

## NUMORSE HELP INDEX

**It is a condition of use that the Author shall not be liable for any direct, indirect, incidental or consequential damages arising from the use of the NuMorse program or any information connected with the program.**

Topics available from the help index:

[Overview of NuMorse](#)  
[Alphabetic listing of topics](#)  
[Keyboard and mouse](#)  
[Procedures](#)  
[Technical reference](#)  
[NUMORSE.INI](#)  
[Instructions on using help](#)  
[A word from the author](#)  
[Program Registration](#)  
[QSO and Test Generation](#)

**NuTest!** Pass your written FCC tests first time! Big discounts for NuMorse registered users!

**Click on an underlined topic.**

## ALPHABETIC LISTING OF TOPICS

(Also see the [QSO generator index](#))

[@](#)

[48 WPM](#)

[accelerator keys](#)

[adapt](#)

[addressing](#)

[Africa](#)

[alphabetic filter](#)

[ascii text](#)

[attempts allowed](#)

[audible warning](#)

[author](#)

[background operation](#)

[buffered input](#)

[calibrate](#)

[character display](#)

[clear source file](#)

[response statistics](#)

[click](#)

[Click Filter](#)

[clock/timer](#)

[closing down](#)

[code filters](#)

[code sent button](#)

[code source, control codes](#)

[code source](#)

[control codes](#)

[control](#)

[credit cards](#)

[custom filter](#)

[custom](#)

[dash](#)

[data to port](#)

[default, restoring to](#)

[default settings](#)

[de installing NuMorse](#)

[display prosigns](#)

[display statistics](#)

[distribution, random](#)

[dot](#)

[drill audible warning](#)

[drill mode](#)

[drill session time](#)

[drill settings](#)

[drill show correct character](#)

[drill tries allowed](#)

[drill visual warning](#)

[Editing custom filter](#)

[editing external files](#)

[editing NUMORSE.INI](#)

[EMail Registration](#)

[embedded control codes](#)

[envelope controls](#)

exiting from NuMorse  
f10  
f1  
f9  
Farnsworth code  
faster  
FCC Character counting  
file, open  
file, save as  
file, source  
file, serial source  
file editor  
file menu  
file open dialogue box  
file random source  
filter code sent button  
filter custom  
filter letters  
filter numbers  
filter presets  
filter prosigns  
filter punctuation  
filters dialogue  
finish  
Flashing Window  
frequency  
generator, random  
groups, random  
guessing code  
hardware  
help instructions  
help  
help me  
history box  
history of program  
hot keys  
incorrect response  
index  
instructions  
invert custom filter  
keyboard, code source  
keyboard and mouse  
keypress  
learning Morse code  
letters, filtering  
load default settings  
log files  
main  
Mixing speech and code  
Morse code drill  
MORSE.INI  
mouse  
note  
number sequence  
numeric filtering  
NuMorse main window

NuMorse overview  
NUMORSE.INI  
NuTest  
open file in file window  
open settings file  
open source file  
opening source file  
output device  
PARIS  
performance  
pitch  
play continuously  
play menu  
play mode  
play settings dialogue  
playing Morse code  
port  
postal registration  
preferences dialogue  
preset  
printer  
procedures  
problems  
prosigns, displaying  
prosigns, filtering  
pseudo random source  
pseudo random  
punctuation filter  
QSO generation  
quit  
random seed  
random stream  
random weighted  
random weighting  
Registration form  
registering NuMorse  
reset source file  
response  
response time thermometer  
responses, reviewing  
retries  
review  
running NuMorse  
running other applications  
save as in file window  
save settings file  
save  
seed  
select output device  
select run path  
sent code, reviewing  
sent code  
serial source file  
session time  
session timeout  
settings, default

[settings, opening file](#)  
[settings menu](#)  
[shortcut keys](#)  
[Sine wave](#)  
[showing character](#)  
[slower speed](#)  
[SoundBlaster™](#)  
[sound files](#)  
[source, serial](#)  
[source filename, clearing](#)  
[source for code](#)  
[source setting](#)  
[South Africa](#)  
[spacing, character](#)  
[spacing, word](#)  
[Speech from source files](#)  
[Speech Output, drill mode](#)  
[speed, adaptive](#)  
[speed, adaptive](#)  
[speed, setting in timing dialogue](#)  
[speed control](#)  
[Square wave](#)  
[starting NuMorse](#)  
[statistics](#)  
[stream, random](#)  
[summary](#)  
[technical reference](#)  
[test generation](#)  
[testing](#)  
[text file for code source](#)  
[Text speed](#)  
[thermometer](#)  
[time for responding](#)  
[time out](#)  
[timer](#)  
[timing calibration](#)  
[timing character spacing](#)  
[timing dash length](#)  
[timing dot length](#)  
[timing frequency](#)  
[timing parameters](#)  
[timing word spacing](#)  
[tone](#)  
[training](#)  
[tries allowed](#)  
[typeahead, keyboard](#)  
[using the mouse](#)  
[Version number](#)  
[view](#)  
[Voice Output](#)  
[Volume control](#)  
[visual feedback of errors](#)  
[wave files](#)  
[warn](#)  
[whats new in this version](#)  
[window](#)

windows

wpm

wrong

wrong

## Overview of NuMorse

NuMorse is a Windows application designed to help you to learn Morse code. It has many features intended to make this task as painless as possible. Here is a selection:

Alter timing to send code as slowly as you like, speed up as you become more proficient.

Start with just a few characters, add extra characters whenever you wish using filter feature.

Play characters from random stream, disk file or keyboard using character source feature.

Use timing options to alter code style from standard Morse to enhance the clarity of the code.

Morse speed, characters sent, style can all be saved to file and then reloaded.

Most Prosigns are supported.

Several drill modes, guess the character sent. Configure how NuMorse responds to your guesses.

NuMorse will gather detailed statistics on your performance while training, you can then use the statistics to adjust code settings.

Adaptive mode where NuMorse adapts code speed to your abilities.

Log files to record code output as well as user responses, cut and paste log file contents to other applications such as word processors. Print out log file contents.

A large number of code filter options. You may apply random number weighting so that certain characters are sent more often than others.

Send code output to a hardware device such as a tone generator or semaphore window.

Control speed and display your own messages during a Morse practice session using embedded control codes in a code source file.

Mix speech with code in source files.

## **NuMorse help Instructions**

For more complete instructions press the F1 key while the Help window is active.

### **To return to the index**

Click the Contents button at the top of the Help window.

### **To use the Help Index**

Click an underlined topic.

or Press the TAB key until you highlight the topic you want, and then press the ENTER key.

### **To scroll in the Help window**

Press PageUp or PageDn keys.

or Use the scroll bars with the mouse.

### **To return to the previous topic**

Click the Back button or press the B key.

### **To open the Topic Search feature in Help**

Click the Search button or press the S key.

### **To close the Help window**

From Help's File menu, choose Exit (ALT, F, X).

or Press the ALT F4 key.

or Click on the minimise icon.

or Double-click the Control-menu box in the upper-left corner of the help window.



## Technical Reference

### Windows 3.1 Multimedia Sound Functions.

Windows 3.1+ Multimedia Sound Functions are supported from NuMorse V1.20 onwards. Using the PC's internal speaker you are normally stuck with one volume level, there is a way round this problem without having to purchase a sound board. If you install one of the freeware sound board emulation drivers (such as the Microsoft™ Speaker Driver) and then set up NuMorse to send to sound board using the device select dialogue you may be able to control the PC speaker volume using the volume control in the Sound Board Dialogue box. The Microsoft™ Speaker Driver has given very variable results in the machines I have tested it on, don't expect brilliant results from it. There is also a facility for driving an external tone oscillator via a user selectable hardware port.

### The development system.

The hardware used:

Multiplex™ 386DX running at 33Mhz with 8Mbyte Ram, Math co-processor, and Doublespaced™ 500Mbyte HD.

S3 V24 BitBlaster SVGA board, 1Mb VRAM, running in 640X480, 800X600 and 1024X768modes.

Microsoft™ mouse and driver

Escom™ Multimedia Sound Card V2.0

Software:

Windows™ for WorkGroups running on top of MSDOS 6.2.

Source is mainly Pascal sprinkled with a little assembler. Compiler is Borland Pascal 7 for Windows™. Resources were designed using Borlands Resource Workshop™. This help system was set up using the Microsoft™ Help Compiler V5.05 with the help text entered into Microsoft Word for Windows™.

## **Keyboard and Mouse**

Using a mouse

Using the keyboard

Keyboard Hotkeys

## **Procedures**

Starting up

Playing Morse code

Learning Morse code

Exiting NuMorse

## Hotkeys

Keyboard hot keys allow rapid movement around the program, instead of selecting menu items you just press a hotkey combination. Keyboard hotkeys are displayed alongside menu items where relevant. Here is a list of available hotkeys:

Hotkeys used in most Windows applications, for explanations refer to your Windows Users Guide:

Alt+Space	Open the control menu.
F10	Go to files menu.
F1	Get help.
Alt+F4	Close window, exit from NuMorse if in main window
Ctrl+Esc	Bring up task list

Hotkeys specific to NuMorse

F3	<u>Open a settings file</u>
F2	<u>Save current settings file</u>
Ctrl+F9	<u>Continuous play dialogue box</u>
F9	<u>Drill dialogue box</u>
Alt+F9	<u>Statistics dialogue box</u>

## **Using the Mouse**

Standard mouse usage is implemented throughout the program, only the left mouse button is used. To select a menu item point to it and click.

To get help when using a mouse click on the help menu, then click on the required item. Many of the dialogue boxes in NuMorse have a help button, click on this to summon help.

A quick way to exit from the current window is to double click on the control menu icon. This is the small box that normally appears at the top left corner of a window.

In a dialogue box click on the required control to alter its state. Some controls have associated scroll bars, click and drag on these to bring hidden text into view in list boxes.

## **The File Menu**

Here is a list of the commands available from the File menu:

- Source open
- Source clear
- Settings open
- Settings save
- Settings save as
- Review characters sent
- Review user responses
- Exit

## **The Play Menu**

Here is a list of the commands available from the Play menu:

Play  
Drill  
Statistics

## **The Settings Menu**

Here is a list of the commands available from the Settings menu:

Filters

Timing

Drill

Play

Preferences

Defaults

Calibrate



## Starting NuMorse

To start up the program double click on the NuMorse icon in the NuMorse program group. Another way to start up the program is to type WIN NuMorse at the DOS prompt.

NuMorse can also be run via File Manager by double clicking on the NUMORSE.EXE directory entry.

When NuMorse launches it will spend a short period loading the file NUMORSE.INI from your \WINDOWS directory. If this file is not detected NuMorse will start up anyway and then initialise NUMORSE.INI automatically with a set of predefined values.

As soon as the main menu is displayed you may go straight to the Play menu, or use the file menu to recall previously saved settings.

## Exiting from NuMorse

To exit from NuMorse get back to the main menu. If you are in a dialogue box there are various ways of getting back to the main menu:

1. Click on the cancel button, any changes made in the box will be discarded before the box is closed.
2. Press Esc, this is equivalent to clicking on Cancel.
3. Double click on the system menu icon, top left corner, equivalent to cancelling.
4. Click on the system menu to open it, then click on Close.
5. Click on OK, changes will be saved before the box is closed.
6. If the Cancel or OK button has the focus (indicated by a thick black outline) then pressing the ENTER key will close the dialogue box

Here are some ways to exit from the program from the main menu:

1. Select File/Exit
2. Double click on the system menu icon, top left corner.
3. Press Alt+F4
4. Click on the system menu to open it, then click on close.

If any program settings were altered you will be asked if you wish to save the new settings before leaving NuMorse. If you select OK with the Save as default checkbox checked then the current settings will be saved to the NUMORSE.INI file and will be the settings used by default the next time the program runs.

## **Playing Morse Code**

To play Morse code select Play/Continuous or Ctrl+F9 and the Play dialogue box will open. Click on the Play button to start. Click on the Stop button for a temporary pause. You may need to use the Settings menu or Speed control to adjust the Morse code output to your needs. When playing NuMorse always sends its output to a log file and any user responses will also be sent to a further log file.

## Learning Morse code

Here are few suggestions for learning the Morse Code:

When you start dont set the speed TOO slow. You need to get a feel for the rhythm of the individual characters and a very slow speed can hinder this process.

Experiment with word and character spacing, Morse code sent at moderate speed, with long gaps between characters is good to get you started. You can then reduce the spacing as you get more proficient.

Familiarise yourself with a few letters first, then add new characters in small groups of 4 or 5 at a time. Use the [Custom Filter](#) to select a small group of characters.

NuMorse has the ability to record your keystrokes along with its own output in [log files](#) which can subsequently be viewed or printed out, you may find these useful.

Once you have learned most of the characters then it is possible to concentrate on the ones you are finding difficult. NuMorse has a feature which allows incorrect characters to be output more frequently in subsequent practice sessions. See the section on [Custom Filters](#) for more details.

Once you are fluent in Morse Code then use the [Increase Speed](#) and [Adapt Speed](#) features to get up to speed.

## **Open source file**

The Open source file command displays a File Open dialogue box.

In this dialogue box, you select the file you want to use when playing Morse code from a file. The option to play Morse from a file is chosen in the Source Settings options in the Play/Drill dialogue box. NuMorse is not too fussy about the type of source file selected.

Most word processors and text editors should produce suitable files. The built in log file editor is ideal as it produces a minimum of extraneous control characters.

NuMorse will respond to certain embedded control codes which can easily be added using any text editor.

**Clear source file**

If a source file is open then after confirmation the name will be removed and the file closed; the file itself is not affected. Source files are never written to or altered by NuMorse (unless you edit them using the built in [log file editor](#)) and they do not need to be explicitly closed.

Before exit from NuMorse all open files are automatically closed.

For further details on source files see the [Source Settings](#) topic.

## **Open settings file**

When NuMorse starts up it obtains its default program settings from a file in the WINDOWS directory called NUMORSE.INI. You may load alternative program settings using this menu option. The settings will have been saved from previous program runs using the Save settings file menu option.

The Open settings file command displays a File Open dialogue box. In this box you select which group of settings you wish to reload from file.

All of the Settings menu options are stored. If a code source file was open when the settings were saved then this file will be reopened.

## **Save settings file**

The Save command saves the settings in the current Settings Menu to disk.

If the file has no name then NuMorse opens the File Save As dialogue box so you can rename the file and save it in a different directory or on a different drive.

If you use an existing file name to name the file, NuMorse asks if you want to overwrite the existing file



## **Save settings file as**

The Save as command saves the settings in the current Settings Menu to disk.

NuMorse opens the File Save As dialogue box so you can rename the file and save it in a different directory or on a different drive.

If you use an existing file name to name the file, NuMorse asks if you want to overwrite the existing file.

## **Morse code play**

This dialogue box displays the character currently being sent as Morse code, as well as the last few characters sent. When NuMorse is displaying this dialogue box it is in **Play Mode**. A visual interpretation of the code itself is also shown, which may help speed up the learning process. The on-screen keyboard display is used mainly in the Drill dialogue; in the play dialogue box; the keys corresponding to the current Morse code character will be seen to press. All characters played as well as any key presses are recorded by NuMorse in a log file. This means that you can practice reading Morse code in a situation similar to a real life, without any pauses or distractions from the program. When you have finished you may open both and compare what NuMorse sent with your own guesses. If you wish to remove all clues as to the character being sent by NuMorse then a set of options, play settings, is available which allow you to remove any visual indication about the character being sent.

An on-screen timer/clock is displayed in Play and Drill modes.

## Morse code drill

This dialogue box appears exactly the same as the [Play](#) dialogue box , the difference is that the character being played is not displayed (unless you have set an option to display it in the [Drill settings](#) dialogue box), you have to guess it and enter it. To input your guess:

1. Enter the character by pressing the corresponding key on your keyboard
2. Use a mouse to point at the on-screen keyboard, then click.

If you guess correctly then the character is displayed and the next character is sent. If your guess is wrong then the response from NuMorse depends on the [Drill Settings](#) options. A typical response might be to allow one or more incorrect guesses before moving onto the next character, with an audible and visual indication that each guess was wrong. NuMorse gathers [Statistics](#) on your progress while in Drill mode. The statistics that were gathered can be used to set up the [custom filter](#) to emphasise those difficult to learn characters in future sessions.

The group of keys shown separated from the keyboard are for some of the characters that are normally accessed as a shift-key combination. They have been arranged like this to allow mouse selection without the use of the keyboard shift key. The > and # characters do not correspond to true Morse code characters; they can be used to generate the prosigns (CT) and (HH). See the [Prosign Mapping](#) help topic for more details.

As in the Code Play dialogue all characters played as well as any key presses are recorded by NuMorse in a [log file](#).

As well as the [clock/timer](#) a response time [thermometer](#) is optionally available.

## **Display statistics**

Statistics are gathered whenever NuMorse is sending Morse code in the Play or Drill dialogue box; they are reset every time one of these boxes is re-opened. Items such as response time are only appropriate when the drill dialogue box has been active, otherwise they will be set to zero. Other items such as session time will be valid for both Play and Drill sessions. The only active control in the statistics dialogue box is a list box containing statistics for each character sent in the session. Scroll the list box using the scroll bar or use the cursor control keys.

The session time shown is for the LAST code session before the Play or Drill dialogue box is closed. So if you do repeated code stops and starts while in these dialogue boxes then only the last of the series will appear in the Statistics dialogue box.

The statistics for the most recent session can be fed into the custom filter to allow the user to concentrate on the characters that have been incorrectly guessed.

## Select source for code

Open the Play or Drill dialogues. To choose how the Morse code characters are generated click on the required button to select the source. Click on the Play button to actually hear the Morse code.

Here are the available options:

Serial file  
Random File  
Random  
Groups.  
Keyboard.  
Top window  
QSO

Whichever option you choose the Morse code output will be subject to the effect of a built in character filter mechanism.

**Random stream.**

Characters are randomly generated in a continuous stream. Actually, the stream is the same pseudo random sequence starting from the same seed number each time the Morse play dialogue is activated. If you would prefer a more random sequence then read the section on NUMORSE.INI.

If the custom filter box has been checked in the Filters dialogue then the random number generator operates somewhat differently. It will emphasise those characters with the greatest filter values.

**Random Groups.**

This is the same as Random stream, except that a pause is inserted every five characters. The pause is shown as a space in the character History Box in the Morse Play dialogue.

**File serial.**

This causes NuMorse to read characters from a text Source File, if opened. The characters are read in sequence starting at the front of the file. When the file end is reached you are given the option to start again or finish Morse code output. If you break off in the middle of the file and then resume an option in the Preferences dialogue determines whether transmission resumes from this break or starts again at the beginning of the file. The action taken at end of file depends in an entry in NUMORSE.INI



**File random.**

This causes NuMorse to read characters from a text Source File, if opened. NuMorse searches the file at random for whole words, that is strings of characters separated by spaces (and any other control codes). Each complete word is sent as Morse code.

**Keyboard.**

This option allows you to send Morse code by typing in the required characters from the keyboard in real time, use the space bar for inter-word spacing. The CCITT recommendations are used for certain prosigns which are not represented on a standard PC keyboard. It is fairly easy to type faster than Morse code can be transmitted, and so a fairly large (100 characters) type ahead buffer is provided. See the Play Morse dialogue for further details on keyboard input.

## **Top window**

Text is read from the topmost (or active) file window on the NuMorse desktop and played as Morse Code. You can make any file window topmost by clicking on its Title Bar and the topmost window can be identified by a different colour for its title bar. The character currently being played will be highlighted in the topmost file window. If you wish to play just a part of the text then select (by clicking and dragging) the required part of the text before opening the Drill or play Dialogue boxes.

## **QSO**

The [QSO generator](#) will write QSO messages to the [QSO window](#) and then play them as Morse Code.

See also:

Step by Step [QSO generation](#)

## Select code character filter

This dialogue box is where you select what characters are output as Morse code. This feature is useful for those learning the code who wish to practice with a subset of the available characters. Think of this feature as filtering out undesired characters, only the characters selected will be output. Bear in mind that the character filter operates on characters from any source. If you are entering characters from the keyboard and some of the keys do not seem to function then check what characters are being filtered. If the character source is a file it is possible to set up NuMorse to filter ALL of the file characters out before they get converted to Morse code; it can appear as if nothing is happening. In this situation NuMorse will eventually time out with an error message.

NuMorse will not allow exit from this dialogue box if you try to filter out all characters!

Here are the available character filter options:

Letters  
Numbers  
Punctuation  
Prosigns  
Custom

Except for the Custom option you can combine the above to provide various mixtures of character types.

There are 5 buttons in this dialogue box that control the contents of the custom filter edit control:



Preset Custom button is used to preload a group of characters into the custom filter.



Invert Custom button alters all of the custom filter values.



Code Sent button uses statistics on code sent during the last drill or play session to set up the custom filter.



Responses button uses statistics on user responses gathered from the last drill session.



Standard Filter preloads a group of characters into the custom filter. This group represents the relative frequencies of letters in English text.

When any of these 5 buttons are pressed the custom filter check box is also set.

## Choose code timing parameters

In this dialogue box you set the timing of the various elements of the Morse code generated by NuMorse. To adjust a timing scroll bar click and drag on the slider, or click on the small arrow symbols at the scroll bar ends.

The WPM display is based on dot length and assumes standard spacing. Other spacings will give inaccurate results, but may be useful in the early stages of learning the code.

Element lengths are displayed in milliseconds. With the speed calibration factor set at 100% a dot length of 80 units corresponds to a speed of about 15 W.P.M. on the Authors system when continuously sending the word PARIS at standard Morse code spacing.

Here are the parameters that you can alter:

Dot length  
Frequency  
Dash Length  
Character spacing  
Word spacing

Two buttons are provided for automatically setting up the timing parameters, this is how they work:



sets standard spacing based on a dot length of 80 milliseconds.



sets standard spacing based on currently displayed dot length.

The radio buttons affect Morse code output in the following way:

Constant speed means just that, code speed will not vary through a practice session; unless the code source is a serial file and contains certain embedded control characters of the type that NuMorse recognises.

Increase speed means that at certain intervals, determined by an entry in NUMORSE.INI, the speed will increase by a small amount up to the maximum. This will occur in Play and Drill modes.

Adapt speed works differently, it is based on number of characters output instead of a set time period. In Play Mode the speed will increase by a small amount after outputting a number of characters as determined by an entry in NUMORSE.INI. In drill mode the speed will adjust itself depending on how well the user guesses the characters. Bad guesses will slow down the code, good guesses will cause it to speed up.

## Select drill settings

This group of options affects how NuMorse functions when in Drill mode. Here are the options:

Tries allowed

Response time

Session time

Visual warning

Audible warning

Show correct character

Display Morse Code

Display character sent

Display response thermometer

Speech Output

## **Speech output**

This feature may be useful for people with sight problems. It functions in drill mode when the Drill Settings speech output option is checked and a sound board is the output device. When an incorrect response is given NuMorse normally issues a Hu-Hu sound to indicate your mistake. If the speech output feature is active then an incorrect response will cause the expected character to be spoken, the speech is read from a set of .WAV files



**Display Morse Code**

If this option is checked then as the Morse character is sounded the code is displayed graphically as a set of dots and dashes. If you would rather not see the code characters then switch this option off.

**Display character sent**

If this option is checked then the character that NuMorse is waiting for is displayed. This makes drill sessions rather easy since the character that NuMorse is waiting for you to enter is shown on the screen! Nevertheless this feature may be useful to people just starting to learn the Morse code and who getting familiar with the way that NuMorse works.

## **Preferences dialogue**

The options in this dialogue are ones that dont seem to fit into any other specific category, if NuMorse continues to evolve then this dialogue box may start to fill up as users ask for extra features! Here are the current options:

Reset source file  
FCCCharacter counting

If an option does not appear here then check the section on NUMORSE.INI the option may be one that must be set by direct editing of this file.

## **Reset Source File**

This affects the behaviour of NuMorse when the Play or Drill dialogue box is cancelled during reading of a text file. When the dialogue box is reopened the file will start from the beginning if this box is checked, otherwise the file read will continue from where it was interrupted. There is an entry in NUMORSE.INI which determines the action to be taken when end of file is encountered.

### **Load default settings**

After confirmation all of the settings menu options will be restored to their default state. This is done by loading a set of default values from the file NUMORSE.INI which will have been automatically set up in your WINDOWS directory. With care you can alter these default values, before you do this read the Section on [NUMORSE.INI](#)

## Calibrate code timing

This dialogue box contains two scroll bars and a small edit box. The setting of the WPM control affects the WPM readout in the Timing dialogue box, it does not affect actual code speed. At the default setting of 100% a speed of 15 WPM corresponds to a dot length of 80 milliseconds. This was arrived at by a little trial and error using the standard Morse sequence PARIS. Other methods of checking code speed in WPM will give slightly different results; for example the Morse sequence CODEX might be more appropriate under certain circumstances. Feel free to alter the WPM calibration setting as required.

The default button in the Timing dialogue always sets the dot length display to 80 milliseconds: this will not correspond to 15 WPM if the WPM calibration setting is not 100%. This can be put to use if you need to use units other than W.P.M. For example some European countries use letters per minute which is 5 times the W.P.M. value, the correct WPM calibration factor in this case would be 20. If you need to use speed units other than W.P.M. you can also alter the displayed text from W.P.M. to any other text, up to a maximum of 6 characters using the edit box in this dialogue.

The other scroll bar affects actual code speed for a given dot length. Thus for a displayed dot length of 80 milliseconds the actual code speed can be somewhat more or less than 80 ms. depending on the setting of this scroll bar.

Do not overlook the use of the windows clock. When doing calibration checks it is handy to have a continuous display of time on screen. Look for a code source file called PARIS.TXT among the sample files provided. This file is provided to help you check NuMorse code speed on your system.

The Index contains a list of Help topics available in the NuMorse Help System. For information on how to use Help press F1 or choose Using Help from the Help menu. You can get back to the Index from any point in the help system by clicking on the Toolbar Contents key.

The Windows Help system gives general information about using Microsoft™ Windows and can be accessed by pressing the F1 key while the NuMorse Help Window is active.



## **File Open dialogue box**

The File Open dialogue box is where you open a file by typing the file name in the input box or using the directories list box or file name list box to find and open the file.

The File Name input box is where you enter the name of the file , or the file-name mask to use as a filter for the Files list box.

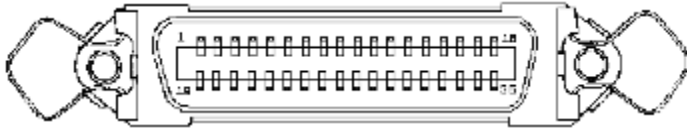
The Files list box lists the names of files in the current directory that match the file-name mask in the File Name input box, plus the parent directory and all sub directories.

You view the contents of different directories by selecting a directory name in the Directories list box.

Control characters are used by many word processors to control the layout and appearance of text. They are usually not displayed on screen, but will be present in document files. Within reason, NuMorse will try to ignore such characters.

## Printer port pinouts.

Here are the data pinout details for a Centronics parallel printer port, this view is looking into the rear of the PC.



Pin	Signal
2	Bit 0
3	Bit 1
4	Bit 2
5	Bit 3
6	Bit 4
7	Bit 5
8	Bit 6
9	Bit 7
33	GND

### **File Save As dialogue box**

The File Save As dialogue box is where you type in the new name in the file name input box (you can include a drive and directory path) or use the directories list box to select a new path.

If you choose an existing file name, NuMorse asks if you want to overwrite the existing file.

## CCITT

The CCITT have made certain recommendations regarding conversion of ASCII codes to Morse and other codes. NuMorse provides support for most of the prosigns. NuMorse optionally displays the dual letter combination or the ANSI substitute when outputting Morse code. You may select how prosigns are displayed in the Prosign Mapping Dialogue box.



Prosigns (Procedure Signals) are signals used in Morse Code to indicate such phrases as start of transmission, here is and so on. The FCC code tests require knowledge of:  
AR,SK,BT,DN

Type-ahead is a feature that allows typed characters to be stored until the system can deal with them, it can be thought of as a queue in which the typed characters are held in the exact sequence in which they were typed.

A character History box contains a list of the most recent 15 or so characters output by NuMorse. As new characters are added the oldest ones are discarded.

**Filter letters**

This option removes all characters except for the letters A to Z, and a to z, The lower case letters are the same as upper case in Morse code; ASCII lower case letters are accepted and then converted to upper case before code translation occurs.

**Filter numbers**

All of the characters except 0 to 9 are discarded by this filter.

## Filter punctuation

This option allows the following punctuation symbols to be translated to Morse code:

. , ? - : ; [ ] { }  
( ) " \$ \_

The various brace characters are translated to ( and ).

## **Filter Prosigns**

If the display as prosign option in the Prosign Mapping Dialogue Box is checked for this character it will be displayed as a character combination according to CCITT recommendations, otherwise it will be displayed as the ANSI equivalent. This filter option causes only the characters currently designated as prosigns to be allowed through the character filter software.

## Filter Custom

This option allows you to use a filter designed by you to vary the content of code output by NuMorse, and is an alternative to the preset filters available in the [Filters dialogue](#) box. The custom filter can contain as few or as many characters as you wish. To edit the filter select a character entry in the custom filter control, then drag the thumbwheel on the filter adjust control to the required setting. Filter values can be any multiple of 5 between 0 and 100. For other ways of setting filter values see the [Filters dialogue](#) topic.

If code is being read from an external file with the custom filter active then the filter acts like an on-off switch. If a particular character has a filter setting above zero then that character will always be allowed, otherwise it will never be allowed.

When characters are being produced by the random number generator the relative filter values will determine how often each character occurs. For example if the letter A is set to 100 and the letter S is set to 50 then over a sufficiently long session A will be output about twice as many times as S.

By default the custom filter and standard filter is set to produce characters at about the same relative frequencies as standard English text. Here is a list of the relative letter frequencies in a piece of English text containing about 40000 words, the values are relative to letter E the most frequently occurring letter:

A	64
B	16
C	36
D	34
E	100
F	11
G	24
H	18
I	71
J	1
K	7
L	45
M	23
N	60
O	51
P	25
Q	1
R	61
S	68
T	59
U	29
V	9
W	7
X	2
Y	14
Z	1

This list was used to provide the default custom filter and standard filter values used by NuMorse and is present in coded form in [NUMORSE.INI](#). I decided to set all the valid characters not on the list to a value of 5; this means that numbers and punctuation will occasionally be transmitted.

The custom filter settings will be saved to disk when you save the [program settings](#). Selecting the filter custom option will automatically deselect any other filter options. The standard filter is not written to by NuMorse, treat it as a default filter to reload in at any time. It can be altered by editing the entry in NUMORSE.INI if required.



## **Filter Preset**

This button causes the current preset filter options to be loaded into the custom filter edit control. If the Filter custom option is active when the preset button is pressed then the custom edit control is cleared. The characters loaded are the groups checked just before the button is pressed. So to preset the filter for numbers, for example, check the numbers button first then press the preset button.

This feature can save typing effort. For example, to specify a group of 15 alphabetic characters:

1. Select Filter letters, unselect any other filter preset options.
2. Click the preset button to load the entire alphabet into the custom edit control, they will all be set to 100.
3. Edit the letters not required to a value of 0

### **Load Standard Custom Filter**

This button causes the standard custom filter settings to be loaded into the custom filter edit control. These settings correspond to the relative character frequencies for English text.

### **Filter custom edit control**

Select the characters that are to be output as Morse Code and use the custom filter adjust control to set to the required value by dragging on the thumbwheel. Another way of getting text into this control is by using the Filter preset or From Disk buttons.

Here is the default code spacing used by NuMorse

Pause between elements	1
Pause between words	7
Pause between characters	3
Dot length	1
Dash length	3

### **Drill Tries Allowed**

This parameter determines the programs response to an incorrect input. You can adjust the scroll bar to allow no second attempts. If no second attempts are allowed then you will given a warning that you entered an incorrect response and then the next character will be displayed.

At the other extreme of the scroll bar any number of attempts are allowed, you will still be warned if an incorrect input is detected, but the same character will then be repeated until you get it right.

At intermediate settings a certain number of guesses are allowed before proceeding to the next character.

## **Drill Response Time**

At the default setting of <Any> there is no time limit when attempting to guess the Morse code character. Other settings cause NuMorse to wait for a certain interval, if no character is entered by the end of the interval it is treated as incorrect input. What happens next depends on the other settings in the Drill Settings dialogue box.

## **Drill Session time**

If this scroll bar is set to maximum then drill sessions will continue until the dialogue box is cancelled. Setting the scroll bar to any other position allows you to practise against the clock. After the specified time interval a special audible signal will indicate end of session, and a message will appear. Clicking on the message OK box will then take you back to the main menu. Consult the statistics dialogue box to see how you did.

This feature combined with the other drill setting options gives a very large range of possible test situations. For technical reasons (connected with the audible and visual warnings) the session time shown in the statistics box after the drill session may be a second or two longer than the original time specified.

If you click the Stop button during a timed Session then restarting by clicking on the Play button will have the effect of restarting the session timing clock from zero.

The setting of the session time scroll bar also has an effect on the clock display shown in drill and play modes. If the scroll bar is set to maximum, that is no specified time interval, then a clock is shown. If a time limit has been set then a count down timer will be displayed instead.

### **Drill Visual Warning**

By default this option is checked. When a wrong character is input a symbol is shown which resembles those crosses your primary grade teacher scribbled across your first attempts at math!



**Drill Audible Warning**

If this option is checked then NuMorse responds to wrong input by outputting a Hu-Hu sound instead of the next Morse code character. Unless Speech output is activated, in which case the correct character will be spoken.

**Drill show correct character**

If the wrong guess is made for a character one of the program responses is to briefly show the correct character. This option turns the feature off.

## Timing Dot Length

The dot length controls the fundamental speed of the Morse code generated by NuMorse. To set up code at the required speed, with standard spacing, adjust the dot length scroll bar until the WPM display reads the correct code speed. Then press the Setup button; this has the effect of adjusting all of the other Timing scroll bars to give code at the displayed speed with correct character and word spacing.

**Timing Frequency**

This scroll bar allows you to adjust the pitch of the generated tone. This tone is generated using standard Windows 3.0 sound generation procedures if the PC speaker option is selected in the device select dialogue box. The new Windows 3.1 MCI calls are supported by selecting Soundboard as the output device, however the method of selecting the frequency remains the same.

## Timing Dash Length

You may wish to set the dash length longer than standard spacing in the early stages of learning Morse code to make the characters easier to decode.

### **Timing Character Spacing**

This option allows you to set the character spacing to a value greater than the default of 3 dot lengths. This can be useful when you are trying to increase your code speed. Set dot and dash length to the higher speed, but increase character spacing to longer than 3 dots in duration until you are comfortable with the new speed.

## Timing Word Spacing

You might like to increase word spacing above the default of 7 dot lengths for the same reasons as increasing character spacing.

**MCI calls.**

This is a feature introduced in Microsoft™ Windows 3.1 which allows control of multimedia files and hardware devices, including sound generation hardware. MCI calls are supported in this version of NuMorse.



## Play Settings

This dialogue box controls the appearance of the Play Dialogue during Morse code play sessions. There are 4 checkboxes:

1. Show current character. The current character is the large one that appears as each piece of code is sent.
2. Show Morse character. The Morse character is the code displayed in visual form when the code is sent aurally.
3. Display keypress. When playing you will notice that the simulated keyboard keys press as the code is output. This checkbox can be used to disable the keypresses.
4. Display sent characters. These characters are the ones that are displayed as a horizontal list.

You might decide to turn some or all of these feature off to avoid getting visual clues when trying to decode Morse using Play Mode.

## Speed Settings

The 2 vertical slider controls in the Play/Drill dialogue box allows changes to code and text speed as an alternative to the method using dot length in the Timing Dialogue box. The code speed based on dot length is called character speed. The speed based on the time to send a block of text is called text speed and can be set lower than character speed by lengthening word and character spacing. Code with greater than normal character and word pauses is often called Farnsworth code and is a useful technique for achieving higher code speeds. In NuMorse the character and text speed can be set independently; although text speed can never be greater than character speed.

**Save File As**

This allows you to save a file in a file edit window using an alternative filename to RESPONSE.TXT or SENT.TXT. You will need to do this to avoid losing information when the Drill or Play dialogue boxes are re-opened since this action causes these files to be erased and restarted.

## **Open File**

This allows you to open a file other than RESPONSE.TXT or SENT.TXT and enhances the usefulness of the file edit windows by allowing viewing and editing of any text file.

## NUMORSE.INI

NUMORSE.INI is an ASCII file in \WINDOWS that defines certain NuMorse default values. There is a number of important program parameters in NuMorse that can only be altered by editing this file. The reason for this is that I have received so many suggestions regarding options and preferences that to control them all from within the program would have resulted in a confusing array of dialogue boxes each with many buttons and checkboxes. My solution was to put some of the more obscure parameters into NUMORSE.INI where they can be altered by direct editing of the file. No doubt some users will disagree with my decisions about which items should be controlled directly from NuMorse!

If a control exists within NuMorse for a particular program setting then there is no need to edit NUMORSE.INI at all. On program exit there is an option to automatically save such settings to NUMORSE.INI.

If you edit NUMORSE.INI then make sure that the text editor used is a pure ASCII text editor such as the review file editor that is part of NuMorse.

Here is list of NUMORSE.INI file contents as copied directly from the file, since the values depend on program settings you may find differences between the list here and your own NUMORSE.INI file.

```
[Morse]
SourceFlags=16
FilterFlags=1
Dot=25
Freq=1000
Pause=75
WordPause=175
Dash=75
Retries=1
TimeOut=9
DrillFlags=127
PlaySettings=15
ReviewFiles=20
Restart=1
PreferencesFlags=30
DisplayCalibration=100
SessionTime=610
SourceFileName=g:\morse2\paris.txt
SettingsFileName=g:\bp\doc\*.mrs
Seed=0
SpeedUnits=WPM
CustomFilter=ABABAABBBBBBBBBBBBBBBBBBBBBBABABAMDHHUCFEBCJFMKFBMOMGCCB
DBAAAAA
StandardFilter=ABABAABBBBBBBBBBBBBBBBBBBBBBABABAMDHHUCFEBCJFMKFBMOMGCCB
DBAAAAA
SpeedModeFlags=2
DeltaTime=10
DeltaChars=10
Port=888
ToneOn=255
ToneOff=0
SpeedCalibration=100
SelectDevice=4
Volume=50
```

```
Attack=100
Decay=100
RunPath1=
ItemName1=
RunPath2=g:\morse2\alarm.wav
ItemName2=Wav player
Waveform=17
PauseCalibration=100
WavPath=G:\MORSE\
ProsignMap=*+/=
WordWrap=1
QSO=6
CheckCharacters=*+.,/0123456789=?ABCDEFGHIJKLMNPOQRSTUVWXYZ
QSOType=1
TEST=4
Topics=NUMORSE.TOP
Phrases=NUMORSE.PHR
```

If you need to restore your NUMORSE.INI file to the above settings then here is a simple way to do it:

- 1 Bring up this help screen.
- 2 Use Help Copy to copy the above list to the Windows clipboard.
- 3 Close help and select a review file editor from the NuMorse File menu.
- 4 When the Review Edit window is displayed select File New to clear the existing file.
- 5 Use Paste to send the clipboard contents to the Review edit window. You should see the above list and now is the time to make any changes if required.
- 6 Now use File Save As to replace your own NUMORSE.INI file. You will need to specify the file as \WINDOWS\NUMORSE.INI

As an alternative to the above you could copy and paste selected parts of the above list to your own NUMORSE.INI file.

Here is a line by line explanation of the file

**SourceFlags.** This line controls the default source for generated code. Valid values are 1,2,4,8,16. You should not need to edit this line, the default source can be set from within NuMorse and source settings can also be saved and loaded.

**FilterFlags.** Controls the default filter checkbox settings. Valid values are 1 to 31 which can be set from within NuMorse so there should be no need to directly edit this line. If you wish to work out the correct value then add together the following for each of the checkboxes to be checked.  
custom=1,prosigns=2,punctuation=4,numbers=8,letters=16

**Dot.** Default Morse Code dot length, normally 80, settable from within NuMorse

**Freq.** Default Frequency, normally 1000Hz, settable from NuMorse

**Pause.** Default character pause length, normally 240, settable from NuMorse

**WordPause.** Default word pause, normally 560 units, settable from within NuMorse

**Dash.** Default dash length, normally 240 units, settable from within NuMorse

**Retries.** Used in drill settings. Normally 1, meaning no retries, settable from NuMorse

**TimeOut.** A drill settings option. If set to 9 seconds or greater there is no TimeOut.

**DrillFlags.** Controls the default drill settings checkboxes. Valid values are 1 to 15 which can be set from within NuMorse so there should be no need to directly edit this line. If you wish to work out the correct value then add together the following for each of the checkboxes to be checked. Show correct character=1, Audible warnings=2, visible warnings=4 Display Morse character =8

**PlaySettings.** Controls the default Play settings checkboxes. Valid values are 1 to 15 which can be set from within NuMorse so there should be no need to directly edit this line. If you wish to work out the correct value then add together the following for each of the checkboxes to be checked. Show current character 8 Show Morse character 2 Display keypress 1 Display sent characters 4

**ReviewFiles.** This value is normally 20. It specifies the maximum size, in Kbytes, of a review file. If this maximum size is exceeded then NuMorse will display a warning before closing the Drill or Play dialogue box. This is needed because if NuMorse were left running in Play Mode for several hours the Send Review file would become very large. There is no control within NuMorse to alter this setting, it must be edited directly.

**Restart.** Acceptable values in this optional line are 0 or 1. If the line exists and is set to 1 then the review file warning will not be displayed, instead it will be deleted and restarted. There is no control within NuMorse to alter this setting, it must be edited directly.

**PreferencesFlags.** Controls the default drill settings checkboxes. Valid values are 1 to 31 which can be set from within NuMorse so there should be no need to directly edit this line. If you wish to work out the correct value then add together the following for each of the checkboxes to be checked. FCCCharacter counting=1 Reset source file=2, Save to Init=16 (in exit dialogue.)

**Display Calibration.** Speed calibration factor, although normally set to 100 it can be adjusted over a wide range, settable from NuMorse.

**SessionTime.** Drill settings default session time out value. If set to 610 seconds then no time limit is imposed, settable from NuMorse.

**SourceFileName.** Specify a default source file here if required, settable from NuMorse. This line is optional

**SettingsFileName.** Specify a default settings file here if required, this line is settable from NuMorse and is optional. Normally this line can be omitted or left blank.

**Seed.** Random generator seed. If this is set to a non zero value then the random number generator will output the same stream of numbers each time it starts. Set to 0 if a more random character stream is required that produces a different series of numbers each time it is started. This line MUST be present in NUMORSE.INI and must be directly edited if you need to alter it. The value 255 is reserved do not use.

**SpeedUnits.** This line is editable from the calibrate timing dialogue. It specifies the text following the various speed displays in NuMorse. Normally it is set to the string W.P.M., but can be altered to any other string up to a maximum of 6 characters.

**CustomFilter.** This line represents the default values in the custom filter, and is settable from within NuMorse via the Custom filter control in the Filters dialogue box. If you edit it directly then make sure that the resulting line contains exactly the correct number of characters. The first character

represents <!> ASCII code 33 decimal while the last character in the line represents <\_> the underscore character which has ASCII code 95 decimal. The letters <A> to <U> represent filter values from 0 to 100 in steps of 5. If you need to restore the original custom filter settings then cut and paste from this help file as explained above.

**StandardFilter.** This line contains custom filter settings which represent the frequency of characters in English text, a control in the Filters dialogue loads the values from this line into the custom filter. It is a quick and easy way of setting the custom filter so that NuMorse outputs characters at the same relative frequency as English text. This line is not editable from within the program so if you decide to edit it for languages other than English then please note the comments as for the CustomFilter line.

**SpeedModeFlags.** Controls the default checkbox settings in the Timing dialogue box. Valid values are 1,2 and 4 which can be set from within NuMorse so there should be no need to directly edit this line.

**DeltaTime.** This line can be edited from within the Timing dialogue box. It controls how fast the code speeds up during Play and Drill sessions when the Increase speed radio button in the Timing Dialogue box is checked. If set to 20 then the speed will increase every 20 seconds.

**DeltaChars.** This line can be edited from within the Timing dialogue box. It determines the responsiveness of NuMorse to user errors when in Drill mode and the Adapt speed radio button in the Timing Dialogue box is checked. If set to 10 then the speed will increase/decrease with every 10 correct/incorrect guesses when in Drill mode. If in Play Mode the speed will increase for every 10 characters sent.

**Port.** This line and the two following does not need to be edited since appropriate edit controls exist within NuMorse. This line is where the hardware port address is stored, the value is in decimal. For more information consult the hardware help topic.

**ToneOn.** This line is a decimal representation of the bit state of the selected port during code tone on periods. For more information consult the hardware help topic.

**ToneOff.** This line is a decimal representation of the bit state of the selected port during quiescent or silent periods. For more information consult the hardware help topic.

**SpeedCalibration** This line affects the speed for a given displayed dot duration, it is normally set to 100. This line is alterable from the Calibration dialogue box.

**SelectDevice.** This is where the currently selected output device is stored. Permitted values are 1 2 and 4 representing PC Speaker, Hardware Port and Sound Board. This value is adjustable from the Select Device dialogue box.

**Volume** If a sound board (or sound board emulation) is fitted then this value is the output volume level as set in the Sound Board options dialogue.

**Attack.** If a sound board is fitted then this line determines the shape of the leading edge of the tone envelope. A value of 100 indicates a square envelope. This is settable from the Sound Board options dialogue.

**Decay.** If a sound board is fitted then this line determines the shape of the trailing edge of the tone envelope. A value of 100 indicates a square envelope. This is settable from the Sound Board options dialogue.

**Waveform.** If a sound board is fitted then this line determines the shape of the tone waveform and also whether the key click filter is active or not. It is settable from the Sound Board options dialogue.



**RunPath1,RunPath2.** These are paths to executable files which are accessed from the main menu RUN option when you click on the user defined items. Settable from the Paths item in the Run menu option.

**ItemName1,ItemName2.** These are names which you give to the user defined items in the Run main menu option. Settable from the Paths item in the Run menu option.

**PauseCalibration.** Code text speed setting can be altered by changing this value from its default of 100, used when code speed is different from text speed. You should not need to alter this value but if you do then it must be edited directly.

**ProsignMap.** This defines which characters are displayed as prosigns and is settable from the Prosigns dialogue box.

**WavPath.** By default all sound files used by NuMorse are in the same directory as the executable files (MORSE.EXE) if you wish you can copy them to another directory and specify the sound file directory path on this line.

**WordWrap** If set to 1 then automatic word wrap is active in the file windows. Set it to 0 to disable automatic word wrap.

**QSO** This is a set of flags representing various options in the Generate/QSO dialogue box.

**CheckCharacters** The characters on this line are the ones that are required in FCC tests and are present in QSOs and tests generated by NuMorse. They can be set from the Generate/QSO dialogue box.

**QSOType** Defines the length of QSOs generated by NuMorse and is settable from the Generate/QSO dialogue box.

**TEST** Defines the type of test produced by NuMorse.

**Topics** This is the name of the Topics script file used by NuMorse

**Phrases** This is the name of the Phrases script file used by NuMorse.

An ASCII text editor embeds no hidden control characters in the text file. Windows Notepad is an example. The [file edit windows](#) supported by NuMorse are also suitable ASCII text editors.

## **Adaptive Speed changes**

NuMorse is capable of automatically altering code speed during a play or drill session. There are 3 ways in which this can be achieved.

1. Using embedded control codes when the code source is an external file.
2. Checking the Adapt speed checkbox in the Timing dialogue box. The effect of Adaptive speed in Play Mode is to speed up the code every few characters sent, this is different from Increase speed mode when the speed increase occurs after specific time intervals. In Drill sessions NuMorse will continuously adjust the code speed depending on the users performance. An entry in NUMORSE.INI called DeltaChars determines how sensitive NuMorse is to errors when adapting code speed. A series of correct guesses will cause the speed to increase a little, while a few bad guesses will cause the code speed to decrease. At the end of a session the new speed will be retained by the system and can be saved as a program setting for future use.
3. Checking the Increase speed checkbox in the Timing dialogue box causes the speed in both drill and Play Modes to increase regardless of user input, the rate of increase is determined by an entry in NUMORSE.INI called DeltaTime. The final speed will be retained by the system and can be saved as a program setting for future use.

## File windows

File windows can be used to edit text files just like a word processor. The NuMorse file windows are used to display and edit special files generated during normal operation and also to edit code source files for playing as code. They are also suitable for editing the NuMorse start-up file that is in your Windows directory with the filename NUMORSE.INI.

The file windows are not intended to be full word processors, they are limited in function as well as maximum file size, which is about 24K. You will be warned if the maximum file size is exceeded. The file windows in NuMorse support automatic word wrap, and any unformatted text files are word wrapped as they are loaded, this means that fairly large files will take several seconds to load. Several features of NuMorse use the concept of a topmost, or active, window. This can be identified by the coloured title bar, the other ones being grey; a file window can be made topmost by clicking on its title bar.

NuMorse always stores its output and any user responses. Output is sent to a file called SENT.TXT and user keyboard responses are sent to a file called RESPONSE.TXT. Two special purpose file edit windows are provided for viewing and editing these review files.

New log files are started whenever the Play or Drill dialogue windows are opened, so you need to make sure that any information to be saved for future use is saved as a different filename before opening the Drill or Play dialogue boxes. The file edit windows provide a File Save as function for this purpose.

A small file header is always sent to each log file at the start of a Play or Drill session giving the time and date of the session In Play Mode the program sends code without pausing to check user responses, however any responses **are** sent to RESPONSE.TXT.

Each log file edit window opens with the SENT.TXT and RESPONSE.TXT files opened. This allows you to open both windows, and size or move them so that your responses can be compared with NuMorse output.

The editors contain all of the usual windows style text editing and file manipulation features, as well as the ability to print and cut and paste to other applications and help systems. Here are a few ideas and suggestions:

- Cut and paste from one file window to the other to put code sent and response information into the same file, then save this file under a new file name.

- Copy parts of the file, via windows clipboard, to a word processor.

- Copy text from the help system into a text file.

- Type in code source files and save them for future use.

## The random number generators

There are several random number generators used in NuMorse.

- 1 Uniform pseudo random1, Numbers within the range 32 to 95 are generated which are then filtered and converted to characters. The numbers are uniformly distributed, this means that any number in the specified range has the same chance of being generated as any other number. This generator is used by the Random stream and Random groups features when the custom filter is NOT active.
- 2 Uniform pseudo random2. Numbers within the range 1 to filesize(code source file) are generated which are then converted to file positions and adjusted to read in whole words from a code source file. This generator is used by the File random feature regardless of the custom filter setting.
- 3 Weighted pseudo random. Numbers within the range 32 to 95 are generated which are then converted to characters. The numbers are weighted so that certain characters will appear more often than others, the weighting depends on the custom filter values. This generator is used by the Random stream and Random groups features when the custom filter is active. When reading characters from an external file the weighting acts more like a switch, none zero filter values allow the character to be output by NuMorse.

A pseudo random generator is based on a mathematical trick which allows a sequence of numbers to be generated given a starting number called a SEED. In the case of NuMorse the seed number is stored in NUMORSE.INI and can be edited. The sequence is not truly random since, for a given seed, the same sequence will be generated each time. If you change the seed number to 0 then the generator will behave more like a true random number generator, because a different seed number will be obtained by the program each time the generator starts. For your information the seed number is obtained from a memory location associated with the system timer which is constantly changing.

Weighting of random numbers is achieved by a technique called cumulative probability distribution sampling. An analogy is a group of targets some of which are bigger than others. Throwing a ball at random causes the bigger targets to be hit more than the others, thus achieving weighting.

## Sending code to a port

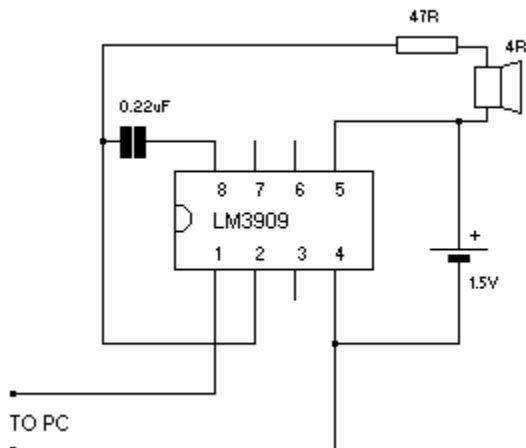
If you wish to attach a code practice oscillator to your PC then you will need to define a hardware port address and also what electrical conditions are to appear on the port when code is sent. This dialogue box is where you do this.

A representation of the logical levels of bits 0-7 of the selected port is shown together with the corresponding value (in decimal) that NuMorse will send to the port during tone on and tone off periods. You may adjust the values by editing the decimal value or you may toggle the logical levels by clicking on the bit map. Be aware that the bit map shown refers to logical bit numbers NOT physical pinout numbers, which will almost certainly be different.

It must be pointed out that an application which directly accesses PC hardware breaks all the rules of Windows programming, but it seems to work OK in this case! To use the feature you will need to identify, from your system hardware documentation, which are valid ports. Most systems will have the printer data port at 888 decimal available and so you could set the Port address value to 888 to use this. Using the printer output to drive an oscillator means that printed output from the program (the log file windows) will not be available unless you add a printer switch box to the system.

A figure of 255 for ToneOn would cause all the port bits to be set to HIGH during the time that a dot or dash were being sent. A figure of 0 for ToneOff in would set all the port bits to LOW during spaces. Setting ToneOff to 254 would result in the port bits all staying HIGH except for bit 0. The permanently HIGH bits could, with care, be used to power an external device such as a code practice oscillator. Check your documentation for maximum current drain allowed from your PC printer port, exceeding this value could result in damage to your machine.

The only port tested by the Author is the printer data port, although there is no reason why any port that physically appears to the outside world cannot be used. There is no software checking on port addressing or validity so make sure you know what you are doing before attempting to set up NuMorse to send code to a port. Here is the tone oscillator circuit that was used for testing on the authors system:



## Control codes in code source text

NuMorse recognises <linefeed> <@> in a code source file as a control sequence and interprets the following text as a command instead of characters to be output as code:

In practice the above sequence is a <@> character at the start of a new line (you dont type the < and the > symbols). Four commands are currently supported, heres how they work.

<@><NN>,<MM> where NN is a number between 10 and 80. This causes NuMorse code speed to alter to the new character speed of NN W.P.M. (or whatever speed units you have set NuMorse to work with). The optional comma and second number <MM> determines a new text speed. Here is an example:

@12,5

This line will cause the new character speed to be 12 WPM and the new text speed 5 WPM

<@><string>, where string is a line of text. <string> will appear on screen as a message up to 80 characters in length. The user will have to click on an OK button to continue. This feature could be used to warn the user of a coded speed change.

<@><space>, this command causes the rest of the line to be ignored.

<@><ABCDE.WAV>, this command will cause NuMorse to look for a sound file called ABCDE.WAV and if it is found it will be played through the sound card. The filename extension must be .WAV for this feature to work.

In the above cases any further text on the same line will be ignored.

The embedded codes feature will only work when code is being read from an external file that is being scanned in a serial manner. It will not work with files being scanned in random mode.

A demonstration file called EMBEDDED.TXT should be present in your NuMorse directory.

**Code sent button**

This button causes some of the statistics gathered during the most recent Play or Drill session to be loaded into the Custom Array. The values will be adjusted so that the maximum is 100, and all the other values will be relative to this and rounded to the nearest multiple of 5. The statistics used are the numbers of each character sent by NuMorse.



## **Responses button**

This button causes some of the statistics gathered during the most recent Drill session to be loaded into the Custom Array. The values will be adjusted so that the maximum is 100, and all the other values will be relative to this and rounded to the nearest multiple of 5. The statistics used are the numbers of each incorrect response made by the user. This feature is particularly useful for people wishing to improve their performance on less well learned code characters, since NuMorse will tend to use more of these after this procedure. Remember that the custom filter contents can be saved and used in future sessions.

**Invert custom filter button**

This button affects all of the custom filter values. It subtracts each value from 100, this means that the lower filter values will become the higher ones and vice-versa.

## Program History

1992

First version upload. Program based on a single dialogue box.

---

1992-1993

Development of menu based program using feedback and suggestions from users of first version.

Many features added and various beta versions uploaded during this period.

Final beta 0.95 upload 1 April 1993

---

Enhancements for version 1.0, CIS upload Autumn 1993

A number of bugs squashed and features added. Main ones only are listed here.

Options to suppress various on screen items during play and drill modes.

Log files, for program output as well as user responses.

Integral log file editor allowing cut and paste to Windows clipboard as well as printing and editing of files.

Variable weighting in custom filter

Custom filter weighting from  
    user direct  
    recent user responses  
    recent program output

Auto increase speed during play and drill sessions

Adaptive speed: faster if no errors  
                  slow down when error

Sample code source files included

Added ability to optionally save to .INI file instead of settings file.

Added a speed control to main menu.

Improvement to random character generator, alpha were appearing twice as often as other chars, due to a spurious lower to upper case conversion

Custom filter invert button in filters dialogue

Feature to output code to a hardware port added  
if hardware port active then PC speaker is off

Embedded control codes in source file, change speed and display messages from code source text

---

Enhancements for version 1.01, CIS upload December 1993

From this version onwards Windows™ 3.1 is required.

Help file now in Win 3.1 format, and some Win 3.1 API calls are now used.

Some major timing problems when using an external code oscillator now fixed.

MORSEBTN.DLL is no longer required, users can remove it from their systems if it was installed for previous versions of NUMORSE.

Improvement to appearance of some screens in EGA modes.

On-line registration begins

---

Enhancements for version 1.20, CIS upload February 1994

SoundBlaster™ support added, at last! With this comes a new dialogue box containing a volume control as well as envelope and waveform controls.

Several items previously only alterable by editing NUMORSE.INI can now be adjusted from within the program. These include:

- Port addressing and values, for driving an external code oscillator
- Parameters connected with automatic speed changes in certain modes.

Timing rationalised throughout program, dot length is now always displayed as milliseconds.

In previous versions a crash could be provoked by repeatedly pressing the PLAY button in the Play dialogue box whilst code was being output. This and some other minor bugs fixed.

Registered users can obtain a software key which will suppress the registration fee request in the opening screen, from this version (1.20). Cosmetic changes have been made to the About box; if the program is not registered then this fact is emphasised.

A hint line added at the bottom of the screen.

The program will issue a warning and halt if run under Windows™ 3.0, Version 3.1 is required.

Code element length is now in milliseconds throughout program.

Prosigns are displayed correctly in various lists.

---

Enhancements for version 1.30, CIS upload early 1995

Some new waveforms available to users with sound cards

Character and text speed settable from the speed dialogue box

Maximum character speed available is now 48 WPM (Registered version only)

Minimum character speed 5 WPM, minimum text speed 3.2 WPM

Source file embedded code feature extended:

Character and text speed is settable within source files.

Sound file names embedded in source files will be played, mix speech with Morse!

A visual semaphore option added.

Beginners can set up drill mode so that the required character is displayed on screen.

File window operation now conforms to the Microsoft MDI style. File windows can be sized, tiled, cascaded etc.

A key click filter provided.

Options to launch other applications from the main menu.

A main menu toolbar added.

Clock/timer display added.

Response time thermometer Gizmo added.

Prosign support extended to cover most prosigns.

Speech output option added to drill mode.

Printer setup option added.

QSO generator added.

Test generator added.

Further bug fixes in sound board drivers.

MDI operation, file windows can be cascaded/tiled on NuMorse desktop

Farnsworth code feature added.

User registration screen added.

---

Enhancements for version 1.40, CIS upload May 1996

This was a minor maintenance release to fix the soundboard problems which occurred on some Windows 95 machines. Some information on NuTest was added to the help system

## A word from the Author



Hi, if you have got this far into the help system you are obviously interested in the program and would perhaps like to know a little more about me. I live in a village in Staffordshire, Central England about 2 hours drive from London. Apart from computing my other interests include cycling, SCUBA diving, gardening, chess and Mensa meetings.

Compared to the products of Borland™ and Microsoft™, NuMorse may seem rather trivial. But it represents a lot of effort on the part of one individual, rather than a small project by some programming team.

I would like to thank the following:

The folk on CompuServe Borland™ Programming forum A. Without TeamB and the other gurus on BProgA the project might have foundered many times over.

My wife Shirley for not taking an axe to the PC.

Mike Guest (G0VYN), for his enthusiasm, valuable comments and patience during program testing.

Donald B. Bell (KI5YT) for his professional approach to program testing and his dry humour.

You people who took the trouble to E-mail me with comments and suggestions about earlier versions of the program.

Keep up the constructive comments. Many of the current features are a result of suggestions from users.

Important note:

You **must** purchase a license to use NuMorse by Registering your copy if you find it useful.

Thanks, Tony Lacy, G4AUD, CIS 100030,157

## Select device

This is the dialogue that allows you to select the device used for playing Morse Code. You can select any option even if the associated device is not present. If you attempt to send to a sound board and one is not fitted there will be no error message, the result will simply be...silence!

If you select the PC speaker then Morse Code is emitted from the small speaker usually fitted to PC's. NuMorse contains it's own built-in PC speaker driver and so a third party speaker driver should NOT be installed. If such a driver is fitted then NuMorse will not be able to send code directly to the PC speaker. If a speaker driver that emulates a sound board is installed then select the SoundBoard option and NuMorse will send to the emulation driver which will then operate the speaker. Although this method works don't expect too much from a sound board emulation speaker driver.

Selecting Hardware Port will direct the output of NuMorse to an external port which will allow operation of an external oscillator. There is a dialogue box that allows you to set up the hardware port to drive the oscillator.

Selecting Flashing Window will activate an extra semaphore window when the Play or Drill dialogues are open.

## **Flashing window**

If Flashing Window is selected as the output device Code output will be sent to this window which will flash on and off. This is intended as an aid to those wishing to learn semaphore and for people with hearing difficulties. There is a certain time overhead in updating the screen, especially in Windows programs, so you may find that the code timing is slightly disturbed. This extra semaphore window is resizeable, but if you make it fill a large portion of the screen then the on/off update time becomes fairly obvious, so try to keep it as small as possible or try minimising it.



## Sound Board Options Dialogue

This dialogue box controls how code will sound when played via a Sound Board. The software was developed on a 100% SoundBlaster™ compatible sound board.

Here is a description of the controls in this dialogue box:

Volume is controlled by a slider bar. I suggest that you set this at 50 and then set the physical volume control on your sound board to a suitable level. If this slider is set at high levels (more than 50) then the envelope controls may not work correctly.

Leading edge and trailing edge controls affect the waveform envelope. These controls (like many others in NuMorse!) are not strictly necessary for the learning of code. They are provided for interest and experimentation, in the spirit of Amateur Radio. The author had much fun listening to the weird code sounds that NuMorse can produce by varying the envelope from the nominal levels of 100 for each control. A test button is provided so that you don't need to go to the Play dialogue to hear the results of adjusting these controls. If the volume control is set to a high level then you may find that the envelope shape is 'clipped' under certain extreme settings of the envelope controls.

A click filter is provided; the time constant associated with this filter varies according to the code transmission speed setting.

The 4 radio button controls give square wave, sine wave, sawtooth and rectified sine wave output; the difference is interesting...

## The .WAV files

There is a number of .WAV files shipped with this version of NuMorse. By default they are placed in a subdirectory of the NuMorse working directory called WAV. If you wish to copy them to a different directory then be sure to set the WavPath entry in [NUMORSE.INI](#) to the new directory. These files are the source of various sounds produced during program operation when a sound board is installed and [selected](#), except for:

- Morse code, the sounds for generated code are produced 'on the fly' during code transmission.
- Sounds associated with message boxes are standard Windows™ system sounds, the \*.WAV files for these will be in your Windows™ directory.

Here are the supplied \*.WAV files and their functions, if you do not like the sounds I have provided (highly likely!) then feel free to substitute your own \*.WAV files.

INTRO.WAV is the source of the sound produced when the program starts up and during subsequent display of the About box.

GROAN.WAV is the source of the sound produced in [drill mode](#) when the user presses an incorrect key.

ALARM.WAV produces the sound at the end of a session time out if the [time out](#) option is enabled.

The speech output option in NuMorse Drill Mode requires an optional set of WAV files, one for each Morse Code character. They may be present on your install disk, in which case they will be transferred to the correct location during installation. If they are not present then there may be several sources for these files. Any sets of WAV files that I upload to CIS will be called NUCHARS.ZIP; or you may record your own using Windows Sound Recorder. The WAV filenames are all of the format:

CHARXX.WAV, where XX is a pair of numbers.

These will be used instead of GROAN.WAV if in drill mode and [Speech Output](#) has been selected in the [drill options](#) dialogue box. Each of these files contains a spoken representation of a character, in phonetic form in the case of alphabetic characters. For example CHAR65.WAV will contain the phonetic Alpha, since 65 is the ANSI code for A. If you prefer (for example your language may not be English) then you may substitute your own files. To make the WAV files available to NuMorse all you need to do is to place them into your WAV directory if they haven't already been put there by the MINSTALL program.

The WAV files provided were all recorded as 8 bit mono files at a fairly low sampling rate, and so will be nowhere near hi-fi quality. One reason for this was the extra file space that would be required for, say, 16 bit WAV files. The [problems](#) help topic contains information on problems you may encounter when getting the speech output option to work.

If you have a sound board and a suitable microphone then you can produce your own .WAV files to play through NuMorse by using the Windows Sound Recorder program. Place the WAV files you produce in the same directory as the other NuMorse files and use the file naming procedure described above.

Some other miscellaneous .WAV files may be present as samples to demonstrate how speech and code can be mixed together in code source files. See the [Control Codes](#) help topic for more details on how to mix speech and code in source files.

## Problems

Here are some of the queries I have received, mainly via CIS E-mail.

**Q** NuMorse starts up and appears to work fine, except that no sound emerges from the PC speaker.

**A** The most common reason for this is that you have a third party PC speaker driver installed, which prevents NuMorse from accessing the speaker. There are various solutions, apart from de-installing the driver. One solution (offered by a user) is to open the sound section of Windows control panel and select Sound (not drivers) and turn the sound off by unchecking the sounds box. When you have finished with NuMorse simply turn the sound on again.

If your PC speaker driver is a full emulation of the SoundBlaster™ sound board then try selecting SoundBoard for your output device. If this works then you will have the advantage of a volume control as well as envelope and waveform controls, but don't expect fantastic results from this method.

Check that you have the correct output device selected, NuMorse will happily send code to a hardware port or none-existent sound board if they are selected in error.

Check the code filter settings. If you try hard enough it is possible to set up the code character filter to filter out ALL of the characters.

Check the Code Source setting it may be that the code source is set to keyboard, and NuMorse is simply waiting for you to enter a keyboard character.

If you are using a sound board then check that the software volume control in the sound board options dialogue is turned up.

**Q** The code from NuMorse sounds dreadful, the timing is hopeless.

**A** This fooled the author for a time. The problem is to do with the way Windows shares out processor time between applications. NuMorse, in common with all Windows applications, politely hands control back to Windows during pauses in program execution, such as during code spaces. If a time hungry application grabs control then all other applications, including NuMorse, have to wait. Hence code timing suffers. A typical offender is File Manager which, if left maximised but hidden, can spend lots of time rewriting its (hidden) file lists. The solution is to close all the applications you don't need while running NuMorse. This is one reason why I do not emphasise the NuMorse capability for background operation, the foreground program can sometimes disturb the timing of the generated code.

**Q** I would like to be able to control the volume of the output from the PC speaker.

**A** See the section above regarding PC speaker drivers and selection of the Sound Board option.

**Q** When using the automatic speed feature I get occasional extra bleeps from the speaker.

**A** This was a bug and will not happen in this version!

**Q** When I perform certain operations the program crashes.

**A** Let me know about this!

**Q** Im using the Microsoft Speaker program to emulate a SoundBoard. The code sound quality is poor and speech output is almost unintelligible.

**A** Sorry, cant help much here. The main problem is that the CPU is trying to perform the functions of a sound card as well as run NuMorse and the rest of your system. The constant switching between these tasks is what is making the sound quality so poor.

**Q** Ive selected the speech output option in the Drill Settings dialogue box, but I cant get any speech.

**A** You need a SoundBlaster™ compatible SoundBoard to be installed and the output device set to Sound Card. A set of .WAV files in the WAV subdirectory is also required. These are called CHARXX.WAV where

XX is a pair of numbers representing the character code, and will normally be installed automatically from your install disk. If they are not present you may find them on CompuServe and other BBSs with the filename NUCHARS.ZIP, or you can produce your own using Windows Sound Recorder supplied with your Windows system. Disks distributed by me will contain a full set of these files and I will be pleased to send a set to registered users of NuMorse.

**Q** I get speech output OK. but there seem to be a lot of pauses and my hard drive access light seems to be flashing all the time during speech output.

**A** The speech output feature requires good system performance. This means that at least a 25Mhz 386, preferably DX, is required. And Windows needs to be working efficiently on your system, the use of Smartdrive™ or a similar disk caching utility is highly recommended. Although NuMorse will multitask successfully you might find that other running applications will upset the timing, so if you are experiencing timing problems try closing down all other applications.

## Shell to program.

You may enter details of up to two applications in this dialogue box. The item names will appear in the main menu and can be selected just like any other menu items. The path details tell NuMorse where the items are on your system.

To some extent the Shell-to feature duplicates the existing Windows Alt\Tab applications switching feature with one main difference; using the NuMorse shell feature will load an application from disk if it is not already in memory. The shell-to feature is aware of any file associations that may be set up, for example if you have set up an association for \*.TXT files to be edited by Windows Notepad then shelling to a .TXT file will result in Windows Notepad being launched and the \*.TXT file displayed ready for editing.

Click on the controls in this dialogue box for details.

Name for menu item1		<input type="text" value="Dos prompt"/>
Path to item 1		<input type="text" value="C:\WINDOWS\DOSPRMPT.PIF"/> <input type="button" value="B"/>
Parameters (optional)		<input type="text"/>
Default directory (optional)		<input type="text" value="C:\WINDOWS"/>
Name for menu item2		<input type="text" value="Calculator"/>
Path to item 2		<input type="text" value="C:\WINDOWS\CALC.EXE"/> <input type="button" value="B"/>
Parameters (optional)		<input type="text"/>
Default directory (optional)		<input type="text" value="C:\WINDOWS"/>
<input type="button" value="✓ OK"/>		<input type="button" value="✗ Cancel"/>
		<input type="button" value="? Help"/>

Type text here to describe the application that is to run. The text will appear as a menu item. If you wish to remove the menu item then delete this text.

This is a file browse button that allows you to choose the required file for the application from a standard file selection list.

The path (including filename) is typed in here. Alternatively, you may select from a file dialogue by pressing the adjacent file browse button.



Some applications accept command line parameters. If the application to run needs command line parameters then type them in here, otherwise leave this line blank.

Many applications load and save data from a default working directory. This is often the same directory as the executable file is stored in and this is automatically entered here when the application path is selected using the file select button. You may amend this line or leave it blank if the default directory is not important. The directory specified here is NOT necessarily the same as the path to the application as specified in the path details line.

Selecting this button will bring up a file dialogue. Choose the application that is to run and the full path will be entered in the path description. The working directory description will also be updated to the same directory as the application.

Changes to the data are stored and the dialogue box is closed.

Any changes are discarded and the dialogue box is closed.

The help system is invoked.

**NuMorse versions; click on any part of the version number to see the significance of the digits:**

**NuMorse V1.23x**

The major version digit; a complete rewrite of the program would result in a change in this number.



Minor version digit; this number changes when a number of significant enhancements are made to the program. It is usually worth obtaining an update if this digit changes.

This digit changes when a small change is made to the program. It is not usually worth obtaining a version update unless the change is particularly significant to you.

A letter appended to the end of the version number indicates that one or more small bug fixes have occurred in this release. Unless the bug is important to you it will not be worth updating on the basis of a change in this letter.

## Prosign mapping dialogue

Prosigns (PROcedural SIGNals) were introduced because the original Morse code used letters and numerals only. Punctuation characters were synthesised by combining together existing Morse characters without a character space and this concept was extended to codes that signify common procedures used during code transmission and reception. For many of the prosigns there is a one to one correspondence between the ANSI code used internally by the computer and each prosign character. The correspondence has been defined internationally by ISO/DIS 9636 and the mapping shown in this box is loosely based on this (but see below).

In the Prosign Mapping dialogue box you choose which characters are to be displayed as prosigns and which as standard ANSI characters. The FCC test requires knowledge of the following four prosigns:

Prosign	Equivalent ANSI character	Meaning
(AR)	+	End of message
(SK)	*	End of work
(BT)	=	Double dash
(DN)	/	Fraction bar

There are 2 prosigns supported in NuMorse that do not yet have ANSI equivalents; they are:

- (HH) error.
- (CT) Start, signal, otherwise known as (KA).

Until there is a recognised conversion for these symbols NuMorse will use the following ANSI representations:

Prosign	Represented by
(HH)	# the hash, or pound, symbol ANSI code 035d
(CT)	> the greater than symbol, ANSI code 062d

Two new keys (# and >) have been added to the Play Dialogue keyboard; they will become active if the mappings for (HH) and (CT) are selected. Any # and > symbols in a code source file will be correctly translated to prosign symbols and sounds.

The rest of the prosign mappings shown in the dialogue are for punctuation signs. Here are some of the prosigns that are NOT supported:

(SN)	message understood
(AL)	new paragraph.
(TTTTT)	break in
(SOS)	Emergency distress

I have decided not to add Esperanto, Japanese Hebrew or Russian characters to the NuMorse character set!

## **Clock/Timer display**

In drill and play modes a clock is displayed; the time shown is obtained from the system timer and is updated every second. If the drill mode screen is displayed then setting a time limit to the session (see the [Drill Settings](#) dialogue for details) will cause a timer to be displayed which will count down the time remaining for this session.

Click on the clock display to bring up an alter time dialogue box; changes made to the displayed time will alter your system clock settings.

## **Response time thermometer**

To relieve the boredom of those endless practice sessions Ive introduced a little gimmick that Ive called the response time thermometer. This device is active when in drill mode and the Response thermometer option is checked in the Drill Settings dialogue box.

In drill mode NuMorse waits for your keypresses in response to the Morse character that has just been played. If you respond fast enough the liquid in the thermometer begins to rise, and as it warms up it also changes colour. Fumble a little and the temperature will start to fall. See how high and for how long you can keep up the thermometer display!

## **FCC Character counting**

If FCC Character counting is enabled in the Preferences dialogue box then, in compliance with the FCC testing rules certain characters are counted as two characters. This will make a difference to the statistics dialogue box that keeps track of transmitted characters and your responses. The characters affected are numerals prosigns and punctuation signs.

## **Background operation**

Like most Windows programs NuMorse will run in the background, so you can listen to Morse Code while getting on with another task on your PC. I have not emphasised this feature before because there are some potential problems when using certain features of NuMorse while it is running as a background task. If you wish to try it then here is one way to do it:

Get NuMorse running and playing code.

1. Press ALT-Tab, in other words press the Alt key and the Tab key together.
2. While holding the Alt key down press and release the tab key until Windows displays either the name of the application you wish to run or Program manager.
3. You should now be either in Program Manager or in your chosen application, with Morse Code still being output. If in Program Manager then select and run your application in the normal way.
4. Use the Alt-Tab method to get back to NuMorse again when required.



Alternatively:

1. Get NuMorse running.
2. If NuMorse is enlarged to full screen then click the Restore button (top right corner).
3. Size NuMorse to a convenient size occupying part of your display.
4. Click on other applications or Program Manager icons to start/run other applications.

Click on the NuMorse Window to alter NuMorse settings or start/stop playing code.



## The NuMorse Desktop

The first thing you will see when you start up NuMorse is the Main Window. The number of menu items may look a little daunting at first. To get started, though, all you need to do is click on the  toolbar button. You will now see a window with a set of typewriter style keys. Select  Random from the Source list and then click on



. You should now be hearing Morse Code!

The best way to explore the many features of NuMorse is to try out each main menu item. Each one gives you a pull-down menu. Click on items in the pull-down menus. Dialogue boxes will appear. Click on the Help button in a dialogue box for a description of what the box does.

See also:

[Overview of NuMorse](#)

Windows Program Manager help system glossary gives full details of this item

## **Farnsworth code**

Farnsworth code is the name given to Morse code that has additional spacing between characters and words to provide an effective text rate that is lower than the sending rate of the individual characters. Practice with Farnsworth code is believed to be a good method of achieving higher code speeds. The idea is to learn the rhythm of a character rather than the sound of the individual code elements and this is easier at higher character speeds with exaggerated spacing between characters. As proficiency increases the character speed remains constant and the spacing is reduced to increase the text speed.

The preferred text and character speeds varies between different authorities. Donald Bell (KI5YT) obtained and forwarded an F.C.C. public notice of June 22, 1982 entitled SPECIFICATIONS USED BY FCC FOR AMATEUR RADIO MORSE CODE TEST TAPES. This specifies a 5 word per minute text speed and 13 words per minute character speed. The British RSGB 5 wpm code test uses a 12 wpm character speed and other groups use different combinations of text and character speeds. To cater for these requirements NuMorse allows the storage of 2 text/character speed combinations. Use the slider controls in the Farnsworth Dialogue box to set these speeds up as required. To access these speeds when playing code simply click on the Farnsworth control in the Play Dialogue. One click will give you the first stored speed pair, a second click will give you the second speed pair. To restore to normal Morse code move the text speed slider to its maximum position.

## **Who was Mr. Farnsworth?**

I am again indebted to Donald B. Bell (KI5YT) for obtaining the following information (via the ARRL) from a letter published in *QST* April 1988 written by Bill Fisher (W2OC).

The late Donald Russ Farnsworth was first licensed in the mid-1930s as W9SUV and later held the calls W6TTB and W0JYC. Together with Bart Bartlett (W6OWP) he developed a code course which maintained the text speed constant throughout the course with a gradual increase in text complexity.

## **De installing NuMorse**

To remove NuMorse from your system perform the following steps using File Manager:

- 1) Delete the NuMorse working directory and its subdirectories
- 2) If you do not intend to reinstall any version of NuMorse then delete the file NUMORSE.INI from your WINDOWS directory. If you intend to install a later version of NuMorse then you may leave this file in place.

## Purchasing a license to use NuMorse (Registering)

NuMorse is not free software. This copy has been entrusted to you so that you can decide if it suits your needs. If you find it useful then you **must** purchase the right to use it. There are several ways to do this as detailed below.

The unregistered version of NuMorse is converted to the full speed (48 WPM) registered version by means of a software key. This software key is only available to registered users.

Advantages of registering NuMorse:

- I will provide help on program usage via CompuServe. Or by mail if you prefer this.
- I will be more likely to continue enhancing and developing NuMorse.
- The registration fee request can be switched off in the registered version..
- The registered version of NuMorse can output code at up to 48 WPM
- You will feel better.

Please select your preferred method of payment:



Credit card



E-Mail



Post

Attention, users in Africa!

## Purchasing by credit card



If you have a valid Visa or MasterCard then contact [NorthStar Solutions](#) via any of the following methods:

Voice 1-800-699-6395(10:00 a.m.-10:00 p.m., Eastern Standard Time. Toll free U.S. only)

Voice 1-803-699-6395(10:00 a.m.-10:00 p.m., Eastern Standard Time.)

FAX 1-803-699-5465(Available 24 hours, international and business orders welcome.)

CIS E-Mail 71561,2751 Internet 71561.2751@compuserve.com

Please have the following information ready when you call:

The program and version number (NuMorse version 1.40)

Your mailing address

Your Visa or Mastercard number and expiry date.

Important note: [NorthStar Solutions](#) processes registrations only. Please contact the [Author](#) for product/technical support.

See also: [Purchasing methods](#), [Users in Africa](#)

## **Purchasing via E-mail**

Registration is just a few keystrokes away if you subscribe to CompuServe:

GO SWREG.

Locate the entry for NuMorse, the Registration ID is 1721.

When the registration details appear answer 'y' to register your copy of NuMorse. An amount of \$29 will automatically be debited from your CIS account.

See also [Purchasing methods](#)

## **Purchasing via Post**

You may register your copy of NuMorse by mailing a registration fee of \$29.00 or equivalent to the address shown below. Acceptable methods of payment are Postal Order, cash or Cheque (payable to A. Lacy). If you wish you can print out a [Registration form](#) to include with your payment.

Non U.S. residents may pay in local currency (the amount to be based on prevailing exchange rates), U.S. dollars or Pounds Sterling (preferred). Checks drawn on Non U.K. banks are perfectly acceptable. Just write your check in the usual way using your own currency, send it to me and I'll do the rest. Your order will ship within one week of receipt, but your check may require up to four weeks to clear your bank.

For UK residents the required amount is £20.00 until further notice.  
Please send payment to:

A. Lacy.  
58 Bilbrook Road.  
Codsall.  
Wolverhampton.  
WV8 1ER  
United Kingdom.

See also  
[Purchasing methods](#)  
[Registration form](#)  
[Users in Africa](#)



## NuMorse Registration form

This form is provided for your convenience if registering by post. You can print it out by selecting File\Print Topic.

NuMorse single copy: Quantity: \_\_\_\_\_ @ \$ 29.00 ea. = \_\_\_\_\_

If you are ordering multiple copies, please list the first and last names of each additional user so that the author can supply them with their security codes as well.

Name : \_\_\_\_\_ Date : \_\_\_\_\_

Company : \_\_\_\_\_

Address : \_\_\_\_\_

City, State, Zip : \_\_\_\_\_

Country : \_\_\_\_\_

Daytime Phone : \_\_\_\_\_ Evening : \_\_\_\_\_

Electronic Mail address : \_\_\_\_\_

How did you hear about NuMorse? : \_\_\_\_\_

Any comments : \_\_\_\_\_

\_\_\_\_\_

See also  
[Purchasing methods](#), [Users in Africa](#)

## **Attention South African users!**

In South Africa registrations can be done via Registered Shareware for Africa. Orders can be placed by e-mail using the following address:

gerhard@bixsa.core.org.za.

Orders can also be placed making use of the on-line shopping mall on BIXSA on-line service telephone number 341-0352. BIXSA is also accessible via Telnet and Internet (address: 196.11.62.201)

Payments can also be made directly into the following bank account:

Account holder :	Registered Shareware for Africa
Account type:	Transmission
Branch:	Sunnyside
Account number	90-4453-0905

Payments can be made at United, Allied, Volkskas or Trustbank. Please make payments over the counter so as to be able to supply your name and telephone number on the deposit slip. If cheques are deposited orders will be processed after the cheque has been cleared.

Cheques can also be sent to:  
Registered Shareware for Africa  
Las Palmas No. 66  
Troye str. 175  
Sunnyside  
Pretoria  
0002

Regrettably, Registered Shareware for Africa cannot accept payment by credit card.

The latest releases of all the shareware available for registration at Registered Shareware for Africa can be downloaded from BIXSA.

## NuTest for Windows

The NuTest program is designed to help you prepare for the FCC (Federal Communications Commission) examinations for all classes of amateur radio licenses. It presents you with on-screen tests using questions and answers taken from the FCC question pools. Printed practice examinations can also be generated. These have a format very similar to the FCC examinations administered by the VEC (Volunteer Examiner Coordinator) system. NuTest runs under Microsoft Windows 3.1 or Microsoft Windows 95.

Here are the prices:

<u>Number of classes</u>	<u>Fee</u>
One (1)	\$14.95
Two (2)	\$21.95
Three (3)	\$29.95

If three (3) classes are purchased then the fourth one is included FREE!

Special price for NuTest (all classes) and NuMorse purchased together is \$45.00!

If you are an existing registered user of NuMorse you may register all four license classes in NuTest for the price of two. Registered users of NuMorse **must** contact me first to take advantage of this offer.

The latest version of Nutest is always available in the HAMNET Compuserve forums and will soon be available for ftp download from many other sites. You may also obtain a copy direct from the author.

Address for NuTest enquiries:

A. Lacy.  
58 Bilbrook Road.  
Codsall.  
Wolverhampton.  
WV8 1ER  
United Kingdom.

Email to:

100030.157@compuserve.com

Or

CIS 100030,157

## **Windows Keys**

Choose from the following list to review the keys used in Windows applications:

[Cursor Movement Keys](#)

[Dialog Box Keys](#)

[Editing Keys](#)

[Help Keys](#)

[Menu Keys](#)

[System Keys](#)

[Text Selection Keys](#)

[Window Keys](#)

## Cursor Movement Keys

<b>Key(s)</b>	<b>Function</b>
DIRECTION key	Moves the cursor left, right, up, or down in a field.
End or Ctrl+Right Arrow	Moves to the end of a field.
Home or CTRL+Left Arrow	Moves to the beginning of a field.
PAGE UP or PAGE DOWN	Moves up or down in a field, one screen at a time.


## Dialog Box Keys

<b>Key(s)</b>	<b>Function</b>
TAB	Moves from field to field (left to right and top to bottom).
SHIFT+TAB	Moves from field to field in reverse order.
ALT+letter	Moves to the option or group whose underlined letter matches the one you type.
DIRECTION key	Moves from option to option within a group of options.
ENTER	Executes a command button. Or, chooses the selected item in a list box and executes the command.
ESC	Closes a dialog box without completing the command. (Same as Cancel)
ALT+DOWN ARROW	Opens a drop-down list box.
ALT+UP or DOWN ARROW	Selects item in a drop-down list box.
SPACEBAR	Cancels a selection in a list box. Selects or clears a check box.
CTRL+SLASH	Selects all the items in a list box.
CTRL+BACKSLASH	Cancels all selections except the current selection.
SHIFT+ DIRECTION key	Extends selection in a text box.
SHIFT+ HOME	Extends selection to first character in a text box.
SHIFT+ END	Extends selection to last character in a text box

## Editing Keys

<b>Key(s)</b>	<b>Function</b>
Backspace	Deletes the character to the left of the cursor. Or, deletes selected text.
Delete	Deletes the character to the right of the cursor. Or, deletes selected text.

## Help Keys

<b>Key(s)</b>	<b>Function</b>
F1	<p>Gets Help and displays the Help Index for the application. If the Help window is already open, pressing F1 displays the "Using Windows Help" topics.</p> <p>In some Windows applications, pressing F1 displays a Help topic on the selected command, dialog box option, or system message.</p>
SHIFT+F1	<p>Changes the pointer to  so you can get Help on a specific command, screen region, or key. You can then choose a command, click the screen region, or press a key or key combination you want to know more about.</p>



## Menu Keys

<b>Key(s)</b>	<b>Function</b>
Alt	Selects the first menu on the menu bar.
Letter key	Chooses the menu, or menu item, whose underlined letter matches the one you type.
Alt+letter key	Pulls down the menu whose underlined letter matches the one you type.
LEFT or RIGHT ARROW	Moves among menus.
UP or DOWN ARROW	Moves among menu items.
Enter	Chooses the selected menu item.

## System Keys

The following keys can be used from any window, regardless of the application you are using.

<b>Key(s)</b>	<b>Function</b>
Ctrl+Esc	Switches to the Task List.
Alt+Esc	Switches to the next application window or minimized icon, including full-screen programs.
Alt+TAB	Switches to the next application window, restoring applications that are running as icons.
Alt+PrtSc	Copies the entire screen to Clipboard.
Ctrl+F4	Closes the active window.
F1	Gets Help and displays the Help Index for the application. (See <a href="#">Help Keys</a> )

## Text Selection Keys

<b>Key(s)</b>	<b>Function</b>
SHIFT+LEFT or RIGHT ARROW	Selects text one character at a time to the left or right.
SHIFT+DOWN or UP	Selects one line of text up or down.
SHIFT+END	Selects text to the end of the line.
SHIFT+HOME	Selects text to the beginning of the line.
SHIFT+PAGE DOWN	Selects text down one window. Or, cancels the selection if the next window is already selected.
SHIFT+PAGE UP	Selects text up one window. Or, cancels the selection if the previous window is already selected.
CTRL+SHIFT+LEFT or RIGHT ARROW	Selects text to the next or previous word.
CTRL+SHIFT+UP or DOWN ARROW	Selects text to the beginning (UP ARROW) or end (DOWN ARROW) of the paragraph.
CTRL+SHIFT+END	Selects text to the end of the document.
CTRL+SHIFT+HOME	Selects text to the beginning of the document.

## Window Keys

<b>Key(s)</b>	<b>Function</b>
ALT+SPACEBAR	Opens the Control menu for an application window.
ALT+Hyphen	Opens the Control menu for a document window.
Alt+F4	Closes a window.
Alt+Esc	Switches to the next application window or minimized icon, including full-screen programs.
Alt+TAB	Switches to the next application window, restoring applications that are running as icons.
Alt+ENTER	Switches a non-Windows application between running in a window and running full screen.
DIRECTION key	Moves a window when you have chosen Move from the Control menu. Or, changes the size of a window when you have chosen Size from the Control menu.



# NuMorse QSO and test Generation

These QSO generator help topics and [QSO generator index](#) have been kept separate from the rest of the NuMorse help topics because they are fairly extensive but will probably not be required unless you intend to alter the QSO generator script files.

The [test](#) and [QSO](#) generation features of NuMorse have been designed to produce authentic seeming QSOs by assembling various phrases stored in two editable script files [NUMORSE.PHR](#) and [NUMORSE.TOP](#)

Before you modify the script files it is important to understand how they are organised and how QSO messages and tests are automatically assembled from them.

See also:

[Step by Step QSO generation](#)  
[Step by Step test generation](#)

# QSO generation step by step

1. Optionally, set QSO message parameters in Generate\QSO [dialogue](#) box.
2. Go to [Play](#) or [Drill](#) dialogue box.
3. Select QSO as [code source](#).
4. Press the Play button.

For more information see [QSO generator overview](#)

# Test Generation step by step

1. Optionally, set QSO message parameters in Generate\QSO dialogue box.
2. Go to Generate\Tests dialogue box
3. Select type of test
4. Press the GO button to generate the test in the form of 3 minimised test windows
5. Print the contents of the question and answer windows if you wish. You may do this while they are minimised if you do not wish to see the contents.
6. Select the QSO.TST window before opening the Play or Drill dialogue box.
7. Set the code source to Top Window and press Play

For more information see [Test generator overview](#)



# QSO generator Topic Index

(Also see the [main index](#))

[Character checking](#)

[Character counting](#)

[Characters in QSO](#)

[Characters Missing](#)

[Check Phrase File](#)

[Check Topics File](#)

[Distracters in tests](#)

[Hints and Tips](#)

[Overview of test generation](#)

[QSO generation parameters](#)

[QSO generation step by step](#)

[QSO overview](#)

[QSO Window](#)

[Random character specifier](#)

[Random](#)

[Structure of Phrase File](#)

[Template file structure](#)

[Test file windows](#)

[Test generation step by step](#)

[Test parameters dialogue](#)

[Test settings](#)

[Topics file structure](#)

[Worked example](#)

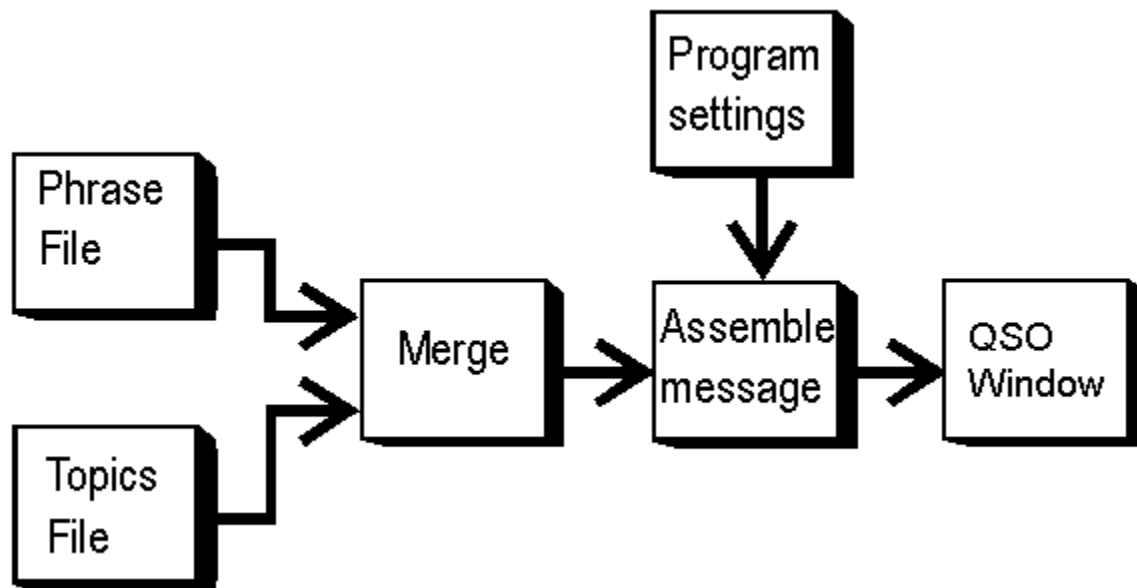
## Overview of QSO generator.

The QSO generator. is a feature designed to generate text files containing self-consistent authentic sounding Morse Code messages to give practice in the decoding of messages similar to those heard on the Amateur Radio bands. An additional feature is the ability to generate tests that can be made to conform to the requirements of many testing authorities.

The QSO generator. has been designed so that users can access and customise almost of the QSO and test generator parameters. A unique feature of the QSO generator. is the provision of a simple script language whereby users can edit a set of built-in script files to alter the sort of QSO messages and tests that are produced. The user can also select the length of generated messages. Assembled messages are written to the topmost window on the NuMorse desktop and then played as code in the normal way.

Most testing authorities require the inclusion of specific characters in a code test. To facilitate this the QSO generator will attempt to make sure that every message produced contains all of the characters specified in the list box in the QSO generate dialogue box and display a characters missing error if it cannot do this.

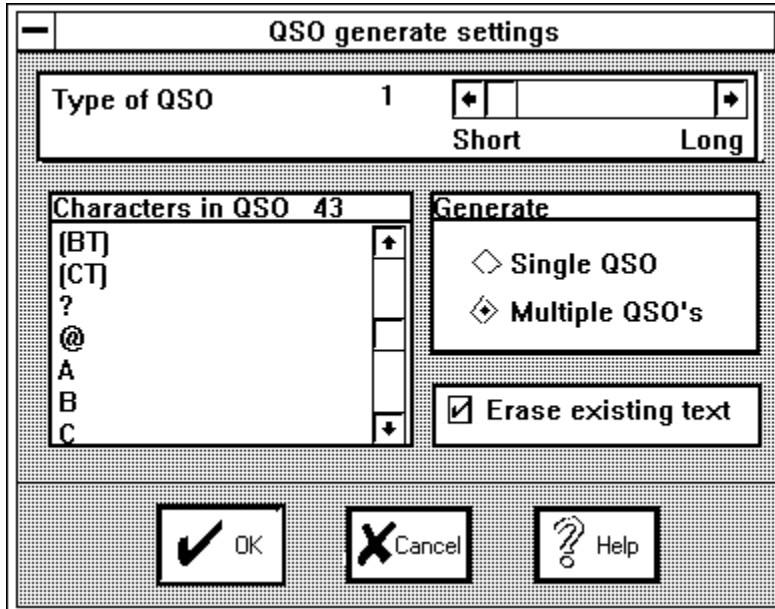
Here is a schematic diagram showing how messages are produced. Select items on the diagram for further details.



# Generate QSO dialogue box controls.

This dialogue box is accessed from the Generate menu item and determines how QSO message generation is carried out in Play and Drill modes when the selected code source is set to QSO.

Select items on this diagram for an explanation of the controls:



This slider control allows you to select the length of generated QSOs. Here are the approximate lengths of QSOs produced at various slider settings using the supplied script files:

<b>QSO type setting</b>	<b>Approximate QSO Length</b>
1	6 minutes, played at 5 WPM
2	6 minutes played at 13 WPM
3 to 9	6 minutes played at 20 WPM

The [Phrase Files](#) topic explains how the QSO type setting determines what phrases are assembled into QSOs.

If this control is checked then when generated QSO messages are inserted into a file window any existing contents are removed first. If the box is not checked then the generated QSO is appended to the end of any existing text.

If single QSO is checked then code generation will halt after one QSO message has been produced and sent as code. If multiple QSO is checked then messages will be generated and sent until the pause or cancel buttons in the Play and Drill dialogues are pressed.

This scrollable list box is where you can specify the characters that must be included in QSOs. Many testing authorities require all of the alpha and numeric characters as well as certain punctuation symbols and prosigns. The [Characters Missing](#) help topic gives further details of how NuMorse ensures that the required characters are present.

This button exits from the dialogue box and retains any changes.



This button exits from the dialogue box discarding any changes.

This button invokes the help system.

## Topic file errors.

If you open the Topics file using the Generate/Topics file window and save the file then a check for errors will be made. If errors are found then an error message will be displayed. You are given the option to ignore the error and save the file anyway. If you choose to re-edit the file then selecting the edit option will re-open the file window at the line containing the error. In the case of multiple errors only the first error will be detected. Here is a list of the current error messages that are related to topic files.

Maximum line length is 40 characters.  
'[' must be at start of section header.  
No topics in previous section.  
Section header must start with 'A' to 'Z'.  
Missing ']'.  
Spaces not allowed in section headers.  
Badly formed cross reference.  
Cross reference needs at least 2 characters.  
Invalid cross reference.  
No section headers found.  
Section header needs at least 2 characters.  
Section header expected.  
Duplicate section header.  
Section header must end with ']'.  
'#' is not allowed in topic section headers.

In addition to the above there are a few errors that can only be detected during message generation, these errors will be displayed when a test or QSO message is being assembled. An example of this is when the message generator is unable to compose a message containing all of the alphanumeric characters and prosigns that have been specified in the Generate/QSO dialogue box. In this case a characters missing error message will be displayed.

## Phrase file errors.

If you open the Phrase file using the Generate/Topics file window and save the file then a check for errors will be made. If errors are found then an error message will be displayed. You are given the option to ignore the error and save the file anyway. If you choose to re-edit the file then selecting the edit option will re-open the file window at the line containing the error. In the case of multiple errors only the first error will be detected. Here is a list of the current error messages that are related to phrase files.

Maximum line length is 80 characters.

Phrase section header expected.

'[' must be at start of section header.

Sequence number 1-99 expected.

':' expected.

Phrase class number 1-9 expected.

Space missing before header title.

Section title must begin with 'A' to 'Z'.

Missing ']'.  
'

Spaces are not allowed in header title.

No phrases in previous section.

Misplaced ','.

Bad character in topic reference.

Can't find topic reference.

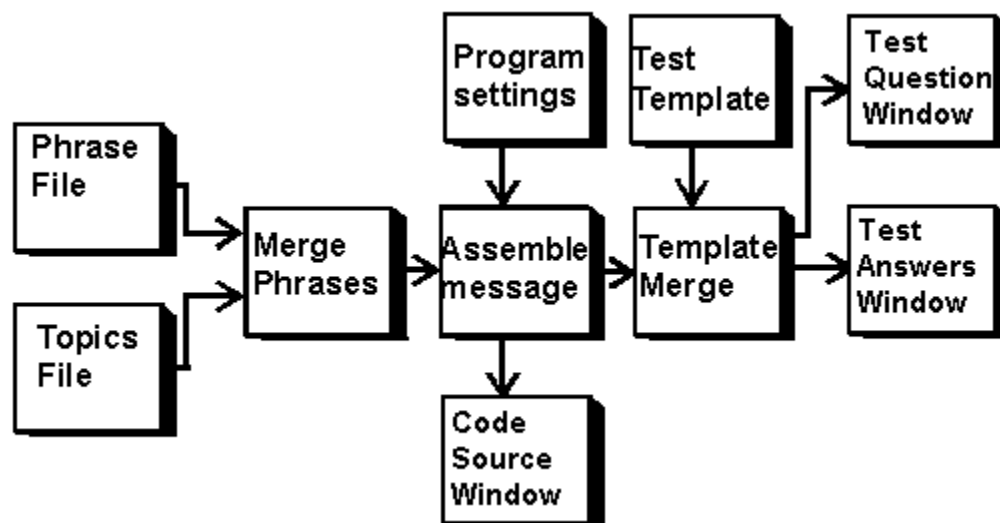
In addition to the above there are a few errors that can only be detected during message generation. These errors will be displayed when a test or QSO message is being assembled. An example of this is when the message generator is unable to compose a message containing all of the alphanumeric characters and prosigns that have been specified in the Generate/QSO dialogue box. In this case a characters missing error message will be displayed.

# Overview of test generation.

Code tests can be generated in a variety of formats; for example multi-choice tests or simple question and answer tests can easily be produced. The test format is set in the Generate Tests dialogue box. After generation the tests and answers are automatically combined with an editable template file to produce the final result which is written to a set of windows ready for you to edit, print or save as required.

Most testing authorities require the inclusion of specific characters in a code test. To facilitate this the QSO generator will attempt to make sure that every message produced contains all of the characters specified in the list box in the Generate QSO dialogue box and display a characters missing error message if it cannot do this.

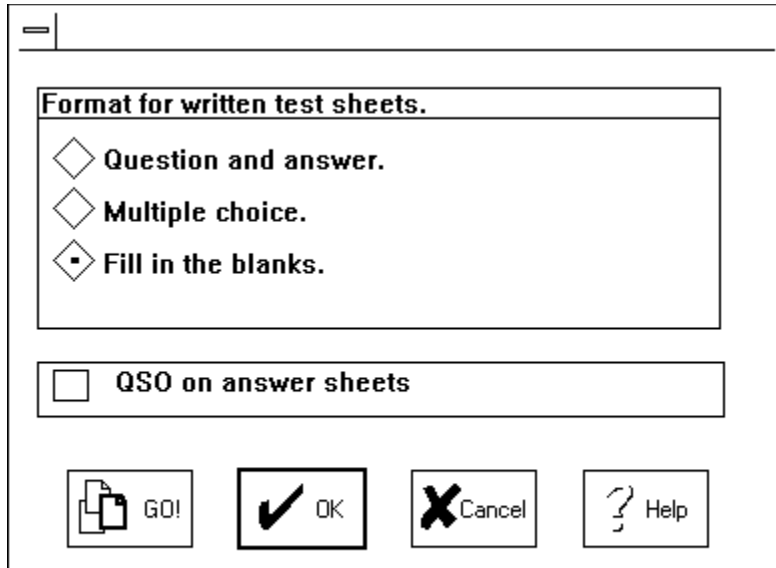
Here is a schematic diagram showing how tests are produced. Select items on the diagram for further details.



# Test dialogue controls

This dialogue box is accessed from the Generate menu item and determines how test generation is carried out.

Select items on this diagram for an explanation of the controls:



Questions are presented, candidate writes the answers. Answers are printed in answer sheet.

Multiple choice with distracters. Answer sheet contains an answer matrix



Message with key items blanked on question sheet. Complete message is printed on answer sheet.

The generated QSO is shown on the answer sheet in addition to lists of answers

This button starts up the test generator and opens 3 test file windows.

# Test windows.

If you select GO from the Generate/Test dialogue box then test generation will start. Three new file windows will be opened, or if file windows with matching filenames are already open then these will be used instead. The file windows will be minimised before text is written to them in case you wish to test yourself and therefore do not wish to spy on the test answers!

The test file windows have the following filenames:

**QSO.TST** A QSO message intended to be used as source text by NuMorse. The format will depend on the settings in the Generate/QSO dialogue box.

**QUESTION.TST** A list of questions about the message in QSO.TST, the format depends, among other things, on the setting in the Generate/Test dialogue box. The page layout will be determined by an editable file called TEMPLATE.TST

**ANSWER.TST** A list of answers to the questions in **QUESTION.TST** based on the same page layout.

The file window **QSO.TST** will be selected ready for you to play using Drill or Play dialogues with code source set to Top Window; you do NOT need to restore or maximise this window to play code from it. You may restore the file windows for viewing and editing or you may choose to save or print them while they are minimised.

## QSO window

The QSO window is a file window that is written to by the QSO generator prior to code being read from it by the code generator. If one or more file windows are open on the NuMorse desktop then the topmost one is used as the QSO window. If no windows are open then a new one is opened. You can make any file window the topmost one by simply clicking on its title bar before opening the Play or Drill dialogue boxes. Settings in the Generate\QSO dialogue box determine how QSO messages are written to the QSO window.

# Check For

Most testing authorities require that certain characters are present in code tests. The Character Checking list is where the characters to be checked are selected. If you select certain EXTRA characters then you will probably start getting error messages during code generation due to The QSO generator being unable to synthesise messages with the extra characters because they may not be present in the script files. If extra characters are really required then it may be necessary to add them to the script files. Before you do this consult the [Phrase file](#) help topic. To select/deselect individual characters click on them. Use click and drag to select character groups.

# Characters Missing

The characters missing error box appears when the assembled message lacks some of the characters that are required to be in a QSO message as defined in the Characters in QSO list box in the QSO generate dialogue box. As shipped the characters checking list box contains the following characters:

A..Z	26
0..9	10
,.?	3
+*=/	4

Giving a total of 43 characters. The phrases and topics script files are set up so that these characters will always be available. If you add to the characters checking list then NuMorse may not be able to assemble messages containing the required characters and the characters missing error message will be displayed. You will need to either remove characters from the characters checking list or add the characters to the script files.

If you decide to modify the script files then making sure that the resulting messages contain a variety of alphanumeric characters is a demanding part of the task. The QSO generator is quite intelligent at getting all the required characters into a message. At points where the actual character is not important to the sense of the message (the transmitter power being run is a typical place where the precise number is not important) the QSO generator will try to insert characters that are required, but have not yet been used in the message or test. But there is a limit to what can be achieved so you need to make sure that text in the phrases and topics files contain a good variety of the less common letters (such as Z,X,Y).

# Structure of Phrase File.

See also:

[Worked example](#)

[Script file hints and tips](#)

[Structure of Topics File](#)

The script files are text files. Although they could be edited using any ASCII based text editor; the built-in editors accessed from the Generate menu item are recommended because error checking is carried out as the files are saved to disk. In both types of script file any lines beginning with a semicolon will be ignored, they can be used as comment lines.

The phrase file is made up of a number of sections called phrase sections. Each phrase section starts with a line called a phrase section header which must be surrounded by square brackets. After the phrase section header is a number of phrase lines containing text and the placeholder symbols > and #. When a phrase line is selected by the message generator for inclusion in a message it is scanned for > and # symbols. Wherever these symbols are encountered another file called the topics file is used as a source of text. The steps below show how this works:

- 1 Choose a phrase line from the next phrase section.
- 2 Select each word in turn from the phrase line.
- 3 If the word starts with a > or # then get a substitute word from the topics file if it exists.
- 4 Copy the word to the QSO text.
- 5 Go back to step 1

Here is a typical phrase section header taken from the supplied Phrase file, click on any part of it to see further details:

## [20:1 EQUIPMENT USED BY STATION]

After the phrase section header there is normally a number of phrase lines.

Normally there must be at least one phrase in a phrase section, typically there are around 5. One phrase from each phrase section is chosen at random for assembly into the final QSO message.

The exception to the above rule is where you wish to always repeat a section of a message, for example the call signs of the transmitting and receiving stations might be repeated several times. In this case if you use an exact copy of a previous phrase section header title then there is no need to provide any section phrases. The previous ones will be used instead.

In the examples that follow (taken from the supplied phrase file) you may click on the components for more detailed explanations.

Several examples of typical phrase lines from a phrase file are shown below. A mouse click on the various parts of the examples will display further explanations. Each example builds on the previous ones, so you will need to work through them in sequence.

Example 1:

## SO HW COPY



Example 2:

NICE TO WORK YOU AGAIN >NAME2

Example 3

>499 WATTS.

Example 4

#499 WATTS.

Example 5

>QTH IN >STATE

Example 6

=UR RST IS #RST

I hope the examples above will give you enough of an understanding to look at the supplied phrase file where you will see more complex ideas based on them. See also the [worked example](#) and [Script file Hints and Tips](#) help topics.

The section in the topic file entitled RST will be found and a topic entry chosen for substitution into the above text. In addition, due to the # placemaker, the phrase section header title will be used as a question if a test is being generated. The chosen topic entry will be nominated as the correct answer and will appear on the answer sheet and in the code test message. If a multi choice test is being constructed then 3 other topic file lines will be chosen at random to insert into the test as distracters.

There are 2 place holders here, and so 2 text substitutions will be made, one from the QTH topics list and one from the STATE topics list. If you have managed to follow the logic in these examples then you may ask how the correct state for a QTH can be chosen if the substitutions are truly random. That is explained in the section on topics files, for now all you need to know is that different topics can be linked together within the topics file. In the case of the current example if Texas is chosen from the STATE topics section in the topics file then only cities in that state will be chosen from the QTH topics section.

This example is the same as Example 3 except that the alternative place holder symbol (#) is used. The effect on the assembled code message is the same, but additional processing occurs if a test is being generated. The phrase section header for this phrase will be used as a question and printed on a question sheet, and so sections containing the # symbol need to have their header titles phrased as questions. The text assembled into the code message is also used as the <correct> answer on the answer sheet. If a multi-choice test is being generated then 3 other distracter values are generated for inclusion on the test sheet.

The greater than symbol causes a look up for a topic called 499. No such topic exists in the supplied topic file and so this text is treated as a group of random character specifiers. In the case of numbers each character is replaced by a random digit between 0 and the digit specified. (An exception is made in the case of leading numeric digit specifiers where the substitution is between 1 and the value of the specifier) Where the character specifiers are letters the substituted character is a random letter between A and the character specifier value.

This part of the phrase will appear in the message unchanged.

This part of the phrase will appear as shown in the message, try clicking on the greater than sign or the text following it.

The greater than (>) symbol is one of two place markers used in phrase lines. It means that the following text is NOT to be assembled into the message; a text substitution is to be made instead. The following text can be either a label for a topic, or if the topic does not exist then it is a specifier for a sequence of random numbers or letters.



In the supplied topics file there is a topic called NAME2 and this means go to that topic and select a name at random, then substitute the name in to the assembled message in the place of the label NAME2. Note that wherever the label NAME2 is used throughout the current message the same text substitution will be made.

This is the simplest example of a phrase, this phrase will appear exactly as shown here when assembled into a message. Note the leading space at the start, phrases are assembled without automatic insertion of spaces. Because of this any spaces required in the assembled message must be explicitly inserted into the source phrase lines.

This left square parenthesis must be the first character on the line and denotes the start of a phrase section header.

This is a sequence number and determines the order in which selected phrases will be assembled this number must follow the bracket with no spaces allowed. All phrases in the following section have sequence 20 in this example. Phrases which must be near the message start have low sequence numbers and vice versa. Allow for insertion of additional sequence numbers by using steps of, say, 5 for sequence numbers. If two or more phrase headers have the same sequence number then the generator assembles phrases from their sections in random order; this is a common occurrence. A sequence number of 0 means that phrases in the section must always follow on from those in the immediately preceding section.

This colon separates the priority number from the sequence number; no spaces are allowed between the colon and each number

The priority number together with the QSO type and test type sliders determines which phrase sections are to be used. For example setting the test type to 3 means that phrases with priority 1, 2 and 3 would be assembled to produce the test message. The priority number must be followed by 1 space. Sections with priority 1 will always be used.

This space must separate the priority number from the phrase title.

This is the section title and must always be present. It can contain numbers, letters and spaces and must be at least 4 characters long. If a test is being generated then the questions that will appear are obtained from the section titles.



This bracket denotes the end of the phrase section header.

# Structure of Topics File.

See also:

[Worked example](#)

[Script file hints and tips](#)

[Structure of Phrase File](#)

The topics file is made up of a number of sections called topics sections containing text. This text is selected and incorporated into phrases derived from the phrase file. Each section starts with a topic section header which must be surrounded by square brackets. Lines beginning with a semi-colon will be ignored and can be used as comment lines.

Here is a typical Topic file section based on the supplied files. When the text #STATE occurs in the phrase file then a state name will be chosen at random from the list below. Click on any part of it to see further details:

```
[STATE]
ALABAMA
ALASKA
ARIZONA
ARKANSAS
[next topic header here]
```

The internal consistency of generated messages is a major feature of The QSO generator. The next example shows how this can be achieved, note that the number of topic lines in the following example is the same as the number in the previous example. The > (greater than) symbol indicates that instead of a random selection the choice of QTH is to depend on the choice previously made in the [STATE] section.

```
[QTH>STATE]
BIRMINGHAM
FAIRBANKS
PHOENIX
LITTLE ROCK
[next topic header here]
```

The only problem with the above construct is that the QTH selected for a given state is always the same. For example when the first item ALABAMA is selected from the STATE list then BIRMINGHAM will always be selected from the QTH list. But everyone knows that there is more than 1 city in Alabama and the above setup would add an undesirable amount of predictability into test and practice messages. The following example demonstrates how this problem can be avoided. There are now TWICE as many items in the QTH list as in the STATE list and a random selection is made from the two QTH items corresponding to ALABAMA. If three times as many QTH items were present then each QTH choice would be made from three items and so on.

```
[QTH>STATE]
MONTGOMERY
BIRMINGHAM
JUNEAU
FAIRBANKS
PHOENIX
TUCSON
LITTLE ROCK
CAMDEN
[next topic header here]
```

The next example shows a situation where we wish to generate plausible seeming signal reports. The idea here is to select a signal strength at random, and then use this to select valid RST reports as well as appropriate comments on the signal. Take special note of the number of topic lines in each topic file section shown below. An item from the STRENGTH section is selected at random. In the COMMENT section there is a > symbol in the header indicating that selection of a COMMENT item is to be based on the STRENGTH selection. But there are 6 items in the STRENGTH section and only 2 items in the COMMENT section so what happens?

The following rule is used:

If items 1 to 3 in STRENGTH are selected then item 1 in COMMENT is used.

If items 4 to 6 in STRENGTH are selected then item 2 in COMMENT is used.

And so strengths of 4 to 6 will have a QRN comment and strengths 7 to 9 will have a SOLID COPY comment. The RST section also depends on the STRENGTH section, but it has twice the number of entries. Here is a summary of how the selections will be made:

STRENGTH	COMMENT	RST
4	/QRN	348 or 349
5	/QRN	349
6	/QRN	448 or 449
7	/SOLID COPY	459 or 569
8	/SOLID COPY	579 or 589
9	/SOLID COPY	599

[STRENGTH]

4  
5  
6  
7  
8  
9

[COMMENT>STRENGTH]

/QRN  
/SOLID COPY

[RST>STRENGTH]

348  
349  
349  
349  
448  
449  
459  
569  
579  
589  
599  
599

[next topic header here]

A suitable phrase file line that would use the above topics relationships would be something like ...UR RST >RST >COMMENT.... This would result in messages containing phrases like:

...UR RST 349 /QRN....

...UR RST 579 /SOLID COPY....

...UR RST 599 /SOLID COPY...

And so on.

I hope that examples above will give you enough of an understanding to look at the supplied topic file where you will see more complex ideas based on them. See also the [worked example](#) and [Script file Hints and Tips](#) help topic.

There are 12 lines in topic section RST and so a STRENGTH selection of the first item will give a RST value of 348 or 349, selection of the second item will give a RST of 349, and so on.

This section header specifies that item selection in RST is to be based on the previous selection in STRENGTH.

There are 2 topic lines in this section, but we have specified that selection is to be based on the STRENGTH section that contains 6 lines, so what happens? This is an example of a many-one dependency. Selection of any of the first 3 positions in STRENGTH will cause selection of the first item in COMMENT. Selection of items 4-6 in STRENGTH will result in selection of the second item in COMMENT.

This section header specifies that item selection in COMMENT is to be based on the previous selection in STRENGTH.



The items here are not intended to be displayed in a message, and so their values are not important. What is important is that the position of the selected item is to be used in subsequent topic sections.

This topic section is a simple type, since no dependencies are specified selections will be made at random.

The crucial point here is that there are now TWICE as many topic lines in the QTH topic section as there are in the STATE topic section. In this case when ALABAMA is selected then MONTGOMERY or BIRMINGHAM will be chosen, the choice between them being at random. If ARIZONA was selected from STATE then PHOENIX or TUCSON will be selected from QTH. This is called a one-many dependency. This mechanism will work for any other multiple, for example 3 or 4 cities from each state could be included in QTH.

This topic SECTION header has a > symbol. This tells the message generator that, instead of a random selection, the selection is to be based on which item was selected in a previous topic, in this case the selection is to be based on the selection in the STATE topic in the previous example. A typical line from the Phrase file to utilise this construct would be...QTH IS >QTH IN >STATE...  
The resulting assembled message would then be ...QTH IS BIRMINGHAM IN ALABAMA... or ...QTH IS PHOENIX IN ALABAMA... and so on.

If ALABAMA had been chosen from the STATE topic section then BIRMINGHAM would be chosen from the QTH topic section. Similarly if ARIZONA had originally been chosen then PHOENIX would be selected. This is called a one-one dependency.

An item (in this case the names of States in America) will be selected from the topics list at random. So a phrase from the phrase file such as .... WE ARE IN >STATE..... will be converted into ....WE ARE IN TEXAS... or perhaps .....WE ARE IN ALABAMA..... and so on.

The topic header must be enclosed in square brackets. No spaces are allowed. The topic section header must not be the same as any of the topic lines in the associated section. To select a name from the NAME2 topics section the phrase must reference it by means of a place holder symbol like this:

NAME HERE IS >NAME2

# Script files, A Worked Example.

This help topic assumes that you have read the Phrase and Topics script files. The example here is taken from the supplied files with a few small modifications. I wanted the QSO messages to contain a comment about the transmitting stations QTH. Here is a list of requirements, click on the highlighted parts of the list to see how each requirement was met:

- 1) The comment should appear around the middle of the message but not always in the same place.
- 2) The individual parts of the comment should be grouped together as a sentence.
- 3) The amount of detail should depend on the program settings for test and QSO type.
- 4) The message should be realistic, with cities in correct states.

Here is an extract from a phrase file which fulfils the above requirements; it contains 3 phrase sections each one with a phrase section header:

```
[20:1 QTH PREAMBLE]
QTH IS
QTH
[0:2 DISTANCE]
1 MILE >DIRECTION OF
>DIGITS26 MILES >DIRECTION OF
[0:1 THE TRANSMITTING STATION QTH]
#QTH IN >STATE
#QTH >STATE
```

The following are extracts from relevant sections of the topics file and are accessed by the above phrase lines:

```
[STATE]
ALABAMA
ALASKA
ARIZONA
ARKANSAS
CALIFORNIA
```

```
[QTH>STATE]
MONTGOMERY
BIRMINGHAM
CLANTON
GADSDEN
JUNEAU
FAIRBANKS
ANCHORAGE
VALDEZ
PHOENIX
TUCSON
MESA
FLAGSTAFF
HOPE
PINE BLUFF
```



LITTLE ROCK  
CAMDEN

[DIGITS26]

2  
3  
4  
5  
6

[DIRECTION]

NORTH  
SOUTH  
EAST  
WEST  
NORTHEAST  
NORTHWEST  
SOUTHEAST  
SOUTHWEST

Here are some examples of messages generated by the above example.

With QSO type set to 1 the phrase sections with priority of 2 are not used:

QTH IS MONTGOMERY IN ALABAMA  
QTH BIRMINGHAM IN ALABAMA  
QTH IS JUNEAU ALASKA  
QTH JUNEAU IN ALASKA

With QSO type set to 2 or greater to obtain longer messages, because more phrase sections are being used:

QTH IS 1 MILE SOUTH OF MONTGOMERY IN ALABAMA  
QTH 6 MILES NORTH OF BIRMINGHAM IN ALABAMA  
QTH IS 1 MILE NORTH OF JUNEAU ALASKA  
QTH 3 MILES SOUTHEAST OF JUNEAU IN ALASKA

Here are some examples of what parts of a test would look like.

If a fill-in blanks test format is generated part of the test would look something like this:

...QTH IS \_\_\_\_\_ IN ALASKA...

While a question/answer test would look something like:

⋮  
⋮  
3) THE TRANSMITTING STATION QTH \_\_\_\_\_  
⋮  
⋮

In the case of a multi-choice question the corresponding part of the test would look like:

⋮

3) THE TRANSMITTING STATION QTH

A MONTGOMERY

B BIRMINGHAM

C CLANTON

D GADSDEN

⋮

(The precise layout may not be exactly the same as this.)

In the case of a multi choice question format you will have included instructions on the test template to ring or tick the correct answer. The answer sheet will contain a single letter corresponding to the item that was assembled into the test message. It is important to make sure that sufficient alternative choices are available in the topics file for selection of the distracter items.

This is where the candidate is to put the city name as included in the code message and, of course, on the answer sheet.

These are the bearings from the local city, to be selected at random. I guess that occasionally this will put a station location a few miles out to sea!

As well as a distance from the local city I wanted the message to contain a direction from that city, this section contains suitable bearings.

These are the digits which end up as distances in the final message.

I wanted a QTH description that included a distance of a few miles from a local city. The random number feature would have included the values 0 and 1. A message ...QTH IS 0 MILES FROM... would not be acceptable, neither would ...QTH IS 1 MILES FROM...

So I needed selection of a small number, but avoidance of 0 and 1. Hence the small topic section containing the digits from 2 to 6 inclusive, selected at random.



This is the header for the topic section QTH and contains names of cities. Due to the > symbol and the reference to STATE the selection here will be based on what item was selected in STATE.

In this example 4 cities have been entered in the QTH section for each entry in the STATE section. Thus the QTH section is 4 times larger than the STATE section. A random choice will be made from the 4 correct cities for the state.

The entries in the STATE topic section are names of states, to be selected and included in the message being generated, an item will be selected and used to replace the text >STATE in the phrase file.

This is the header for the topic section containing names of states, no > symbol is used to force selection of specific items and so state names will be selected at random.

In this phrase section the city and state are selected from the QTH and STATE topic sections. These two sections are linked so that the selected QTH will be correct for the selected state. In addition any tests produced will, because of the # symbol contain a question derived from the QTH phrase section header.

This priority 2 section will be used when type 2 to 9 messages are specified. It includes extra location details. One problem here was to get correct usage of MILE and MILES. There are various ways of doing this, I chose to use 2 phrases to achieve the desired result. The second phrase where MILES is used obtains distance values of 2 to 6 randomly from the topic section DIGITS26. To get a realistic direction description I created a topic section called DIRECTION, items from this are selected at random.

The first or second phrase line is selected at random so the message will sometimes contain QTH IS and sometimes just QTH.

This is achieved by specifying in the topics file that QTH should be dependant on STATE.



The priority numbers follow the colon in the phrase section headers. In this example the first and third phrase sections will be used in type 1 QSO messages while the second section with priority 2 will be switched off. For type 2 QSO messages all three sections will be used.

The sequence numbers of the second and third phrase headers are set to 0. Phrases from these sections will always follow on from the first section of the group that has none zero sequence number.

The sequence number of the first phrase header is set to 20. If you examine the complete phrase file you will see that a number of phrase sections have this number, and there are other sequence numbers preceding and succeeding it. Thus the group of phrases with sequence number 20 will be around the middle of the message, but with random placement within the group.

# Script files, Hints and Tips

Here is a selection of hints and tips that you might find useful if you decide to modify or rewrite the script files.

Make sure you have read the Phrase and Topic script file descriptions.

Save a copy of the original script files as supplied in case things go badly wrong!

If arbitrary numbers or figures are needed then use random character specifiers wherever possible. The message generator keeps track of what characters must be included in the message, but it can only insert them where random character specifiers are placed.

Automatic substitution of required punctuation and prosigns does not occur (although you will be warned of their absence). You must make sure that they occur sufficiently frequently in the script text.

If you keep the phrases in each Phrase section similar in length then generated messages for a given QSO or test type will be consistent in length.

Phrases and the topic references in them do not need to be separated by spaces. This allows such things as callsigns to be fabricated from items in several different sections.

Special numeric formats such as dates are best done by selection from topic sections, but bear in mind the tip above regarding automatic substitution of random characters.

Problems with missing characters in messages will mainly occur in short messages. Try to include topic items with a good variety of numbers and letters. Lists of operators names can usefully include such names as ZOLA and LUCY to get uncommon letters included in the finished message.

A phrase line can consist solely of one or more topic references

Comment lines can be placed in script files, start a comment line with a ; character and it will be ignored by the message generator.

When writing script files intended to generate multi choice tests sufficient items must be present in each section to allow 3 distracters to be chosen as well as the correct answer. In practice this means that selection must be from at least 4 items in the case of random selection.

The script file error check does not pick up all possible errors. Watch out for incorrect or misspelled topic references, they will be interpreted as random character or digit specifiers.

# Structure of Template Files.

A template file is an ASCII file with filename TEMPLATE.TST which specifies any additional text to accompany the questions and answers. It contains a place holder symbol (> greater than) to indicate where the test or answers should appear on the final output. Lines starting with a semicolon are ignored and can be used as comment lines.

Typically, a test template will begin with instructions to the candidate. Spaces for candidate details might come next, you could use the \_ (underline) symbol to indicate where candidate name and address are to be written. It is a good idea to place a line across the full page width before and after the > symbol to make it clear where the candidate is to write the answers. Many authorities allow the candidate to directly transcribe a number of consecutive characters instead of answering the questions; you might decide to leave a space and some comments if your testing rules permit this.

The Phrase text contains placeholders, items from the Topics file are inserted into these placeholders to produce complete phrases. The QSO generator follows a set of rules to decide what text is inserted into these placeholders, details of these rules are in the [Phrase File Structure](#) topic.

When the phrases are properly constructed they are assembled together to form a complete message. The QSO generator follows a set of rules to decide the order in which the phrases are assembled, details of these rules are in the [Phrase File Structure](#) topic.

Various program settings in the QSO generate dialogue can affect the way that the message is constructed. The most important one is the QSO type slider control. In the Phrase File each phrase has a phrase type associated with it which determines if it is to be used for message construction. For example if a type 2 message is required then phrase types 1 and 2 will be used and (in the case of the supplied script files) a medium length message will be produced.



The assembled message is checked to make sure that all of the required characters are present and then the message is sent to the topmost open file window.

Various program settings in the Test Generate dialogue can affect the way that the message is constructed. In the Phrase File, each phrase has a phrase type number associated with it which determines if it is to be used for message construction, this is also called a priority number. For example if a type 2 message is required then phrase types 1 and 2 will be used and (in the case of the supplied script files) a medium length message will be produced. The QSO Type slider control in the Generate\QSO dialogue determines what sort of test is constructed.

When the phrases are properly constructed they are assembled together to form a complete message. The QSO generator follows a set of rules to decide the order in which the phrases are assembled, details of these rules are in the [Phrase File Structure](#) topic. The assembled test is checked to make sure that all of the required characters are present and in the case of a multi choice test a set of plausible [distracters](#) as well as correct answers is chosen. A properly formatted set of answers is also generated.

Distracter is a term used to describe the incorrect, but plausible, answers that are presented in a multi choice test.

The raw tests and answers are merged into a template file before being written into file windows on the NuMorse desktop.

Characters are counted using the following rules:

Letters and numbers count as 1 character.

Prosigns, represented by + \* / and = count as 2 characters.

Punctuation symbols count as 2 characters.

Spaces and other ASCII control codes are not counted.

Random characters used in The QSO generator are not truly random. The message generator knows what characters should be included in a code test and gives priority to these if they are not already present.

Random character specifiers are used to produce random letters and numbers in The QSO generator. One use for random text is the generation of callsigns. Random character specifiers are placed in the Phrase script file and the following explains how they operate.

In a phrase section line the > or # symbol is used to indicate that the following text is either a topic section reference or a random character specifier. A lookup is performed in the topics file and if a matching topic section header is not found then it is assumed to be a random character specifier and the following rules will be used to determine a letter or number to substitute into each character in the specifier.

If the specifier character is between A and Z then a random character between A and the specifier value will be used.

If the specifier character is between 0 and 9 then a random character between 0 and the specifier value will be used unless this is the first character of a group. In this case a random character between 1 and the specifier value will be used; this avoids the unnatural use of leading zeros in generated messages.

Here are some examples (the # symbol would give the same results):

Specifier	Typical Result
>999	238
>222	120
>ZZZ	AGZ
>Z45	D23
>QTH	LISBON (topic file contains section header entitled QTH, with an entry LISBON)
>QTH	GTA (in the case of no matching section header in topics file)





