NuGen !

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 NuGen program or any information connected with the program.

Topics available from the help index:

Click on an underlined topic.

Topic Index

Overview of NuGen

Nugen is an application designed to generate text files containing self-consistent authentic sounding Morse Code messages. These files can be used in Morse Code trainer programs, such as NuMorse, 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.

NuGen has been designed so that users can access and customise almost of the QSO and test generator parameters. A unique feature of NuGen is the provision of a simple script language whereby users can edit a set of built-in script files to alter the sort of messages and tests that are produced. The user can also select the length of messages to be produced.

The tests can be generated in a variety of formats; for example multi-choice tests or simple question and answer tests can easily be produced. After generation the tests and answers are automatically combined with editable template files to produce the final result called a test set.

Up to 26 test sets (all different) can be produced in one session.

For practice purposes up to 26 messages can be combined into a file for subsequent play back by most Morse code trainer programs.

Most testing authorities require the inclusion of specific characters in a code test. To facilitate this NuGen will check the content of every message produced, and take action if the required characters are missing. In keeping with the philosophy of the program, the action taken and characters checked can be set by the user. For example a warning message can be displayed, or a replacement message can be automatically generated.

From the Author

Hello.

This program is designed to work with my other creation, NuMorse for Windows although it should be OK with most code trainer programs. Ive made this prototype available as a result of my experiences with the evolution of NuMorse, which started life on CIS as a fairly miserable single dialogue affair. Many of the features currently offered by NuMorse are a result of feedback from people who downloaded earlier versions; it is truly shareware in the sense that users had a share in its development.

So, any constructive comments about NuGen V0.0 will be appreciated, especially from VEs in the USA where NuMorse seems to be especially popular. It might be YOUR idea that gets added to the next version.

Best Wishes, Tony Lacy G4AUD, July 1994

Program Registration

They say that the best things in life are free. Im not sure about that, but NuGen V0.0 certainly is free!

While I retain all rights over NuGen V0.0 you are permitted to use the NuGen V0.0 package for evaluation purposes free of charge. You are also permitted to distribute it in its entirety provided any fees charged by you are for media/media packaging/delivery only.

The next version will be shareware, and a registration fee will be required. For your convenience I plan to use the CIS on-line registration service for collection of registration fees. For further details of on-line registration GO SWREG after logging on to CIS.

Edit Windows

The Edit Windows in NuGen offer many word processor like functions including the ability to load from and save to files. Like most Windows based editors they conform to the IBM Common User Access (CUA) guidelines which should be familiar to most users. A full set of search/replace functions is provided as well as the all important undo command. If text is altered then you are asked if you wish to save changes on exit from an Edit Window.

Maximize Icon

For information on this topic get to the Windows Help window and click the Glossary button on the toolbar.

Minimize Icon

For information on this topic get to the Windows Help window and click the Glossary button on the toolbar.

System Menu

For information on this topic get to the Windows Help window and click the Glossary button on the toolbar.

Title Bar

For information on this topic get to the Windows Help window and click the Glossary button on the toolbar.

Sizing Border

For information on this topic get to the Windows Help window and click the Glossary button on the toolbar.

Procedures

For information on how to:

Produce a file of QSOs.
Produce test sets.
Play the code that has been produced by NuGen.

Click on an underlined topic.

Generate QSO

To generate one or more messages and write them to a file follow these steps:

Check that the required *.QSO file is selected or created using the Select QSO item in the files menu.

Check that the required script files are selected using the Select Script item in the files menu.

Move the top slider bar until the required number of messages is selected. You may produce up to 26 QSOs in one session.

Select the type of QSO to be produced. The type of QSOs generated will depend on how the script files have been set up. If the supplied script files are used the following QSOs will be produced:

QSO type setting	**Approximate QSO Length**
1	6 minutes, played at 5 WPM
2	6 minutes played at 13 WPM
3 to 9	6 minutes played at 20 WPM

Now click on the 

button and the QSO file will be generated. You may view or edit this QSO file by selecting the QSO\View QSO menu item. NuGen checks the characters in the file and if any characters are missing from a QSO action will be taken depending on the settings in the Character checking\Action dialogue.

If you need to rewrite the script files then you will need details of how messages are generated from script files. A good place to start is the Overview of QSO Generation. help topic

View QSO

This editor allows you to check and alter QSO files. The file initially selected for editing is determined by the File\Select QSO menu item, and is the same file that will be written to by the QSO generate feature. If the Embed Speed Codes checkbox in the code speed dialogue was checked during QSO generation then you will see something like @5,5 on the first line of the text. This is a NuMorse control code, and sets the NuMorse code trainer program to the required speed and character spacing (NuMorse V1.30 and later) if this file is used as a source code. There are several other codes that NuMorse will respond to; see the NuMorse help file for further details.

View Phrases

This editor allows you to view and edit the Phrase file. This file and the topics script file is used by the test and QSO generators to assemble realistic seeming code messages. The NuGen program has been designed so that users can tailor script files to produce messages to their own requirements. Bear in mind that you do not HAVE to get involved in rewriting script files in order to use NuGen. The files provided may be perfectly adequate for your purposes.

To successfully write your own script files requires an understanding of their structure and how they are combined together to produce code messages and tests. Click on one of the following for essential information on writing your own script files.

Overview of QSO generation.
Overview of Test generation.
Structure of phrase file.
Structure of Topics file.
Structure of template files.

View Topics

This editor allows you to view and edit the Topics file. This file and the phrase script file is used by the test and QSO generators to assemble realistic seeming code messages. The NuGen program has been designed so that users can tailor script files to produce messages to their own requirements. Bear in mind that you do not HAVE to get involved in rewriting script files in order to use NuGen. The files provided may be perfectly adequate for your purposes.

To successfully write your own script files requires an understanding of their structure and how they are combined together to produce code messages and tests. Click on one of the following for essential information on writing your own script files.

Overview of QSO generation.
Overview of Test generation.
Structure of phrase file.
Structure of Topics file.
Structure of template files.

Check Topics

Clicking on this item causes the currently selected topics script file to be checked for certain errors. If an error is found a message is displayed and you can choose to open an edit window, with the cursor positioned at the error line. For an explanation of the required syntax of the Topics file please refer to the topics file help topic.

Check Phrases

Clicking on this item causes the currently selected phrases script file to be checked for certain errors. If an error is found an error message is displayed and you can choose to open an edit window,, with the cursor positioned at the error line. For an explanation of the required syntax of the Phrases file please refer to the Phrases file help topic.

Test Format

3 test formats are currently supported by NuGen.

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

2) Multiple choice with distractors. Answer sheet contains an answer matrix

3) Message with key items blanked on question sheet. Complete message on answer sheet.

The questions and distractors are obtained from the script files. The complete message can optionally be printed on any format of answer sheet. A fourth test format can be obtained by simply omitting the place marker symbol from the test sheet, with the candidate being required to transcribe the message.

Generate Tests

To generate one or more tests follow these steps:

Check that the required template files are selected using the Select Template item in the files menu.

Check that the required script files are selected using the Select Script item in the files menu.

Move the top slider bar until the required number of test sets is selected. You may produce up to 26 test sets in one session.

Select the type of test to be produced. The type of tests generated will depend on how the script files have been set up. If the supplied script files are used the following tests will be produced:

Test type setting	**Average test Length**
1	6 minutes, played at 5 WPM
2	6 minutes played at 13 WPM
3 to 9	6 minutes played at 20 WPM

Now click on the 

button and the test files will be generated. You may view or edit the test files by selecting the Written tests\View edit item and choosing the file to view or edit. Test file names are automatically produced as tests are generated.

NuGen checks the characters in the file and if any characters are missing from a test then action will be taken depending on the settings in the Character checking\Action dialogue.

The format of the resulting test and answer sheet will depend on two things:

The settings in the Written test\Format Dialogue
The contents of the Test and Answer template files.

Since it is possible to generate up to 26 X 3 = 78 files per session a batch file delete option is provided, to use this select the Written tests\Print delete menu item.

If you need to rewrite the script files then you will need details of how messages are generated from script files. A good place to start is the Overview of Test Generation. help topic

Code is output as a plain ASCII file with extension *.QST. The filename is automatically derived from the template file name. You may view and edit any of the files output by NuGen.

A test file is output as a plain ASCII file with extension *.TST. The filename is automatically derived from the template file name.You may view and edit any of the files output by NuGen.

An answer sheet file is output as a plain ASCII file with extension *.ANS. The filename is automatically derived from the template file name. You may view and edit any of the files output by NuGen.

Delete/Print Tests

You may delete individual files from File Manager, and printing is available from any of the edit windows which appear in this program. The set of features provided here is useful when groups of files need to printed or deleted. Select the files to be printed or deleted by clicking on the names in the list box. Select groups of adjacent files by mouse dragging across the required items. For separated items Shift-click on each file to be selected.

Select the required file operations and then click on OK.

Check For

Most testing authorities require that certain characters are present in code tests. The Character Checking\ Check For dialogue box is where the characters to be checked are selected. If you select EXTRA characters then you will probably start getting error messages during code generation due to NuGen being unable to synthesise messages with the extra characters. If extra characters are really required then it will 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 Character Checking\Action dialogue is where you determine what action is to be taken if an assembled message lacks the characters required to be used in a Code test. The most useful setting is the one which allows automatic discard and retry if characters are missing. You can watch this process in action during generation of a sequence of tests. As the generator progress slider bar moves you will see the number of tests indicator incrementing. Occasionally you will see a number in square brackets appear, this indicates that a message reassemble is taking place due to missing characters in the original message. This number should never exceed 3 or 4 retries when good script files are in use. If you decide to rewrite the script files then making sure that the resulting messages contain the correct characters is the most demanding part of the task. If the number of retries exceeds 9 then the generator will halt with an error message.

NuGen 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) NuGen will try to insert characters that are required, but have not yet been used in the message or test..

The other options are useful for debugging purposes if you are rewriting the script files. They enable you to either ignore the missing characters or to halt message generation and see which characters are missing.

This error box appears when the assembled message lacks some of the characters that are required to be in a message as defined in the Character Checking\Check for dialogue box AND the Character Checking\Action dialogue is set to Display errors. The Display errors option is useful if you are rewriting script files because you can obtain obtain information on what characters are causing problems. Normally you would set the Character Checking\Action dialogue option to automatic retry, in which case this error box is never displayed. Here is what the buttons do:

The message is written to file and assembly of the next message proceeds.

Message generation is halted and the generate test dialogue box is closed.

The message is discarded and another attempt at message assembly is tried.

Code Speed

This feature is intended to be used when NuMorse V1.30 or later is used to play the code files produced by NuGen. Set the sliders to the required speed and check the Embed Codes checkbox. There are two results:

1) NuMorse character and code speed control codes will be placed at the start of generated code files. For example, in the case of character speed 12 WPM and code speed 5 WPM the codes would be: @5,12

2) Character and code speed information will be appended to generated answer sheets, together with a character count and an estimated code transmission time. The estimated time is based on number of characters, code speed and average characters per word.

Select Template

Choose what templates are used to format your test and answer sheets. Not used for QSO generation. Test set filenames are derived from template filenames. If you select a template file name that does not exist you will be given the chance to create a new pair of template files. These newly created files contain a few dummy lines which you will need to replace.

Select Scripts

Choose what script files are used as a source of text for your practice messages. See also the overview of test generation for the purpose of script files. Script files must always be available in pairs, a topics file (extension .TOP) and a phrase file (extension .PHR). This menu option opens script files in pairs. To create a new pair of script files type a file name that does not exist and confirm your choice. The newly created script files will contaion a few dummy lines which will need to be replaced by your own lines.

Select Qso

Choose the file to which messages will be written. Unlike files generated for tests a QSO file is intended primarily for code practice and up to 26 different messages can be combined into 1 file.

Custom Menu

This menu is customised by the user using the Custom Path item to add items, normally these items will result in the launching of a Windows application. Once set up it is used just like any other menu, double click on an item to run it. By making use of the corresponding menu item in NuMorse (versions 1.30 on) it is possible to switch seamlessly between the two applications. You can set up to 2 Custom Menu items which will allow 3 different applications to function as one. (Yes I have a third Morse Code related application in the pipeline)

If you do not wish to use the Custom Menu method to switch between applications then you may use one of the standard Windows methods to achieve the same thing.

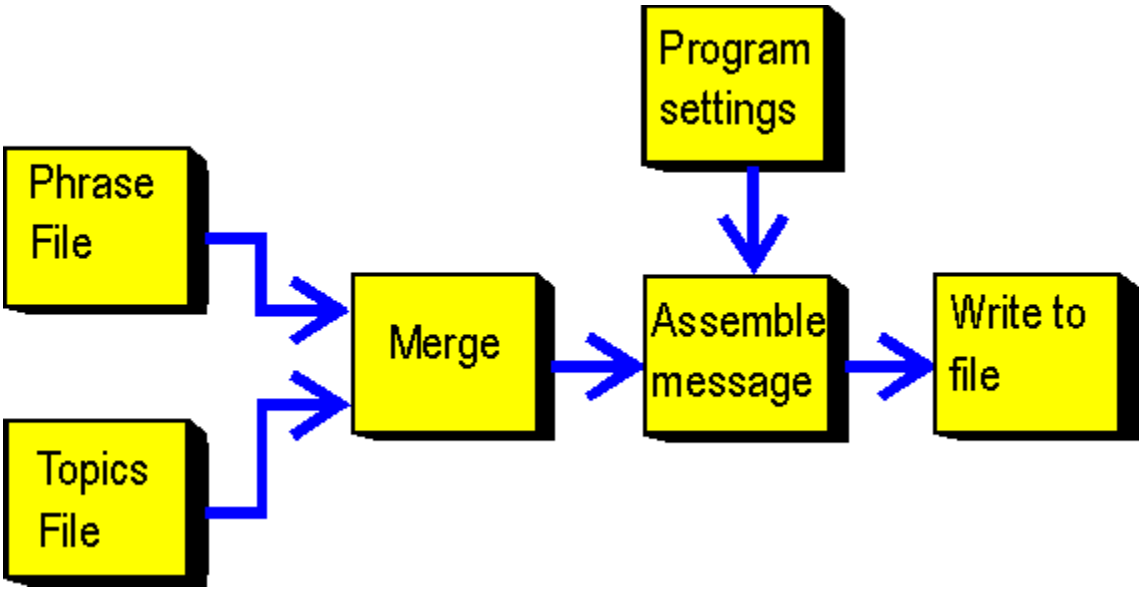
Custom Path

In this dialogue you may define the MSDOS path to any windows application. If you enter text into one of the name fields then an extra menu item will appear in the RUN menu list. Clicking on that item will execute the application that you defined in the path field. To remove a menu item delete the corresponding name field, there is no need to remove the path text. An error message and error number will be displayed if the path is not valid or if other problems occur. Here is a list of the more likely numbers and what they mean:

0	System was out of memory, or executable file was corrupt.
2	File was not found.
3	Path was not found.
5	Attempt was made to dynamically link to a task, or there was a network-protection error.
8	There was insufficient memory to start the application.
10	Windows version was incorrect.
11	Executable file was not a Windows application.
12	Application was designed for a different operating system.
13	Application was designed for MS-DOS 4.0.
14	Type of executable file was unknown.
15	Attempt was made to load a real-mode application.
19	Attempt was made to load a compressed executable file.
20	One of the DLLs required to run this application was corrupt.
21	Application requires Windows 32-bit extensions.

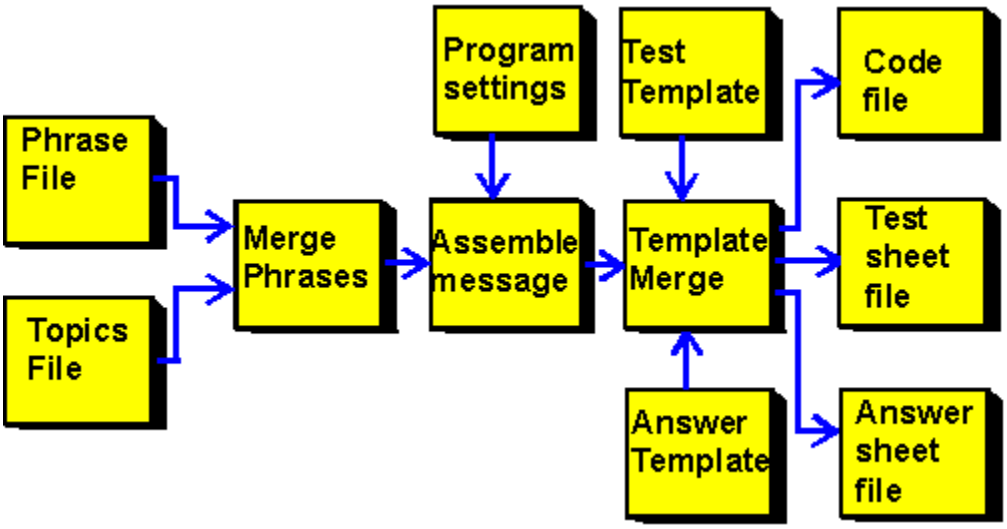
Overview of QSO Generation.

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



Overview of Test Generation.

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



Structure of Phrase File.

Before you read this topic I suggest you take a look at the help topic Apologies

The script files are text files. Although they could be edited using any ASCII based text editor; the built-in editor is recommended. 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. Werever 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.

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 component 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 message.
The exception to the above rule is where you wish to always repeat a section of a message. 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 phrase lines from the supplied phrase file are shown below. 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 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&Tips help topic.help topics.

The topic section entitled RST will be found and a topic entry chosen for substitution into the above text. In addition, due to the # placemarker, 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. 3 other topic entries will be chosen at random to insert into the test as distractors.

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 printed on any generated 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 distractor 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.

Before you read this topic I suggest you take a look at the help topic Apologies

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 section based on the supplied topic file, click on 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 NuGen. The next example shows how this can be achieved, note that the number of topic lines in the this example is the same as the number in the previous example..

[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 ALABAMA is selected then BIRMINGHAM will always be selected. 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:

[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 section.

[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

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&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 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 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) It 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 phrase file which fulfills 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 or test 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 or test 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 on the printed sheet 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 distractor 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 messages while the second section with priority 2 will be switched off. For type 2 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 these 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 distractors 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.

Template files are text files that contain a place holder for tests or answers produced by NuGen. They give you the flexibility to design test and answer sheets in any format. The place holder, which indicates where the test or answers should go is a > (greater than) symbol at the start of a line. Examine the supplied *.TPL file for an example of a test sheet template. The supplied *.TPA file is an example of an answer sheet file.

Typically, a test template will begin with instructions to the candidate. Spaces for candidate details will 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.

An answer template uses the same placeholder; any comments you write here will be directed to examiners. Under certain program settings NuGen will append 4 or 5 lines of information on character and code speed to the end of an answer sheet, so you may need to leave space for this when designing your answer sheet templates.

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

The following files are used or produced by NuGen:

File type	Explanation
*.TPL	A text file used as a template during production of test question sheets.
*.TPA	A text file used as a template during production of test answer sheets.
*.PHR	Script file containing phrases designed for assembly into QSO.
*.TOP	A topics file containing words for inclusion in QSO.
*.TST	A test sheet, to be printed and given to the candidate.
*.QST	A file containing a single QSO to be used as source for a code test.
*.ANS	An answer sheet, to be printed and used for assessing test results.
*.QSO	A code source file containing QSOs to be used for code practice sessions.

Several other temporary files are produced, but should be erased during normal program operation, they generally have file extension *.TM? and can safely be deleted if found.

The Phrase text contains placeholders, items from the Topics file are inserted into these placeholders to produce complete phrases. NuGen 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. NuGen 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 appended to a text file previously selected from the menu option Files\Select QSO

Various program settings in the Test Generate dialogue can affect the way that the message is constructed. The most important one is the test type slider control. 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 Written Tests\Format dialogue determines how the test set is constructed.

A test set is made up of:

1) A code file containing a single message.
2) A test sheet file, for completion by the test candidate
3) An answer sheet file, to be used by the examiner.

When the phrases are properly constructed they are assembled together to form a complete message. NuGen 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 distractors as well as correct answers is chosen. A properly formatted set of answers is also generated.

Distractor 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 previously selected test and answer template files before being written as 3 files making up a test set.

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.

NuMorse!

A Morse code trainer for Windows. The latest shareware version is available for downloading from CIS GO HAMNET CW forum. None-shareware versions are also marketed in some countries. The files output by NuGen are plain ASCII and should work with most code trainer programs, however NuGen is designed to integrate smoothly with NuMorse and so this program is highly recommended.

Test set file names are automatically produced using the following scheme:

The Template filename is truncated to 7 characters in length if it is longer than this. A single letter from A to Z is appended to the name and the following extensions are added:

Test code files	.QST
Answer files	.ANS
Test files	.TST

When you select a script file name that does not exist you will be given the chance to create a new pair of files. The newly created files contain just enough structure for the program to run, although the messages produced are pretty boring. Before you write your own script files you will need to understand the structure of them.

Random characters used in NuGen 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.

Apologies

As I was writing the script file help topics and trying to explain how it all fitted together I began to realise that some people are going to wonder why there should be all this complexity. After all other programs simply require the press of an OK button to churn out code practice text.

Well here are my thoughts:

You dont HAVE to understand how the script files are constructed. You can use them as they are and if you choose to do this then usage of NuGen is as simple as selecting a few options and then pressing the GO! buttons.

But I wanted to write a program that could output messages that were like the messages that you would hear on the Amateur Radio Bands. At the same time I wanted it to be configurable by the user so that the output could be altered to suit a broad range of requirements, even to the extent of generating text in none-English languages. And this is how I came up with the idea of using a sort of programming language in the form of script files to specify how code messages are put together by NuGen.

Random character specifiers are used to produce random letters and numbers in NuGen. 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 (no matching section header in topics file)

In these help topics we refer to any text intended to be converted to audible Morse Code as a MESSAGE. In the case of test generation a message will be a single QSO in a file with extension .QST. In the case of practice QSO generation a number of messages can be combined into 1 file which will have extension .QSO.

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

Key(s)	Function
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

Key(s)	Function
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

Key(s)	Function
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

Key(s)	Function
F1	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. In some Windows applications, pressing F1 displays a Help topic on the selected command, dialog box option, or system message.
SHIFT+F1	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.

Menu Keys

Key(s)	Function
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.

Key(s)	Function
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 Help Keys)

Text Selection Keys

Key(s)	Function
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

Key(s)	Function
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.