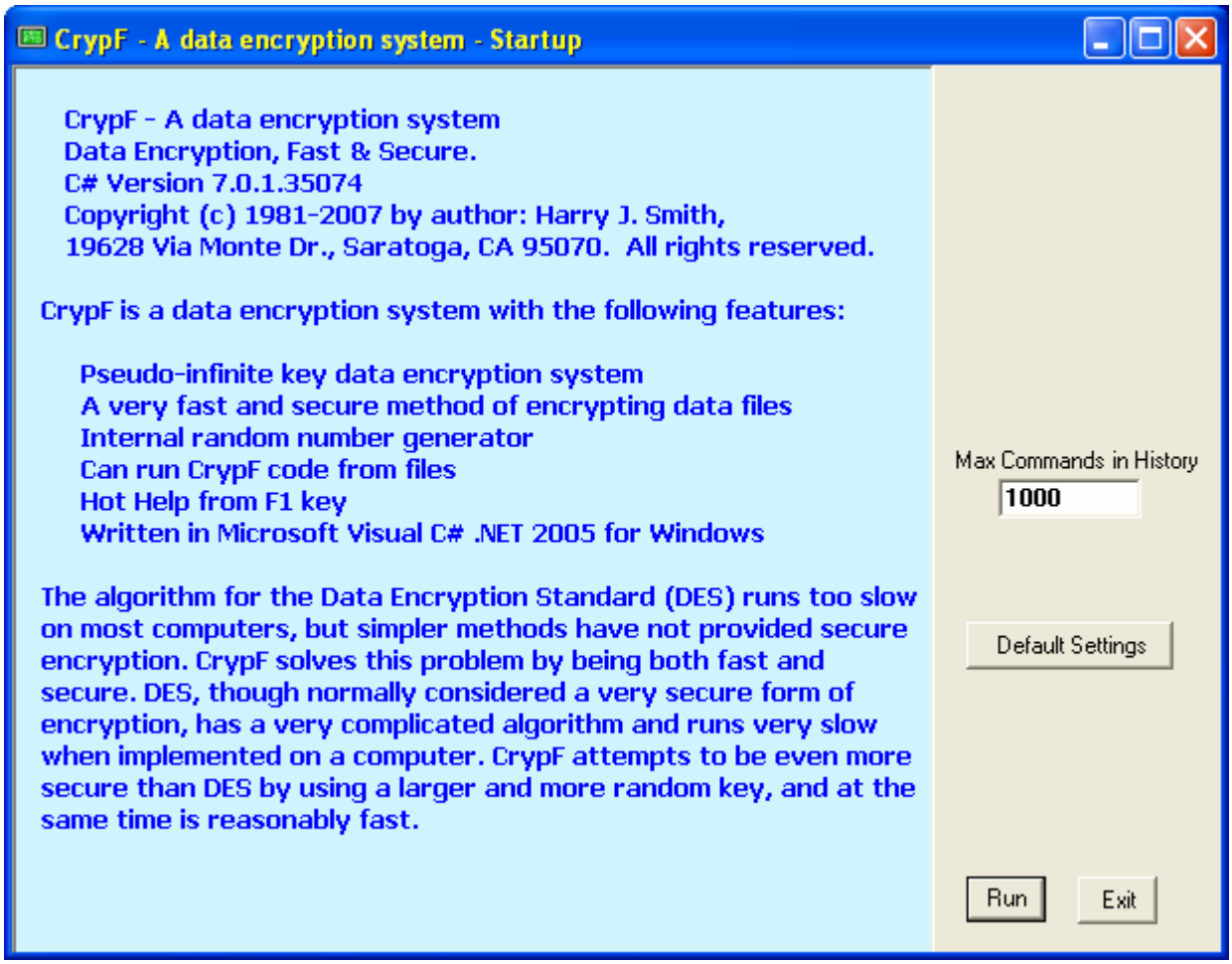
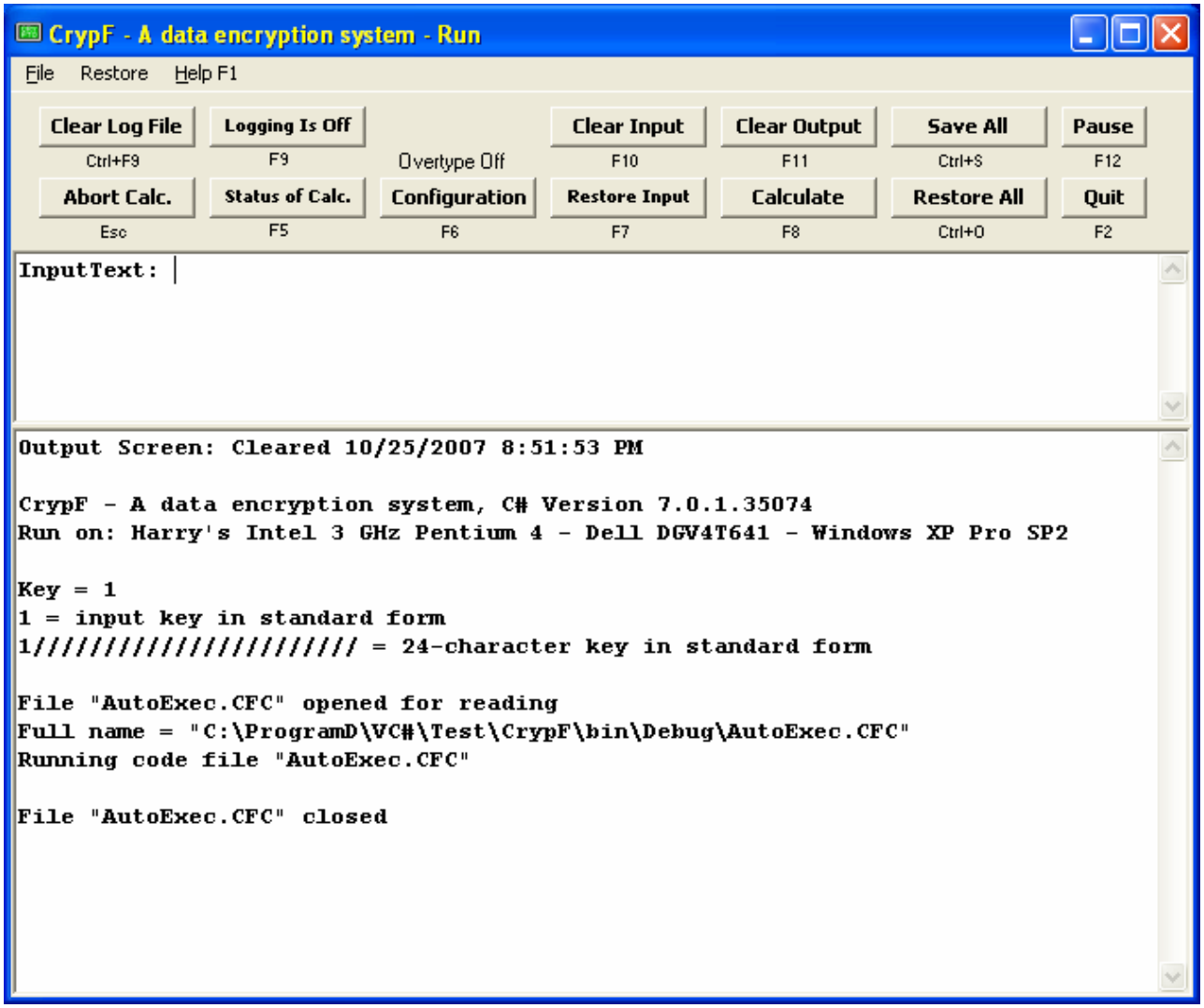
This document applies to the C# Version **7.0.1.35074** of CrypF,

Introduction -

When you execute the program CrypF.exe it responds by displaying the Startup form:



Just click the Run button and the Run form is displayed:



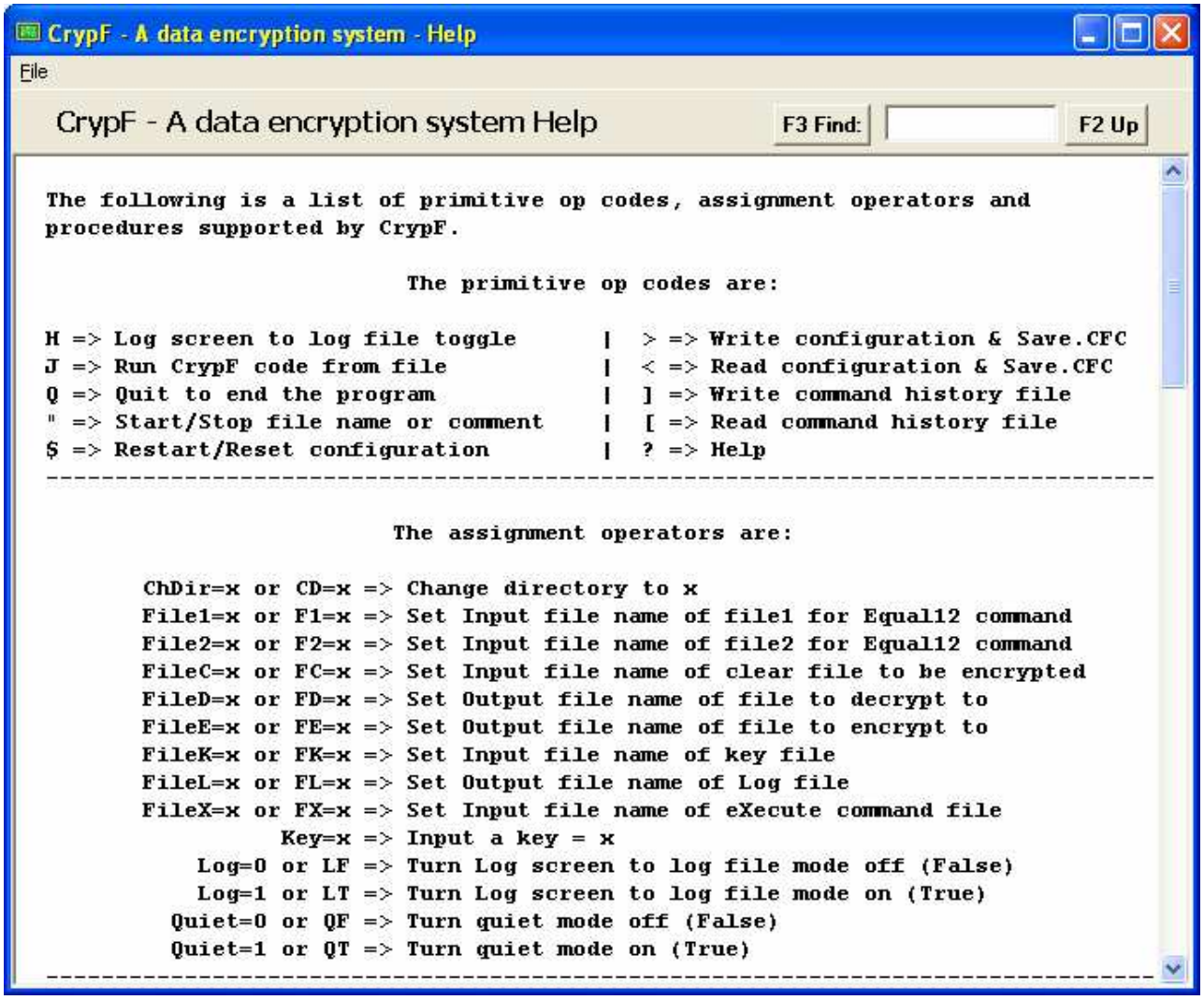
The Run form has two large text boxes. The upper one is for command input. Commands may be entered one at a time each followed by a return, or several separated by one or more blank spaces or semicolons. A separator is never needed between primitive op codes and is only needed otherwise to prevent ambiguity of meaning. Commands are not case sensitive, upper and lower case letters are always interpreted the same.

There are three basic types of commands: 1) Execute a primitive op code, 2) Do an assignment, and 3) Do a procedure.

If the program is executed from the DOS prompt with one or more arguments, the initial Startup form is not displayed and the arguments are taken as an initial CrypF command line. This allows you to control the execution of CrypF from batch files and CrypF code files with no operator intervention. The CrypF code file AutoExec.CFC is always run first, even before the DOS command line commands. The ASCII Tab character (9) is allowed in CrypF code files and in the initial command arguments and treated as a blank space.

Special handling is given to the first argument on the DOS command line. If it ends in .CFC, it is changed to Run("... .CFC") so this CrypF code file will be run. This allows CrypF to be run by Windows or a program, like ZTree, by associating the file CrypF.Exe with the extensions CFC, and then opening a file with this extension.

? => Help: the "?" primitive causes the Help form to be displayed:



The Help form is scrollable and resizable. It contains a list of primitive op codes and procedures supported, Function key actions, and commands used in CrypF code files:

The following is a list of primitive op codes, assignment operators and procedures supported by CrypF.

The primitive op codes are:

```
H => Log screen to log file toggle      |  > => Write configuration & Save.CFC
J => Run CrypF code from file           |  < => Read configuration & Save.CFC
Q => Quit to end the program            |  ] => Write command history file
" => Start/Stop file name or comment    |  [ => Read command history file
$ => Restart/Reset configuration        |  ? => Help
```

The assignment operators are:

```
ChDir=x or CD=x => Change directory to x
File1=x or F1=x => Set Input file name of file1 for Equal12 command
File2=x or F2=x => Set Input file name of file2 for Equal12 command
FileC=x or FC=x => Set Input file name of clear file to be encrypted
FileD=x or FD=x => Set Output file name of file to decrypt to
```

```
      FileE=x or FE=x => Set Output file name of file to encrypt to
      FileK=x or FK=x => Set Input file name of key file
      FileL=x or FL=x => Set Output file name of Log file
      FileX=x or FX=x => Set Input file name of eXecute command file
                Key=x => Input a key = x
          Log=0 or LF => Turn Log screen to log file mode off (False)
          Log=1 or LT => Turn Log screen to log file mode on (True)
        Quiet=0 or QF => Turn quiet mode off (False)
        Quiet=1 or QT => Turn quiet mode on (True)
--------------------------------------------------------------------------------

                        Encryption related procedures are:

         Decode or De => Decode FileE to FileD using key
         Encode or En => Encode FileC to FileE using key
          Equal or Eq => Test for contents of FileC == to FileD
      Equal12 or Eq12 => Test for contents of File1 == to File2
        Execute or Ex => Execute command file FileX
          InitC or Ic => Initialize the Continuation File Cry.con
           KeyF or KF => Input a key from FileK
          State or St => Show state of Encryption inputs
--------------------------------------------------------------------------------

                         Other procedures supported are:

              CFCIn(F) => Enter file name F = "ccc...c" for J command
             CFLOut(F) => Enter file name F = "ccc...c" for H command
              ChDir(F) => Change directory to F = "ccc...c", F optional
          ClearHist    => Clear history of previous operator entries
     ClearLog or CL    => Clear the log file
               Diag(X) => Set diagnostic mode on or off, (X) is optional
               Exit    => Totally quit the program with no questions asked
        Help or He     => Show this Help
             HelpH(X) => Set Height of Help form in pixels
             HelpW(X) => Set Width of Help form in pixels
             HistH(X) => Set Height of History form in pixels
             HistW(X) => Set Width of History form in pixels
        Init or In     => Initialize the program from the beginning ($)
         LogScreen(X) => Log screen to log file mode, on or off
             Pause    => Pause the processing to free the processor
             Quiet(X) => Set the quiet mode on or off, (X) is optional
         Rem or //     => Comment follows, can also be on a line after commands
      Restore or Re    => Restore Configuration, History, & Save.CFC
               Run(F) => Run CrypF code from file F, F is optional
              RunH(X) => Set Height of Run form in pixels
              RunW(X) => Set Width of Run form in pixels
        Save or Sa     => Save Configuration, History, & Save.CFC
              SetC(X) => Set max commands in history
              Time     => Set timing mode on without other diags
             Write(X) => Output X = "ccc...c", X is optional
           WriteLn(X) => Write(X) and a line feed
--------------------------------------------------------------------------------

             +------------Function Keys on Run form------------+
             |   F1  => Display Help form (?)                  |
             |   F2  => Totally Quit/end the program (Q)       |
             |   F3  => Restore previous input command         |
             |   F4  => Restore previous input and accept      |
             |   F5  => Get Status of processing               |
             |   F6  => Display Configuration form             |
             |   F7  => Display Restore Input History form     |
             |   F8  => Accept input and Calculate             |
```

```
            |   F9  => Toggle Logging to Log file on/off (H)  |
            |   F10 => Clear input text box                   |
            |   F11 => Clear output text box                  |
            |   F12 => Pause (Pause)                          |
            |   Ctrl+F2 => Restart ($)                        |
            |   Ctrl+F9 => Clear Log File (ClearLog)          |
            |   Ctrl+S  => Save All (Save)                    |
            |   Ctrl+O  => Restore All (Restore)              |
            |   ESC => Clear Run form Input Text Box          |
            |   ESC => Interrupt/Abort a long process         |
            |  PgUp => Display previous newer command         |
            |  PgDn => Display previous older command         |
            +-------------------------------------------------+
--------------------------------------------------------------------------------

                Commands used in CrypF code files:

        GoTo {label} (label is a name without a colon)
        GoUpTo {label}
        Labels: A name followed by a colon (:)
        Continuation lines ending with + or -
        Batch Commands (Echo, @Echo, Pause, and Rem)
        Echo On/Off
        @Echo On/Off
        Pause
        Rem ... or //... (for remarks)
--------------------------------------------------------------------------------
```

Primitive op codes -

H => Log screen to log file toggle:

The H command causes all output to the screen to be logged to a disk file. See the CFLOut(F) procedure for specifying a file name for this purpose. Each time the H command is given the selection status of this option is reversed/toggled.

The output text box on the Run form is referred to as the output screen or merely as the screen when the context is clear.

J => Run CrypF code from file:

The J command will use the last entered comment as a file name and read this file as a text file. Each line of the file will be interpreted as a CrypF command line and executed. If comment commands and J commands exist in the text file, these other referenced files will be opened and processed. The only limitation to this nesting of code files is the availability of memory and buffers. If a comment has not been entered, the file name NoName.CFC is used. The CFCIn procedure can be used to give an override file name for the J command. The files input by the J command are assumed to have all ASCII text characters with a numerical value less than 128.

Q => To totally Quit/end the program:

The program exits after displaying a message block to allow the operator to OK or Cancel the request.

" => Start/Stop file name or comment:

Comments can be entered anywhere on the command line. The comment is started with a " mark. The comment is ended with a " mark or the end of the line. All blank spaces between the " marks become part of the comment. Comments are also used as file names; see the CFCIn and CFLOut procedures.

$ => Restart/Reset configuration:

This reinitializes the program, the same as reloading except total running time is not reset, the history of previous operator entries is not cleared, and the log screen to log file state is not changed.

> => Write configuration to files Config.CFC and Save.CFC:

The files Config.CFC and Save.CFC are written to disk. They contain the CrypF commands that will restore the configuration of CrypF to its current state.

An example of the contents of a Config.CFC is:

```
@Echo Off
Rem  Start of file Config.CFC
SetC(1000)
Diag(0)
Time
ChDir("ChDir("C:\Program Files\CrypF 7.0\testOutput")
CFCIn("NONAME.CFC")
CFLOut("NoName.CFL")
LogScreen(0)
Quiet(1)
Pri(0)
Rem  End of file Config.CFC
Echo On
```

An example of the contents of a Save.CFC is:

```
FileC = FileC.txt
FileD = FileD.txt
FileE = FileE.txt
FileK = FileK.txt
FileX = FileX.CFC
Key = 1
```

< => Read configuration from files Config.CFC and Save.CFC:

The files Config.CFC and Save.CFC are read and run as CrypF code files. This will restore the configuration of CrypF to its configuration when the files were written by the > command.

] => Write entry command history to file CrypFHist.txt:

The file CrypFHist.txt is written to disk. This is a text file and contains a copy of the current history of operator entries.

[=> Read entry command history from file CrypFHist.txt:

The file CrypFHist.txt is read and used to restore the history of operator entries with the history when the file was written by the] command. The current history is not cleared, but some or all of it may be lost since only “Max

Commands in History" entries are saved. Duplicate commands are deleted as the new copy is entered.

? => Display Help form as presented above.

Assignment operators –

ChDir=x or CD=x => Change directory to x:

Same as the ChDir(F) command.

File1=x or F1=x => Set Input file name of file1 for Equal12 command:

Here x specifies the name of file one that will be used by the Equal12 command. For all commands with a file name, if a file name contain embedded spaces or delimiter characters, the name need to be quoted. For example, File1="";= ()@".

File2=x or F2=x => Set Input file name of file2 for Equal12 command:

Here x specifies the name of file two that will be used by the Equal12 command.

FileC=x or FC=x => Set Input file name of clear file to be encrypted:

Here x specifies the name of the file that will be encoded by the Encode command.

FileD=x or FD=x => Set Output file name of file to decrypt to:

Here x specifies the name of the file that will be used to store the results of the Decode command.

FileE=x or FE=x => Set Output file name of file to encrypt to:

Here x specifies the name of the file that will be used to store the results of the Encode command and used as the name of the file that will be decoded by the Decode command.

FileK=x or FK=x => Set Input file name of key file:

Here x specifies the name of the file that will be read by the KeyF command.

FileL=x or FL=x => Set Output file name of Log file:

Same as the CFLOut(F) command.

FileX=x or FX=x => Set Input file name of eXecute command file:

Here x specifies the name of the file that will be read by the Execute command.

Key=x => Input a key = x:

This uses x to specify the encryption key and computes the “input key in standard form” and the “24-character key in standard form”. Everything following the first = sign is taken as the Key. For example,

```
Key = //~ as @!~;"= // "  ~//
```

Leading and trailing spaces are removed from the key. So, the / character must be used for these.

Log=0 or LF => Turn Log screen to log file mode off (False):

When logging mode is on, all output to the screen is logged to a disk file. See the FileL=x command for specifying a file name for this purpose.

Log=1 or LT => Turn Log screen to log file mode on (True):

When logging mode is on, all output to the screen is logged to a disk file. See the FileL=x command for specifying a file name for this purpose.

Quiet=0 or QF => Turn quiet mode off (False):

When the quiet mode is off, all of the status messages all displayed.

Quiet=1 or QT => Turn quiet mode on (True):

When the quiet mode is on, some of the status messages all not displayed.

Procedures -

Procedures are invoked by a statement starting with a procedure name followed by its argument. Arguments are numbers like -1, 0, 1, ...,712, ... For the procedures Write and WriteLn, arguments are optional and are literal like: WriteLn("Now is the time"). For some procedures like Save, arguments are not allowed.

Encryption related procedures –

Decode or De => Decode FileE to FileD using key:

This decodes the file specified in the FileE=x command and stores the results in the file specified in the FileD=x command using the encryption key specified in the Key=x or KeyF command.

Encode or En => Encode FileC to FileE using key:

This encodes the file specified in the FileC=x command and stores the results in the file specified in the FileE=x command using the encryption key specified in the Key=x or KeyF command.

Equal or Eq => Test for contents of FileC == to FileD:

This reads in the file FileC specified by the FileC=x command and compares it byte-for-byte with the file FileD specified by the FileD=x command. It tells

you if the two files are exactly the same or not. This is mainly for testing to see if the decoded file is the same as the original clear file

Equal12 or Eq12 => Test for contents of File1 == to File2:

This reads in the file File1 specified by the File1=x command and compares it byte-for-byte with the file File2 specified by the File2=x command. It tells you if the two files are exactly the same or not. This is mainly for testing.

When files are read by the Equal12 or the Equal command, the first file is read from the home directory only and the second file is read from the current directory/file path only.

Execute or Ex => Execute command file FileX:

This reads in the file specified in the FileX=x command and executes each line as a CrypF command. Blank lines are ignored.

InitC or Ic => Initialize the Continuation File Cry.con:

This uses the current encryption key specified in the Key=x or KeyF command to initialize the contents of the Cry.con file in the home directory. The Cry.con file is always read from and written to the home directory.

KeyF or KF => Input a key from FileK:

This uses the first line of the file specified in the FileK=x command to specify the encryption key and computes the "input key in standard form" and the "24-character key in standard form". Leading and trailing spaces are deleted.

State or St => Show state of Encryption inputs:

This shows the home directory, file path, and the state of the six encryption commands that would be saves by the Save command. For example:

```
 Home Dir: C:\Program Files\CrypF 7.0\run
File path: C:\Program Files\CrypF 7.0\testOutput

 File1 = File1.txt
 File2 = File2.txt
 FileC = FileC.txt
 FileD = FileD.txt
 FileE = FileE.txt
 FileK = FileK.txt
 FileX = FileX.CFC
 KeyIn = 1
KeyStd = 1
 Key24 = 1//////////////////////
```

Other procedures –

CFCIn(F) => Enter file name F = "ccc...c" for J command:

This establishes the file name of the CrypF code file that will be read by the next J command. If the ("{filename}") is missing, the file name input with the

last "comment" command will be used. If no comment entered, the name NoName.CFC will be used.

CFLOut(F) => Enter file name F = "ccc...c" for H command:

This establishes the file name of the CrypF log file that will be opened by the next H command that opens a file. If the ("{filename}") is missing, the file name input with the last "comment" command will be used. If no comment entered, the name NoName.CFL will be used. If the log file name is changed while it is open, the file name of the open file will not change until the log file is closed and reopened.

ChDir(F) => Change directory to F = "ccc...c", F optional:

Changes the directory used for commands H, J, and Run(F). The original directory when the program starts is called the Home directory and is always used for the commands >, <,], [, ?, Restore, and Save, commands. A ChDir without F will not change the directory, but will tell you how it is currently set.

It is probably easier to use the Configuration form "Change File Path" command button to change the disk directory. This procedure was included so it could be used by the Config.CFC code file for the >, <, Save, and Restore commands.

ClearHist => Clear history of previous operator entries:

The history of up to "Max Commands in History" previous operator entries are saved and can be retrieved by selecting the "Restore Input" button on the Run form. The ClearHist procedure removes all operator entries currently saved and makes this memory available to the program. Even though no argument is needed for this and some other procedures, it is usually better to use the parentheses, e.g., ClearHist() or ClearHist(to prevent unexpected results if the procedure name is misspelled.

ClearLog or CL => Clear the log file

If the log file is open, the file is closed and reopened. If it is currently closed, it opened and then closed. In either case it is cleared. Initially the log file name is NoName.CFL.

The cleared log file will have up to three lines of data like:

Log file NoName.CFL Cleared 9/2/2007 8:56:08 PM
CrypF - A data encryption system, C# Version 7.0.1.20200
Run on: Harry's Intel 3 GHz Pentium 4 - Dell DGV4T641 - Windows XP Pro SP2

The third line is generated by having something like:

SET SYSTEM=Harry's Intel 3 GHz Pentium 4 - Dell DGV4T641 - Windows XP Pro SP2

in your AutoExec.Bat file.

Diag(X) => Set diagnostic mode on or off:

The diagnostic mode is turned on if x != 0 and is turned off if x = 0. When the diagnostic mode is on, all command line executions will be timed by the computer clock and the time spent executing the command will be displayed. The timing data is displayed as:

```
T = xxx.xx  DT = xx.xx sec.  Start execution
{command output, if any}
T = xxx.xx  DT = xx.xx sec.  End of execution
```

The DT value on the End of execution line is the time spent executing the command. The DT on the Start execution line is the time spent waiting for the operator to compose the command. The T values are the total running time since the program was started and can only be reset by terminating and reentering the program. The Quit button followed by the Run button will do it.

To turn diags on Diag(1) can be shortened to Diag. If the argument is < 0 like diag(-1) the debug mode is also turn on and even more diagnostic (debug) messages will be displayed.

Exit => Totally quit the program with no questions asked.

Help or He => Show this Help.

HelpH(X) => Set Height of Help form in pixels:

It is handy to put these commands in the AutoExec.CFC file. For example, the lines:

```
HelpW(674) HelpH(900)
RunW(674) RunH(950)
```

Will set the forms to larger than the default size for a 1280 by 1024 screen resolution.

HelpW(X) => Set Width of Help form in pixels.

HistH(X) => Set Height of History form in pixels.

HistW(X) => Set Width of History form in pixels.

Init or In => Initialize the program from the beginning ($).

LogScreen(X) => Log screen to Log file, on or off:

Same as the H primitive op code, but instead of being a toggle, sets Log screen to Log file on if x != 0, and sets it off if x = 0. This procedure was called EchoScreen(X) in my VPCalc DOS program, and that command still works.

Pause => Pause the processing to free the processor. See function key F12:

Pause is actually a batch command. All commands on the line after Pause are ignored.

Quiet(X) => Set the quiet mode on or off:

The quiet mode is turned on if x != 0 and is turned off if x = 0. When the quiet mode is on, some of the status messages all not displayed. The (X) is optional, Quiet, Quiet(1, and Quiet(1) are interpreted as an on command.

Rem or // => Comment follows, can also be on a line after commands.

Restore or Re/Save or Sa => Restore or Save Configuration, History, & Save.CFC:

Save will write the entry history file CrypFHist.txt like the] command and write the configuration files Config.CFC and Save.CFC like the > command.

Restore will read the entry history file CrypFHist.txt like the [command and run the configuration files Config.CFC and Save.CFC like the < command. Restore does not clear the command history before it executes, so it may grow larger than it was at save time.

Run(F) => Run CrypF code from file F, F is optional:

This will use the argument F = "ccc...c" as a file name and read and run this file as a CrypF code file. To see examples of how CrypF primitives, assignments, and procedures are used, inspect the delivered *.CFC files. It will be noted that they are in plain text.

If the file name does not have a period, the extension .CFC is added. If no argument is given, a comment has not been entered, and the CFCIn(F) command has not set the file name, the file name NoName.CFC is used. The files input by the Run command are assumed to have all ASCII text characters with a numerical value less than 128.

RunH(X) => Set Height of Run form in pixels.

RunW(X) => Set Width of Run form in pixels.

Save or Sa => Save Configuration, History, & Save.CFC:

Same as Ctrl+S, Save All.

SetC(X) => Set max commands in history [1, 100000]:

The SetC(X) procedure sets the maximum number of commands that will be saved in the command history list. If X evaluates to a number x Greater than 100000, the max commands is changed to 100000. If x is less than 1, the max commands is not changed and the status message “Max commands in history not changed from {max}” is displayed. This message is also displayed if x is the same as the value already being used.

If the value is actually changed, the message “Max commands in history changed from {old max} to {new max}” is displayed. If there are more than x commands already in history, all but the latest x commands are removed.

Time => Set timing node on without other diags:

This selects the timing information that you get when diags are turned on, but without the other diagnostic messages. Turning diags off will turn timing off also.

Write(X) => Output X = "ccc...c", X is optional:

The Write(X) procedure outputs the evaluated value of X to the console. The H command and the LogScreen(X) procedure can be used to log this output to the Log file. This procedure is mainly useful in CrypF code files.

WriteLn(X) => Write(X) and a line feed:

The WriteLn(X) procedure is the same as the Write(X) procedure except that the output generated is followed by an end-of-line indicator.

Function keys -

The function keys described here or used on the Run form.

F1 => Display Help form, same as the "?" primitive.

F2 => Totally Quit/end the program, same as the "Q" primitive:

If a process is running, indicated by the word **Running** displayed on the Run form, when F2 is pressed, it is treated as an Abort Calc. request.

F3 => Restore previous input command:

The F3 key normally will restore the command input text box to the value of the previously executed command input. After the B, K, or X command is executed, this key will restore the command line with the learned line. If the learned line changes, when it executes, the previous value of the learned line will be restored by this key.

F4 => Restore previous input command and accept:

The F4 key is the same as F3 except that the previous command is executed without the Enter key being required.

F5 => Get Status of processing:

Press the F5 key while a computation is being performed and a message like "Integer Multiply: 50 Percent done" will be displayed in the output text box. This is the same as the "Status of Calc." command button.

F6 => Display Configuration form:

Press the F6 key and the Configuration form will appear. F6 is the same as the "Configuration" command button.

F7 => Display Restore Input History form:

Press the F7 key and the Restore Input History form will appear. F7 is the same as the "Restore Input" command button.

F8 => Accept input and Calculate:

This will cause a non-empty contents of the input text box on the Run form to be accepted as command input. Pressing the enter/return key when the cursor is at the end of the command input will accomplish the same thing. F8 is the same as the "Calculate" command button.

F9 => Toggle Logging to Log file on/off:

If the "Logging screen to log file mode is off it will be turned on, if on it will be turned off. F9 is the same as the "Logging is On/Off" command button and the same as the "H" primitive.

F10 => Clear input text box:

This clears the input text box so you can start typing a new command. The box is also cleared when a command is accepted for execution, so this key is not needed in that case. F10 is the same as the "Clear Input" command button. In Windows the F10 key is used to activate the file menu button in the upper left hand corner of the form. Just press F10 twice to avoid this small nuisance.

F11 => Clear output text box:

This clears the output text box. F11 is the same as the "Clear Output" command button.

F12 => Pause:

This causes the program to suspend all operations until the operator clicks the "OK" button to resume. This frees up the computer if the processor is needed for a priority task. The timing function provided by the Diag command is also suspended so it will continue to give good timing information. F12 is the same as the "Pause" command button and the CrypF Pause command

Ctrl+F2 => Restart ($).

Ctrl+F9 => Clear Log File (ClearLog).

Ctrl+S => Save All (Save).

Ctrl+O => Restore All (Restore).

ESC => Clear Run form Input Text Box:

If a process is not currently running, the escape key on the RUN form clears the input text box. On the Startup, Help, About, Restore Input History, Configuration, and Change Disk Directory form, the escape key exits the form.

ESC => Interrupt/Abort a long process:

The ESC key from the Run form while a process is running is the same as the "Abort Calc" command button. It can be used after the program has been asked to perform a task that is taking longer than the operator is willing to wait. Press the ESC key once and the message box with the message:

```
*** INTERRUPT: INTERRUPT: To Continue select Ignore or Retry
To Abort Computation select Abort
```

will appear.

If Ignore or Retry is selected, the interrupt is ignored.

If Abort is selected or Space bar, Enter, or the "A" key is pressed, the message:

```
Computation aborted by operator!
```

will appear.

Page Up key => Display previous newer command:

The page up and page down keys can be used on the Run form to move newer and older previous commands into the input text box for execution.

Page Down key => Display previous older command:

The page up and page down keys can be used on the Run form to move newer and older previous commands into the input text box for execution.

Commands used in CrypF code files:

The following commands are primarily for use in CrypF code files, but can be used from the operator command input text box.

GoTo Command:

The GoTo {label} command will skip all statements following the GoTo until {label}: is found and then start executing the statements following the {label}:. If the GoTo command is in a CrypF code file, lines of input also will be skipped until the {label}: is found or until an end-of-file. It is not an error if the {label}: is not found, but a GoTo end-of-file or end-of-line will be performed in this case.

GoUpTo Command:

The GoUpTo {label} command will skip all statements following the start of the current line until {label}: is found and then start executing the statements following the {label}:. If the {label}: is not found on the current line and the GoUpTo command is in a CrypF code file, the file will be reset to the first line of the file and lines of input will be skipped until the {label}: is found or until an end-of-file. It is not an error if the {label}: is not found, but a GoTo end-of-file or end-of-line will be performed in this case.

Labels:

A label is a name followed by a colon (:). When encountered as a command, a label is a no-op. When searching for where to go from a GoTo {label} or from a GoUpTo {label} command, the {label}: is used to determine where to restart execution. If duplicate labels are on a command line or in a code file, the first one encountered is the one that is effective. Labels are alphanumeric with the first character alphabetic, have all characters significant but not case sensitive. Other non-delimiter characters can be used in labels, but this is not recommended. The delimiter characters are:

; = / : ( ) @ "

Continuation lines:

Continuation lines are indicated by the last non-blank character of the line being a + or - character. A + says, this line is to be continued by adding the next line, but a blank character should be included between them if it is needed to separate fields. A - says, this line is to be continued by adding the next line, but no extra blank characters should be included between them.

Batch Commands (Echo, @Echo, Pause, and Rem) -

Echo Command:

Normally, commands from a CrypF code file are displayed on the screen as they are executed. This can be turned off by the Echo off command and turned on by the Echo on command. If something other than or more than on or off follow the word Echo, it is considered a message and is output to the screen.

@Echo Command:

The @Echo command is the same as the Echo command except that, if it is the first command on a line, it is executed before the command line is echoed to the screen. Thus, an @Echo off at the beginning of a line will do an Echo off without the command being echoed to the screen.

Pause Command:

The pause command will output the following message to a Message Box and wait for operator to resume:

CrypF is Paused. OK to resume?

The escape key, space bar, enter/return key, or selecting “OK” will clear the pause action.

Rem Command:

The syntax of the Rem command is Rem {remark}. It is a no-op command and everything on the line following the word Rem is skipped.

// Command:

The // command is the same as the Rem command except that it does not need any spaces or delimiters before or after it.

Command buttons -

There are 13 command buttons on the Run form:

Abort Calc.: Same as ESC key.
Calculate: Same as F8 key.
Clear Input: Same as F10 key.
Clear Log File: Clears the log file, same as the ClearLog command and Ctrl+F9.
Clear Output: Same as F11 key.
Configuration: Same as F6 key.
Logging Is On/Off: Same as F9 key.
Pause: Same as F12 key or Pause command.
Quit: Quit the Run form and go back to the Startup form.
Restore All: Restore Configuration, History, & Save.CFC, same as Restore command and Ctrl+O.
Restore Input: Same as F7 key.
Save All: Save Configuration, History, & Save.CFC, same as Save command and Ctrl+S.
Status of Calc.: Same as F5 key.

If the Quit button, the File menu Exit, or the form's Close button is selected when a process is running, indicated by **Running** being displayed on the Run form, the process is automatically aborted. The message "When **Running**, select Quit twice to quit" is displayed in the text output text box and a second quit request is required to quit.

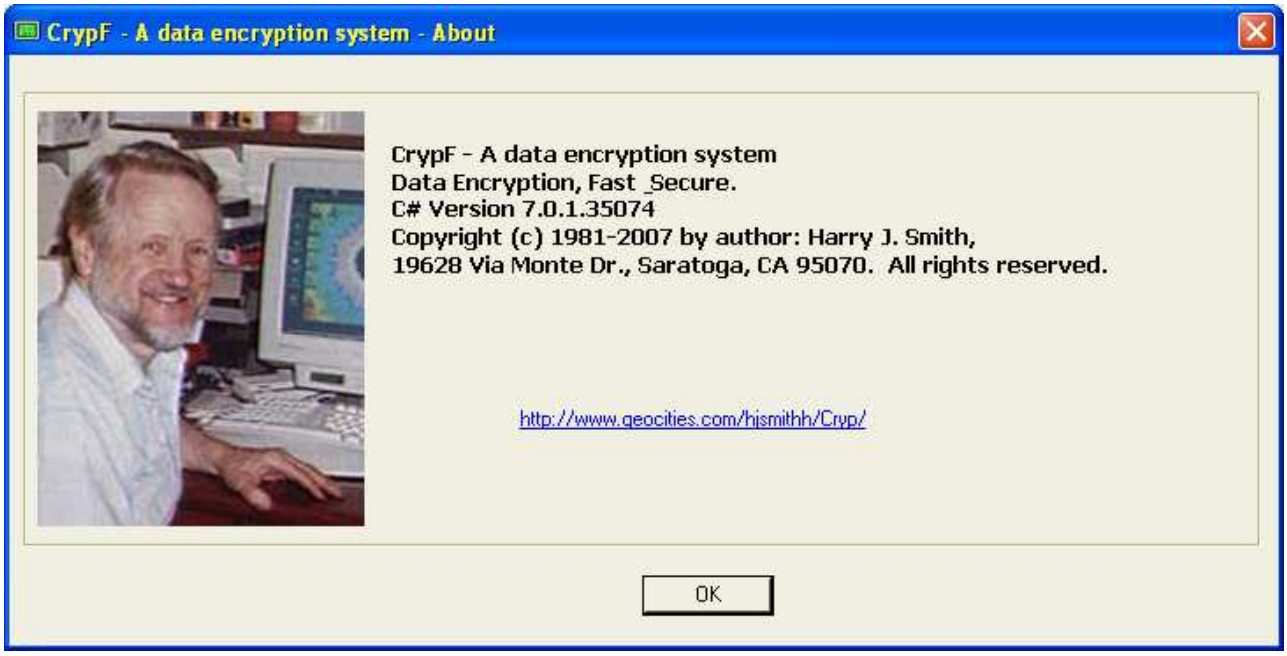
The Run form's Close button, Close menu item and Exit menu item will totally end the program without stopping at the Startup form.

The Run form has a File, a Restore, and a Help menu button. The File menu has Pause, Restart, and Exit. The Restore menu has Restore previous input command and Accept previous input command. The Help menu has Help Form, On Top and About...

Pause is the same as the Pause button.
Restart is the same as the $ command.
Exit is the same as the Quit button.

Restore previous input command is the same as function key F3.
Accept previous input command is the same as function key F4.

Help Form is the same as the ? primitive and causes the Help form to be displayed.
On Top is a check box that, when checked, will keep the Run form on top of other windows.
About will bring up a dialog box with Application Title, Version, Application Description and URL to obtain the latest version of the program:



Click the OK button or hit the escape button to remove this box.

Error reports -

There are many different error reports that are a result of directly or indirectly requesting an operation that cannot be performed (or conceivably an error in the CrypF program). Another type of error is the syntax error, where a command cannot be interpreted.

The processing error reports are:

```
Cannot Decode file "{filename}" to itself
Cannot Encode file "{filename}" to itself
Cannot open file "{filename}" for input
Cannot open file "{filename}" for output
Cannot read file "{filename}"
File write error {error-message}
Not in current directory "{directory}"
Not in home directory "{directory}"
OpenAppend: {error-message}
OpenRead: {error-message}
OpenWrite: {error-message}
Path not found!
ReadHelp: {error-message}
RestoreHist: {error-message}
```

The syntax errors are:

```
( expected: {procedure}|{string}
Directory name expected: CHDIR(|{string}
File name expected: {procedure}(|{string}
Unknown operation, Command line discarded: |{string}
```

The vertical bar | always shows the start of the string of characters that cannot be interpreted.

Other informational messages:

```
{function}: {error-message}
{key} = 24-character key in standard form
{key} = input key in standard form
{n} Commands read from History file, {m} total
{n} Commands written to History file
Aborting file input...
Aborting file write...
Command history cleared
Command line was:  {command}
Computation aborted by operator!
Continuation file "Cry.con" updated
Continuing...
Directory changed to "{directory}"
Directory name = "{directory}"
Directory not changed "{directory}"
False, files "{filename}" and "{filename}" are not equal
False, files "{filename}" and "{filename}" are not the same length
File "{filename}" closed
File "{filename}" opened for reading
File "{filename}" opened for writing
File "{filename}" verified and closed
File is corrupted: "{filename}"
Full name = "{directory\filename}"
Home directory is "{directory}"
I/O operation aborted by operator!
Key = {key}
Label skipped = {label}:
Log file "{filename}" Cleared {date} {time}
Log file "{filename}" Closed {date} {time}
Log file "{filename}" Opened for Append {date} {time}
Max commands in history changed from {old max} to {new max}
Max commands in history not changed from {max}
Priority is {priority}
Reading Commands from History file - Begin ... Aborted
Reading Commands from History file - Begin ... End
Reading Help File: "{filename}"
Running code file "{filename}"
Save file "Save.CFC" updated
T = x.xx  DT = x.xx sec.  End of execution
T = x.xx  DT = x.xx sec.  Start execution
True, files "{filename}" and "{filename}" are equal
True, files "{filename}" and "{filename}" are the same file
Went to {label}:
When **Running**, select Quit twice to quit
Writing all Commands to History file - Begin ... End
```

Other forms that can be displayed to help in using the program are the Startup, Help, Restore Input History, Configuration, Change File Path, and the Select a Fie forms. On all forms except the Run form, pressing the ESC key will remove the form.

Startup form -

The Startup form is shown above and has a parameter that can be set before the program starts accepting commands:

"Max Commands in History" can be set to a value from 1 to 100000. CrypF will allocate an array of this length to save command history. The nominal starting value is 1000 and it can be changed by the SetC(X) command

This value is tested as it is edited. Spaces on the left or right are removed so you will not see them. These numbers are displayed in **bold** face font iff they are acceptable values. If the value is unacceptable, the Run button is disabled.

A "Default Settings" button is provided to reset the parameter to its original values.

The Run button is used to bring up the Run form. It can be used after a Run form exits to reinitialize CrypF and start running again. The Exit button will terminate the program.

Help form -

The Help form was shown and partially described above. It has a File menu button that has a menu with Print Setup, Print..., Print Preview and Exit. The Print Setup will bring up the Windows printer setup dialog box so you can select the printer to use. The Print... will start the printing process to start printing the help file. This will normally bring up the printers dialog box. The Print Preview does just that, it can also be used for printing. The Exit does the same thing as pressing the escape key; the Help form is removed.

This form also has a "F3 Find:" button, a "F2 Up" button and a small text box for entering a string of characters to find. Pressing F3 is the same as clicking the "F3 Find:" button, the text in the small text box is searched for. Pressing F2 is the same as clicking the "F2 Up" button, same as F3 except the search is done from the current location towards the top of the large text box.

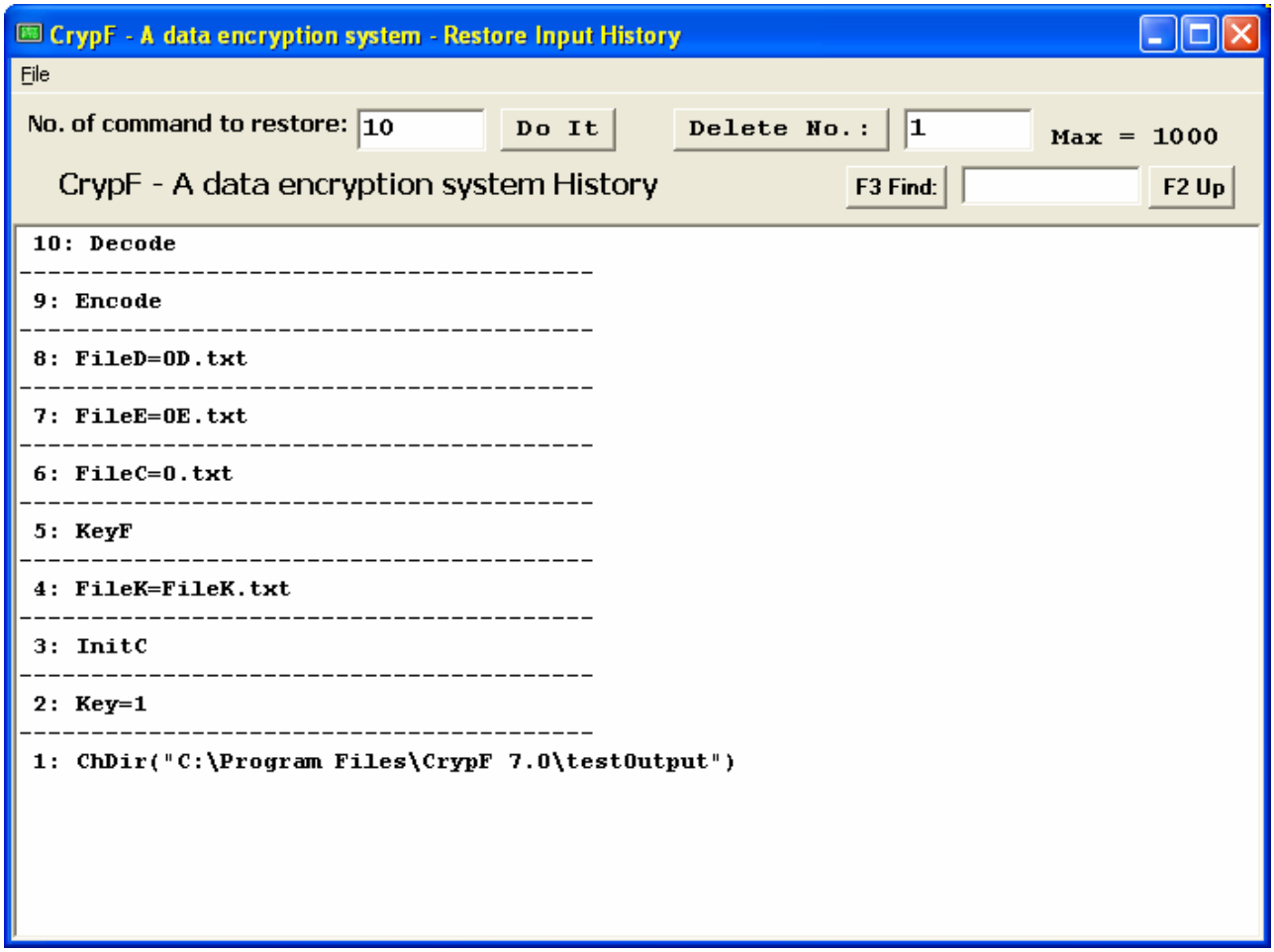
If during a find the text is not found, the sound from NotFound.wav is heard. If it is found, but only by starting over from the other end of the text, the sound from Wrap.wav is heard. This sound process uses the system file winmm.dll. If that file cannot be found, there will be no sound.

A short beep is also heard when the text is found. The beep is 277Hz = middle C# for a normal search and 554Hz one octave above for a search up. The beep process uses the system file kernel32.dll. If that file cannot be found, there will be no beep.

The Help form is unique in that more than one copy can be brought up at the same time. This may or not be useful, but it illustrates a technique in Windows programming.

Restore Input History form -

The Restore Input History form is displayed when function key F7 is pressed or the "Restore Input" command button is selected on the Run form:



The history of up to "Max Commands in History" previous operator entries are saved and can be retrieved by this form

While this form is up and the cursor is in the output text box, use the Up, Down, PgUp, and PgDn keys and the keypad + and - keys to select an entry and then use the Enter key or space bar to accept it. The command accepted will be put into the Command entry text box on the Run form, and there it can be edited before it is executed. The ESC key will remove the History form without changing the Command input text box.

The Ins key will toggle the locked status of the entry containing the cursor and move the cursor to the next higher command. When an entry is locked it cannot be deleted or scrolled off the bottom of the list. A # will be displayed to the left of a locked entry.

The Del key will delete the entry containing the cursor, if it is not locked, and move to the next higher command. The command numbers in the two command number entry text boxes are updated as commands are locked, unlocked, deleted or scrolled to by the + and - keypad keys. The - key is up and + is down due to layout of the keypad, - above +. Only the keypad + and - keys are used here.

The mouse can also be used. A left mouse click will select a command and a right mouse click will accept it.

The "Number of command to restore:" text box and the "Do It" command button can be used to select a command to be restored to the command input text box. The "Delete Command No." command button and its text box can be used to delete a given command by number. These controls act in the same way. Its text box can be

edited and a click on the command button will accept the entry. A space at the right of the text or a Return/Enter key will also accept it.

When the Delete Command No.: button is selected the command referenced is deleted if it is not locked. If locked the command number field will be increased to point to the next command or a set to 1 if at max number.

The Restore input History form has a file menu with Clear History, Restore history, Save History, and Exit.

Selecting Clear History will remove all commands saved in memory. This is the same as the ClearHist command.

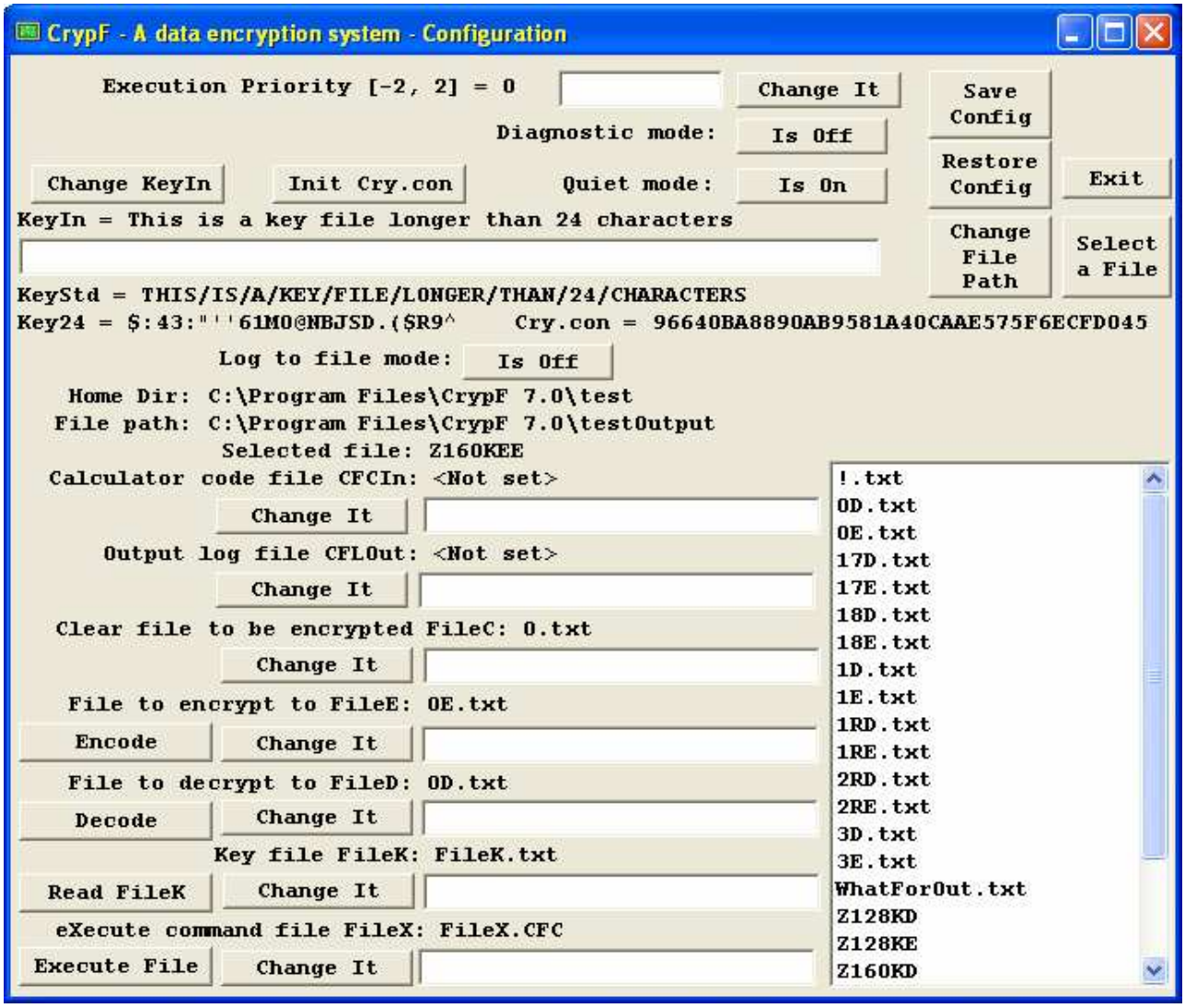
Selecting Restore History will read in the file CrypFHist.txt and add its commands to the ones in memory. This is the same as the [command. Duplicate commands are deleted as the new copy is entered, but the lock state will be as set on the copy of the command in memory.

Selecting Save History will write all of the current command history to file CrypFHist.txt. This is the same as the] command. If a command history file already exists, it will be renamed CrypFHist.Bak.

Selecting Exit does the expected, same as the escape key.

Configuration form -

The Configuration form is displayed when function key F6 is pressed or the “Configuration” command button is selected on the Run form:



This form allows you to change:

Execution Priority [-2, 2] = {p}, same as the Pri(X) command
Diagnostic mode, a combination of Diag(X) and Time commands.
Quiet mode, same as Quiet(X) command.
KeyIn, same as the Key=x command.
Log to file mode, same as the LogScreen(X) command.
Name of CrypF code file CFCIn, same as the CFCIn(F) command.
Name of Output Log file CFLOut, same as the CFLOut(F) command.
Name of Clear file to be encrypted FileC, same as the FileC=x command.
Name of File to encrypt to FileE, same as the FileE=x command.
Name of File to decrypt to FileD, same as the FileD=x command.
Name of Key file FileK, same as the FileK=x command.
Name of eXecute command file FileE, same as the FileE=x command.

It also shows the Home Directory, File path, and the name of the log file if the LogScreen mode is on. The Browse For Folder form can be brought up by selecting the "Change File Path" command button, and the Select a file form can be brought up by selecting "Select a File" command button. The Save Config button is the same as the > command to save the configuration to files Config.CFC and Save.CFC. The Restore Config button is the same as the < command to restore the configuration.

All eight "Change It" and the "Change KeyIn" buttons act in the same way. Clicking one when its text box is empty will bring up the current value. It can

be edited, and a second click with the text box not empty will accept the entry. A Return/Enter key will also accept it. A space at the right of the Execution Priority text box will also accept it.

These text boxes do not allow a space as the first character, and the accepted value will not have spaces on either end. If a KeyIn with spaces on either end is desired, the character '/' should be used, like ///1//. The final file names can have internal spaces, but will never start or end with a space. File names can also have directory information like C:\X.CFC.

Clicking the "Init Cry.con" button is the same as the InitC command, the Cry.con file is initialized using the current Key. The current 18 bytes of the Cry.con file is always displayed in hexadecimal. For example, a KeyIn = 1 gives

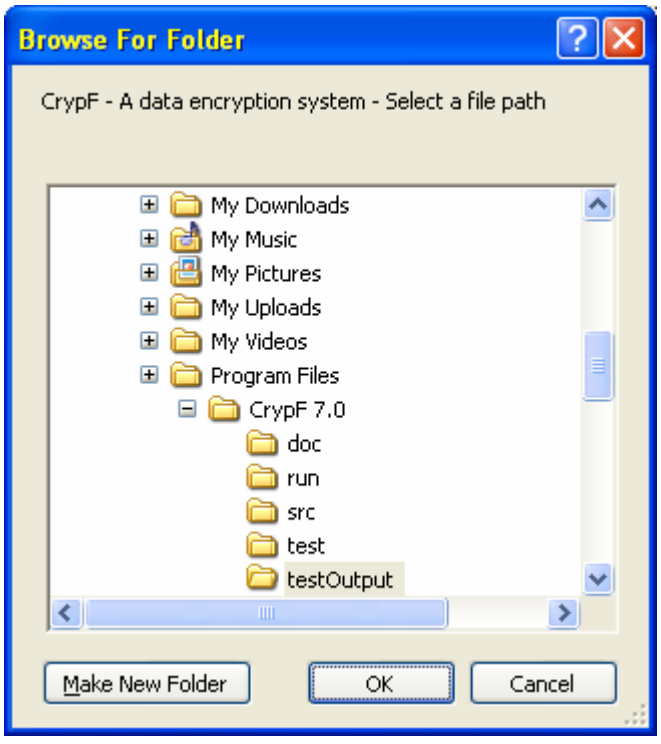
Cry.con = 8170968691454FD7293A61FB27015BCBCBD9

The three modes can be changed/toggled by clinking the corresponding button. If a mode is on, the button will says "Is On", then clicking it will turn the mode off and the button will change to "Is Off". If a mode is off, the button will says "Is Off", then clicking it will turn the mode on and the button will change to "Is On". The Diagnostic mode differs, it has four states, Is Off, Time, Diag, and Debug.

The Log to file mode and Name of output log file CFLOut interact in that if the file name is changed while a log file is open, the file name of the open file will not change until the log file is closed and reopened.

Browse For Folder form -

The Browse For Folder form is displayed when the "Change File Path" command button is selected on the Configuration form:



To change the current directory to be used by the H, I, J, W, ReadN(F), and Run(F) commands, use this form to select a path. This form is resizable as is the Run, Help, History, and Select a file forms.

Select a File form -

The Select a file form is displayed when the "Select a File" command button is selected on the Configuration form:

This provides a way to browse and select a file. If a file is select, it can then be used to set a file name on the Configuration form by clicking one of the file name "Change It" buttons. The first click enters the file name in the text box and a second click will accept it. This form can also be used to change the file path.

The list box in the lower right hand corner of the Configuration form is a list of the files in the currently selected file path. The wild card selection for these files (*.XI?, *.txt, or *.*) is the same as last used on the Select a file form.

The distribution files can be downloaded from my website:

http://www.geocities.com/hjsmithh/

in the **Files to Download** section

http://www.geocities.com/hjsmithh/download.html#CrypF.

When you install the program using the distribution file CrypF70?.exe or CrypF70?.zip, a folder is created with 5 subfolders:

```
├──CrypF 7.0
│   ├──doc
│   ├──run
│   ├──src
```

```
│   ├──test
│   ├──testOutput
│   └──testVerify
```

The main folder has two files sseexec.dat and SSEun.dat. These are needed to be able to uninstall the program

The doc subfolder has the following 7 files:

```
!.txt      (Information on version)
CRYP.doc (Basic documentation for the earlier programs, still applies)
CRYP.txt (Plain ASCII text copy of the CRYP.doc file)
CrypF.doc (This documentation in a Microsoft Word file)
CrypF.txt (Plain ASCII text copy of the CrypF.doc file without graphics)
Fixes.txt (A list of features and fixes added to each version)
Note.txt  (Old note written 7/10/1994)
```

There is also available on my website a copy of the .doc file in an Adobe Acrobat Reader file CrypF.pdf.

The run subfolder has the following 10 files

```
!.txt
AutoExec.CFC
AutoVista.CFC
AutoXP.CFC
Cry.con
CrypF.exe
CrypF.exe.config
CrypFHelp.txt
NotFound.wav
Wrap.wav
```

The .exe file is executed from there.

The src subfolder has all the source files needed for development:

```
!.txt
00Note.txt
AboutForm.cs
AboutForm.resx
app.config
AssemblyInfo.cs
Calc.cs
Common.cs
ConfigForm.cs
ConfigForm.resx
COPYING.txt
Cryp.cs
CrypF.csproj
CrypF.sln
CrypF.suo
CrypF.csproj.user
cs.ico
Global.cs
harry.jpg
HelpForm.cs
HelpForm.resx
HistoryForm.cs
HistoryForm.resx
MultiID.cs
RunForm.cs
```

RunForm.resx
StartupForm.cs
StartupForm.resx
UpgradeLog.XML

The test subfolder has the files I use for testing the program. They are:

!.txt
0.txt
1.txt
17.txt
18.txt
1R.txt
2R.txt
3.txt
AutoExec.CFC
AutoVista.CFC
AutoXP.CFC
Config.CFC
Cry.con
Cry1.con
CrypFHelp.txt
CrypFHist.txt
FileC.txt
FileD.txt
FileE.txt
FileK.txt
FileX.Bat
FileX.CFC
FileX.CFL
K.txt
NoName.CFL
NotFound.wav
Save.CFC
Test.Bat
Test1.CFC
Test1.CFL
TestArgs.Bat
WhatForTst.txt
Wrap.wav
X.Bat
X.CFC
X.CFL
Z
Z128K
Z160K

The testOutput subfolder has the output files I use for testing the program. They are:

!.txt
0D.txt
0E.txt
17D.txt
17E.txt
18D.txt
18E.txt
1D.txt
1E.txt
1RD.txt
1RE.txt
2RD.txt

```
      2RE.txt
      3D.txt
      3E.txt
      AutoExec.CFC
      Cry.con
      TestV.Bat
      TestV.CFC
      TestV.CFL
      WhatForOut.txt
      Z128KD
      Z128KE
      Z160KD
      Z160KE
      Z160KED
      Z160KEE
      ZD
      ZE
```

The testVerify subfolder has old output files I use to verify that CrypF still works correctly, by using the CrypF code file TestV.CFC. These files were actually generated by:

```
      CRYP - A data encryption system written in C, with pipes
      Version 6.00, 1992-11-15, 1400 hours
      Copyright (c) 1987-1992 by author: Harry J. Smith
```

The files are:

```
      !.txt
      0D.txt
      0E.txt
      17D.txt
      17E.txt
      18D.txt
      18E.txt
      1D.txt
      1E.txt
      1RD.txt
      1RE.txt
      2RD.txt
      2RE.txt
      3D.txt
      3E.txt
      WhatForOut.txt
      Z128KD
      Z128KE
      Z160KD
      Z160KE
      Z160KED
      Z160KEE
      ZD
      ZE
```

The program can generate the following files:

```
      Config.CFC
      Cry.con
      NoName.CFC
      NoName.CFL
      CrypFHist.Bak
      CrypFHist.txt
      Save.CFC
      {encrypted-file}
      {decrypted-file}
```

The end -

Report any errors by sending me a letter, an e-mail or call me at my home phone.

-Harry

Harry J. Smith
19628 Via Monte Dr.
Saratoga, CA 95070-4522, USA

Home Phone: 1 408 741-0406
E-mail: hjsmithh@sbcglobal.net
Website: http://www.geocities.com/hjsmithh/