

1.5 seconds to load 336 bytes configuration text file. Note the text files; both configuration and language files can be exported directly from my Access 4000 database program, also my Access 4000 text editor can be used to edit or create new language text files.

Screen shot of main dialog shown below: -

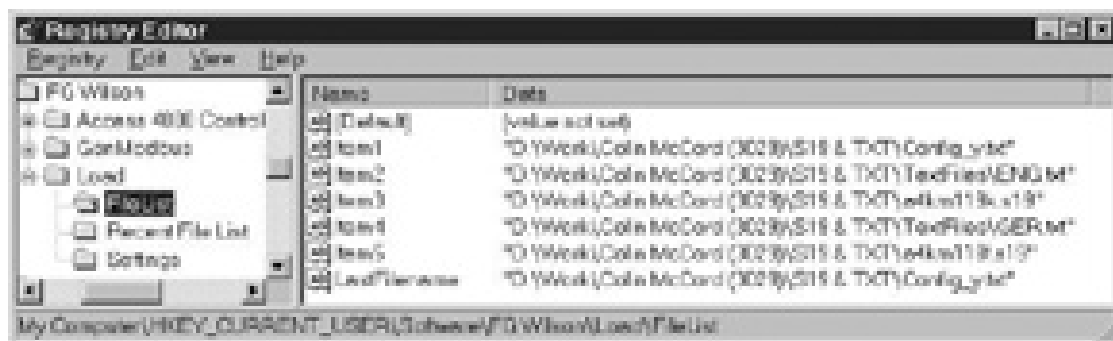


The program was designed to be simple to use, the user first needs to specify the S19 or TXT file to load to the Access 4000 control panel. There are four ways to specify the file: -

1. Type filename and path into combo box.
2. Click [Browse...] and select filename using standard Windows open dialog.
3. Combo box contains last 5 files, select using downward arrow.
4. Drag file into dialog.

Next, user reboots the Access 4000 control panel and while in bootstrap mode user clicks [Load], loading dialog appear and file starts to load.

The last 5 files are stored in the Windows registry, screen shot below: -



The loading dialog using progress meter shows the percentage complete during the loading process, also the bytes send and total number of bytes are also shown. Screen dump below: -



If load was successful the [Cancel] button is change to [OK] and the [Retry] button remains disabled. If load was unsuccessful the [Rety] button is enabled.

The about dialog: -



Program tested on Access 4000 control panel, and after a view modifications the program seems to work fine. The computer I'am using is running Windows NT and has a PIII proccessor with 128MB of RAM, so it was important to test on other operating systems and older PCs.

Started testing on Windows 95 running on P133 with 32MB of RAM, S19 files worked fine, but I had a problem with the text files. At about 50% the communication slowed and the Access 4000 control panel timeout and load failed. At frist I did not understand why this was happening, but then discovered that I was incrementing the process bar after every byte hence creating a delay between the bytes. The reason why the S19 files works was because there is a delay between blocks when waiting for the ACK, hence the processor was able to catch up.

Once it was discovered what was causing the problem, the problem was easily fixed. The solution was to change to step size of the processor bar to 50, and step it every 50 bytes instead of every one. This solution worked fine. Although there was no problem with the S19 files, it is bad and wasteful programming to increment the process bar after every byte. The process bar was updated after every block, hence no delay between bytes within a block.

Before the changes where made, loading S19 files on my computer (PIII – Windows NT) used 10% of the processor, it now uses 2% of the processor. Before the changes where made, loading TXT file on my computer (PIII – Windows NT) used 45% of the processor, it now uses 4%. These values where taken from the "Window NT Task Manager".

Created a small help file using "Help workshop" describing how to operated the program. After a couple more hours of testing, I passed the program file to another team member who will test it. He will go about his normal tasks, and if he requires to load TXT or S19 files to the Access 4000

control panel my program will be used. If program has been proven to run reliably, it will replace the existing MS dos version.

Sometime in the future the bootstrap code in the Access 4000 control panel will be modified to allow uploading as well as downloading. My program can easily and quickly be modified to accommodate any new features, as they're required.

On Thursday,

There a customer in Saudi Arabia was having a problem with the Oil Pressure (variations), they where using Access 4000. The customer was told to put a resistor across the Oil Pressure pins on the Access 4000 control panel, and if a constant value of oil pressure was displayed there was nothing wrong it the Access 4000 control panel. The customer try this and a problem with the Access 4000 control panel was ruled out.

The customer was given a number of possible solutions; one included putting a 0.1uF capacitor in parallel with the oil pressure sender. But the customer wanted use to test the possible solutions. FG Wilson has sold this type of Gensets all over the world and never had any problems with the oil pressure before, so in order for us to fix the problem we need to be able to recreate it.

So I was given the task of finding a Genset in Larne with the same problem, and if one could be found investigate. Took a walk to bay F (Genset test bay), asking the technicians if they had come across this oil pressure problem during testing. They had never had a problem with the oil pressure, they seemed to think that the problem was a faultily oil pressure sender.

There is not a Genset in Larne with the same problem; hence we cannot test any of the possible solutions. Email explaining this was sent, and they will have to try to fix the problem themselves using are suggestions as a guild.

Week 30: Monday 19/02/2001 to Friday 23/02/2001

Carried out some work on the new SCI Annunciator, this is a redesign of the existing Access 4000 – remote SCI Annunciator. Using RS422 communications, the Annunciator can display up to 20 faults (e.g. Low Battery Voltage, High Battery Voltage, High Frequency, Low Frequency etc...). The Annunciator only receives the information it does not transmit, so only two of the communication cables are required RX+ and RX-. As well as the 20 LEDs for the faults there is another one that flashes On/Off when communications is working, also there is a "lamp test" push switch which when pressed all LEDs light. The push switch also shuts off the buzzer when an alarm is active.

A PIC is used, for the prototype stage this PIC contains flash memory which can be changed as many times as necessary, when in production the PICs used will have PROM memory (write once) the program cannot be changed. The main reason for not using flash in production is cost.

We need to have two prototypes ready for the end of the week, so was given the task to check both prototype PCB boards making sure there are no broken tracks. This was achieved using multimeter in buzzer mode, using schematic as a guild checked every connection. Next after it has been confirmed that the PCB are correct, I started to populate both boards with all their components, hence getting some much needed soldering practice.

After the boards had been finished, I tested them. Connected to 12V-power supply, connect RX+ and RX- to TX+ and TX- of the Access 4000 and switched on. I noticed straight away that a view LED's where constantly on, using mulitmeter it was discovered that a couple of BS170 transistor

where dead (Always a circuit between collector and emitter). Replaced faulty transistors and tested again, both prototype SCI Annunciation where working.

There was a slight worry the voltage regulator will over heating (mounted to small heat sink) a high voltages. I left the SCI Annuciator running on 28 Volts for 6 hours, to seek how hot this voltage regulator got. After 6 hours there was a small amount of heat, but the heat sink seemed to be doing a good job in displacing the heat and the tested was seen as a success.

Next using AutoCAD R14, I design label strips. These label strips are inserted behind the main label displaying in English each fault. Next given the task to get a rough price of all the components using RS I priced each component, which where then added for a total.

Week 31: Monday 26/02/2001 to Friday 02/03/2001

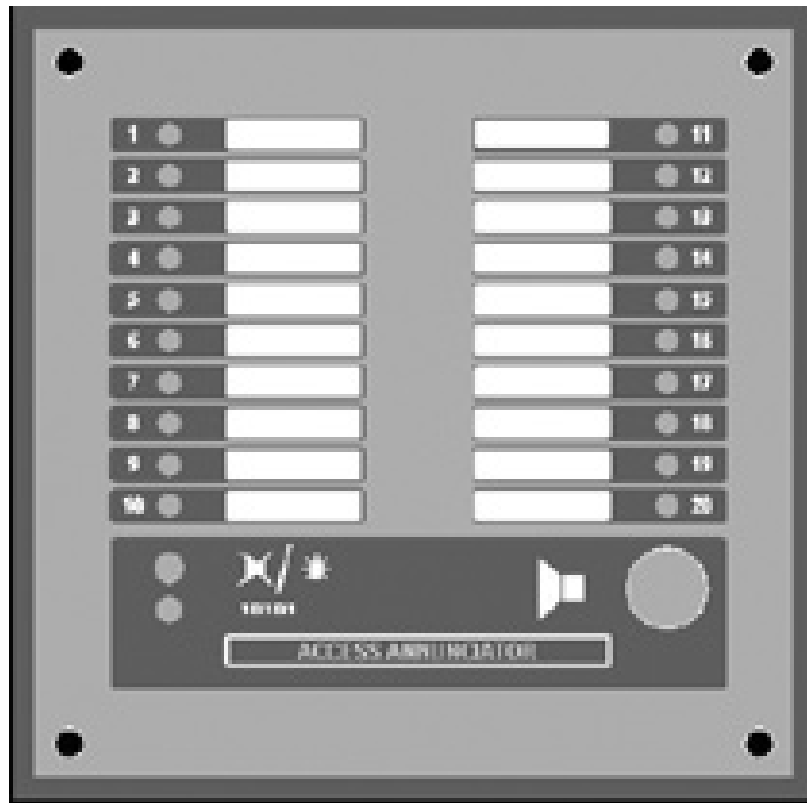
Started to write the technical report for my Access 4000 database program. The Access 4000 database program I have created is only the first step of many. At a later date Internet communications and other functions will be added, another engineer will properly complete these future modifications. So it's important that the code is easy to read, with a fixable programming structure. The technical report is to make life easier for any engineer who wants to modify the database program; it will contain a detail description on every class with a brief description of each function.

I completed the detailed description of each class along with a brief description of each function: -

- CMainFrame
- CPanelApp
- CPanelDoc
- CPanelView
- CAdminToolsView
- CAboutDlg
- CAboutSerialNumberDlg
- CAddNewRecord
- CAdmin_SelectUserDlg
- CAdminAddUserDlg
- CAdminEditUserDlg
- CAdminSelectWonDlg
- CChangePassDlg
- CDeleteRecDlg
- CImportTxtDlg
- COptionsDlg
- CPageSetupDlg
- CPasswordDlg
- CCalibrateView
- CCommercialView
- CInputsView
- COutputsView
- CSecurityView
- CSetpointsView
- CSystemView
- CPanelSet
- CTextFileSet
- CUserHistSet
- CUserStatusSet

Detail description and screen shots of all panel resources (Accelerators, Bitmaps, Dialog boxes, Icons, Menus, String Table and Toolbars) where added to the technical reported.

Carried out more work on the new SCI Annunciator, design changes where made to the plastic cover. Using AutoCAD R14 changes were made to the design of the plastic face, see screen shot below: -



Notice name has been changed from "SCI Annunciator" to "Access Annunciator", the reasons for the change is that the new remote Annunciator only works with the Access 4000 control panel. It is foreseen that it will also be compatible with the new Access 3000 control panel, which is at the early stages of development.

These design changes included increasing the size of the metal cover by 20mm at all edges, the reason of this change is to allow mounting to the wall from the front. 4 additional holes will be drilled in the metal cover; the size of the plastic face will be increased which will cover these additional holes. Hence if customer wishes to front mount the Annunciator they can punch a hole in the plastic cover and insert their screws.

On Thursday 1st March 2001 at 14:30 had my first visit from my academic placement officer Mr. George Jackson. Answered a view questions and he filled in a few forms, give details on his marking scheme and written course work.

Week 32: Monday 05/03/2001 to Friday 09/03/2001

Carried more work on the Access Annunciator, using RS and Farnel compiled a complete PCB component price list, in each case the price was taken if ordering 100+ at anyone time. This give a idea on how much it's going to cost, although FG Wilson will properly get special discounts so

the actual price is almost certain to be less than my estimated price. Below is a copy of the price list I compiled: -

Part Number	Name	Description	Qty.	Comps	Price	Total (Price X Qty)
237-7100 (RS)	0.1uF	CAPACITOR CERAMIC	11	c1-c4, c7-c11, c14, c17	£0.04	£0.47
244-3825 (RS)	10-WAY IDC	RIBBON CONNECTOR	2	EC3-4	£0.33	£0.66
214-1002 (RS)	100 - 5% - TR4	100R TR4 METAL OXIDE FILM 5%	1	R29	£0.08	£0.08
228-6903 (RS)	100UF 50V	CAPACITOR ELECTROLYTIC	2	C12-13	£0.08	£0.15
140-530 (RS)	10K - SIL9	RESISTOR NETWORK 9-WAY	7	R21-26, R31	£0.15	£1.05
214-1024 (RS)	150 - 5% - TR4	150R TR4 METAL OXIDE FILM 5%	1	R27	£0.08	£0.08
148-556 (RS)	1K6 - 5% - TR4	1K6 TR4 METAL OXIDE FILM 5%	21	R1-20, R32	£0.02	£0.50
251-732 (FA)	1N4007	DIODE	1	D1	£0.02	£0.02
211-4909 (RS)	33PF - 5% - S2	33PF 100V COG ML CERAM 5%	2	C15 - 16	£0.16	£0.32
214-1210 (RS)	3K3 - 5% TR4	3K3 TR4 METAL OXIDE FILM 5 %	1	R28	£0.08	£0.08
220-4658 (RS)	4 - WAY KLIPPON	PCB CONNECTOR	1	EC1	£0.67	£0.67
492-917 (FA)	4MHZ CRYSTAL	FARNELL 492-917	1	X1	£0.33	£0.33
228-1532 (RS)	41.P06P120A	IMO PIEZO SOUNDER	1	ALM	£1.94	£1.94
672-385 (FA)	6-WAY HARWIN	FARNELL 672-385	1	X1	£0.29	£0.29
325-727 (FA)	6N135	OPTO - ISOLATOR DIP-8	1	ISO1	£0.50	£0.50
217-5695 (RS)	74HC595	8 BIT SHIFT REGISTERS	3	U4 - 6	£0.52	£1.56
382-140 (FA)	74HCT165N	8-BIT SHIFT REGISTER	4	U1 - 3, U10	£0.23	£0.92
166-2227 (RS)	750 - 5% - TR4	750R TR4 METAL OXIDE FILM 5%	2	R91 - 92	£0.18	£0.36
103-0771 (RS)	91F08	DIP SWITCH 8-WAY	4	SW1-4	£0.53	£2.11
336-747 (RS)	9633CD	MORS SWITCH SPNO	1	SW5	£1.15	£1.15
932-840 (FA)	BS170	MOSFET N-CHANNEL	22	Q1 - 22	£0.09	£1.94
192-0983 (RS)	DDE105S05	DC-DC CONVERTER 5V - 5V	1	REG2	£7.51	£7.51
415-014 (RS)	FUSEHOLDER	FUSE-HOLDER	1	F1	£0.18	£0.18
590-547 (RS)	HLMP4700	LED RED 5MM LOW CURRENT	20	L1 - 20	£0.25	£5.00
590-553 (RS)	HLMP4740	LED GREEN 5MM LOW CURRENT	1	L21	£0.25	£0.25
298-8508 (RS)	L7805ABV	5V, IAMP REGULATOR	1	REG1	£0.33	£0.33
244-0624 (RS)	MAX3082	RS-485/422 TRANSCEIVER	1	U8	£1.13	£1.13
324-3115 (RS)	MM74HC04	HEX INVERTERS 6 OFF	3	U9, U11 - 12	£0.15	£0.45
328-2429 (RS)	PIC16LC63A-04I/P	PIC MICROCONTROLLER	1	U7	£1.74	£1.74
					TOTAL	£31.78

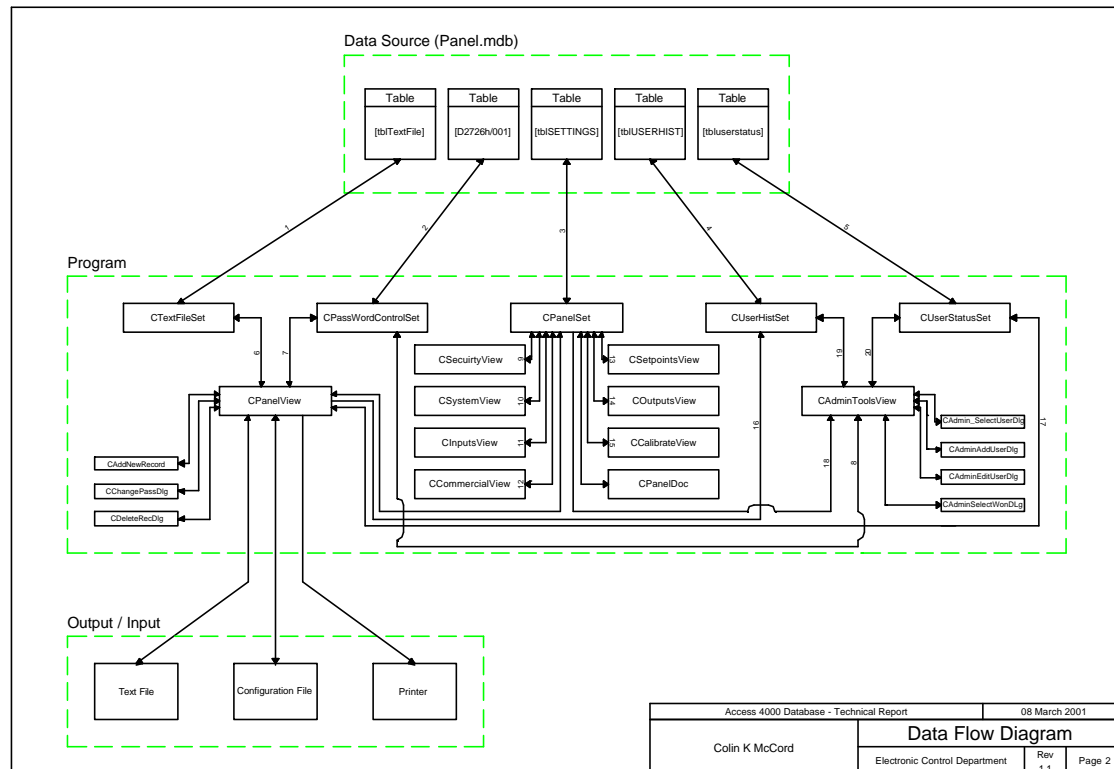
(FA) = Farnell, (RS) = RS Catalogue

My Access 4000 database program has been given out to a few engineers to use in replace of the existing database program, any problems, bugs, or suggestions will be relayed to me. This trial period may last a couple of months; any required changes to the program will be made. Once test period is over and all problems are fixed the program will be released and a copy of the program will be given to all concerned.

Finished Access 4000 database program technical report, the report was 60 A4 pages in length (excluding source code) and contains all technical information required by any engineer/programmer to understand how my program works quickly and easily. Using this technical report as a guide he/she will be able to modify the program, quickly and easily.

Added a number of additional topics including; Front Cover, Introduction, Data Flow Diagram, Data flow paths, Class structure, Class Description, Appendix 1 - *MDB data source description, Appendix 2 – Hierarchy Chart, Appendix 3 – CDaoRecordSet Class Members.

The data flow diagram shows how data is transferred throughout the program, it was drawn in AutoCAD Release 14, screen shot is shown below: -



Week 33: Monday 12/03/2001 to Friday 16/03/2001

Cleaned up source code for my Access 4000 control panel program, adding additional commits and simplifying the code. Also a view minor modifications very made most of which improve speed, the database is quite large and is stored on a network drive, network drives are much slower than the hard drives. Startup time was decreased by half just by making the code more efficient, improvements made through the program. A bug that caused the program to crash was discovered and fixed.

After all the modifications very made, the source code was added to a 3½ floppy disk along with the source code for the serial generator. This disk was then stuck to the back of my technical report, which was then filed (Current program version is 1.99a).

Small amount of work was carried out on my GenModBus program, Access Annunciator, Access 4000 and the new Tel Panel.

Access 3000 is in the early stages of development (Technical Specification not finished) it will be somewhere in between the Access 2000 and Access 4000. It will be cheaper and have fewer features than the Access 4000, but be more expensive than the Access 2000 but have more features.

My task was to think of several methods of configuration the panel. The access 4000 has a configuration text file that is exported from a database, which is then loaded to the controller using load.exe. This file is very low level and cannot be easily be modified manually.

Possible methods for configuring of Access 3000: -

1. Download configuration file using load.exe, while controller is in bootstrap mode or/and maincode. The configuration will be exported form a database.
2. Upload panel configuration to a file using load.exe, while controller is in bootstrap mode or/and maincode. This file can then be imported back to the database.
3. Download panel configuration directly from the database. The database program will have included COM communications (user will select "Export to Access 3000").
4. Upload panel configuration directly to the database. The database program will have included COM communications (user will select "Import from Access 3000").
5. Configure panel in real-time. Communications Protocol could be included in the maincode. Activated by the user selecting it from within the menus of the controller or the controller is able to automatically leave GenAccess or Modbus when a certain signal is received and return when configuration is complete.

Configure panel in real-time

Windows based program displays current configuration, when the user changes a setpoint a communication message block is sent automatically updating the setpoint. There is continuous communication with the program and the controller at all times, if communication is stopped/unreadable for a certain period of time the controller will leave this configuration mode (return to Modbus or GenAccess).

This program will come in useful for customers without access to the database.

This program should also be able to: -

- ◆ Upload and Download Configuration files.
- ◆ Upload and Download Language files.
- ◆ Download maincode.

All this functions should be included in one program with a user-friendly interface and must be easy to use. Maybe consideration should be taken on including these functions in future versions of GenAccess.

Security is very important, it recommended that some sort of password is used to protect the remote configuration of any panel (maybe level 3).

Example Access 3000 configuration text file: -

```
[ID Block]
Works Order Number = D2336A/001

[System]
Communication = 1
Unit Address = 1
Data Speed = 9600
Comms = 4
```



```
[Calibrate]
VT Phase A Offset = 0
VT Phase A Gain = 1.000
VT Phase B Offset = 0
VT Phase B Gain = 1.00

[Setpoints]
Voltage High Status = 1
Voltage High Setpoint = 457
Voltage High Time = 5
Voltage High Action = 2

[Inputs]

[Outputs]

[Security]
Level 1 Password = 1111
Level 2 Password = 2222
Level 3 Password = 2004
Remote Password = 9999
```

Similar to Windows INI files, this has one main advantage over the Access 4000 configuration text files: -

- ◆ Customers can easily modify the setpoints without any technical knowledge, then load the modified configuration to their controller.

But there is one disadvantage the file cannot be directly transmitted to the controller; it must be simplified at the PC end before transmission. Hence the code in the load program is more complex.

It is foreseen that the seam load program can load access 4000 and access 3000 text and S19 files, seen it's not as simple as transmitting the access 3000 configuration directly you must be able to detect which you're sending. Access 4000 text files already have conheaders of CCC and TTT, which can be used to detect existing Access 4000 files, different conheaders will be added to the top of the access 3000 text file.

It is also possible to change the access 4000 text file to a similar format, without changing the bootstrap code. The text file will be in the new high level format, the load program detects that the file is a new access 4000 text file and converts it back to the old format and stores it in a CString, which is then transmitted.

Load program should be able to: -

- ◆ Upload Access 4000 and Access 3000 S19 Maincode files.
- ◆ Upload and Download Access 4000 existing formatted text files.
- ◆ Upload and Download new Access 4000 high-level text files.
- ◆ Convert between old and new access 4000 configuration text files.
- ◆ Upload and Download Access 3000 text files.

I've also drawn a few high level flowcharts for possible communication protocol for uploading and downloading access 3000 configuration text files. Some screen shots show below: -

