



Invest to Save Budget

Improved Air Quality Forecasting  
Invest to Save Report ISB52-09

**Visualisation of Dual Doppler Lidar data  
Software User Handbook**

**By**

**S Siemen and AR Holt**

University of Essex

19<sup>th</sup> February 2004



© QinetiQ Copyright 2004

## Authorisation

**Prepared by** Dr RI Young  
**Title** Visualisation of Dual Doppler Lidar data

**Signature**

**Date**

**Location** QinetiQ Malvern

**Principal authors** Dr S Siemen  
**Appointment** Previously Senior Research Officer, Dept. Of Mathematical Science  
University of Essex

**Principal authors** Prof AR Holt  
**Appointment** Professor, Dept. Of Mathematical Sciences  
**Location** University of Essex

## Record of changes

Issue	Date	Detail of Changes
0.10	19/02/04	First Draft Release
0.12	25/02/04	Revised Draft
1.0	17/03/04	First Release

## **ABSTRACT**

This report ISB52-09 was produced under Project 52 of the Invest to Save Scheme, or ISB. The objective of this project is to improve the ability of air quality forecasts and thus impact on the quality of life in and around urban areas.

As part of the project, the *Propagation and Remote Sensing Research Laboratory* at the University of Essex undertook a study to determine how the measured lidar data and the retrieved products can be best visualised. For this purpose a software suite, called *DAViS*, was developed, which could load lidar data and then visualise it by two-dimensional plots and on an interactive three-dimensional display. The software also allows interactive processing and display in real-time. This software was first described in report ISB52-08 and is now ready for general release.

This report provides an illustrated user manual for the DAViS software.

## Table of Contents

<b>1.</b>	<b>Hardware platforms and DAViS installation</b>	<b>6</b>
1.1	Hardware platforms and DAViS installation requirements	6
1.2	Running DAViS	6
<b>2.</b>	<b>Loading data</b>	<b>7</b>
<b>3</b>	<b>Visulisation</b>	<b>9</b>
3.1	Two-dimensional visualisation	10
3.2	Changing the colour tables	11
3.3	Saving data into an ASCII file	11
3.4	Three-dimensional visualisation	11
<b>4.</b>	<b>Data Analysis</b>	<b>13</b>
4.1	Velocity Azimuth Display (VAD)	13
4.2	Line of Sight (LoS)	14
<b>5.</b>	<b>Conclusion</b>	<b>15</b>
	<b>Appendix A How to install and compile DAViS</b>	<b>16</b>
	<b>Appendix B The DML data format</b>	<b>16</b>

# 1 Starting DAViS

## 1.1 Hardware platforms and DAViS installation requirements

As described in milestone report 8, a software package, called DAViS, was written in ISB-52 [1] to visualise the recorded lidar data and retrieved products. To be able to run this software package, there are some requirements on the computer system running the software which are detailed below.

The software was initially written to run on a Unix system, but has now operated successfully on SGI graphics workstations (this is the platform on which it was developed) and on a Linux PC with a modern graphics cards. To visualise a large amount of data, such as the measured lidar data, in real-time and in three-dimensions, an optimized OpenGL graphics card is required. While these are standard on SGI graphics workstations (which was the reason for choosing them as the development platform) they are not standard on cheaper PC Linux workstations. However, in the last years the price of hardware accelerated graphics cards for PCs has dropped significantly.

To speed up the development of DAViS some third party software packages were used. These have to be installed to enable to run DAViS properly. The packages are:

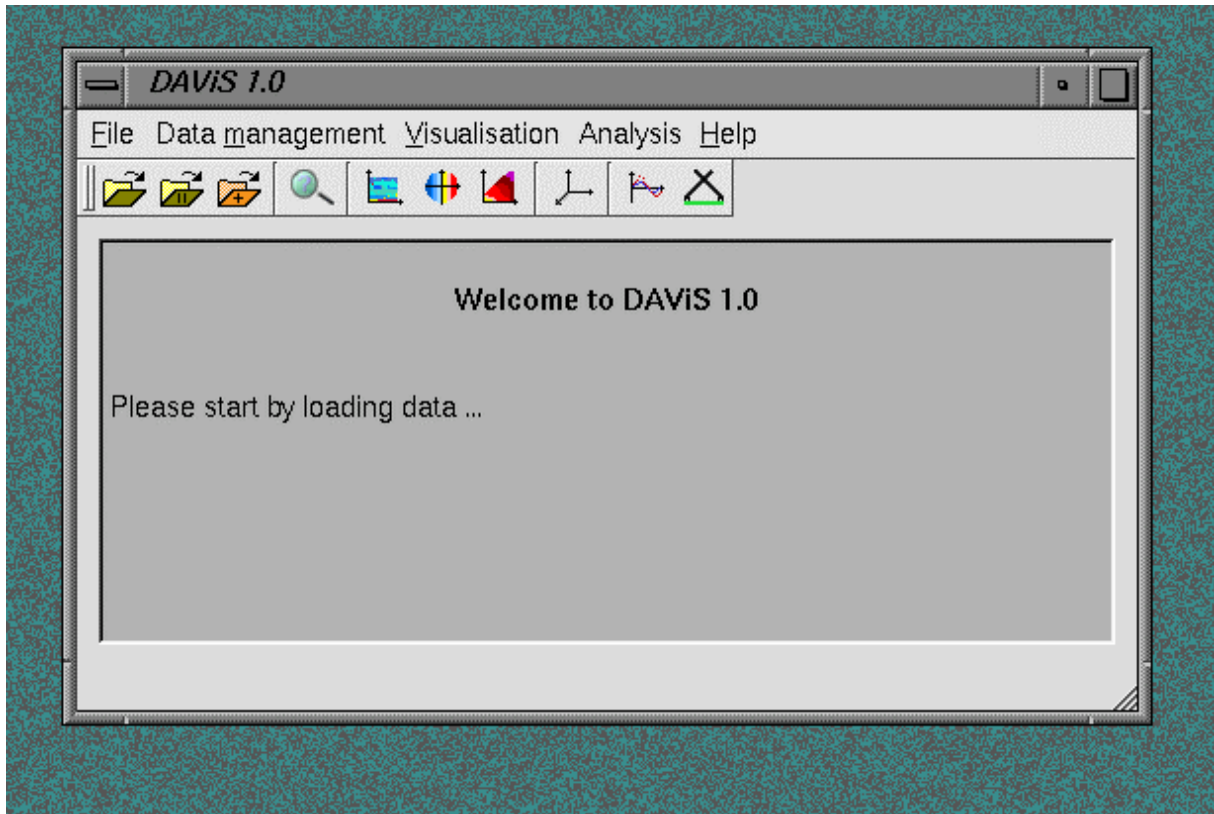
- **Trolltech's Qt library** – version 3.2 or newer for the GUI and the 2D graphics. The library is free as long software for commercial profit is not produced – [www.trolltech.com](http://www.trolltech.com).
- **Motif or Lesstif** – version 2.0 and newer. Needed by Open Inventor for the GUI.
- **SGI Open Inventor** – version 2.0 or newer. Library to visualise three dimensional scenes. Older version free from SGI on SGI Irix and Linux (<http://oss.sgi.com/projects/inventor/>). Newer versions available through Coin ([www.coin3d.org](http://www.coin3d.org)).
- **OpenGL** – standard library for three-dimensional graphics. Needed by Open Inventor. Installation is often part of graphics card installation.

To use DAViS it is recommended that the source code is recompiled, to take advantages of any optimisation the system is providing. To do so, a C++ compiler is needed which supports templates and the STL (Standard Template libraries). The GNU Gcc compiler of versions 3.0 and higher does so on most Unix systems, as does the MipsPro Compiler under SGI Irix.

## 1.2 Running DAViS

Instructions to install DAViS are given in Appendix A. Once installed DAViS can be started by either typing *DAViS* & on the console or double clicking an icon on the desktop. An icon on the desktop can be achieved by a link in the *Desktop* sub-directory in the home area.

To use underlying (street-)maps in the three-dimensional display the file containing the map needs to be at a fixed location. Preferably in the directory where the software is started.






**Figure 1:** DAViS main window.

When DAViS is started a main window from which to call all functions to load, visualise and analyse the lidar data is presented. There are two ways to use DAViS, either by using the top menu or using the icons in the toolbar. For most actions DAViS will print feedback in the main area of the window.

## 2. Loading data

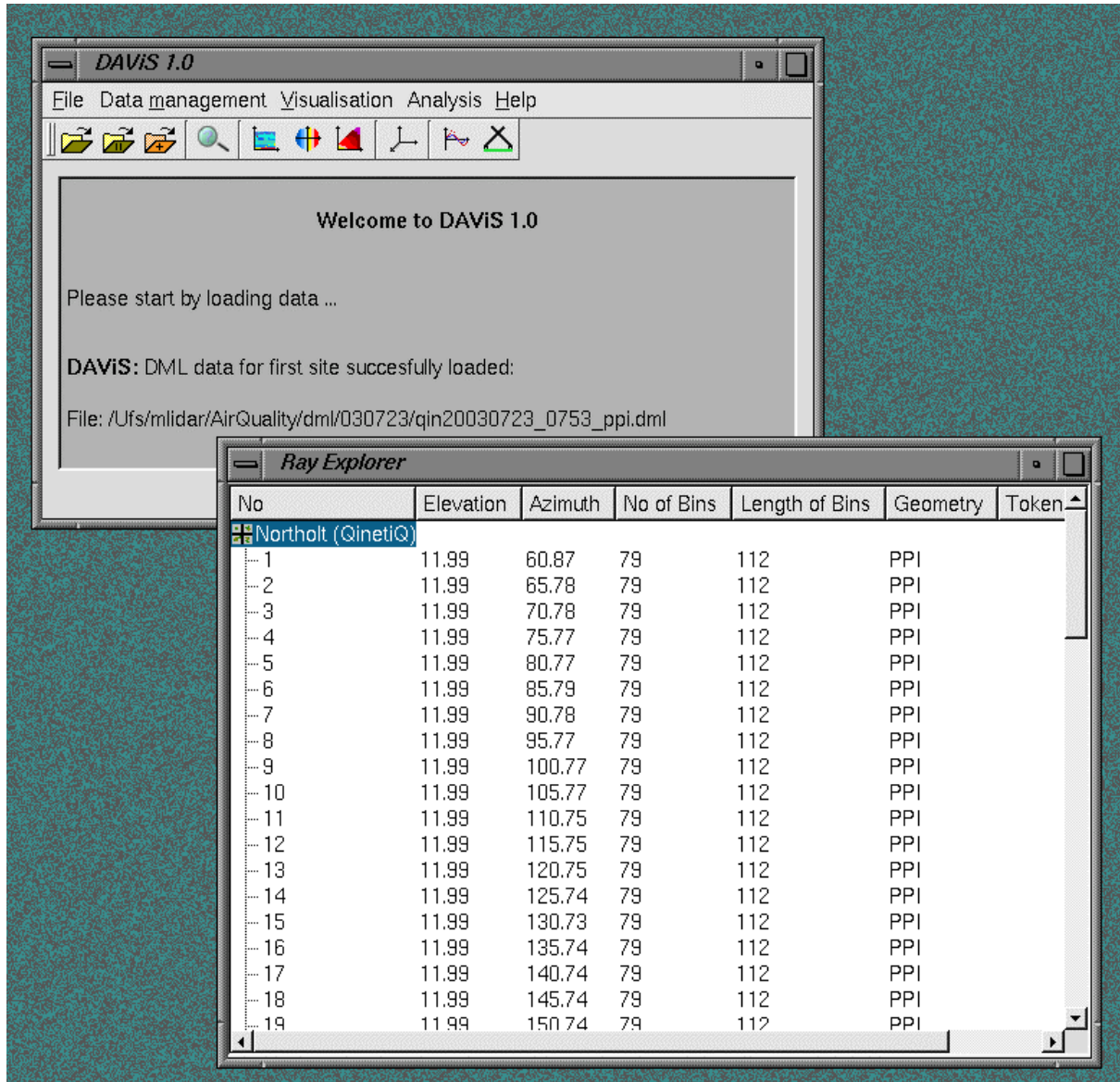
The first thing to do, after DAViS is started is to load the data belonging to the measurement.

There are three *Open-File-Dialogs* which have the icons , , and . The first icon should be used to load the lidar data of the first site and information retrieved from the first site. The second icon (indicated by a "II") should be also only used for the lidar data measured by the second lidar and its retrieved products. The third button should be used for other data files, such as wind information or NAME model outputs which are not related to the two lidar sites. After pressing the buttons a window opens to select a file name. Select file from directory *dml*, since only DML files with the ending *\*.dml* can be loaded.

**If a mixture of PPI and RHI scans are to be loaded then it is necessary to load the PPI first.**

After a lidar data file is selected another dialog box will pop-up and ask for an offset for the radial Doppler velocity data. This offset should lie between -1 and 1 m/s and will be **added** to all velocity data of the previous selected file [2].

It is possible to check that the correct data has been loaded through the option of the *Data management* menu or by pressing in the main window. Figure 2 shows an example of the *Ray Explorer* display. To enable the software to detect the start of a new scan in a multiscan dataset, tokens are used. Every token signals the start of a new scan. **If it is intended to start a new data set and the previously loaded data is to be discarded, then select the *clean up all data* option from under *Data management* in the main menu.**



**Figure 2:** The ray explorer can be used to check that the data was loaded correctly.



### 3. Visualisation

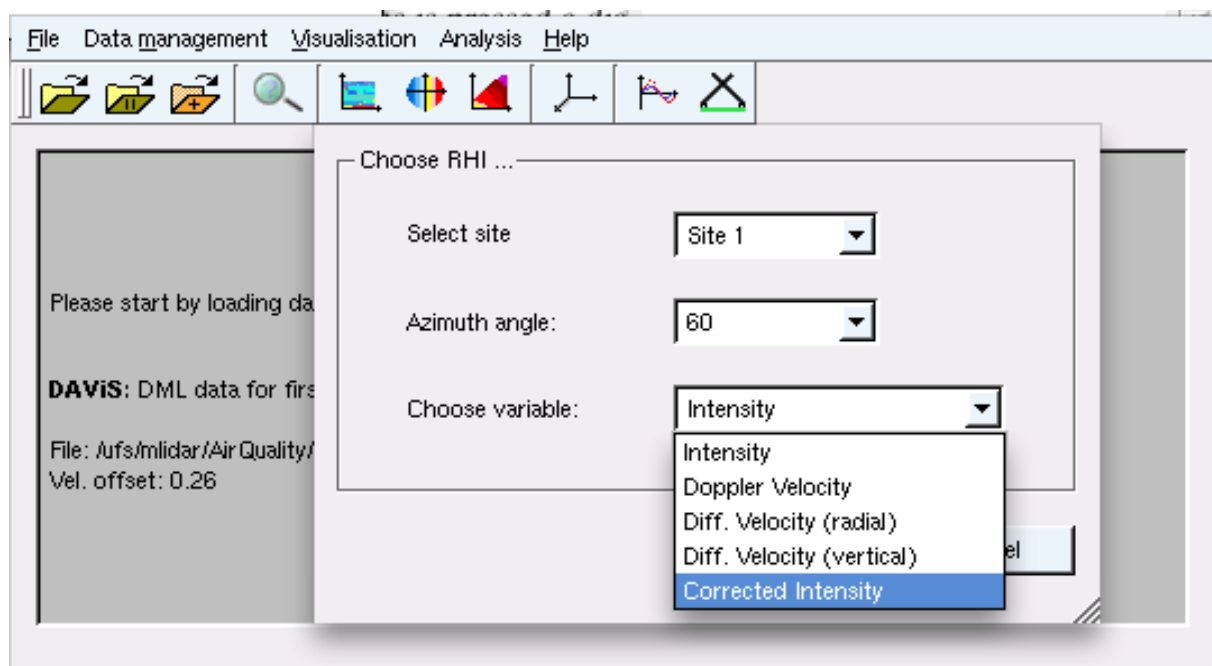
The main function of DAViS is to visualise measured lidar data or outputs from numerical models. The visualisation is divided into two groups: two dimensional displays of data from one lidar and three-dimensional displays of dual lidar data.

#### 3.1 Two dimensional visualisation

DAViS offers various options to view the loaded data in two dimensions:

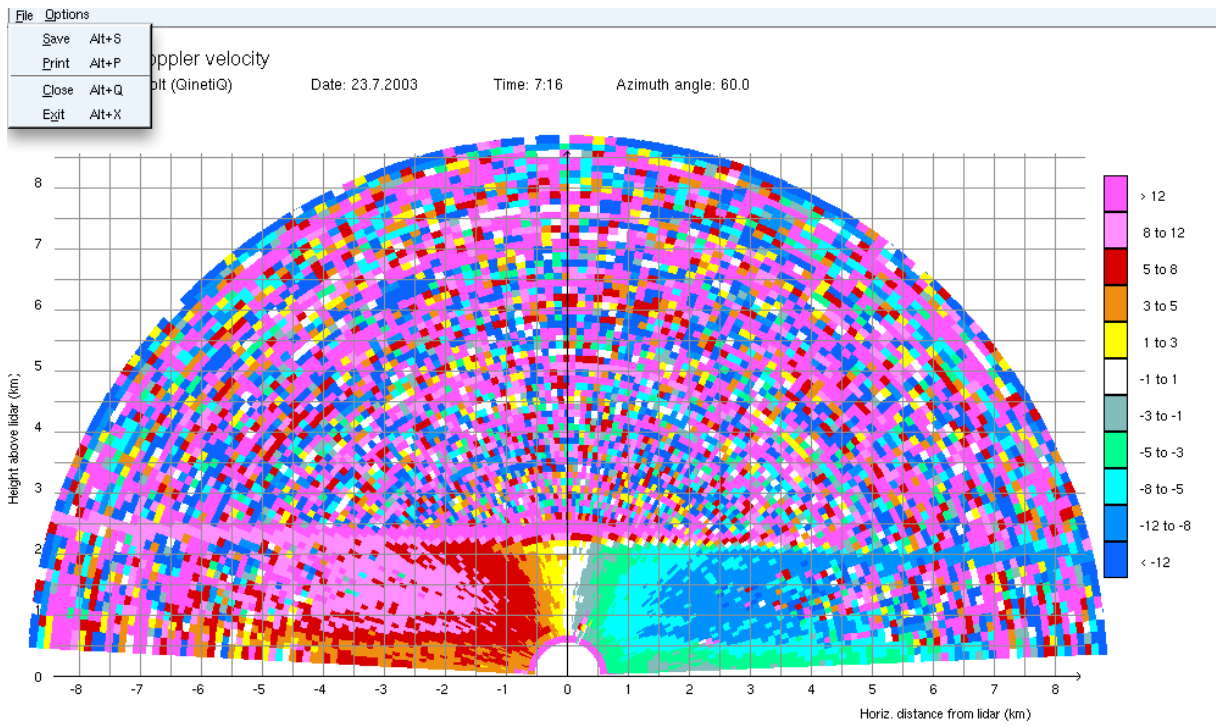
- Time series plot of rays.
- PPI plot (cone geometry)
- RHI plots (vertical plane)

After any of these options is selected a dialog box appears to ask from which lidar ( *Site 1* or *Site 2* ) the user wants to display the data and which variable ( *Intensity* ,*Doppler velocity* etc.) should be displayed. After completing the dialog box options a separate window opens which shows the requested data. For all of these windows the user can choose to print or save the pictures by using the window menu, or the menu which appears after pressing the right mouse button whilst the mouse pointer resides over the window surface. Some data sets also contain more than one sweep. The user can switch between these sweeps by using the menu that appears when the right mouse button is pressed.

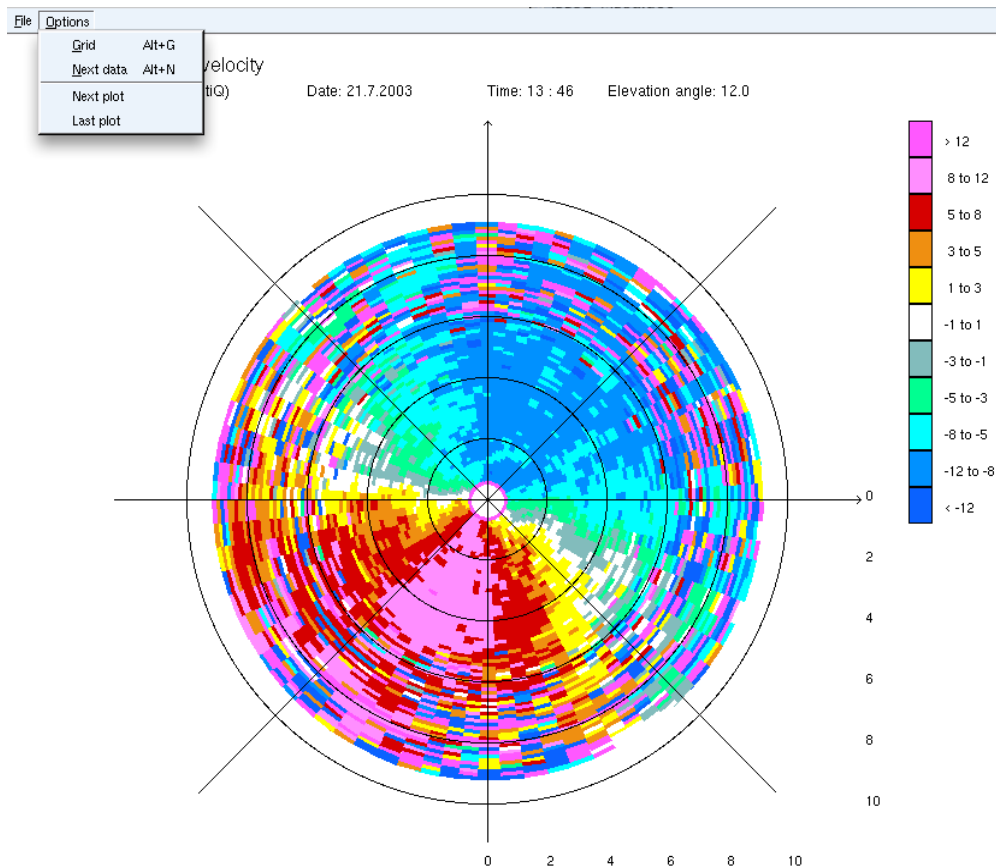


**Figure 3.** Example of RHI dialog

Figure 4 shows an example of a RHI display of Doppler Velocity with a grid superimposed. If a display without the grid is required this can be obtained under by selecting “options” and toggling “grid”. Please note that other choices under options have not been implemented. The figure shows the choices available when “file” has been selected. Figure 5 shows an example of a PPI display with “options” selected.



**Figure 4** Example of RHI display with grid selected

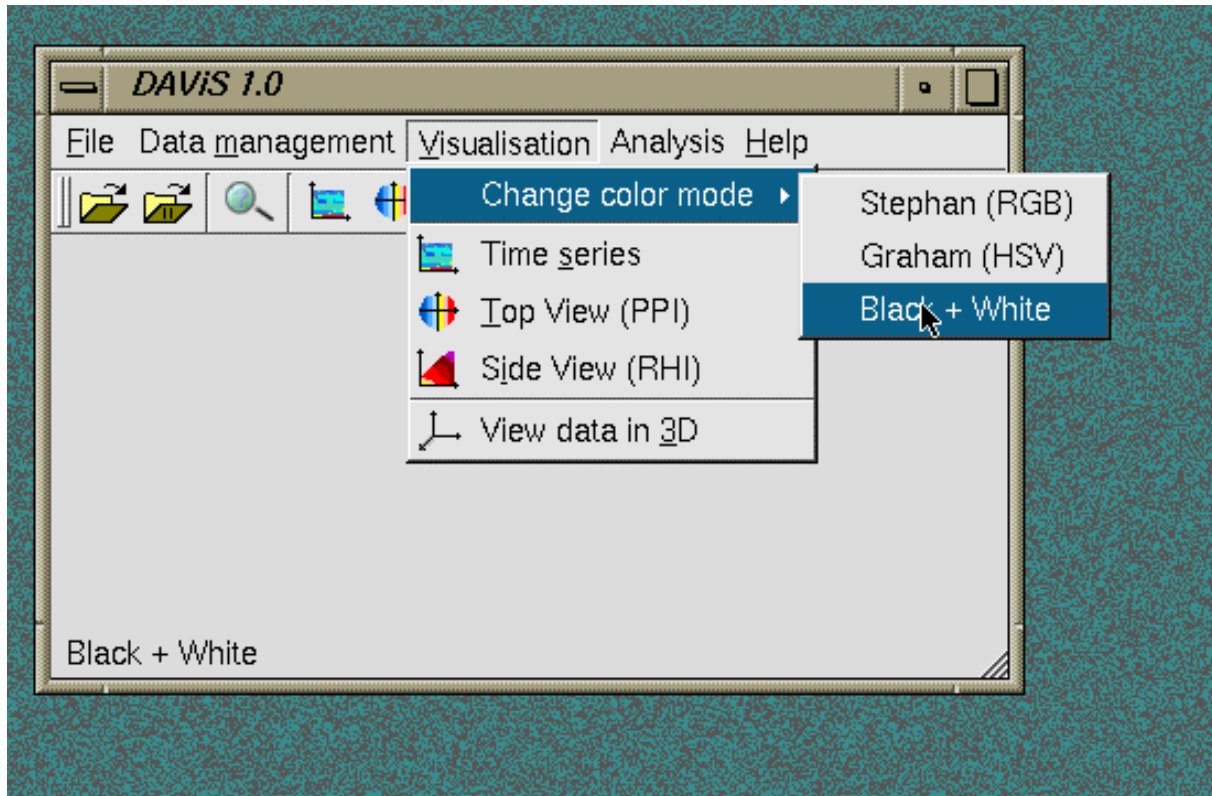


**Figure 5.** Example of a PPI display

### 3.2 Changing the colour tables

For the two-dimensional plots it is possible to change the colour tables used to plot the Velocity and Intensity data. The user can choose between 2 colour and one black and white colour tables.

To select one of these colour tables the user should select one of the tables in the sub-menu *Change color mode* under *Visualisation* in the main menu. This is shown in the figure below.




**Figure 6:** Menu to select a colour table in DAVIS.

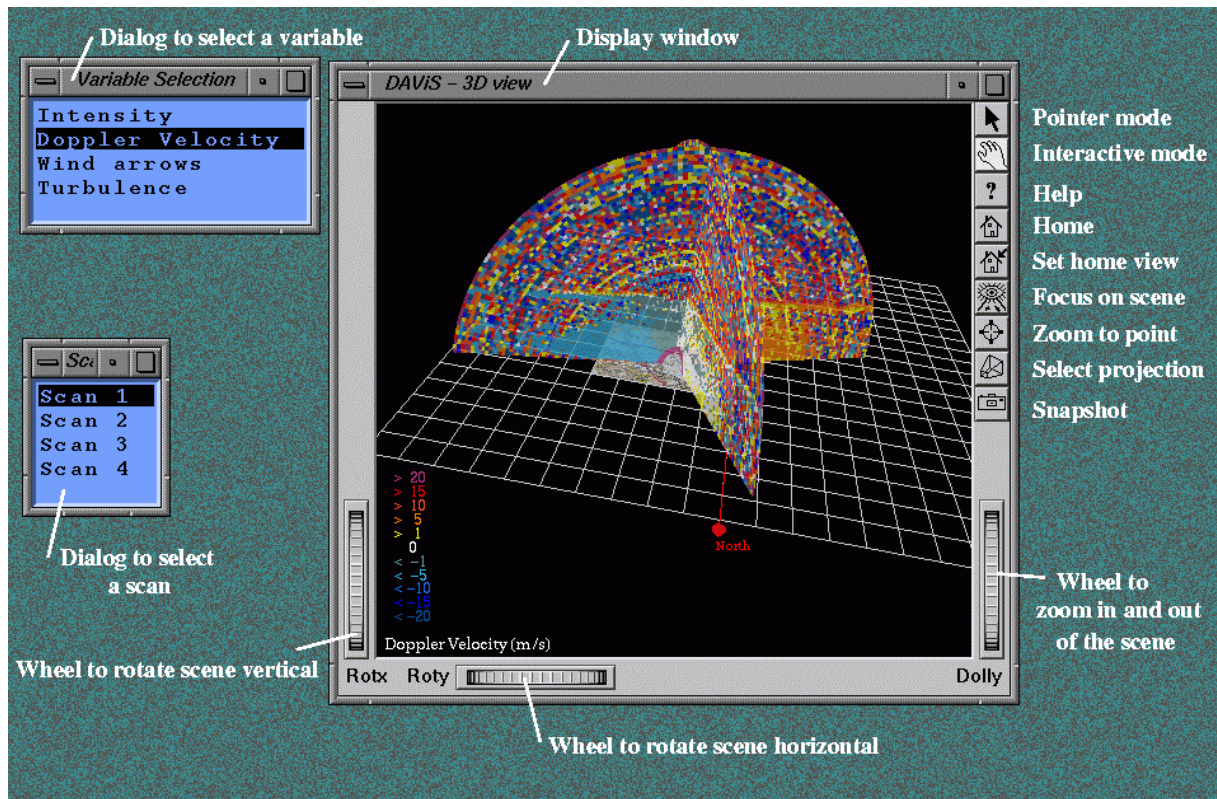
### 3.3 Saving data into an ASCII file

Every two-dimensional display offers the option to save the observed data in a simple ASCII file for further processing of the data outside DAVIS. To save the data the user has to choose the option *Save data* in the right-mouse-click-menu. First a window appears to ask for a file name under which the new data file should be saved. The format in which the data is stored consists of one header line and one line for every ray. The header line contains first a string describing the kind of data stored and then the site, date and time at which the measurement was taken. The last number of the row gives the number of bins for each ray. All following lines contain the elevation angle as the first element followed by the azimuth angle. All following numbers represent the values of the single bins.

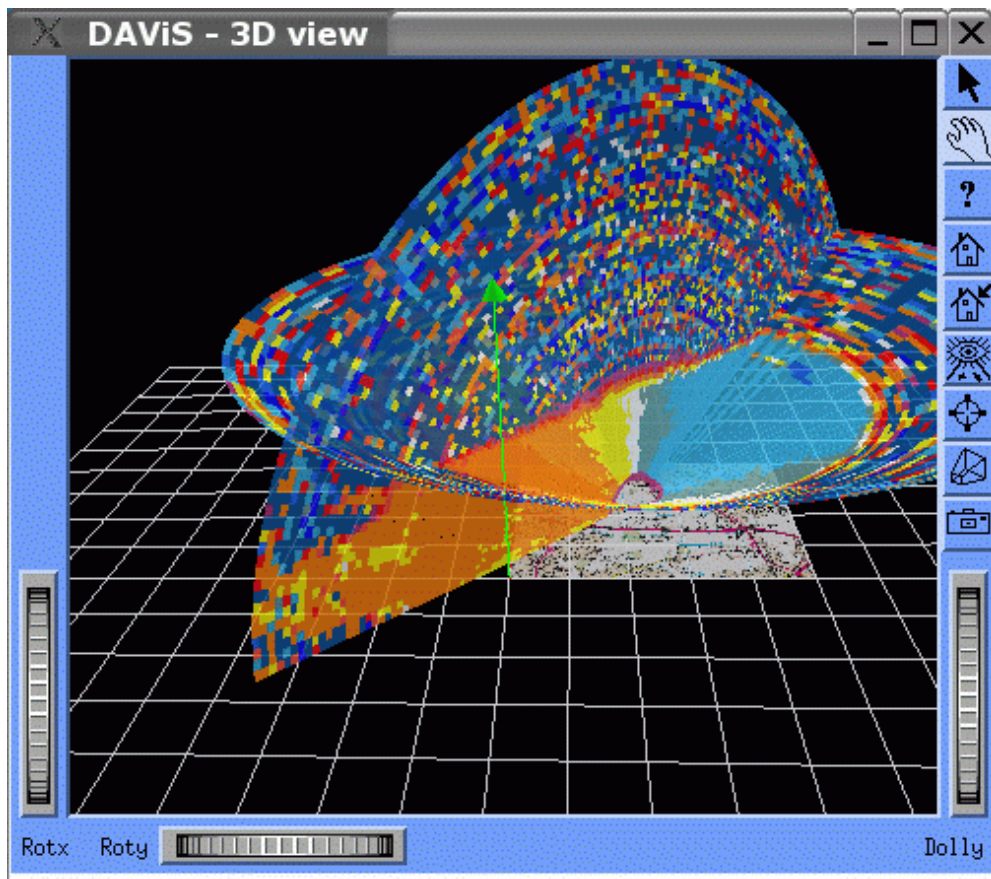
### 3.4 Three dimensional visualisation

To open the three-dimensional display the  icon has to be clicked. Three new windows will open. One is the display window in which the three-dimensional scene is to be displayed and the other two are windows to select either the variable that is to be displayed or the scan. Figure 7

shows a snapshot of the three windows and explains some of the most important controls. Only PPIs and RHIs can be displayed in 3D. It should be noted that the scale and title in the bottom left hand corner are only available in the SGI version; it was not possible to implement them in the Linux version.



**Figure 7:** The three windows to view and control the three-dimensional interactive display.



**Figure 8.** Example showing that RHI and PPI data can be shown together.

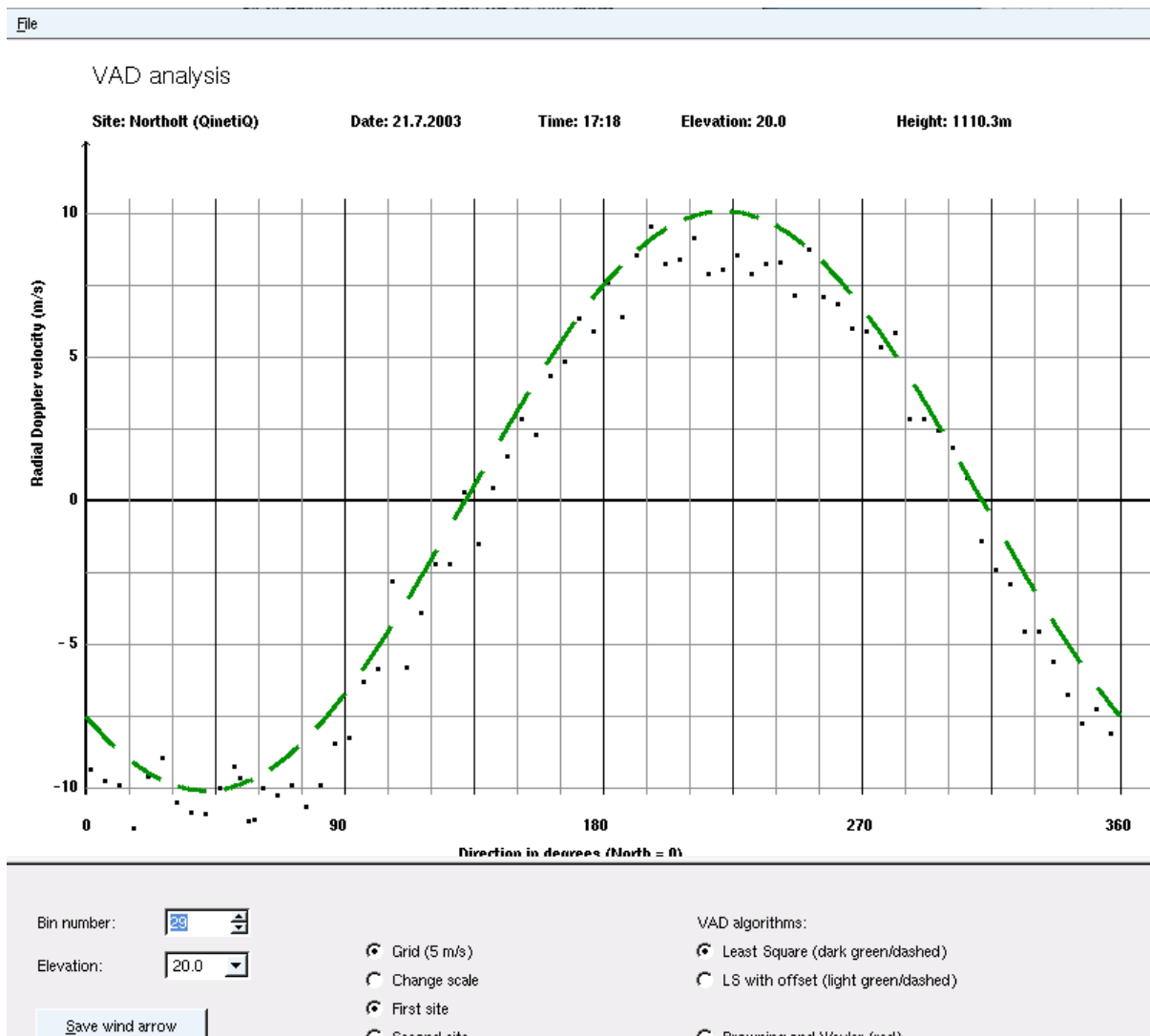
## 4. Data Analysis

DAViS includes the following options to analyse the lidar data:

- Velocity Azimuthal Display (VAD)
- Line of Sight (LOS) analysis in a vertical plane including the two lidar sites.

### 4.1 VAD

A very common method to retrieve wind information from single Doppler lidar or radar data is the VAD. In a VAD all measured radial velocities are plotted depending their location in azimuth around the lidar/radar. From the resulting signature information about the wind field can be obtained. A perfect sine curve represents a uniform wind flow, where the maximum defines the wind speed and the location of the maximum of the sine curve in azimuth gives the wind direction.




**Figure 8** Example of a VAD display

In figure 8 an example of a VAD display is shown. The bin number is selected by the user. This defines the range, and, through the elevation, the height of the scan. A number of different options are available including three different VAD curves. The first option is the standard least-squares fit; the second is a least-squares fit with offset. The third option uses the method of Browning & Wexler.

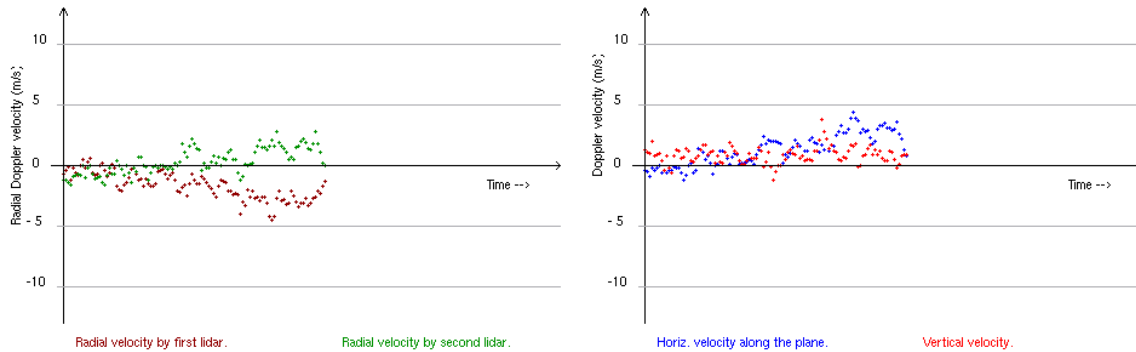
## 4.2 LOS analysis in a vertical plane

In the case, where line of sight (LOS) measurements are made in a plane containing the two lidars, the horizontal wind vector along the plane and the vertical wind component can be retrieved.

To do this the two data sets of LOS data have to be loaded. By pressing the  icon a small dialog is opened to ask which of the bins in the ray are at the crossing point. After pressing *OK* in this dialog box the result is shown in a new window. An example is given in figure 9.

## LOS in plane analysis

Site: Northolt      Date: 23.7.2003      Time: 17 : 2      Azimuth: 118.75      Duration: 10 min      Height: 711 m  
 1st lidar: Elevation: 42.00 Bin: 9      2nd lidar: Elevation: 42.00 Bin: 9



**Figure 9** Example of the analysis of line of sight data.

## 5 Conclusion

The display software was built to display all the types of data there were collected in the two project experiments. The analysis section of the software covers extraction of wind information from PPI scans under the VAD model, and also the extraction of vertical, and horizontal velocities in the plane containing the two lidars. It should be noted that the latter is only approximate, since the two lidars took slightly different times per scan. In principle, if a complete two-lidar scan of the vertical plane containing the lidars were made, it would be possible to visualise the whole 2-D wind field in that plane. We believe that the visualisation system enables features of the data to be seen, and this is an important first step in data interpretation.

## References

- [1] Visualisation of dual Doppler lidar data. S Siemen & AR Holt. ISB52-08 November 2003.  
 [2] Description of the RAF Northolt ISB-52 Dual Doppler Lidar Trial. RI Young, GN Pearson, C Collier, F Davies & K Bozier. ISB52-07 August 2003.

## Appendix A: How to install and compile DAViS

First the source code has to be downloaded from <http://prswwww.essex.ac.uk/lidar/davis.html> and unpacked with these commands:

```
gunzip davis_10_src.tar.gz
```

```
tar -xvf davis_10_src.tar
```

The source code and documentation will be in a directory called *DAViS-1.0* . In that directory are the various subdirectories in which the code, documentation and binary files are. To compile DAViS change into the src subdirectory and type

```
qmake DAViS.pro - o Makefile  
make clean ; make
```

To build the documentation you need doxygen . Type *doxygen DAViS.doxygen* in the *DAViS* main directory.

## Appendix B: The DML data format

The two lidars used in ISB-52 had different data storage facilities because of the difference in scanner design. This meant that there was no consistent data format for the measurements and



some measurements were stored in several data files. (One file for scan angle and one for lidar data). This made it difficult to write an automatic procedure to read and process the data. To overcome this problem a utility to retrieve raw lidar data and store in a consistent data format was needed. This format does not only have to store all recorded lidar data and its parameters, but also had to be able to store retrieved information about the observed wind field.

The result of this, was the Doppler Markup Language, or DML. This format is based on the XML- Markup-Language which is commonly used to describe data formats in the World-Wide-Web. The format has the advantages that it is ASCII and can be edited with any standard editor under Unix or Windows.

Given that DML is derived from XML it is necessary that the first line of the object has to identify the data format. This is done by the line `<!DOCTYPE dml>`. XML is based on objects which start and end with a tag. Tags are symbolised by `< >`. An object starts with a tag containing the object name and finishes with a tag where the object name has a `'` in front of it. This signals the reader that the object is finished. The whole data set itself is also an object of type "dml". This means that a minimal DML file looks like this:

```
<!DOCTYPE dml>
<dml>
.... list of objects ....
</dml>
```

The advantage of having an object-oriented format is that it is very flexible when manipulating different types of data and can be easily extended to handle further objects if necessary. In the following the various objects which are currently defined in DML are described.

### **B.1 Describing the lidar site - `<site>`**

The object `<site>` defines in the data file the site at which the lidar was positioned and other information about the measurement. Each DML data file should only contain one `<site>` definition. The information which can be stored in an `<site>` object are:

```
<name> Sets the name of the location
<time> Sets the date and time of the measurement. Attributes are:
year - integer value of the year
month - integer value of the month
day - integer value of the day
hour - integer value of the hour
minute - integer value of the minute
duration - integer value giving the length of measurements in minute
<position> (or <location>) Sets the position in metres from a origin. Attributes are:
x - floating-point number of location in the x-axis (towards east)
y - floating-point number of location in the x-axis (towards north)
z - floating-point number of location in the z-axis (height)
Example for a description of a lidar site:
<site>
  <name>Malvern (QinetiQ)</name>
  <time year="2003" month="3" day="18" hour="17" minute="50" duration="13" />
  <location x="-432.0" y="2923.0" z="7.0" />
</site>
```

This example describes the site called Malvern, with giving in the brackets the owner of the lidar. The data was measured on the 18th of March in 2003 at 17:50 o'clock. The measurement took 13 minutes to make. The grid position was -432 m in the x (west-east-axis) direction and 2923 m in the y (south-north-axis) direction. The height is given as 7m (which is the difference to the height above sea level of the second lidar).

## B.2 Examples of lidar data - <ray>

The lidar data is organised in rays. A ray is defined as measurements along a line for given angles in azimuth and elevation. An <dml> object can contain as many <ray> objects as needed. The definition of the object <ray> is as following:

*<orientation> Set the direction in which the lidar beam is pointing. Attributes are:*

*elevation - floating point number giving the elevation angle*

*azimuth - floating point number giving the azimuth angle*

*<dimension> Sets the date and time of the measurement.*

*Attributes are:*

*beamwidth - integer type*

*binlength - integer type*

*skipbins - integer type*

*type - string defining the type of measurement ("ppi", "rhi" or "los")*

*<velocities> Object to hold all velocity values for a ray*

*<intensities> Object to hold all intensity values for a ray*

The objects <velocities> and <intensities> have both the attribute number which gives the number of bins. Inside the these objects all bin values are stored in an object called b.

Example for the description of a ray:

*<ray>*

*<orientation elevation="25.0" azimuth="2.27" />*

*<dimension beamwidth="1" binlength="112" skipbins="1" type="ppi"/>*

*<velocities number="2"> <b>14.814</b> <b>0.496557</b> </velocities>*

*<intensities number="2"> <b>1141.36</b> <b>2877.65</b> </intensities>*

*</ray>*

This example shows a ray that consists of two bins which were measured using an elevation angle of 25 degrees and an azimuth angle of 2.27 degrees. The ray belongs to a PPI and the first bin can be ignored.

Wind information - <windvector>

Additional to lidar data, DML can also store information about the calculated wind fields. This information is wind direction and speed and is stored in an object of type <windvector>. The definition is as follows:

*<position> Sets the position of the wind vector in 3D. Attributes are:*

*x - floating-point number of location in the x-axis (towards east)*

*y - floating point number of location in the x-axis (towards north)*

*z - floating point number of location in the x-axis (height)*

*<amplitude> Sets dimension of the wind vector. Attributes are:*

*x - floating-point giving the strength of wind along the x-axis (towards east)*

*y - floating-point giving the strength of wind along the x-axis (towards north)*

*z - floating-point giving the strength of wind along the x-axis (vertical)*

*<properties> Object to describe various properties of the wind vector*

*Two attributes exist to describe the quality of the wind information: quality and variance.*

Example for the definition of a wind vector:

```
<windvector>  
  <position x="34.0" y="-102.0" z="23.0" />  
  <amplitude x="-2.0" y="23.0" z="7.0" />  
  <properties quality="0.5" variance="0.1" />  
</windvector>
```

## **ACKNOWLEDGEMENTS**

This work was funded by HM Treasury under the Invest to Save Budget. Department for Environment, Food and Rural Affairs (DEFRA) acted on behalf of HM Treasury. QinetiQ work described herein was supported under Contract Number CU016-0000014438 and this support is acknowledged. The authors also acknowledge assistance from members of the Urban Lidar Project (Met Office, Salford University and QinetiQ).

## **DISCLAIMER**

The authors of this report were/are employed by the University of Essex. The work reported herein was carried out under sub-Contract CU016-00000 14438 placed on 26 October 2001 by QinetiQ Malvern as part of a larger Contract between QinetiQ and the Secretary of State for Environment, Food and Rural Affairs. Any views expressed are not necessarily those of the Secretary of State for Environment, Food and Rural Affairs.

The DAViS software is Crown Copyright, it is freely available to anyone who is interested provided any work conducted acknowledges the original source code and authors.

**© Copyright 2004**