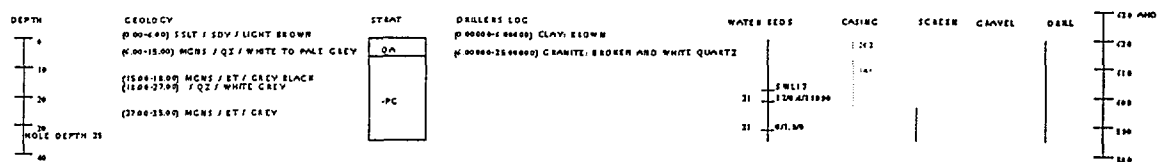
- \* a block diagram of units from the stratigraphy table
- \* text information about lithology from the driller's log
- \* a marker for SWL from the main bore table, and swl/yield/ec values for points down the hole from the water beds table
- \* casing intervals and diameters, colour-coded for casing type
- \* screen intervals, colour-coded for screen type
- \* gravel intervals, colour-coded for gravel type
- \* drill intervals, colour-coded for drill type
- \* a 'ruler' marking reduced AHD downhole.

Colour-coding for the casing, screen, gravel and drill types uses the HYDSYS codes, for which explanations were unavailable. The colour conventions are:

code: 0	colour: black	type: 1	colour: red
2	green	3	blue
4	yellow	5	cyan
6	magenta		

This tool makes no effort to resolve situations where pieces of information overlap on the drawing. It is the least flexible tool and is specific to the set of table names used. To make a better cartographic product graphic text and objects may be selected, moved or re-sized.

Here is an example of a super bore picture:



## General Tools

### Create a Color Palette (an option in the Window menu)

This tool really has two parts: an Arc/Info AML which writes the RGB components of a shadeset to an ASCII file; and an ArcView utility which reads that ASCII file and creates a color palette. However the ArcView utility will read any ASCII file of the correct format, no matter how it was created.

By default, ArcView reads Arc/Info shadesets as fill palettes, not color palettes, hence the need for this utility.

The AML utility is very simple. It must be run from Arcplot:  
&run write\_shades <shadeset> <no\_symbols> {filename}

The arguments are the name of the shadeset (e.g. *agso94*), the number of symbols to be considered ( max 999), and optionally the name of the ASCII file. If that name is omitted, the file will be named from the shadeset, with *.txt* extension.

This utility ONLY works for shadesets defined via RGB or CMYK, NOT those like *colornames* which vary from device to device, and for which Arcplot is unable to return the colour components.

The ASCII file it creates contains one record per shade symbol, where that record contains 3 values representing the Red, Green and Blue components of the symbol, each separated by a blank character.

The ArcView utility prompts the user for the name of the ASCII file described above (which must have a *.txt* extension) and the name of the output palette, which receives a *.avp* extension. It creates an ODB file which is available for the user to load into ArcView via the Palette Manager, just like other available palettes. The utility does NOT load the palette.

### Colour a Theme using a Lookup Table



This tool mimics Arc/Info's ability to symbolise a data set using a lookup table (LUT) which maps field values to symbol numbers. It is intended for colouring themes such as geology, when large amounts of time have been invested creating Arc/Info LUT's.

The tool uses the **current color palette**: the user must have already loaded the correct palette.

The utility expects the View to have one active theme, on which it operates. The user is prompted for the field in the theme table on which the legend is to be based (e.g. *map\_symbol*, *polylabel*, *igds-text*), then the name of the lookup table. The LUT must contain a field with the same name as that just specified in the theme table. Finally, the user must pick an integer field in the LUT which contains the symbol number (e.g. *symbol*)

The theme is classified uniquely on the field picked for the legend, then the LUT is used to set the colour of each different feature by finding the symbol corresponding to each classification.

## The Avenue Scripts

### Generic.SetupLinks

This script is used whenever a new one-to-many data model must be added to a project, or if a data model is changed.

It prompts the user for the name of the text file containing the data model, and reads that file. Each record (line) in the file is stored as a list of strings, for example the first line of the file described above becomes {"View1","bores","rego\_no","elev\_ahd","bed\_sw1"}.

All of those lists are put into another list. This list of lists is stored in the Object Tag of the Project GUI. This is done because Object Tags are persistent, ie are saved with the project, so are available whenever the project is opened. It avoids having to read the text file each time the project is used.

The script also clears and resets all links between the theme and the related one-to-many tables. Links are persistent in a project so this process only needs to be carried out when the data model is created or altered.

While reading the text file, the script checks for the existence of the tables and fields referred to in the file. If any object does not exist in the project, the script stops with an error message.

#### Bore.SelectTable

This script is attached to a Click event for the three utilities in the ToolBar. It allows the user to pick which linked table is to be used for the tool.

The script retrieves the data model from where it is stored as a list of lists in the Object Tag of the Project GUI. From this information it constructs a list of available linked tables which can be presented to the user in a MsgBox window.

When the user has picked a table, a pointer to it is made available to the tool scripts via a global variable called *\_codeno*.

#### Bore.SelectTable2

This script allows a user to pick which linked table of type 2 is to be used for a tool. It is used for tools like Label with Most Recent, which operate only on tables containing a date field.

The script functions similarly to *Bore.SelectTable*, except that it constructs a list containing only type 2 tables.

#### View.IdMany

This script is derived from the View.Identify script provided with ArcView. It is the Apply script for the new Information Tool.

The script enlarges the Identify window to allow easier viewing of the listed information.

It retrieves data model information for the relevant linked table from the ObjectTag of the project, using global variable *\_codeno*.



The tool reports on all bores within a circle of one pixel radius around the location pointed to by the mouse. The information in the bore's attribute table is posted in the Identify window, then all entries in the linked table for that bore are found and listed, differentiated by the value in the "from field".

This script uses some temporary table selections, so on conclusion all the table selections are reset to their status on entry to the script.

### View.DrawInformation

This is the Apply script for the Draw Information Tool.

It retrieves data model information for the relevant linked table from the Object Tag of the project, using global variable \_codeno.

The tool draws pictures of all bores within a circle of one pixel radius around the location pointed to by the mouse. Each picture is created in the graphics layer of a new temporary View named from the linked table, the bore "link field" and the "label field". These views have a RemoveDoc(SELF) shutdown script so that each is deleted when its pin is pulled.

The contents of the picture depend on the type of the table: type1 tables contain depth information so a downhole picture is created; type2 tables contain information measured over time, so a time series graph is created.

To draw a downhole picture, the script extracts the elevation of the bore from its attribute table, then finds all entries in the linked table for the bore. Depth labels are constructed using the "from field" and the "to field" of each record, and the value of the "label field" is placed as the interval label. Each downhole interval is represented as a rectangle with a label at its centre; depth labels are placed to the left of the column; elevation AHD is placed to the right.

The time series picture requires the user to nominate an additional numeric field to be graphed. All entries for the bore in the linked table are graphed, even those with zero values of the variable. A simple bar graph is created, each data marker being labelled with the date ("from field") and time ("to field") corresponding to the reading.

This script uses some temporary table selections, so on conclusion all the table selections are reset to their status on entry to the script.

### View.SelMany

This script is derived from the View.SelectPoint script provided with ArcView. It is the Apply script for the new Select Many Tool.

It retrieves data model information for the relevant linked table from the Object Tag in the project GUI, using global variable \_codeno.

Selection is done by pointing at a single feature with the mouse, or by drawing a box. If the shift key is held down, the selected features are added to the current selection set.

ArcView automatically selects records in linked tables, so the script only needs to ensure that the window for the linked table is open and that the selected records are promoted to the top of the table.

### View.CrossSection

This is the Click script on the Draw a CrossSection Button. It runs the *Bore.SelectTable* script directly (because it is a Button not a Tool), so that the user can pick a linked table for drawing and labelling the downhole intervals.

The script retrieves data model information for the linked table from the Object Tag in the project GUI, using global variable `_codeno`.

This utility operates only on a selected set of features. If there are no points selected in the bore theme, the script exits.

The user must pick which boreholes define the endpoints of the cross-section line. The coordinates of the bores are placed in a list and sorted in ascending order, along with their link field values.

The sorted link values are presented in a MsgBox, so that the user can pick the hole which starts the line. The point with the smallest easting will be the default, being first in the list.

The list is then sorted in descending order, and those sorted link field values presented for the user to pick the hole which ends the line. The point with the largest easting will be the default, being first in the list.

A vertical exaggeration factor must be supplied in a MsgBox.Choice. The default is 1.

The cross-section picture is drawn as graphics in a temporary view named by the user. This view has a RemoveDoc(SELF) shutdown script so that it is deleted when its pin is pulled.

Each hole is represented as a vertical black line, with horizontal ticks marking depth intervals, standing water level and water cuts. There are text labels (which vary in content according to which linked table is represented) on the left side of the vertical line and numeric labels on the right of the vertical line showing hole elevation, depth values, SWL and water cuts. The holes are not joined by horizontal lines. Each hole is labelled with its "link field" value, which should contain some sort of a bore identifier.

The script loops over the selected points in the bore theme, projecting their position perpendicularly onto the cross-section line. The records in the linked table belonging

to each hole are found, and their depth values extracted for drawing. The bore identifier, elevation and SWL are obtained from the bore attribute table.

Finally, simple “rulers” are drawn at the left side and bottom of the view canvas. The vertical values represent depth, and the horizontal values measure distance along the cross-section line from the first hole.

#### View.LabelReduced

This is the Click script on the Label with Reduced Values Button.

It does not make use of the *Bore.SelectTable* script, since the user may wish to use the main bore attribute table rather than a linked table. Therefore the script retrieves data model information from the ObjectTag in the project GUI, and constructs its own picklist of available tables which includes the bore attribute table. The user selects a table from that extended list.

If some bores are selected, only they are labelled. Otherwise all points are labelled. The labels are created as graphic text in the View, using the current text size, font and colour.

The action of the script depends on whether the user picks the bore attribute table for calculating reduced values, or one of the linked tables.

In the first case, a list of all numeric fields in the bore attribute table is created and presented in a MsgBox. The user must pick a field from that list. Then all points are labelled with the reduced value = bore elevation (AHD) - field value.

If a linked table is to be used, a standard ArcView query window is presented, so that the user can create a logical expression. This query defines the records in the linked table which cover a depth interval of interest. For example, the query *[desc] = "CAINOZOIC"* would select all records covering the Cainozoic era in the linked stratigraphy table. For each bore, the record with the largest “to field” value (ie the deepest interval) is found, and the point labelled with the reduced value = bore elevation - to field.

This script uses some temporary table selections, so on conclusion all the table selections are reset to their status on entry to the script.

#### View.LabelThickness

This is the Click script on the Label with Thickness Button. It is almost identical to *View.LabelReduced*.

It does not make use of the *Bore.SelectTable* script, since the user may wish to use the main bore attribute table rather than a linked table. Therefore the script retrieves data model information from the ObjectTag in the project GUI, and constructs its own picklist of available tables which includes the bore attribute table. The user selects a table from that extended list.

If some bores are selected, only they are labelled. Otherwise all points are labelled. The labels are created as graphic text in the View, using the current text size, font and colour.

The action of the script depends on whether the user picks the bore attribute table for calculating thickness values, or one of the linked tables.

In the first case, a list of all numeric fields in the bore attribute table is created and presented in a MsgBox. The user must pick a field from that list. Then all points are labelled with the value of that field.

If a linked table is to be used, a standard ArcView query window is presented, so that the user can create a logical expression. This query defines the records in the linked table which cover a depth interval of interest. For example, the query *[desc] = "CAINOZOIC"* would select all records covering the Cainozoic era in the linked stratigraphy table. For each bore, the records with the smallest "from field" and the largest "to field" value are found, and the point labelled with the thickness value = to field - from field

This script uses some temporary table selections, so on conclusion all the table selections are reset to their status on entry to the script.

#### View.LabelMostRecent

This is the Click script on the Label with Most Recent Button. It labels points with the most recent value of a field, based on fields containing date and time.

It makes use of the *Bore.SelectTable2* script, which retrieves data model information from the ObjectTag of the project and constructs a picklist of available tables which have type = 2. The user selects a table from that list.

A second picklist, containing the fields of the selected table, is constructed. The user selects a labelling field from that list.

If some bores are selected, only they are labelled. Otherwise all points are labelled. The labels are created as graphic text in the View, using the current text size, font and colour.

For each point to be labelled, the record in the linked table with the greatest value of the date field is found. If two identical date values exist, the record with the largest value of the time field is considered the most recent.. The label field of this record is used to label the point.

**Note:** Currently, the date and time fields are considered to be integer numbers, since this is the convention in the existing PAWA water quality table. Date is in the North American format, eg 19890327.

This script uses some temporary table selections, so on conclusion all the table selections are reset to their status on entry to the script.

### RGBPalette

This script is activated from the pull-down *Windows* menu in all document GUI's.

It initiates a dialogue whereby the user supplies the name of a .txt file containing the RGB definition of a Color Palette, and a second dialogue to supply the name of the .avl file which will contain the new color palette.

The script simply reads the ASCII file, extracting Red, Green and Blue values (0 to 255) from each record, These values define a colour. The colours are added to a list, from which an ODB file is made.

### MainAquifer

This script is the click script of the Main Aquifer Tool. It must compare several tables and is quite slow, therefore it only works if the theme has a selected set.

After retrieving data model information from the ObjectTag of the project, the script builds a list of all available linked tables so that the user can pick the table which contains stratigraphy and the table of water beds. For the water beds table, a list of numeric fields is built up, from which the user must pick the field representing the ec values.

Since a new dBase table of 'main aquifers' is created, the file dialogue menu provides a browser for naming and storing that table on disk. The new table contains fields from the water beds table (the bore identifier, the from field, and the ec field) as well as all fields from the stratigraphy table.

A preliminary selection is placed on the water beds table, to reduce it to all records containing either a non-zero "from field" or "to field". This is necessary because the water cuts appear to be arbitrarily assigned to either field.

For all selected points in the theme, records from the water beds table with a non-zero value of the ec field are found, and the smallest value of ec determined. If that record has a non-zero "from field" its depth is used to find an interval in the stratigraphy table; otherwise the "to field" depth is used. The record for the bore in the stratigraphy table which has a depth interval containing the water bed depth is selected as the main aquifer. Its record, along with the relevant fields of the water beds record are added to the new table.

A single record is added to the main aquifer table for every selected bore which has an entry in the water beds table and a corresponding interval in the stratigraphy table.

### SuperBorePicture

This script is the click script of the tool which draws the 'super bore picture'. It uses information from most of the linked tables.

**Note:** The names of the tables used to retrieve the bore information are explicitly stored in the script:

*borehole stratigraphy*

*borehole geology*

*borehole drillers log*

*borehole water cuts*

*borehole casing*

*borehole screens*

*borehole gravel*

*borehole drill*

These names were those decided on at AGSO: if the table names are changed in the project, the script must be modified.

Using the hardcoded table names, the remaining link information from the theme to the tables is read from the data model stored in the project's Object Tag. If any table name is not found in the data model lists, the script stops with an error message.

The tool draws pictures of all bores within a circle of one pixel radius around the location pointed to by the mouse. Each picture is created in the graphics layer of a new temporary View named with "Bore Picture" concatenated to the bore "link field" and the "label field". These views have a RemoveDoc(SELF) shutdown script so that each is deleted when its pin is pulled.

Any entries for the current bore are found in each of the linked tables containing stratigraphy, geology, drillers log, water beds, casing, screens, gravel, and drill. This is a straightforward, repetitive process. For any table, if information is present, it is added to the picture at the depth given by the "from field" or for the interval from the "from field" to the "to field". The information is drawn approximately in columns, arranged across the view from left to right, with an appropriate heading

STRAT is represented as boxes containing text from the nominated "label field".

GEOLOGY information is text from the fields *rock\_type*, *rock\_desc* and *color*, written at the "from" location of the interval..

DRILLERS LOG text is from the *lith\_desc* field, again written at the "from" location of the depth interval.

The WATER BEDS column consists of labels on a vertical line extending to the bottom of the hole. The SWL is marked if there is a non-zero value in the "swl field" of the main bore attribute table. If there are entries in the linked water beds table, each of these is marked and labelled on the right of the vertical line with a string made up of the fields *bed\_swl/bed\_yield/bed\_ec*. If any of these fields is null, its value is represented as "-", eg 25.6/-/2. This label is matched on the left of the vertical line with the value of the "from field", or if that is missing the "to field".

The CASING is drawn as a series of vertical lines for each interval over which casing data is present, with a label for the casing diameter. The colour of the vertical line and

the label represents the casing type (*cas\_type*), which is used as a colour code as described below.

SCREEN, GRAVEL and DRILL are all drawn as vertical lines for each interval over which data is available. The colour of the vertical line segments represents the code from the given table (*scr\_type*, *gvl\_type*, *drl\_type*), which is used as a colour code as described below.

The colour coding of lines and labels is set in a pre-determined colour list at the start of the script: code 0 is black; code 1 is red; code 2 is green; code 3 is blue; code 4 is yellow; code 5 is cyan; code 6 is magenta. More codes will require extra colours.

A vertical line with horizontal tics is drawn at each end of the picture. The ruler at the left represents intervals marking downhole distances, measured from 0 at the top of the hole. The ruler at the right is labelled with AHD values.

#### AddWQAliases

This script is not attached to any button or menu. It simply provides an easy way of adding aliases to fields in the water quality table from a lookup table. It can be run whenever necessary by the data administrator.

The script has the table names inline: *borehole water quality* and the lookup table *wqvarlup.dbf*. If table or field names are changed, the script must be altered and recompiled.

The script creates a list of the fields in the water quality table, and loops over them searching for those names in the *variable* field of the lookup table. Then an alias for that field is created from the values in the *name* and *units* fields of the lookup table.

### Conclusions

Because one-to-many relationships are so common in geology, the methodology developed for this project should find much wider use in many other GIS applications. The one-to-many tools which have been written in Avenue can be modified or extended to help answer other questions we may pose of data of this type.