

Phase One Systems  
2950 Merced Street, Suite 101  
San Leandro, CA 94577

Sales : (800) 971-3733  
Support : (510) 895-0832  
Fax : (510) 895-0828  
BBS : (303) 530-9358

Date : 3 December 1996  
To : CET Distributors and Dealers  
Subject : FREEDOM/32 Report Writer for Windows, Version 2.1

This technical note will provide you with the information you need to configure the FREEDOM/32 report writing system to run under any of the Microsoft Windows platforms. Note that the 32-bit version of the ACCESS executable has been renamed to ACCESS32 to avoid confusion with our DOS product and Microsoft Access.

## Installing FREEDOM/32

The FREEDOM/32 distribution diskette for Windows contains the following files in a .zip file format:

\freedom\bin\access32.exe	FREEDOM/32 executables
\freedom\bin\voc.exe	
\freedom\bin\voclist.exe	
\freedom\bin\access32.b	ACCESS32 program source code
\freedom\bin\acc32win.rc	ACCESS32 W/32 Windows Framework files
\freedom\bin\acc32win.res	
\freedom\bin\acc32win.b	
\freedom\bin\acc32win.obj	
\freedom\bin\mkacc32.bat	Batch file to compile ACCESS32 program
\freedom\bin\acctool.bmp	ACCESS32 program toolbar
\freedom\bin\posicon.ico	Phase One Systems icon
\cetlib\cetacc32.lib	ACCESS32 C function library
\freedom\bin\config.cp	Global configuration file
\freedom\bin\syslib\keyword	Required SYSLIB files
\freedom\bin\syslib\message	
\freedom\bin\syslib\remark.dat	
\freedom\bin\syslib\remark.idx	
\freedom\bin\syslib\subtext	
\freedom\bin\syslib\voc.msg	
\freedom\bin\syslib\voclist.msg	
\freedom\bin\syslib\voc.dat	Sample dictionary file
\freedom\bin\syslib\voc.idx	
\windows\access32.w32	Customized w32app.w32 file for ACCESS32
\postk\config.cp	Sample ACCESS32 local configuration file
\postk\cp\sentence\cust...	Sample sentences
\postk\cp\sentence\ex...	
\postk\cp\data\custname.dat	Sample data files
\postk\cp\data\custname.idx	
\postk\cp\data\month.dat	
\postk\cp\data\month.idx	

Although the FREEDOM/32 executables may be installed in any directory defined by the PATH variable, by default ACCESS32 is already configured to run as installed in the directories shown above. For that reason, we highly recommend that you use the **install.bat** file that is provided on the floppy:

**a:install** or **b:install**

After installing the FREEDOM/32 files, add the directory where the executables are stored (\freedom\bin by default) to your PATH. ACCESS32 is set up to run reports immediately with the demo data files in the \postk subdirectory. Execute the program by typing:

```
cd \postk
access32
```

It is important to note that ACCESS32 will store the temporary work files it uses in the directory set by the TEMP environment variable. If this variable is not defined, it will use the \temp\access directory by default. This directory must exist before executing ACCESS32 or an error will be detected.

## FREEDOM/32 Components

### The syslib Directory

This is the directory that contains the required system files, such as **keyword** and **message**, in addition to the public or global data dictionary. The syslib directory may be installed in the same directory with the ACCESS32 executable or in the path specified by the CONTROL variable in the config.cp configuration file. (Configuration files are explained below.) By default, this directory is **\freedom\bin\syslib**.

### The Dictionary File(s)

Phase One distributes a data dictionary with FREEDOM/32. The dictionary file, **\freedom\bin\syslib\voc.dat|idx**, contains the field definitions for the sample data files. This dictionary may be used as a global (public) dictionary, similar to having a CONTROL.VOC file residing in the SYSTEM account in the THEOS version of FREEDOM.

Note that like the THEOS version of FREEDOM, FREEDOM/32 also supports a multi-tiered dictionary so that you can control access to privileged information without having to maintain separate dictionaries with duplicate entries for the common definitions. (For more information on using multi-tiered data dictionaries and the SYSVOC feature, please refer to the *FREEDOM Reference Manual*.) If you wish to use a multi-tiered dictionary, the dictionary file **sysvoc.dat|idx** would become the global (public) dictionary and must reside in the syslib directory (like the ACCESS.SYSLIB.VOC file in the THEOS version). Then a separate local dictionary file, **voc.dat|idx**, could reside in each subdirectory where ACCESS32 would be executed. ACCESS32 is able to generate reports by reading the definitions in both files; the local voc file is read first.

Please note that ACCESS32 only requires that one dictionary file be present, either syslib\voc, syslib\sysvoc, or a voc file in the current working directory.

### The config.cp Configuration File

The configuration file, **config.cp**, allows for customization of the FREEDOM/32 environment. The file is an ASCII text file used to set the paths for the data, sentence and FREEDOM/32 system files. There may be two configuration files for any user -- a local config.cp file and a global config.cp file.

The local configuration file is the config.cp file residing in the directory where ACCESS32 is executed from. The global config.cp file should be located in the directory set by the environment variable B\_ACCESS. This variable may be set in the autoexec.bat or access32.w32. The access32.w32 file distributed with FREEDOM/32 sets B\_ACCESS=c:\freedom\bin by default. (The access32.w32 file is a version of the CET W/32 App Builder w32app.w32 file used specifically by the ACCESS32 program. This program is covered in a later section in this document.)

The local config.cp file in the \postk directory contains the following information. Lines beginning with ";" or "#" characters are interpreted as comments and may be included for documentation purposes.

```
# CONTROL set to the path for the FREEDOM system files
CONTROL=c:\freedom\bin\syslib
# SENTENCE set to the path for the sentence files
SENTENCE=.\cp\sentence
# DATA set to the path for the data files
DATA=.\cp\data
```

Note that the order of the statements is not important, and no particular variable is required. (The search sequence ACCESS32 uses to find the required files is covered in a separate section later.)

The SENTENCE variable allows you to set the path for the sentence files that will be used. You will find this feature extremely powerful. Since it is similar to having a default sentence library (under THEOS), you can produce reports by entering a command such as

```
access32 cust1 (noreturn
```

ACCESS32 will automatically concatenate "cust1" with the specified path in the SENTENCE variable and use \postk\cp\sentence\cust1 to generate the report.

You can also use the full path name of a sentence file. If the name starts with a drive code or the '\' or '.' characters the file name will be used as-is. For example, no conversion will be done on the following:

```
c:\postk\cp\sentence\cust1
\cp\sentence\cust1
.\cust1
```

ACCESS32 will search for files beginning with '.' first in the current working directory and then, if not found, according to the search sequence specified with the B\_ACCSEARCH variable (to be covered later).

The DATA variable performs the same function with data files. For example, the VOC entries for \postk\cp\data\custname are defined under CUSTNAME. The sample sentences all specify "custname" as the name of the file so ACCESS32 can find the definitions it needs. It then concatenates "custname" with the specified path and uses \postk\cp\data\custname to generate the report.

This feature works the same way when accessing data in a secondary file. For example, under the CUSTNAME file there is a definition for FULL.MONTH that loads data defined under MONTH from \postk\cp\data\month:

```
1) load bill.date[1 2]
2) load (month @1 month)
```

If you are porting your files directly from THEOS, it is important to note that ACCESS32 does not support THEOS file names. We suggest that you use the vocabulary and sentence conversion utilities available from Phase One under THEOS to rename the data file descriptions used in CONTROL.VOC and your sentence files before the files are ported.

Note that generic definitions, where portions of filenames are denoted by curly braces '{' and '}', are supported by ACCESS32 as in the THEOS version. Please refer to the *FREEDOM Reference Manual* for more information about this feature.

The global configuration file that has been provided with FREEDOM/32, \freedom\bin\config.cp, contains only one item by default -- the variable to set the path for the FREEDOM/32 system files:

```
CONTROL=C:\FREEDOM\BIN\SYSLIB
```

In our local config.cp file above, the CONTROL variable is actually unnecessary since we are also using the global configuration file (B\_ACCESS is set to the path of the global config.cp) and the CONTROL variable has been set to the same path in the global configuration file.

## ACCESS32 Search Sequence

There may be times when you will need to generate reports from multiple company databases on the same system. ACCESS32 is flexible enough to handle any search configuration by using the environment variable B\_ACCSEARCH to set the search sequence to meet your specific needs.

B\_ACCSEARCH uses the following codes to designate the search sequence for sentence and data files:

Code	
C	The current working directory
L	The path set by the appropriate variable in the local config.cp file
G	The path set by the appropriate variable in the global config.cp file
P	The path specified with B_FPATH
E	The directory where the global config.cp file resides (the path specified with B_ACCESS)

Note that if you use this feature, you must set the variable to all the paths to be searched. For example, the following setting will look first in the paths set in the local config.cp file, then in the current working directory and nowhere else. An error will be reported if ACCESS32 is unable to find the required files in these locations.

```
B_ACCSEARCH=LC
```

Note that if B\_ACCSEARCH is not set at all, then **CLGP** is used as the default sequence. In addition, the B\_ACCSEARCH variable has no effect on how ACCESS32 searches for the required system files such as keyword and message; it always uses the sequence CLGPE.

Any output files created by ACCESS32 (using INTO, APPEND, SELECT or SSELECT) are written to the directory defined by the first element in the search sequence.

The ACCESS Sentence Editor will also use the first element in the sequence to GET sentence files with <ESC>G (or File-Open) or PUT them to the disk with <ESC>P (or File-Save). For example, if the current working directory is first in sequence, entering GET and an asterisk will display all the files it finds there regardless of whether or not they are sentences.

## Using SELECT and SSELECT with ACCESS32

The SELECT and SSELECT verbs use the B\_USER environment variable to create a unique output file name for each workstation. The output files selected.### and sorted.### are created, respectively, where ### is the zero-filled number indicated by B\_USER. For example, if B\_USER=3, the files selected.003 and sorted.003 would be created on that workstation. When B\_USER is not set, then the file names selected.001 and sorted.001 are used by default.

## The ACCESS32 Executable

The ACCESS32 executable is a 32-bit Windows program created using the CET W/32 Application Builder. It can be executed in the same manner as any other Windows executable: from the Windows File Manager; from an icon in the Program Manager or on the Windows 95 Desktop; from the MS-DOS Prompt in Windows 95; or from another CET W/32 program via the cWinExec or cCreateProcess functions.

The 32-bit version of the ACCESS Report Generator has been made into a special C function called cAccess. ACCESS32 is a text-based program that provides an attractive user interface around the cAccess function. All source code files for the program have been included with FREEDOM/32, including the file **access32.b** and its associated Windows Framework files, **acc32win.b** and **acc32win.rc**. The Windows Framework has been modified to add a custom toolbar and menu bar, giving the program more of a Windows look-and-feel while still maintaining the complete functionality of the THEOS version of the ACCESS Sentence Editor.

ACCESS32 may be used as a standalone ad-hoc report writer as-is, or it may be customized and recompiled by the developer using the CET W/32 Application Builder. The source code files have been provided for this purpose, however, you should become familiar with the information in the *CET W/32 Application Builder User's Guide* before attempting to do so.

A new command line option HOURGLASS (abbreviated HOUR) has been added to ACCESS32 (cAccess) so that an hourglass cursor will be displayed while the sentence is processing instead of the "Read; Processed; Selected" message that is normally displayed. This speeds up processing time when a sentence is executed from the command line, bypassing the ACCESS Sentence Editor. For example:

```
access32 cust1 (noreturn wait hour
```

## The ACCESS32.W32 File

ACCESS32 can, as any CET W/32 program, use the environment variables set in the w32app.w32 file. However, it may be undesirable for ACCESS32 to use the same .w32 file as your other W/32 application(s). The file **access32.w32** has been included so that environment variables can be set specifically for ACCESS32. The contents of the file are as follows:

```
[W32APP]
```

B\_ACCESS=C:\FREEDOM\BIN  
B\_ACCSEARCH=L  
B\_DFLTFCG=0  
B\_DFLTBGC=8  
B\_FFPTR=  
B\_FPATH=  
B\_PRINTER1=DEFAULT  
B\_PRINTER2=DIALOG  
B\_USER=1  
B\_WINPRINT=1  
B\_4DYEAR=

If the variable B\_4DYEAR is set, ACCESS32 will display dates with a 4-digit year (as if the FULLYEAR keyword is used in the sentence).

Note that the current version of ACCESS32 does not support the control keys in the Sentence Editor using B\_EMULATE=THEOS. If your W32 application uses this variable, then you must use the file access32.w32.

### The cAccess Function

The cAccess function may be used from within any CET W/32 program. The syntax is as follows:

```
CALL cAccess(arg$, addrof(retcode%), addrof(errcode%))
```

Where

#### arg\$

is the string of command arguments to be passed. For example, in the command "access32 cust1 (noreturn", arg\$ would contain the string "cust1 (noreturn".

#### retcode%

is the return code from the function. A return code of zero indicates that the function executed successfully. Other return codes are as follows:

Code

1	Message file not found.
2	Voc and Sysvoc files not found.
3	Bad Voc file format.
4	SYSVOC option used, but Sysvoc file not found.
5	Sentence file not found.
6	Keyword file not found.
10	Bad command line option.
11	Sentence execution error.

The ACCESS32 program adds 800 to non-zero return codes to make the function return codes that get displayed by the program distinctive from the error codes generated by the ACCESS Report Generator itself.

Note that a "no records selected" condition is considered to be a sentence execution error. Therefore, retcode% will be set to 11. For this reason, the error routine in access32.b is not executed when retcode%=11; instead the report generator is allowed to process the error.

#### errcode%

is the error code generated by the ACCESS Report Generator. The complete list of codes is included in the ACCESS Command chapter in the *FREEDOM Reference Manual*.

To compile a program which calls cAccess, the name of the library where the function resides (**lcetlib** or **lcetacc32.lib**) must be passed on the obwin command line. For example, the command to compile the access32.b program is

```
obwin -wr acc32win -wo acc32win -o access32 access32.b -lcetacc32
```

For more information about obwin, please refer to the *CET W/32 Application Builder User's Guide*.

## ACCESS32 Printing Support

ACCESS32 printing support is identical to that which is available with the W32 Application Builder. To use the printer interface, set the following environment variable:

B\_WINPRINT=1

When B\_WINPRINT is used, the variable B\_PRINTER $n$  may also be set to specify the how the PRINT keywords will operate. ACCESS32 supports PRINTER1 through PRINTER9. For example:

Variable Setting	Function
B_PRINTER1=DEFAULT	PRINT will output the report on the system default printer specified in Print Manager. (Not setting B_PRINTER1 will have the same effect.)
B_PRINTER2=DIALOG	PRINT2 will display a Print dialog so the user can select a printer from the list specified in Print Manager.

PRINT? is supported by ACCESS32, but it does not bring up the Print dialog and it is not recognized in the access32.w32 file. It operates exactly as it does in the THEOS version of ACCESS.

Note that if B\_WINPRINT is not set or is equal to a value other than one, then ACCESS32 will use the '.bat' file printing method. (For more information refer to the Appendix of the *CET W32 Application Builder User's Guide*.)

## Sample Configurations

The following examples illustrate how ACCESS32 may be configured to run under a network operating system.

### Example 1: All users access the same data

This example shows how ACCESS32 may be configured so that all users access the same data. In this case, all the programs and related files are stored on the network drive indicated here as "n".

n:\ats32\cmd	W32 compiled programs
n:\ats32\cmd\access32.exe	ACCESS32 executable
n:\ats32\cmd\voc.exe	
n:\ats32\cmd\voclist.exe	
n:\ats32\cmd\config.cp	global configuration file
n:\ats32\cmd\syslib	ACCESS32 syslib directory
n:\ats32\data	data directory
n:\ats32\table	data directory
n:\ats32\sentence	ACCESS32 sentence directory

In this example, B\_ACCESS would be set to specify where the global configuration file config.cp is stored:

B\_ACCESS=n:\ats32\cmd

The global configuration file n:\ats32\cmd\config.cp contains the following records:

DATA=n:\ats32	Set to the PATH for the data files
SENTENCE=n:\ats32\sentence	Set to the PATH for the sentence files
CONTROL=n:\ats32\cmd	Set to the PATH for the dictionary and other related files

If the dictionary contained definitions for the files data\cust or table\param, the PATH set with the DATA variable would be added at the front of the name (e.g. n:\ats32\data\cust) so that ACCESS32 knows which file to use.

You can also use full file names starting with .file, \file, d:\dir\file for data or sentence names. If any of these forms are used, the file name will be used 'as is', without any conversion. ACCESS32 will look for these files first in the current working directory and then, if not found, according to the search sequence specified with the B\_ACCSEARCH variable.

Since all the required files may be found in the paths set in the global config.cp, there is no need to use the default search sequence. B\_ACCSEARCH may be set to look first in the path set in that file and then in the current working directory.

B\_ACCSEARCH=GC

## Example 2: All users access data from both private and public files

This example illustrates how to configure a network system so that users read data from both private (local) and public (global) files.

A possible configuration of the public or global files that everyone can access might be:

n:\ats32\cmd	W32 programs
n:\ats32\data	public data
n:\ats32\table	public data
n:\ats32\sentence	public sentences
n:\ats32\cmd\config.cp	public configuration file
n:\ats32\cmd\syslib	directory containing public dictionary and other related files

The B\_ACCESS variable must be set to indicate the PATH for the global configuration file config.cp. In this example, B\_ACCESS would be set to:

n:\ats32\cmd

ACCESS32 will search this directory for the config.cp file that contains the PATH to the public data and dictionary file:

DATA=n:\ats32  
SENTENCE=n:\ats32\sentence  
CONTROL=n:\ats32\cmd

A users on the same network may also have a set of 'private' files on the local drive. For example:

c:\cmi\data	private data
c:\cmi\sentence	private sentences
c:\cmi\config.cp	private configuration file
c:\cmi\control\voc	private dictionary

If ACCESS32 is executed from c:\cmi using the default search sequence, it will find the local dictionary and config.cp files, and search the paths defined for the sentence and data files:

DATA=c:\cmi\data  
SENTENCE=c:\cmi\sentence

If any of the required files (or definitions) are still not found, the program will then look in the path defined by B\_ACCESS for the global config.cp and search those paths. (The B\_FPATH setting will be searched last.)

Remember that the CONTROL, DATA, and SENTENCE variables may be set in one, both or neither config.cp file. The B\_ACCESS and B\_ACCSEARCH variables may also be used to setup the environment to meet you specific requirements.