# Table of Contents

License...........................................................................................................................................6
Recognitions.....................................................................................................................................7
Configuring.......................................................................................................................................9
  Configure Operator..................................................................................................................10
  User Interface Configuration – General.............................................................................11
    Capturing Log data.............................................................................................................12
  User Interface Configuration – Browser................................................................------------14
  User Interface Configuration - Macros................................................................................15
  User Interface Configuration - Contest..............................................................................16
  User Interface Configuration - WF Controls........................................................................17
  User Interface Configuration - Rx Text..............................................................................18
Rig Configuration........................................................................................................................19
  Rig Configuration-Hardware PTT control..............................................................................20
    Right Channel VOX Signal.................................................................................................20
    Serial Port using DTR or RTS............................................................................................20
    Parallel Port (Linux and Free BSD only).............................................................................20
  RigCAT control....................................................................................................................21
  Hamlib CAT control.............................................................................................................22
  Memory Mapped CAT & Xml-Rpc CAT................................................................................23
  Rig Xml Howto......................................................................................................................24
Waterfall Configuration-Display.................................................................................................30
Waterfall Configuration-FFT Processing..................................................................................31
Waterfall Configuration-Mouse..................................................................................................32
Modem Configurations..................................................................................................................33
  Modem Configuration-CW..................................................................................................33
    Timing.................................................................................................................................34
    PROSIGNS.........................................................................................................................37
  Modem Configuration-DominoEX......................................................................................38
  Modem Configuration-Olivia and Contestia.........................................................................39
  Modem Configuration-PSK...................................................................................................40
  Modem Configuration-RTTY / FSK.....................................................................................41
  Using the Fldigi Pseudo FSK (Rt. Channel) function to key a transmitter..........................42
  Modem Configuration-Thor...............................................................................................43
Id Configuration........................................................................................................................44
  Video Text...............................................................................................................................44
  Cw postamble........................................................................................................................45
  Reed Solomon Identifier.........................................................................................................45
    Reed-Solomon Identification Description.........................................................................46
    Description of RSID encoding............................................................................................47
Miscellaneous Configuration.........................................................................................................53
Callsign DB Configuration............................................................................................................55
Colors & Fonts.............................................................................................................................56
Freq Display...............................................................................................................................56
Operating Controls & Displays.................................................................58
  Mode Status Indicators.........................................................................61
Signal Browser..........................................................................................62
Keyboard Operation.....................................................................................64
  Function Keys.........................................................................................65
Mouse & Keyboard Shortcuts......................................................................65
Operating Multiple Copies.......................................................................68
Macros.......................................................................................................69
  Exec Macro............................................................................................74
    Exported variables..............................................................................74
    Detection of existing scripts...............................................................75
    Querying an external database.........................................................75
    Google Earth Map..............................................................................76
    Custom dates/times............................................................................76
Operating fldigi ........................................................................................78
  Operating CW Mode............................................................................78
    Carrier Frequency...............................................................................78
    QSK and the TAB KEY......................................................................78
    Pausing transmit................................................................................78
    Aborting transmit..............................................................................78
    WPM adjustment...............................................................................78
    CW configuration..............................................................................79
    Prosings............................................................................................79
  CW (Morse) Modes Description............................................................79
    General Description...........................................................................79
    Protocol...............................................................................................80
    Coding and Character Set.................................................................80
  Operating Contestia...............................................................................81
    Contestia Modes Description............................................................82
      General Description.......................................................................82
      Protocol..........................................................................................82
      Coding and Character Set..............................................................82
      Contestia Operating Parameters...................................................83
  DominoEX - Operating.........................................................................84
    DominoEX Modes Description..........................................................85
      General Description.......................................................................85
      Protocol..........................................................................................85
      Coding and Character Set..............................................................85
      DominoEX Operating Parameters.................................................86
  Hell Modes.............................................................................................87
    Hellschreiber Modes Description.......................................................88
      General Description.......................................................................88
      Protocol..........................................................................................88
License

Copyright (c)

- 2007, 2008, 2009 Stelios Bounanos, M0GLD
- 2008, 2009 Stephane Fillod - F8CFE
- 2009 John Douyere - VK2ETA

This application and all supporting documentation is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with the source code for fldigi; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
Recognitions

This software would not have been possible without the contribution of many programmers who have
given their best to the open source community. The application is built upon the foundation of the Fast
Light Tool Kit (http://www.fltk.org), a wonderfully fast and efficient graphical user interface design
library. Many have asked what the Fast Light means. There are probably as many answers as there are
programmers using the toolkit. I prefer to think of it as lightning fast and light on the code size. Take a
look at the size of the executable for fldigi and then compare it with similar applications. I think you
will be surprised by how small it is for what it does.

The active current development team consists of:

- Dave Freese - W1HKJ
- Stelios Bounanos - M0GLD
- Remi Chateauneu - F4ECW
- Leigh Klotz - WA5ZNU
- Stephane Fillod - F8CFE
- John Douyere - VK2ETA
- Joe Veldhuis - N8FQ
- Chris Sylvain - KB3CS
- Gary Robinson - WB8ROL

Localization files:

- French: Stephane Fillod - F8CFE
- Italian: Pierfrancesco Caci - IK5PVX
- Spanish: Pavel Milanes Costa - CO7WT

Several authors have placed their digital modem code and signal processing code in the public domain
and their source was either an inspiration or in some cases formed the backbone of the code used in Fldigi.

- AE4JY - WinPsk - a windows application
- Takuya OOURA - a generic Fast Fourier Transform for real valued data streams -
  http://momonga.t.u-tokyo.ac.jp/~ooura/fft.html
- Tomi Manninen, OH2BNS - gmfsk - a great digital modem program for Linux
- Hamish Moffatt, VK3SB - dominoEX code originally for gmfsk
- Dr. Steven W. Smith - author of "Digital Signal Processing", who has kindly placed an entire

If you make a side-by-side comparison between gmfsk and fldigi source code you will see that they
follow the same general structure. The primary difference is that gmfsk is written in the C language and
uses the gnome/gtk libraries for the user interface. Fldigi is a C++ application that uses the Fast Light
Tool Kit (Fltk) gui library. The design of Fldigi puts emphasis on separating the user interface from the
sound card and transceiver input/output operations. Nearly all modern digital modem programs use a
programming paradigm called "threads." Threads are light weight processes that share the same
memory space, but each has its own stack. The use of threads makes the program look and feel
responsive to the user while a lot of code is being executed in the background.

Many of the modem source code files are C to C++ rewrites from the gmfsk application. They say that copying is the best form of flattery and gmfsk simply had the best explanations and the easiest source code to read and understand. The author had also spent several months creating improvements and fixing bugs in the original gmfsk application. That exercise was the impetus to create Fldigi.

The Fast Fourier Transform used by Fldigi is a rewrite of Takuya Ooura's C code. The rewrite is in C++ but you will see the strong resemblance to Takuya's original if you study both. Takuya's FFT code was also used in the Winpsk program. Some of the signal processing algorithms used in Fldigi are from Dr. Smith's book. His on-line publication is sufficient to allow you to become fluent in fft analysis and the creation of digital filters. I printed the relevant pdf files and then purchased the hard bound copy. Improvements to the original gmfsk signal processing algorithms can all be attributed to this excellent source.

And last but certainly not least, I must thank the crew who perform alpha testing and on-line support of the application. These are stalwart amateurs who risk their operating system and radio equipment in testing, testing and more testing. Their only reward is in being able to influence the design of the application and the fun of seeing it work and the bugs disappear. Thank you to:

<table>
<thead>
<tr>
<th>Call</th>
<th>Name</th>
<th>Call</th>
<th>Name</th>
<th>Call</th>
<th>Name</th>
<th>Call</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>4Z5ST</td>
<td>Boris</td>
<td>K3GAU</td>
<td>David</td>
<td>KU1T</td>
<td>Zibi</td>
<td>VA3DB</td>
<td>Dianne</td>
</tr>
<tr>
<td>AA0HW</td>
<td>Chuck</td>
<td>K4XTT</td>
<td>Victor</td>
<td>KV9U</td>
<td>Rick</td>
<td>VE3XI</td>
<td>Dave</td>
</tr>
<tr>
<td>AC7JN</td>
<td>Dave</td>
<td>K6KAR</td>
<td>Kirk</td>
<td>N0NB</td>
<td>Nate</td>
<td>VK2TMG</td>
<td>Brett</td>
</tr>
<tr>
<td>CT1DRB</td>
<td>David</td>
<td>K7BRK</td>
<td>Chris</td>
<td>N2AMG</td>
<td>Rick</td>
<td>VK4BDJ</td>
<td>David</td>
</tr>
<tr>
<td>CX7BF</td>
<td>Walter</td>
<td>K4RE</td>
<td>Brian</td>
<td>N4UM</td>
<td>Tim</td>
<td>W3NR</td>
<td>Ed</td>
</tr>
<tr>
<td>DF4OR</td>
<td>Ekki</td>
<td>K9AO</td>
<td>Rick</td>
<td>N4ZNV</td>
<td>Mike</td>
<td>W4ROS</td>
<td>Ross</td>
</tr>
<tr>
<td>DK1JBE</td>
<td>Tom</td>
<td>KB3FN</td>
<td>Lynn</td>
<td>N6WFL</td>
<td>Jason</td>
<td>W6JVE</td>
<td>Jim</td>
</tr>
<tr>
<td>DL6XAZ</td>
<td>Fred</td>
<td>KD0AR</td>
<td>Mike</td>
<td>N8FQ</td>
<td>Joe</td>
<td>WA3VPZ</td>
<td>Marshal</td>
</tr>
<tr>
<td>DL8FCL</td>
<td>Walter</td>
<td>KD4O</td>
<td>Phil</td>
<td>NN8B</td>
<td>Don</td>
<td>WA4SXZ</td>
<td>Rich</td>
</tr>
<tr>
<td>G0UZP</td>
<td>Paul</td>
<td>KD8DKT</td>
<td>Mike</td>
<td>NT1G</td>
<td>Skip</td>
<td>WB8ROL</td>
<td>Gary</td>
</tr>
<tr>
<td>G3TDJ</td>
<td>Andy</td>
<td>KE3Y</td>
<td>Travis</td>
<td>OZ4KK</td>
<td>Erik</td>
<td>WD4FDW</td>
<td>Steve</td>
</tr>
<tr>
<td>G6CKR</td>
<td>Roger</td>
<td>KH6TY</td>
<td>Skip</td>
<td>PA0R</td>
<td>Rein</td>
<td>WD4FNY</td>
<td>Bill</td>
</tr>
<tr>
<td>G8SQH</td>
<td>David</td>
<td>KL7NA</td>
<td>Rob</td>
<td>PA3GWH</td>
<td>Richard</td>
<td>WU9Q</td>
<td>Bob</td>
</tr>
</tbody>
</table>

and many others whose names are not listed, please accept my apology.

The test team is representative of users on Windows, Linux, Free BSD and OS X operating systems. They have varying interests from very slow speed CW to high speed keyboard full break-in CW, from RTTY contesters to PSK rag chewers. They have insisted that fldigi perform well under all of those operations. I have been amazed by the global distribution of the testing team. It is easy to think that the internet will be the death of amateur radio. On the contrary it opens up so many additional ways for us to be cooperative.
Configuring

The first time you execute fldigi you should resize the main window to suit your screen dimensions. Then adjust the divider line between the Rx and Tx text widgets.

Fldigi contains many configurable items, to specify operator data, user interface, and modem characteristics. The application also saves many state variables between executions. It will start up in the state that it was last used.

You should initially configure the following:

**Operator UI Waterfall Modems Rig Audio Id Misc Callsign DB**

and

**Colors & Fonts**

When the program receives and transmits digital signals and your rig control is satisfactory then you can continue configuring other aspects of the program:

**Operator UI Waterfall Modems Rig Audio Id Misc Callsign DB**

You can configure each modem type to suit your particular operating needs, but the defaults should be satisfactory for most users.

**CW DominoEX FeldHell MT-63 Olivia Contestia Psk Rtty Thor**

To learn more about the characteristics of specific digital modes look here: Digital Modes, Sights & Sounds.

When you have completed the configuration go to the **Configure** menu and select **Save config** or press the “Save Config” button on the configure dialog box. The program will write the file ~/.fldigi/fldigi_def.xml.

Exit the program and restart it to test that your configuration was saved and is working correctly.

Your fldigi install is now ready for you to start receiving and transmitting digital signals.

Fldigi recognizes if any configuration changes are made and not saved. You will then be prompted to save the configuration when exiting the program.
Configure Operator

Enter your personal information on the Operator tab of the configuration dialog. This information is used by some of the macro expanders.

Your locator data is also used for automatically computing Azimuth to a remote locator when that is available from an on-line database Call query.

Discussion on Grid Squares: http://www.arrl.org/grid-squares
User Interface Configuration – General

Fldigi offers tips on the use of nearly every aspect of its operation. These are particularly useful when you first use the program, but after you are familiar with it's operation they tend to get in the way. You can turn them off by de-selecting "Show tooltips"

Some users prefer to not have icons on the menu system. You can turn them off also.

Fldigi offers three different look and feel based on the parent Fast Light Toolkit graphics interface; "base", "gtk+" and "plastic". These can be combined with command line specifiers for the default background and foreground colors to create a user unique look to fldigi. You will probably discover that the default colors and the gtk+ UI scheme are to be preferred.

Fldigi has internationalization files for French, Italian and Spanish. These control the contents of various menu items and labels. Linux users should build and install fldigi from source to gain access to these. Windows users should select the language of choice from the list, press the "Save" button and then close and restart fldigi. The "UI language" selector is only present on the Windows version of fldigi. The percentage indicates the completeness of the translation.
You may not want to use all possible modes when operating Fldigi. Press the "Visible modes" button and open the mode selector dialog.

The use of this dialog should be obvious after a few clicks here and there.

By deselecting all but CW, PSK31, PSK63 and RTTY the Op_Mode menu is decluttered of all the other modes. The WWV and Freq Analysis modes are always visible. When a subset of the entire mode bank is selected a new menu item is added to the Op_Mode menu, "Show all modes". This is a toggle to restore all modes to the menu. The inverse toggle is "Show fewer modes."

Capturing Log data

Fldigi has a built in logbook. You can request to be prompted whenever there is an unsaved entry in the qso log area. You can also elect to whether to clear all of the qso fields when the log is saved or to leave them intact. Auto-fill Country and Azimuth uses the data found in the file "cnty.dat" that you should download and place in the fldigi default folder. You can force the callsign field to be upper case independent of capture or keyboard entry. You enter your default Transmit Power which is used for the logbook record.

Fldigi has various ways to transfer data in the Rx panel to the qso logging fields. The default is to use a Shift-Left-Click paradigm. You can also use a double click method if you prefer. The Shift-Left-Click will still function. Each data item is considered to be a single word normally delimited by the space, tab or end-of-line character. You can add word delimiter characters in the designated text box. The default is to add *,-,; to the normal delimiters. This is a useful tool for extracting contest exchange data. The exchange might be RST, STATE and NAME. The station being worked might send this as 599-NJ-Bozo. Double clicking on the 599 NJ and Bozo would treat each as a separate word.
If you check the "Show callsign tooltips in received text" then the Rx text area will popup an information box whenever the mouse is held over a callsign for more than 2 seconds. The popup will look like one of the following:

If the station was previously worked the operator's name and azimuth/distance will be computed from the logbook gridsquare entry (Loc). Otherwise the azimuth/distance is computed from the data in the cty.dat file.

This file is maintained by, and can be downloaded from the following web site:

http://www.country-files.com/

Download the file and put it in your fldigi default files folder.

You can elect to have the RST out preset to 599 after you clear the QSO entry fields. User Interface Configuration - Browser
User Interface Configuration – Browser

Fldigi can display multiple decoded signals in both PSK and RTTY modes. The multi-channel display is either a separate dialog or an embedded resizable panel.

1) select the number of 100 Hz channels you want visible
2) select the start frequency for the browser (your transceiver may not rx signals below this value)
3) select the inactivity timeout for the browser. After this number of seconds the channel will be cleared and prepared for the next detection cycle.
4) select what kind of label annotation you want on each line
5) select the font and font size to be used in the browser
6) You can enter any text you want to search for in the Seek Regular Expression widget. This text can be a simple text snippit such as "CQ" or any regular expression. When the regex is satisfied in a channel the text color for that channel is changed to red. With a regex you can specify a more generic pattern, which means that you can match more things and your search is somewhat noise tolerant. Here is an example for a CQ from a US station (should match most callsigns):

cq.+?[aknw][a-z]?[0-9][a-pr-z][a-z]{1,2}

This says "cq followed by at least one character, followed by one A, K, N, or W, followed by an optional letter, followed by a digit, followed by a letter that is not q, followed by one or two letters". The search is case-insensitive.

All plain text is a valid regular expression, unless you really had been looking for these metacharacters:

\{|()\}*+?|^$  

These will have to be escaped with a backslash.
7) select whether you want a marquee type of continuous scrolling, or simply clear the line when it is filled.
8) select whether you want the lowest frequency at the bottom (checked) or the top of browser panel
9) select whether you want the audio stream history buffer to be played back when you select an active channel. The first-in first-out audio history represents the previous 2 minutes of received audio.
10) Both the background and slider hilite colors can be selected for the signal browser detection level control. The default colors are shown in these images.
User Interface Configuration - Macros

Access to a macro is gained by pressing the associated function key. There are actually 48 separate macros that can be created. With the default configuration you rotate between the sub-sets of 12 using the numbered button to the right of the macro bar, or by selecting a set with the Alt-1, Alt-2, Alt-3 or Alt-4 key combination. (on OS X use the Option-1 etc.)

The default user interface is a single macro bar of 12 buttons located just above the waterfall panel. There are times when you need ready access to more than 12 macro functions. When a 2 row configuration is selected the original row is forced to the Alt-1, or first sub-set of macros, and it's numeric rotate button is disabled. The second or SHIFTED macro button row can be rotate through Alt-2 ... Alt-4 using either it's numeric button or the Alt-#key combination. The fldigi macro bar positions will change immediately so you can see the selection.

With 2 rows shown you obtain access to the primary set with normal Function key press. The secondary set is accessed by a SHIFT-Function-key press.

You edit any macro definition by using a mouse right-click on it's button.

You can also select to use the mouse wheel to rotate through the macro sub-sets. When checked you simply hover the mouse over the macro bar and roll the mouse wheel.

Fldigi manages multiple files that contain macro definitions. You may want to have the last used macro file be the one available the next time you start fldigi. If so, simply enable the "load last used Macro file on startup" check box. You can also choose to display which macro file was loaded at startup or when a new macro file is loaded. A brief message indicating which file was loaded will be written to the Rx text area if this option is selected.
User Interface Configuration - Contest

Fldigi supports a generic but robust set of contest functions. In addition to serial-in and serial-out you can capture and transmit three exchange sequences unique to a specific contest. Enter the exchange you want to send for each of the three. You can force the RST in/out to always be 599. That seems to be a norm for many contests. When operating in a CW contest you can have fldigi send cut numbers, T for 0, N for nine.

![Fldigi configuration](image)

The serial number can be set to use leading zeros. You can specify the starting number for the sequence and how many digits are sent, ie: 0024. Pressing Reset will set the starting number to the qso logging serial out field.

You can check for duplicates by any combination of the specified named fields. You can also specify that the duplicate had to occur with a given time interval. Some VHF contests allow a duplicate CALL after a given time interval.
User Interface Configuration - WF Controls

You can configure the appearance of fldigi in a variety of ways, including the suppression of unused waterfall controls. The "Oper' Controls" tab contains selection for nearly every waterfall control. Before adjusting these settings it is recommended that you enable this menu item. You can then see the effect of enabling and disabling the various selection boxes.
User Interface Configuration - Rx Text

CW, RTTY (baudot), THROB, and CONTESTIA are modes that only transmit in upper case characters. They also have a very limited set of non-alpha characters. A screen full of UPPERCASE characters can be stressful. Select this option to print all of this text in lower case.
Rig Configuration

**Note:** The same control is also used for both manual entry of the transceiver frequency or with full CAT control. When no CAT is available the control is simply a convenient way of keeping track of the transceiver USB/LSB suppressed carrier frequency, the mode and the audio tracking point. If fldigi is being used with an FM transceiver you probably should enter the simplex frequency or the input frequency of a repeater being used. This frequency value is used with the waterfall audio frequency to compute the logged frequency. The logged frequency value will only be correct for LSB and USB operation.

The frequency/mode pick list is displayed when the book button is pressed. Pressing the book button a second time will restore the original logging panel.

The pick list buttons control selecting, adding and deleting entries in the frequency/mode list.

- add the current frequency / mode / audio track point to the list
- select the current list entry
- delete the highlighted entry from the list
- delete all entries from the list (a warning prompt will appear)
- show active frequencies based on either the entry field to the right or the stations locator, see pskreporter/spotter.
- entry field for active frequencies search, for example "EM."

The browser list contains frequency, sideband, modem type and audio frequency. The list is saved when fldigi is shut down.

The combo box on the left will allow the selection and control of the operating mode of the transceiver. The combo box on the right will allow the selection and control of the transceiver bandwidth.

The frequency display is in fact a set of special buttons. Each digit may be left-clicked to increment in frequency by that digit value, or right clicked to decrement by that digit value. The leading digits will follow suit if a decade rollover occurs. You can also place the mouse cursor on a digit and then use the mouse wheel to roll the frequency up and down.

Manual entry of frequency can be accomplished by clicking on any digit and then entering the numeric value in KHz. Don't forget the decimal point if you are entering a fractional KHz value.

The mode combobox, the bandwidth combobox and the frequency display also announce the current transceiver status. If you change operating mode on the transceiver, that will be announced in the respective combobox and fldigi will adjust any internal parameters accordingly. Fldigi queries the transceiver 10 times per second to maintain a lock step with the transceiver.
Rig Configuration-Hardware PTT control

Right Channel VOX Signal
Fldigi can generate a 1000 Hz tone for the duration of the PTT keydown period. A simple tone detector/filter and transistor switch can be used to generate a PTT signal from this sound card output. The circuit will be similar to that used for QSK control. This might be a convenient way to create a PTT signal for a small notebook or netbook computer that does not have a serial or a parallel port.

Serial Port using DTR or RTS
The simplest rig control is just being able to control the push to talk via an external transistor switch. You set this type of control on the first configuration tab for rig control.

You select this operation by checking the "Use serial port PTT". Select the serial port from the list (fldigi will have searched for available ports). Then specify whether the h/w uses RTS or DTR and whether a + or - voltage is required to toggle PTT on. You can use a serial port for control with the RTS and DTR pins configured for you particular interface. The program allows you to use RTS, DTR or BOTH for the PTT signal. Press the Initialize button to start the serial port.

Parallel Port (Linux and Free BSD only)
Fldidi sets and clears the parallel port pin, PARPORT_CONTROL_INIT, pin 16 on the 25 pin parallel port connector. Keydown sets Pin 16 to +5 volts and keyup sets the voltage to zero.
RigCAT control

RigCAT is a rig control system similar to hamlib that was developed specifically for fldigi. It uses command/response definitions that are found in various rig.xml files. You can use a rig.xml file specific for your transceiver or write and test one yourself. The easiest way is to adapt an existing rig xml file for a rig that is similar to your own. ICOM almost identical command/response strings for all of its transceiver line. Yaesu rigs have nearly all used unique command/response structures until just recently. The TS-450, TS-950 and others share a similar set of commands and responses.

RigCAT commands and responses are defined in a rig specificxml file which contains all of the required queries and responses in extended markup language format. Please read the specification document rigxml to learn more about this new way of building generic rig interface definitions and how they are used with fldigi. fldigi will look for a file in the $HOME/.fldigi/rigs directory for all files with extension".xml". These contain definitions for the transceiver indicated by the file name, ie: FT-450.xml, IC-756PRO.xml, etc. You can download the appropriate xml files from the resource directory tree http://www.w1hkj.com/xmls or from the archives at http://www.w1hkj.com/xmlarchives.html Place the file in your rigs directory and fldigi will find it.

You will need to specify how your PTT will be triggered. This can be using a CAT command, the RTS or DTR pins or none. None would be appropriate if you are using the rig's VOX or an outboard sound card interface such as the SignalLink SL-1+ which produces its own VOX type of PTT. In that case simply leave all of the PTT options unselected.

If you are using a transceiver or a rig interface such as CI-V that echos all serial data you check off the "Commands are echoed" box. That will suppress fldigi trying to respond to a command it just sent to the transceiver.

You may need to try various values of retries, retry interval, and command interval to achieve consistent rigcat control.

Press the Initialize button after setting all of the parameters. If the settings are all correct fldigi should start receiving frequency information from the rig and annunciating them on the rig control frequency display.
Hamlib CAT control

Hamlib is a set of standard libraries for interfacing to a large number of transceivers. The hamlib library system consists of a front end which acts on behalf of all rigs and backends which are specific to each rig. The fldigi implementation of hamlib differs on the various OS for which it is targeted. On the Unix/Linux based systems the hamlib is a shared library which the user must have installed on his or her system. This is the standard way of handling hamlib on Unix/Linux systems. On Windows the entire hamlib library has been compiled and statically linked into the application executable. No additional dynamic link libraries are necessary. This approach simplifies the installation of fldigi on Windows platforms.

Select your transceiver from the list of supported units. Then select the serial port and baud rate. If you are familiar with the hamlib library you can send various startup sequences to the rig using the advanced configuration. PTT control can be achieved using CAT commands or via DTR / RTS on the same port as the control comms. You might also need to specify whether RTS/CTS flow control is used (Kenwood rigs use this quite often) or if Xon/Xoff flow control is used.

You may need to try various values of retries, retry interval, and command interval to achieve consistent hamlib control.

Press the Initialize button after setting all of the parameters. If the settings are all correct fldigi should start receiving frequency information from the rig and announcing them on the rig control frequency display.
Memory Mapped CAT & Xml-Rpc CAT

Memory mapped control is selected if you are operating a Kachina 505DSP using the W1HKJ control software for that rig.

Xml-Rpc allows third party software to control various aspects of fldigi operation including but not limited to rig control. This is the data interface that is also used by the program flrig, a fldigi companion transceiver control program.

If you are using a third party interface such as DxKeeper Bridge you might be instructed to select this method of CAT.
**Rig Xml Howto**

This document describes the contents of the rig definition file “rig.xml”.

A number of transceivers have rig definition files written and tested which you may use. These are found in the xmls directory on this site: http://www.w1hkj.com/xmlarchives.htm You will find subdirectories by manufacturer which contain files named by rig type, ie: TS-850.xml. If you create, test and verify the proper operation for a transceiver not yet posted please share that with others by sending it as an attachment to w1hkj@w1hkj.com and I will post it on the web site. You are encouraged to study the various rig definition files to learn more about how they are organized.

Comments are contained within the tag pair

```xml
<!-
-->
```

and may appear anywhere in the rig definition file The entire rig definition must be contained within the tag pair

```xml
<RIGDEF>
</RIGDEF>
```

The text within the tag pair `<RIG></RIG>` specifies the transceiver to which this file applies, as in:

```xml
<RIG>Kenwood TRX-8000</RIG>
```

The text within the tag pair `<PROGRAMMER></PROGRAMMER>` is not used by the parser, but should as a minimum say who created and who tested the definition file, as in:

```xml
<PROGRAMMER>
Dave Freese W1HKJ Tested by: W1HKJ, Dave
</PROGRAMMER>
```

The text within the tag pair

```xml
<STATUS>
</STATUS>
```

is not used by the parser, but should as a minimum state whether the definition file has been “Verified”, is “Alpha”, what the Version and Date of creation or update, as in:

```xml
<STATUS> Verified Version: 1.0 Date: 2007 Jan 5 </STATUS>
```

The

```xml
<TITLE>
</TITLE>
```

tag pair contains the text which will be displayed on the window decoration bar, as in:

```xml
<TITLE>Rig Control - IC-746 PRO</TITLE>
```

The serial port parameters may be preset in the xml file and also set or changed on the rigcat configuration tab.

```xml
<!--
```
default settings for initial setup

-->

<table>
<thead>
<tr>
<th>XML Tag</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;TIMEOUT&gt;</code> TT</td>
<td>TT in milliseconds</td>
</tr>
<tr>
<td><code>&lt;RETRIES&gt;</code> NN</td>
<td>NN integer</td>
</tr>
<tr>
<td><code>&lt;WRITE_DELAY&gt;</code> TT</td>
<td>TT in milliseconds</td>
</tr>
<tr>
<td><code>&lt;POST_WRITE_DELAY&gt;</code> TT</td>
<td>TT in milliseconds</td>
</tr>
<tr>
<td><code>&lt;BAUDRATE&gt;</code> BAUD</td>
<td>BAUD = 1200, 2400, 4800, 9600, 19200, 38400 ...</td>
</tr>
<tr>
<td><code>&lt;STOPBITS&gt;</code> B</td>
<td>B = 1 or 2</td>
</tr>
<tr>
<td><code>&lt;RTSCTS&gt;</code> BOOL</td>
<td>BOOL = true, false; h/w handshake used for data flow control</td>
</tr>
<tr>
<td><code>&lt;RTSPLUS&gt;</code> BOOL</td>
<td>BOOL = true, false; set RTS signal line to +12 V</td>
</tr>
<tr>
<td><code>&lt;RTSPPTT&gt;</code> BOOL</td>
<td>BOOL = true, false; toggle RTS signal line for PTT</td>
</tr>
<tr>
<td><code>&lt;DTRPLUS&gt;</code> BOOL</td>
<td>BOOL = true, false; set DTR signal line to + 12 V</td>
</tr>
<tr>
<td><code>&lt;DTRPTT&gt;</code> BOOL</td>
<td>BOOL = true, false; toggle DTR signal line for PTT</td>
</tr>
<tr>
<td><code>&lt;ECHO&gt;</code> BOOL</td>
<td>BOOL = true, false; xcvr/interface echos all chars (typical of CI-V interface)</td>
</tr>
<tr>
<td><code>&lt;CMDPTT&gt;</code> BOOL</td>
<td>BOOL = true, false; use command string for PTT (not supported by all rigs)</td>
</tr>
</tbody>
</table>

The transceiver modes are specified within the `<MODES>` tag pair. Each entry or element associated with a mode has a symbol name (text) and a way to specify what the data transfer consists of. The data transfer might be a single byte, multiple bytes, or a string.

Example 1, for the Icom-746PRO

```
<MODES>
  <ELEMENT><SYMBOL>LSB</SYMBOL><BYTE>00</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>USB</SYMBOL><BYTE>01</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>AM</SYMBOL><BYTE>02</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>CW</SYMBOL><BYTE>03</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>RTTY</SYMBOL><BYTE>04</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>FM</SYMBOL><BYTE>05</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>CW-R</SYMBOL><BYTE>07</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>RTTY-R</SYMBOL><BYTE>08</BYTE></ELEMENT>
</MODES>
```

Example 2, for the Kenwood 850

```
<MODES>
  <ELEMENT><SYMBOL>LSB</SYMBOL><BYTE>31</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>USB</SYMBOL><BYTE>32</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>CW</SYMBOL><BYTE>33</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>FOR</SYMBOL><BYTE>34</BYTE></ELEMENT>
</MODES>
```
Example 3, for the FT-100

```
<MODES>
  <ELEMENT><SYMBOL>LSB</SYMBOL><BYTE>00</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>USB</SYMBOL><BYTE>01</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>CW</SYMBOL><BYTE>02</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>CW-R</SYMBOL><BYTE>03</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>AM</SYMBOL><BYTE>04</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>DIG</SYMBOL><BYTE>05</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>FM</SYMBOL><BYTE>06</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>W-FM</SYMBOL><BYTE>07</BYTE></ELEMENT>
</MODES>
```

The modes which are supported by lower sideband in the transceiver are specified in the `<LSBMODES>` tag pair. The string data for the lsb modes must match those given in the modes id specifier. For example in the Icom 746 Pro:

```
<LSBMODES>
  <STRING>LSB</STRING>
  <STRING>RTTY</STRING>
  <STRING>CW-R</STRING>
</LSBMODES>
```

If the transceiver data stream uses identically the same format for the bandwidth data then it is specified in the `<BANDWIDTHS>` tag pair.

Example for the Icom 746 Pro:

```
<BANDWIDTHS>
  <ELEMENT><SYMBOL>50</SYMBOL><BYTE>00</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>100</SYMBOL><BYTE>01</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>150</SYMBOL><BYTE>02</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>200</SYMBOL><BYTE>03</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>250</SYMBOL><BYTE>04</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>300</SYMBOL><BYTE>05</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>350</SYMBOL><BYTE>06</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>400</SYMBOL><BYTE>07</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>450</SYMBOL><BYTE>08</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>500</SYMBOL><BYTE>09</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>600</SYMBOL><BYTE>10</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>700</SYMBOL><BYTE>11</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>800</SYMBOL><BYTE>12</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>900</SYMBOL><BYTE>13</BYTE></ELEMENT>
</BANDWIDTHS>
```
If the bandwidth data stream is unique for send and receive data streams then they are specified separately with the `<BW-CMD>`/<BW-CMD> tag pair for data sent to the transceiver, and the `<BW-REPLY>`/<BW-REPLY> tag pair for data returned to the computer.

Example: FT-100:

```
<BW-CMD>
  <ELEMENT><SYMBOL>300</SYMBOL><BYTE>00</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>500</SYMBOL><BYTE>01</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2400</SYMBOL><BYTE>02</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>6000</SYMBOL><BYTE>03</BYTE></ELEMENT>
</BW-CMD>

<BW-REPLY>
  <ELEMENT><SYMBOL>300</SYMBOL><BYTE>03</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>500</SYMBOL><BYTE>02</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2400</SYMBOL><BYTE>01</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>6000</SYMBOL><BYTE>00</BYTE></ELEMENT>
</BW-REPLY>
```
Fldigi can parse and decode messages returned from the transceiver that define 4 aspects of the transceiver operation:

<table>
<thead>
<tr>
<th>OK</th>
<th>data accepted by the transceiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAD</td>
<td>data rejected by the transceiver</td>
</tr>
<tr>
<td>MODE</td>
<td>current operating mode of the transceiver</td>
</tr>
<tr>
<td>BW</td>
<td>current bandwidth setting of the transceiver</td>
</tr>
<tr>
<td>FREQ</td>
<td>frequency of the active vfo (might be either A or B for example)</td>
</tr>
</tbody>
</table>

These are all contained within multiple `<REPLY></REPLY>` tag pairs. This is an example of a fixed format message with no variable fields. It is the OK message sent back by the Icom-746 PRO:

```xml
<REPLY>
  <SYMBOL>OK</SYMBOL>
  <SIZE>6</SIZE>
  <BYTES>FE FE E0 66</BYTES>
  <BYTE>FB</BYTE>
  <BYTE>FD</BYTE>
</REPLY>
```

The `<SYMBOL></SYMBOL>` pair and the command definition are mandatory. The `<SIZE></SIZE>` field is mandatory and specifies the number of bytes contained in this reply. The above definition could also have been coded as:

```xml
<REPLY>
  <SYMBOL>OK</SYMBOL>
  <SIZE>6</SIZE>
  <BYTES>FE FE E0 66 FB FD</BYTES>
</REPLY>
```

When the reply contains variable data it is specified in a contained tag pair `<DATA></DATA>`. This data field contains specifiers that describe the kind and size of the data. The `<DTYPE></DTYPE>` tag pair may be one of:

- BINARY
- DECIMAL

This is an example for the reply to a mode query that is returned by the Icom-746 PRO:

```xml
<REPLY>
  <SYMBOL>MODE</SYMBOL> specifies the response name
  <SIZE>8</SIZE> 8 bytes of data returned
  <BYTES>FE FE E0 66</BYTES> 4 bytes of preamble
  <BYTE>04</BYTE> 1 additional byte for preamble
</REPLY>
```
<DATA>
<DTYPE>BINARY</DTYPE> binary data field of 1 byte
<SIZE>1</SIZE>
</DATA>
<FILL>1</FILL> a variable field (data) not used
<BYTE>FD</BYTE> 1 byte postamble
</REPLY>

Fldigi rigcat will check for both the preamble and postamble to insure that a valid reply has been sent by the transceiver.
Waterfall Configuration-Display

The waterfall palette or color scheme can be altered to suit your personal tastes and visual needs. When fldigi is first started it creates a wide range of pre-built palettes in the $HOME/.fldigi folder. The "Load" button gives you access to those palettes. You may change any palette by clicking on the various color buttons beneath the palette sample. A color picker opens for you to select the color by various means including specifying the RGB values. If you create a palette that suits you better than any of the prebuilt ones you can "Save" the palette.

The waterfall cursor is a set of markers on the frequency scale that are spaced a signal bandwidth apart. You can add a pair of lines that drop down from those two markers for the full height of the waterfall by selecting Cursor BW. You can add a center line cursor to this pair of BW line by selecting Cursor Center line. You can also add a set of BW lines that straddle the received signal tracking point by selecting Bandwidth tracks. All three of these options are color selectable. Click on the colored button below the check box and a color selection dialog will open.

The frequency scale defaults to RF frequency. You can select to show audio frequencies. You can **monitor the transmitted audio waveform** and also set the level of the monitored signal. This IS NOT your final transmitted signal!
Waterfall Configuration-FFT Processing

**Lower limit** and **Upper limit** control the displayed frequencies in Hz.

Fldigi's waterfall FFT has a bin size of 1 Hz. With an FFT of 8192 and a sampling rate of 8000 it takes almost a second to accumulate enough data to perform the full FFT. A waterfall that dropped at one scan line per second would be hard on the viewer, so fldigi uses a first-in-first-out (FIFO) 8192 byte buffer for the FFT data. 512 byte audio blocks move through the buffer with each successive read of the sound card. The full buffer of 8192 samples is used to compute the FFT. That means that data in the FFT can have a latency of 8 scans. This provides excellent frequency resolution but poor time resolution (the vertical waterfall appearance). The latency control allows you to select the number of 512 byte blocks that are used for the FFT. The default **FFT latency** is set to 4. You should be able to achieve a reasonable compromise between the time and frequency domain resolutions. **FFT averaging** can be used to smooth the waterfall display in the frequency domain.

The **FFT Prefilter** or window function is used to reduce aliasing in the FFT computation. The default prefilter for the Fast Fourier Transform associated with the waterfall is **Blackman**. You can try the other windowing filter. Under some conditions you might prefer one of those. The Blackman window has proven best for my setup.

Note: Changes in the **Waterfall height in pixels** requires the program to be restarted.
Waterfall Configuration-Mouse

The mouse behavior in the waterfall panel can be controlled to suit your particular operating style. You might want to **replay the saved audio history** every time you either left click to select or right click to preview a particular signal. You can move the transceiver frequency in increments of 100 Hz by dragging the waterfall scale. **You can also choose to insert a line of text into the Rx panel each time you left click a waterfall signal.** The text can include expandable macro tags.

The **mouse wheel behavior** can also be tailored to your liking:
- None - no mouse wheel activity in waterfall panel
- AFC range or BW - adjust the AFC range/BW up/down
- Squelch level - adjust the squelch level up/down
- Signal search - search up / down for next signal in current mode
- Modem carrier - adjust the audio tracking point +/- Hz increments
- Modem - select modem type from a full rotary of available modems
- Scroll - move the waterfall left/right in 100 Hz increments (for 2x, 4x expanded waterfall view)
Modem Configurations

Modem Configuration-CW

Fldigi can send and receive Morse code from 5 wpm to 200 wpm. The operating controls for CW are found on the Config/CW tab. You can open that tab by selecting the "Configure/Modems" menu item and the clicking on the Modems/CW tab. You can also open up the CW tab by first selecting CW as the operating mode and then clicking on the left-most item "CW" on the status bar at the bottom of the fldigi main window. During operation the Rx and Tx WPM settings are annunciated on the status bar in the two boxes next to the mode indicator.

The CW decoder has a DSP filter than is implemented with a sin(x)/x impulse response. This is a very steep sided filter that is centered on the received signal in the audio passband. You can control the bandwidth of this filter using the BW control.

Fldigi can track the incoming signal. Enable Rx WPM tracking by enabling the check box "Enable Tx Trkg". The tracking range (+/- Hz around the TxWPM setting) can be set using the "Rx Trkg Rng" control.

The RxWPM controls are indicators and are not used for setting the operation of the CW decoder.

The TxWPM sliding controller is used to set the transmit WPM. To make the setting easier two additional controls are provided. "Lower" sets the lower limit of the slider and "Upper" sets the upper limit of the slider. The resolution of the TxWPM slider is 1 WPM. The Lower/Upper controls are in in 5 WPM increments.
The transmit encoder settings for WPM can also be adjusted with three hot keys:

- Numeric keypad "+" increases the TxWPM by 1
- Numeric keypad "-" decreases the TxWPM by 1
- Numeric keypad "*" toggles between the selected TxWPM and a default WPM

The "Default" control on the CW tab sets that default value. As shown above the TxWPM is 30 and the default is 18. If during a QSO you needed to slow down to give the other op a better chance to copy what you are sending, just hit the "*" on the numeric keypad and the cw code will immediately switch to sending cw at the set default value (18 wpm in this example). Press the "*" again to return to back to the cw speed that you were previously using.

### Timing

Fldigi generates CW by inserting a keyed tone at the current waterfall audio frequency. The transceiver should be operated in either USB (preferred) or LSB mode. The CW signal is completely generated in the software so it is possible to control many aspects of the CW signal. The actual transmitted signal will be at the USB carrier + the audio frequency, or the LSB carrier - the audio frequency. If fldigi is tracking and receiving a CW signal on the waterfall your transmitted signal will be exactly on the frequency of the other operator. The CW generated this way has a nearly ideal attack and decay time, controlled by the software modem. But ... a caveat ... your transmitter must never be overdriven and it should have excellent opposite sideband suppression. Overdriving the transmitter can cause multiple audio signals within the SSB passband, and cause unwanted interference to other ops. The same is true for a poorly designed or adjusted transmitter with bad sideband suppression. I recommend having a trusted and knowledgable operator assist you when first trying A2 CW. Have them carefully look for evidence of your signal above and below your primary signal (by at least +/- 3 Khz). If there is no evidence of extra signals then your are set to go. If there is you might want to have the transceiver adjusted for sideband suppression, or check to be sure you are not over driving the audio.
• Wt % control sets the weight of the CW. Normal CW is at 50% weight, ie: a dot is equal to the interval between dots or between code elements. It has a range of 20 to 80 percent.

• Dash/Dot controls the relative weight between a dash and a dot. The standard for CW is 3 to 1. The dash is 3 times the length of a dot. Some operators prefer the sound of either a heavier or lighter sounding CW. This control can be adjusted from 2.5 to 4.0 in 0.1 increments.

• Edge shape provides two leading/trailing edge shapes (1) Hanning, or raised cosine, and (2) Blackman a modified raised cosine with a steeper attack and decay. Both of these edge shapes give a more narrow bandwidth CW signal than the traditional exponential waveform. They are very easy to listen to even at speeds exceeding 100 wpm.

• The Edge control sets the rise and fall times of the CW waveform. It can be set anywhere from 0.0 to 15.0 milliseconds in 0.1 millisecond increments. DO NOT operate A2 CW with the control set below 4 msec. This is the control that sets the effective bandwidth and sound of your CW. If the edge is too steep you will have a clicky signal and be the bane of the CW bands. The purpose of being able to set the edge to 0.0 or a very quick rise/fall time is explained below. A good setting for nice sounding CW at 40 WPM and below is 4 to 6 milliseconds.

• Edge decreases pulse width, when checked will give a slightly narrower dot length as the edge timing is increased. This is useful when operating QSK and listening between the character elements.

This is what the A2 signal should look like with various settings of weight, Dash/Dot and Edge. The audio frequency is 400 Hz and the TxWPM is 100 WPM.

![Graph of A2 signal with various settings of weight, Dash/Dot and Edge.](image-url)
Changing the weight, dash/dot or edge of the waveform does not change the WPM at which the code is generated. When a conflict occurs between the various settings WPM takes first priority, and Edge second. In the above examples, the Edge setting could not exceed 12 msec even if the control were set higher than 12.0. The figures were generated by capturing the output data being sent to the sound card and then formatting it using Gnumeric. An oscilloscope photo of the signal is virtually identical.

The setting for inter-character and inter-word spacings are fixed at 3 and 7 respectively. The 3 is achieved by sending a silent period of 1 dot (element) length at the beginning of each character and 2 at the end of each character (shown in the figures). This silent period is sufficient for most transceivers to respond to the PTT signal which occurs at the beginning of the transmission so that the first dit or dash is not lost in transmission. QRQ (high speed CW operation)

You may wonder why fldigi can go as high as 200 WPM. It's hard to believe but there are CW operators who can decode 100+ WPM in their head. These operators also usually operate QSK (full breakin). A2 CW and PTT operation and QRQ/QSK are not a natural mix. But fldigi can be used for this type of operation if an external keyer is used. For that purpose the A2 Tx output from fldigi is full wave rectified and detected to create a keyline control. The outboard conversion from A2 to keyline requires a nearly square wave pulse output of audio at the CW keying rate. Setting the Edge control to 0.0 and then the audio frequency to about 1000 Hz provides the needed signal to effect this type of keyline control.

If you are operating QSK with a separate receiver / transmitter you can very quickly stop your transmit signal with the TAB key. In the CW mode only the TAB key causes the program to skip over the remaining text in the transmit text buffer. The text that is skipped will be color coded blue. The program remains in the transmit mode (PTT enabled), but since the buffer is now empty no A2 CW signal is generated. Code transmission will then restart with the very next keyboard closure of a valid CW character. The Escape and Pause/Break keys still can be used to respectively abort and pause transmission.
QSK
You might ask why fldigi doesn't simply provide a keyline output on one of the parallel port pins or on RTS or DTR via a comm port. The answer is quite simple. Linux is a multi-tasking operating system and the interaction between the OS and the application causes the timing to be adversely effected. The driver implementation of the audio sub system must be responsive and so the OS gives that sub system a very high priority in its multi-tasking structure.

Many QSK operators use high speed diode antenna switching between receiver and antenna. fldigi generates a signal that can be used for that purpose. The left audio channel is always the AFCW signal. When selected the right audio channel can be configured to generate a square wave signal that begins earlier and ends later than each of the CW elements. The square wave signal can be rectified and filtered to provide the diode switching signal for the Rx/Tx antenna switching.

The right audio channel QSK signal is selected by checking the box and then adjusting the pre and post timing in millisecond increments. Additional information and a schematic diagram of a QSK keying circuit is described in CW Keying. Setting up a QSK device can be quite difficult. Fldigi helps to ease the adjustment by generating a continuous series of characters. This allows a dual trace scope to be properly synched while making the adjustments to both the software and the associated QSK hardware. You enable continuous characters by selecting the checkbox, and then enabling the T/R button for transmit. The repeated character can be change on the fly with the pick control. It can be one of either E, I, S, T, M, O or V.

PROSIGNS
You can assign keyboard characters to be used for Morse prosigns. The available characters are: ~ % & + = {} <> []

The default assignments are shown above. You can also elect to send and receive the KN prosign as an open parenthesis '('.

This is commonly used on MARS CW operations.

See Operating CW for additional information.
Modem Configuration-DominoEX

Enter the secondary text. This text will be sent during periods when your keyboard is inactive (between letters for slow typists). The default for this text will be your callsign when you have entered that in the Operator configuration tab.

Set the BW factor for the decoding prefilter. 2.0 should be adequate unless you are experiencing nearby continuous wave interference (CWI). You can enable and disable the prefilter with the checkbox. Please note that the filter requires additional cpu cycles. Older and slower cpu models might give better decoding with the filter disabled.

Fldigi can send and receive FEC in accordance with the DomEX-FEC specification for MultiPsk. This type of FEC is achieved by some loss of non printing characters in the primary character set. It is therefore not usable as an FEC mode for ARQ (automatic repeat request) transmissions.

The DominoEX decoder can detect the presence of CWI within the passband set by the BW factor. Increasing the CWI threshold increases the sensitivity to such interference. When the interference is detected the associated data is culled using a technique called puncturing.
Modem Configuration-Olivia and Contestia

<table>
<thead>
<tr>
<th>Operator</th>
<th>UI</th>
<th>Waterfall</th>
<th>Modems</th>
<th>Rig</th>
<th>Audio</th>
<th>ID</th>
<th>Misc</th>
<th>Callsign DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CW</td>
<td></td>
<td>DominoEX</td>
<td>Fedhell</td>
<td>MT-63</td>
<td>Olivia</td>
<td>PSK</td>
<td>RTTY</td>
<td>Thor</td>
</tr>
</tbody>
</table>

- **250 Bandwidth**
- **8 Tones**

Receive synchronizer:
- **8 Tune margin (tone frequency spacing)**
- **4 Integration period (FEC blocks)**

- **8-bit extended characters**

Configuration of Contestia is similar to Olivia as Contestia is a derivative of Oliva. Olivia is a family of MFSK modes with a high redundancy Forward Error Correction system similar to MT63. The family is very large, with 40 or more different options, which can make it very difficult to work out which is which. The mode works well on poor HF paths and has good sensitivity. There are three popular modes, which have 8-FSK, 16-FSK and 32-FSK, thus having three, four or five bits per symbol. These three modes can be selected without additional configuration. The tone frequency spacing and integration period should always be left at 8 and 4 respectively unless you are experimenting with another station running an Olivia modem that can be changed. These must always be the same at both ends of the Olivia QSO. The modes have two serious shortcomings - excessive bandwidth combined with slow typing rate, and excessive latency which is the apparent typing delay caused by the integration period.
Modem Configuration-PSK

![Screenshot of the Fldigi configuration interface for PSK mode]

You should set the acquisition search range for waterfall left click action. As you adjust this control you will see the red mode width change on the waterfall scale. You can also adjust this value by pointing the mouse to the waterfall. Hold down the Control key and rotate the mouse wheel. The search routine which finds the psk signal operates on a s/n threshold detector as well as recognizing the Psk phase modulation. You can adjust the acquisition signal to ratio threshold for the search routine.

The PSK decoder estimates the signal to noise ratio, S/N, and the intermodulation distortion, IMD, of the received signal. This measurement is valid during periods when the other station is transmitting the idle signal. The estimates are displayed on the status bar. You can control how these values are displayed; **clear** or **dim** after NN seconds. Setting the seconds to 0 disables the clear/dim action.

Fldigi has a multi channel browser than can display simultaneous reception of up to 30 psk signals.
Modem Configuration-RTTY / FSK

Fldigi operates RTTY using AFSK and the transceiver set to USB. The RTTY signal can be transmitted anywhere within the USB passband of the transceiver.

You can select from various Shifts, Bauds, Bits, Parity and Stop Bits for both AFSK and FSK keying of the transmitter. You can elect to have fldigi automatically insert a CFLF when it reaches character 72 on a line. You can also have it insert a CR-CR-LF sequence instead of the standard CR-LF sequence. This is very useful if you are communicating with someone using a hardware TTY printer. The extra carriage return will give the physical device time to move to the left margin before new characters arrive.

The RTTY decoder maintains an internal AFC system for tracking the desired signal. Depending on operating conditions you may need to adjust the action of the AFC loop. Select from the Slow, Normal or Fast AFC loop. You can also disable AFC with the AFC button on the main panel.

The Digiscope display can be defaulted to the X-scope or the Signal scope. The X-scope is similar to older hardware scopes that show the Mark/Space channel signals as quadrature signals.

The DSP bandpass filter precedes the decoder. You can adjust the bandwidth with the Receive filter bandwidth control. The bandwidth is reflected on the waterfall tracking indicator.

PseudoFSK selection generates an additional audio signal on the right channel. This signal is a burst tone at the FSK keying rate. You can full wave rectify and filter the signal so that it can be used as the FSK keyline signal to a rig that supports FSK transmissions.
Using the FLdigi Pseudo FSK (Rt. Channel) function to key a transmitter

Select the PseudoFSK check boxes.
FLdigi is now ready to generate a 1000 hertz tone burst signal on the right channel of the stereo audio out of your sound card. This tone burst is on when the RTTY bit is on and off when the RTTY bit is off. The left channel will be the normal AFSK signal.

The following circuit may be used to take the FLdigi PSEUDO-FSK signal from the right channel of your SOUND CARD to key your transmitter's FSK input line. You may find it necessary to invert the sense of the keying signal.

NOTE: L1 Radio Shack has two items that may be used for this isolation transformer.
Catalog # 270-054, and
Catalog # 273-1374

Attach an audio cable from the Rt. Channel out of the your computer's SOUND CARD to the input of this FSK INTERFACE CIRCUIT (input of L1).
Attach another cable from the output of this circuit to your Rig's Keying FSK Jack.
Every PSEUDO-FSK tone that is generated by FLdigi is rectified by this FULL WAVE VOLTAGE DOUBLER circuit. The resultant voltage turns the Q1 transistor on and "grounds" the collector.
Modem Configuration-Thor

The decoder can detect and defeat a modest amount of CWI that is within the BW set by the BW factor. Increasing the CWI threshold increases the sensitivity for this correction. The offending tones are punctured thereby rendering them null to the Viterbi decoder.

Enter the secondary text. This text will be sent during periods when your keyboard is inactive (between letters for slow typists). The default for this text will be your callsign when you have entered that in the Operator configuration tab.

Set the BW factor for the decoding prefilter. 2.0 should be adequate unless you are experiencing nearby continuous wave interference (CWI). You can enable and disable the prefilter with the checkbox. Please note that the filter requires additional cpu cycles. Older and slower cpu models might give better decoding with the filter disabled.

The DominoEX decoder can detect the presence of CWI within the passband set by the BW factor. Increasing the CWI threshold increases the sensitivity to such interference. When the interference is detected the associated data is culled using a technique called puncturing.

Thor has been specifically designed to be used with ARQ text transmissions. It is also an easy to use keyboard chat mode. Thor operations are described in Operating Thor.
Id Configuration

Fldigi offers several ways to identify the operator or mode that is being used. This is particularly useful when using a hard to recognize mode such as Thor, Olivia or MT63.

Video Text

Transmitted video text will appear as a sequence of characters on the waterfall. The text can be a brief mode identifier or some user specified text. You can use a small font that always appears as a 2 character wide sequence or a larger font that can be 1 to 4 characters wide. You should be aware that the video signal is a constant energy signal and the content will be spread across multiple characters. The highest s/n at the receiving end will be for 1 character wide video. Small font at 2 character width is next in s/n performance followed by 2 character large font etc. You can select which modes will include the video text preamble. You can limit the horizontal (frequency width) of the video signal in one of several inclusive ways.

- Number of characters per row of text
- Constrain to be less than or equal to 500 Hz
- Constrain to be within the bandwidth limits of the mode in use

Fldigi uses abbreviated acronyms for the mode and it's characteristics when you are transmitting the mode ID using a video text. Here are two examples, one in small and the other in large font.

Notice that Olivia 16-500 is abbreviated to OL-16/500 and that the number of characters is limited to 8 per row. You might want to use the large characters by default if you routinely have QSO's with operators using older digital mode programs or one whose waterfall visual is not on a par with fldigi's.
Cw postamble
You can transmit your callsign in CW as a postamble to all modes except of CW (a bit redundant to do that). You can select which modes will include the cw postamble.

Reed Solomon Identifier
RSid, Reed Solomon Identifier, is a special transmission designed by Patrick Lindecker, F6CTE, for the modem program MultiPsk. It has been adapted to other modem programs. Fldigi's implementation is compatible with the MultiPsk RSid, but provides a slight variation. You can transmit RSID at both the beginning and end of a transmission. The detection of RSid normally only occurs in the near vicinity of the current waterfall tracking point. This cuts down on extraneous RSid detections when the band is crowded and several RSid signals might be present. If you want fldigi to search the entire waterfall for RSid signals you can do so by enabling the "Detector searches entire passband". You start the search for a signal based on RS Id by using the main panel switch. The RSID detector is a separate decoder that operates in parallel with all other modem decoders. If you select the "Mark previous frequency and mode" a restore link will be inserted into the Rx text upon detecting an RSID signal. Clicking on this link restores the previous frequency and mode of operation. You elect to disable the RSID upon first detection. You also have the option of just receiving notification when an RSID signal is detected. The notification occurs with a pop-up message box.

You can select which modes will include the transmitted RS identifier, and which modes will react to a received and decoded RS identifier.

The mode to identifier relationships are selected by pressing the associated "modes" button.
Reed-Solomon Identification Description

Reed-Solomon Identification (RSID) of digital modes is a creation of Patrick Lindecker, F6CTE, and kindly released to the public domain. It is used in several digital mode programs. Patrick maintains the master list of code / mode assignments in order to maintain compatibility between these programs. He accepts requests for new modes if it can be shown that the mode has a high utility factor. The reason for not simply adding every mode in current use will be explained later in this document.

RSID allows the automatic identification of any digital transmission which has been assigned a unique code identifier. All RSID's are detected by fldigi, but not all are decoded. All detected codes are annunciated. On reception of a RS ID, two events occur: the mode used is detected and the central frequency of the RSID, which is also the central frequency of the identified mode, is determined with a precision of 2.7 Hz. This is sufficient to allow all current modes to begin accurate decoding. This is an excellent way to insure that signals like MFSK are properly tuned and decoded.

The RSID signal is transmitted in 1.4 sec and has a bandwidth of 172 Hz. Detection of the RSID signal is possible down to a Signal to Noise ratio of about -16 dB, so with a sensitivity equal or better than the majority of the digital modes (RTTY, PSK31...), except several modes as PSK10, PSKAM10, THROB, THROBX or JT65.

Note: consequently, it is possible to detect RSID and not be able to decode the ensuing data signal due to it being too weak a signal. fldigi allows the RSID signal to be sent at the beginning and the end of each transmission. The leading RSID is the normal position.

During reception fldigi can decode RSID signals within the entire audio spectrum. It can also be configured to limit the reception to a narrow bandwidth centered on the current audio subcarrier. Detection occurs as a background process and does not interfer with the normal signal decoding. False detection is possible, but statistically rare due to the use of a very strong autocorrelation function associated with the RSID codes.

At the present time, five programs offer RSID:

- PocketDigi, Vojtech, OK1IAK
- FDMDV, Cesco, HB9TLK
- DM780, Simon, HB9DRV
- fldigi, Dave, W1HKJ
- Multipsk, Patrick, F6CTE

RSID programs sources are available C++, from OK1IAK, W1HKJ and HB9DRV. They are also available from F6CTE in Delphi.

The master list of RSID code/mode combinations is maintained by Patrick, F6CTE, and changes made to the list in concurrence with the above list of software developers. This list is distributed on the DigitalRadio Yahoo group at each change.
Description of RSID encoding

Each mode corresponds to a number which is transformed in a particular Reed-Solomon sequence. This RS coding (RS (k = 4, t = 6)) is defined by the parameters:

- \( k = 4 \) (number of bits per symbol),
- \( n = 15 = 2^k - 1 \) (number of symbols by RS sequence) and
- \( t = 6 \) (maximum number of errors which could be theoretically corrected).

Each RS sequence is composed of 15 symbols of 4 bits, among which 3 (n – 2 x t) contain data. In other words, 12 bits (3 x 4) are available to define the mode number. Consequently, the number of possibilities would be equal to 4096 (2^12).

In addition, even if the maximum number of errors which could be fixed is equal to 6, it is limited to one correction so as to have a negligible probability of false detection. This is accomplished by increasing the Hamming distance between any random sequence and the selected sequences.

It has been proposed that two RSID could be sent sequentially or that two RSID could be also transmitted in juxtaposed frequencies. As false RS ID detection with part of one and part of the other (either in the time domain or the frequency domain) must be avoided, it is necessary to identify a subset of the 4096 RSID codes that are mutually independent or orthogonal. That is, two unique RSID sequences can't produce a valid but wrong RSID code. This subset is composed of only 272 unique values. Secondary and tertiary subsets can be identified which can produce false detections. These are only assigned for use during the development of a new modem type.

If every known digital signal were assigned a unique code the list of 272 would quickly be exhausted. It is for that reason that not every combination of baud rate, tone numbers, and bandwidth is currently supported for RTTY, MFSK, OLIVIA, CONTESTIA, DOMINO-EX and THOR. Some RSID codes exist for modes which are currently out of vogue. That is simply due to the development history of the RSID technique. Multipsk was the only digital modem program to support RSID for some time and Patrick merely assigned codes to all of the modes supported by his program at the time that RSID was introduced.

Each symbol is transmitted using MFSK modulation. There are 16 possibilities of frequencies separated by 11025 / 1024 = 10.766 Hz. Each symbol transmission being done on only one frequency for a duration equal to 1024 / 11025 x 1000 = 92.88 ms. The entire RSID sequence of 15 symbols is transmitted in 15 x 1024 / 11025 = 1393 ms. The RSID signal is transmitted at the maximum peak power of the associated modem signal.

Decoding can be accomplished by a brute force algorithm where all the possibilities are tested. The possible found solutions are sorted according to their distance (0 or 1) to a valid solution and according to a pseudo signal-to-noise ratio (peak power/average power). The final detected code will have the lowest distance with the largest pseudo signal-to-noise ratio.

Vojtech, OK1IAK, developed a hashing algorithm which reduces the detector processing load. He kindly made that code available to other developers. Both fldigi and DM780 use Vojtech's decoding algorithm.
For each semi-step of time (46.44 ms) and for each semi-step of frequency (5.38 Hz), the program attempts to detect a RSID extending for the last 1.393 seconds. So each second, about 8500 possible RSID (depending on the selected bandwidth) are tested (depending on the bandwidth).

The analysis is based on a Fast Fourier transform of 2048 points at 11025 samples/sec, regularly done at each semi-step of time (46.44 ms).

There are only two possible outputs from the detection algorithm:
- either the RSID identifier is not received because the signal is too weak,
- or it is received and it is correct

The probability of detecting a wrong RSID is almost nil.

List of assigned RSID codes by RSID code, mode, and whether supported by fldigi:

<table>
<thead>
<tr>
<th>#</th>
<th>RSID Code</th>
<th>Mode</th>
<th>fldigi</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>BPSK31</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>BPSK63</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>BPSK125</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>126</td>
<td>BPSK250</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>173</td>
<td>BPSK500</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>QPSK31</td>
<td>yes</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>QPSK63</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>QPSK125</td>
<td>yes</td>
</tr>
<tr>
<td>9</td>
<td>127</td>
<td>QPSK250</td>
<td>yes</td>
</tr>
<tr>
<td>10</td>
<td>183</td>
<td>PSK125R</td>
<td>yes</td>
</tr>
<tr>
<td>11</td>
<td>186</td>
<td>PSK250R</td>
<td>yes</td>
</tr>
<tr>
<td>12</td>
<td>187</td>
<td>PSK500R</td>
<td>yes</td>
</tr>
<tr>
<td>13</td>
<td>7</td>
<td>PSKFEC31</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>PSK10</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>MT63-500 long interleave</td>
<td>yes</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>MT63-500 short interleave</td>
<td>yes</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>MT63-500 very short interleave</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>MT63-1000 long interleave</td>
<td>yes</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>MT63-1000 short interleave</td>
<td>yes</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>MT63-1000 very short interleave</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>MT63-2000 long interleave</td>
<td>yes</td>
</tr>
<tr>
<td>22</td>
<td>17</td>
<td>MT63-2000 short interleave</td>
<td>yes</td>
</tr>
<tr>
<td>23</td>
<td>18</td>
<td>MT63-2000 very short interleave</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>19</td>
<td>PSKAM10</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>RSID Code</td>
<td>Mode</td>
<td>fldigi</td>
</tr>
<tr>
<td>---</td>
<td>-----------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>25</td>
<td>20</td>
<td>PSKAM31</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>21</td>
<td>PSKAM50</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>22</td>
<td>PSK63F</td>
<td>yes</td>
</tr>
<tr>
<td>28</td>
<td>23</td>
<td>PSK220F</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>24</td>
<td>CHIP64</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>25</td>
<td>CHIP128</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>26</td>
<td>CW</td>
<td>yes</td>
</tr>
<tr>
<td>32</td>
<td>27</td>
<td>CCW_OOK_12</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>28</td>
<td>CCW_OOK_24</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>29</td>
<td>CCW_OOK_48</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>30</td>
<td>CCW_FSK_12</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>31</td>
<td>CCW_FSK_24</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>33</td>
<td>CCW_FSK_48</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>34</td>
<td>PACTOR1_FEC</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>113</td>
<td>PACKET_110</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>35</td>
<td>PACKET_300</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>36</td>
<td>PACKET_1200</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>37</td>
<td>RTTY_ASCII_7</td>
<td>yes</td>
</tr>
<tr>
<td>43</td>
<td>38</td>
<td>RTTY_ASCII_8</td>
<td>yes</td>
</tr>
<tr>
<td>44</td>
<td>39</td>
<td>RTTY_45</td>
<td>yes</td>
</tr>
<tr>
<td>45</td>
<td>40</td>
<td>RTTY_50</td>
<td>yes</td>
</tr>
<tr>
<td>46</td>
<td>41</td>
<td>RTTY_75</td>
<td>yes</td>
</tr>
<tr>
<td>47</td>
<td>42</td>
<td>AMTOR_FEC</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>43</td>
<td>THROB_1</td>
<td>yes</td>
</tr>
<tr>
<td>49</td>
<td>44</td>
<td>THROB_2</td>
<td>yes</td>
</tr>
<tr>
<td>50</td>
<td>45</td>
<td>THROB_4</td>
<td>yes</td>
</tr>
<tr>
<td>51</td>
<td>46</td>
<td>THROBX_1</td>
<td>yes</td>
</tr>
<tr>
<td>52</td>
<td>47</td>
<td>THROBX_2</td>
<td>yes</td>
</tr>
<tr>
<td>53</td>
<td>146</td>
<td>THROBX_4</td>
<td>yes</td>
</tr>
<tr>
<td>54</td>
<td>49</td>
<td>CONTESTIA 8-250</td>
<td>yes</td>
</tr>
<tr>
<td>55</td>
<td>50</td>
<td>CONTESTIA 16-500</td>
<td>yes</td>
</tr>
<tr>
<td>56</td>
<td>51</td>
<td>CONTESTIA 32-1000</td>
<td>yes</td>
</tr>
<tr>
<td>#</td>
<td>RSID Code</td>
<td>Mode</td>
<td>fldigi</td>
</tr>
<tr>
<td>----</td>
<td>----------</td>
<td>----------------</td>
<td>--------</td>
</tr>
<tr>
<td>57</td>
<td>52</td>
<td>CONTESTIA 8-500</td>
<td>yes</td>
</tr>
<tr>
<td>58</td>
<td>53</td>
<td>CONTESTIA 16-1000</td>
<td>yes</td>
</tr>
<tr>
<td>59</td>
<td>54</td>
<td>CONTESTIA 4-500</td>
<td>yes</td>
</tr>
<tr>
<td>60</td>
<td>55</td>
<td>CONTESTIA 4-250</td>
<td>yes</td>
</tr>
<tr>
<td>61</td>
<td>117</td>
<td>CONTESTIA 8-1000</td>
<td>yes</td>
</tr>
<tr>
<td>62</td>
<td>169</td>
<td>CONTESTIA 8-125</td>
<td>yes</td>
</tr>
<tr>
<td>63</td>
<td>56</td>
<td>VOICE</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>60</td>
<td>MFSK 8</td>
<td>yes</td>
</tr>
<tr>
<td>65</td>
<td>57</td>
<td>MFSK 16</td>
<td>yes</td>
</tr>
<tr>
<td>66</td>
<td>147</td>
<td>MFSK 32</td>
<td>yes</td>
</tr>
<tr>
<td>67</td>
<td>148</td>
<td>MFSK 11</td>
<td>yes</td>
</tr>
<tr>
<td>68</td>
<td>152</td>
<td>MFSK 22</td>
<td>yes</td>
</tr>
<tr>
<td>69</td>
<td>61</td>
<td>RTTYM 8-250</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>62</td>
<td>RTTYM 16-500</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>63</td>
<td>RTTYM 32-1000</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>65</td>
<td>RTTYM 8-500</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>66</td>
<td>RTTYM 16-1000</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>67</td>
<td>RTTYM 4-500</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>68</td>
<td>RTTYM 4-250</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>119</td>
<td>RTTYM 8-1000</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>170</td>
<td>RTTYM 8-125</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>69</td>
<td>OLIVIA 8-250</td>
<td>yes</td>
</tr>
<tr>
<td>79</td>
<td>70</td>
<td>OLIVIA 16-500</td>
<td>yes</td>
</tr>
<tr>
<td>80</td>
<td>71</td>
<td>OLIVIA 32-1000</td>
<td>yes</td>
</tr>
<tr>
<td>81</td>
<td>72</td>
<td>OLIVIA 8-500</td>
<td>yes</td>
</tr>
<tr>
<td>82</td>
<td>73</td>
<td>OLIVIA 16-1000</td>
<td>yes</td>
</tr>
<tr>
<td>83</td>
<td>74</td>
<td>OLIVIA 4-500</td>
<td>yes</td>
</tr>
<tr>
<td>84</td>
<td>75</td>
<td>OLIVIA 4-250</td>
<td>yes</td>
</tr>
<tr>
<td>85</td>
<td>116</td>
<td>OLIVIA 8-1000</td>
<td>yes</td>
</tr>
<tr>
<td>86</td>
<td>164</td>
<td>OLIVIA 8-125</td>
<td>yes</td>
</tr>
<tr>
<td>87</td>
<td>76</td>
<td>PAX</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>77</td>
<td>PAX2</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>RSID Code</td>
<td>Mode</td>
<td>fldigi</td>
</tr>
<tr>
<td>---</td>
<td>-----------</td>
<td>------------</td>
<td>--------</td>
</tr>
<tr>
<td>89</td>
<td>78</td>
<td>DOMINOF</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>79</td>
<td>FAX</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>81</td>
<td>SSTV</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>84</td>
<td>DOMINOEX 4</td>
<td>yes</td>
</tr>
<tr>
<td>93</td>
<td>85</td>
<td>DOMINOEX 5</td>
<td>yes</td>
</tr>
<tr>
<td>94</td>
<td>86</td>
<td>DOMINOEX 8</td>
<td>yes</td>
</tr>
<tr>
<td>95</td>
<td>87</td>
<td>DOMINOEX 11</td>
<td>yes</td>
</tr>
<tr>
<td>96</td>
<td>88</td>
<td>DOMINOEX 16</td>
<td>yes</td>
</tr>
<tr>
<td>97</td>
<td>90</td>
<td>DOMINOEX 22</td>
<td>yes</td>
</tr>
<tr>
<td>98</td>
<td>92</td>
<td>DOMINOEX 4 FEC</td>
<td>yes</td>
</tr>
<tr>
<td>99</td>
<td>93</td>
<td>DOMINOEX 5 FEC</td>
<td>yes</td>
</tr>
<tr>
<td>100</td>
<td>97</td>
<td>DOMINOEX 8 FEC</td>
<td>yes</td>
</tr>
<tr>
<td>101</td>
<td>98</td>
<td>DOMINOEX 11 FEC</td>
<td>yes</td>
</tr>
<tr>
<td>102</td>
<td>99</td>
<td>DOMINOEX 16 FEC</td>
<td>yes</td>
</tr>
<tr>
<td>103</td>
<td>101</td>
<td>DOMINOEX 22 FEC</td>
<td>yes</td>
</tr>
<tr>
<td>104</td>
<td>104</td>
<td>FELD HELL</td>
<td>yes</td>
</tr>
<tr>
<td>105</td>
<td>105</td>
<td>PSK HELL</td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>106</td>
<td>HELL 80</td>
<td>yes</td>
</tr>
<tr>
<td>107</td>
<td>107</td>
<td>FM_HELL 105</td>
<td>yes</td>
</tr>
<tr>
<td>108</td>
<td>108</td>
<td>FM_HELL 245</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>114</td>
<td>141A</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>123</td>
<td>DTMF</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>125</td>
<td>ALE400</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>131</td>
<td>FDMDV</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>132</td>
<td>JT65 A</td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>134</td>
<td>JT65 B</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>135</td>
<td>JT64 C</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>136</td>
<td>THOR 4</td>
<td>yes</td>
</tr>
<tr>
<td>117</td>
<td>137</td>
<td>THOR 8</td>
<td>yes</td>
</tr>
<tr>
<td>118</td>
<td>138</td>
<td>THOR 16</td>
<td>yes</td>
</tr>
<tr>
<td>119</td>
<td>139</td>
<td>THOR 5</td>
<td>yes</td>
</tr>
<tr>
<td>120</td>
<td>143</td>
<td>THOR 11</td>
<td>yes</td>
</tr>
<tr>
<td>#</td>
<td>RSID Code</td>
<td>Mode</td>
<td>fldigi</td>
</tr>
<tr>
<td>----</td>
<td>-----------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>121</td>
<td>145</td>
<td>THOR 22</td>
<td>yes</td>
</tr>
<tr>
<td>122</td>
<td>153</td>
<td>CALL_ID</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>155</td>
<td>PACKET PSK 1200</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>156</td>
<td>PACKET PSK 250</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>159</td>
<td>PACKET PSK 63</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>172</td>
<td>188 110A 8N1</td>
<td></td>
</tr>
</tbody>
</table>
Miscellaneous Configuration

The sweet spot is the audio frequency at which your transceiver provides the best filtering for a particular signal type. You can specify the value of the sweet spot for CW, RTTY and all others. You can also elect to have the audio cursor placed at the sweet spot when changing modes. The sweet spot is used for the QSY function.

Fldigi allows you to automatically participate in a spotting network maintained by Philip Gladstone. You can see what the web based reporter looks like by visiting this web site: http://pskreporter.info/pskmap?W1HKJ or by simply selecting the menu item "Help / Reception reports."

Fldigi will continuously scan for spotted callsigns in the decoded text and send reports in the background if you check the "Automat..." option.

Reports will also (or only) be sent when you log the QSO into the logbook.

If you have rig control enabled the reported rig frequency will also be sent to the spotting network. Do not change the Host and Port numbers unless these are changed by Philip.

You need to press the Initialize to begin reporting spot information. You will receive a warning message if you did not enter your antenna information on the Operator tab.
When fldigi is executed for the first time it does some tests to determine the performance factor for your central processor unit. If it determines that the cpu is below a critical speed it tries to compensate by modifying some of its timing and algorithms. If you are using a "slow" cpu the "Slow cpu" check box will be enabled. You can also manually check this box if you find that fldigi is not performing well on some of the more esoteric modes such as PSK250, MFSK32, etc.

Fldigi can perform automatic capture of the Rx text stream. The simplest is to simply capture all incoming text to a file. Select this from the lower of the two frames. The Rx file is named "textout.txt" and is written to the directory as shown above. The file can be used to review an execution session, or it can be accessed by an external program. For example it could be parsed to provide a text to speech conversion.

The NBEMS suite of programs, fldigi, flarq, flwrap and flmsg provide the emergency operator with a set of tools to assist in the transfer of data files over HF and VHF radio.

The reception of a flwrap and flmsg files can be automated by selecting the "Enable detection & extraction" option. The wrap program can then be used to test for validity and data extraction at some later time. fldigi can recognize flmsg data files and automatically open the flmsg program with the newly received data stream. It can also transfer the data stream to flmsg and instruct flmsg to save the data file, unwrap and decode it, display the data in a fully formatted html page and then exit. Pressing "Locate flmsg" performs differently on the different OS that are supported.

- Linux - a file finder is opened to the /usr/local/bin/ folder. Select the flmsg executable and the entry box is correctly populated.
- Windows - a file finder is opened to the "C:\Program Files\" folder. Drill down to the most current flmsg folder and then select the flmsg.exe file. The entry box will be correctly populated.
- OS X - a file browser is opened to the "Applications" folder. Locate the flmsg icon, and right click on it. Select "Show Package Contents". Double click "Contents". Double click on "MacOS". You will be viewing an icon labeled "flmsg". Drag and drop the icon on to the "flmsg:" entry box and the the value will be correctly entered.
## Callsign DB Configuration

<table>
<thead>
<tr>
<th>Operator</th>
<th>UI</th>
<th>Waterfall</th>
<th>Modems</th>
<th>Rig</th>
<th>Audio</th>
<th>ID</th>
<th>Misc</th>
<th>Callsign DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>○ Not available</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>○ QRZ online via default Internet Browser</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>○ HamCall online via default Internet Browser</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CDROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>○ QRZ at:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Paid online subscription</th>
</tr>
</thead>
<tbody>
<tr>
<td>○ QRZ.com</td>
</tr>
<tr>
<td>○ Hamcall.net</td>
</tr>
</tbody>
</table>

Fldigi will open a web browser to either QRZ.com or Hamcall.net with the contents of the QSO Call field used as a query string to the on line service. You may find that your default browser needs to be triggered twice on the first such query. That behavior seems to be associated with IE7 but not IE6 for example.

If you have a CD with the QRZ database installed you can use that CD or its' stored contents on a hard drive. Simply specify where the CALLBK directory can be found and enable the QRZ radio button.

If you are a paid subscriber to either QRZ or Hamcall xml database service then you can specify that fldigi use that service for all Callsign data base queries.
Colors & Fonts

"System colors" are set by command line switches. The default is black on a white background.

From the Menu Configure/Defaults select the menu item Colors and Fonts and then select one of the following tabs. The text widget used for Rx, Tx and Event log displays has been improved to give better performance with proportional fonts. Fixed width fonts still give better performance and are not as demanding on the cpu. There are several very good fixed width fonts that include a slashed zero which are available for both Windows and Linux. If you are using a proportional font and find that the Rx text display gets unresponsive with large amounts of text then you should change to a fixed width font. Do a search on the internet for "Andale Mono". It is an excellent font for this use.

Freq Display

The rig control panel uses a special button for each digit the represents the transceiver frequency. The buttons are responsive to mouse clicks on the upper and lower half with corresponding changes it that unit's value. Unit value is also controlled by the mouse wheel when the cursor is over a particular digit. Select the background and foreground colors to please your overall color scheme and for best visual acuity. The System colors are the same ones that are used by all input and output text controls.

Func keys

You can color code the macro (function key) buttons in groups of 4, F1-F4, F5-F8, and F9-F12. The background color for each group is adjusted by clicking the respective Bkgnd button. The text color for the button labels is adjusted by clicking on the Label Txt button. The colors will change on these buttons and also on the main dialog as you make these adjustments. The Defaults button restores the colors as shown in this view.
Text Ctrls

The initial color, font and font-size for the Rx and Tx panel are the default values. You can always return to these by pressing the Defaults button. The background color, font and font-size are independently selectable. The Rx panel displays text in one of 5 colors:

- normal received text - "Rx font" button
- transmitted text - XMIT button
- control characters - CTRL button
- skipped characters (Tx ON/OFF in Tx pane) - SKIP button
- quick view characters - ALTR button
- select text highlight - SEL button

Tab Colors

Adjust the color of all tabs to suit your personal taste.

Light Buttons Colors

- Adjust the "on" color of Spot, RxID, TxID, Tune, Lk, Rev, T/R and AFC button
- Adjust the "enabled" and "on" colors of the Sql button
Operating Controls & Displays

The button **WF** toggles the display between a waterfall, spectrum display and an oscilloscope. This button acts as a rotary. Left clicking moves the display selection in one direction and right clicking in the other direction. The three display modes are **WF** - waterfall, **FFT** - spectrum (Fast Fourier Transform) and **Sig** - oscilloscope time domain. Let the mouse cursor hover over any one of the controls and a small hint box will open to help you navigate the various controls.

The next two controls to the right of the audio frequency control are for the **receive signal processing**. The one that reads -10 is the max signal level for the waterfall/spectrum display. The one that reads 40 is for the range over which that control will display signals. Both of these are in dB. The default of -10 / 40 is a good starting point, but you need to adjust these for band conditions. You can see the impact of these controls most easily by putting the main display area in the spectrum mode. Changes in these controls will effect the waterfall instantly and for all past history displayed on the waterfall. You do not have to wait for new signal data to observe the effect.

The **scale control** (X1, X2, X4) expands or contracts the view into the fast fourier transform that is displayed on the waterfall or the FFT display. fldigi always computes the FFT to a 1 Hz resolution, and displays the results according to the scale control.

The next three controls are **positional controls** for the waterfall. The waterfall can display 4096 data points, where each one can be thought of as a spectral line at the equivalent Hertz. The ratio is actually 8000/8192 and is related to the ratio of sound card sampling rate to Fast Fourier Transform length. This ratio changes for some modems that require a sampling rate other than 8000 Hz. The left arrow key will shift the display to the right (displays a lower section of the spectrum). The right arrow key moves the display higher in frequency. These two buttons are repeating buttons. Hold them down and the display slews at about 20 shifts / sec. The center button with the two vertical block lines is a "center the signal" button. The current cursor (red signal cursor in the waterfall) will be centered in the display area. **NOTE:** these controls are only functional if the current waterfall or spectrum view is smaller than the full view available. This is usually the case when the X2 or X4 expansion is selected. But it also might be the case when the width of the main dialog is reduced so that the waterfall display does not extend over the entire available width.
Try moving the cursor around in the waterfall area. You will see a set of yellow cursor blocks that show the center point and bandwidth of the current operating mode (psk31 = 31.25 Hz for example). To capture a received signal just click near the signal and the AFC will perform a multi-step acquisition. This will be very fast and should not require additional operator intervention. **Casual tuning** You can take a look at any received signal on the waterfall by right-clicking and holding the mouse button on or near the signal. The modem will begin to decode that signal if it is in the currently selected mode. The text will be a unique color on the Rx text widget so that you can discern the difference between casual and normal tracking. Release the mouse button and the tracking returns to the previously selected normal tracking point.

**Audio History** Fldigi maintains a history buffer of the received audio. This buffer is approximately 2 minutes in duration. After tracking commences on a signal you can decode the audio history for that signal. The audio history is invoked by a Ctrl-Left click anywhere on the waterfall. You can also invoke the audio history for the casual tuning mode by pressing Ctrl-Right click on the waterfall.

The **Norm** button controls the speed of the waterfall drop. This is also a rotary type of button control. The speeds available are SLOW, NORM, FAST and PAUSE. The load on the cpu will be directly proportional to this selection. If your cpu is slow you might want to select the SLOW or PAUSE option for the waterfall.

![Waterfall Controls](image)

The next control is your transceiver audio frequency. In the display above you can see that the audio signal is 1000 Hz. The arrow key pairs move up/down in cycles and tens of cycles. You can fine tune the receive point using this control.

The **QSY** button is very specific to rigs interfaced with either hamlib or the memory mapped i/o. Each rig has a sweet spot associated with its bandwidth controller. For the Argonaut V this is 1100 Hz. For the the Kachina it is 1000 Hz. As the transceivers bandwidth is changed the changes occur centered at this frequency. So .... let's say that I just started copying a rare dx at 1758 Hz and I wanted to put the signal at the sweet spot so I could easily narrow the receiver bandwidth. Click on the signal on the waterfall. Let the AFC capture and then press the QSY button. The tranceiver frequency will be shifted and the fldigi audio tracking point shifted in unison such that the signal is now at the receivers sweet spot. Very fast and very convenient! If you do not have hamlib enabled for your transceiver this button will be dimmed and not activated.

The **Store** button allows you to store, recall and manage mode/frequency pairs. If you want to save the current mode and frequency simply left click the button. A right click will enable a popup menu from which you can select a previously stored set. You can quickly move between modes and audio sub carrier using this technique. A shift-left click will clear the memory. When the popup menu is visible you left click on an entry to select it. You can shift-left click on an entry to delete that single entry.
The **T/R** button should be self-explanatory. It's your transmit/receive button. Action is immediate, so if you were transmitting some text and hit the button the PTT is disabled, the transmit text area cleared and the program returned to receive mode. The T/R button is a "lighted button" that shows RED when transmitting. All other lighted buttons show YELLOW when they are in the active state.

The **Lck** button locks the transmit audio frequency to its present value. You can then continue to QSY around your transmit position. I have used this to reply to a DX station that wanted a +500 Hz response. The DX was at 690 Hz audio, and wanted a response at +500. I moved the display cursor (or the audio frequency control) to 1190 Hz. Hit the Lck button and then went back to 690 with the waterfall cursor. Now the program is receiving on 690 Hz and transmitting on 1190 Hz. Caught him on the first try. Use this button also as a **Master Station** control. Not all rigs are equal in their VFO performance. Some exhibit a shift between receive and transmit. If this occurs then the stations find themselves chasing each other with every t/r exchange. Locking your transmit frequency with this control will inhibit that from happening. Be sure to disable the control when that QSO is over or **you may forget and transmit over top of another QSO!**

If the "Lck" is enabled the TX frequency does not follow the AFC action applied to the RX frequency. For transceivers which are either hamlib or memmap enabled, if the "Qsy" button is pressed BOTH the RX and TX frequencies are changed to synchronize to where the RX was positioned. Perhaps some numbers will help to make that a little clearer.

<table>
<thead>
<tr>
<th>&quot;Lck&quot;</th>
<th>Before &quot;Qsy&quot;</th>
<th>After &quot;Qsy&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RX</td>
<td>TX</td>
</tr>
<tr>
<td>OFF</td>
<td>1002 / 7071.002</td>
<td>1002 / 7071.002</td>
</tr>
<tr>
<td>ON</td>
<td>1002 / 7071.002</td>
<td>1000 / 7071.000</td>
</tr>
<tr>
<td>ON</td>
<td>1000 / 7071.000</td>
<td>1800 / 7071.800</td>
</tr>
</tbody>
</table>

With "Lck" off the TX audio frequency is always synchronized with the RX frequency. With "Lck" on the TX audio frequency is fixed with respect to the RX frequency UNLESS the "Qsy" button is pressed in which case it shifts to the RX frequency, the Transceiver VFO is shifted and both the RX and TX audio frequencies are shifted to put both into the middle of the transceiver passband. The TX continues to be locked, but at the new audio frequency. If the "Lck" is ON moving the cursor around will ONLY EFFECT the RX frequency and NOT the TX frequency.
The AFC (Automatic Frequency Control) button enables the tracking of the signal's frequency in RTTY, MSFK, PSK, THROB and WEFAX modes. SQL buttons enable or disable decoding of signals below the squelch level. The slider just above the AFC & SQL controls is the **squelch level control**. The bar indicator just above it is the equivalent of received signal level and relates on a 1:1 basis with the squelch level slider. The SQL button illuminates **YELLOW** when the SQL is selected, but the signal is below the squelch level. It illuminates **GREEN** when the the SQL is selected and the signal is above the squelch level.

The indicator just to the left of the AFC button is the **overload indicator**. It will be **GREEN** if your audio drive to sound card is satisfactory, **YELLOW** if the audio signal is marginally high and turn red when it is in overload. Back down the mixer control or the audio pad from the rig to computer. Fldigi will not perform well if the sound card is over driven. You will see ghost signals on the waterfall and the modem decoders will not work correctly.

Receive audio level should be adjusted so that the overload indicator does not illuminate red. When observing the received signals on the oscilloscope view you should expect that they do not exceed a peak-to-peak amplitude of 3/4 of the full display height.

**Mode Status Indicators**

The lower left corner of the main display (OLIVIA 16/500) in the view to the right is actually a button disguised as a status panel. This button responds to the mouse in several ways:

- Left Click - opens a **quick pick list** of associated modem types; you can switch to a new modem type from this popup menu
- Right Click - opens the **configuration dialog** at the tab associated with the current modem type
- Scroll Wheel - rotates forward and backwards through the various **modem types** in accordance with the modem menu hierarchy. Stop at the one you want and you are now in that mode

The next status indicator to the right provides **information relative to the current modem**, for PSK it indicates the received signal strength in dB. The third status indicator from the left provides additional information relative to the current modem, IMD for PSK measured in dB. Note that for PSK these values are only measured during periods when the PSK idle signal is being received.
Signal Browser

PSK and RTTY signals can be viewed in a multi-channel context. You can open an embedded or a separate dialog to access the browser. These browsers can help to locate a signal of interest on a busy band. The browsers can be visible at any time, but are only active when fldigi is in one of the PSK or RTTY modes. Open the external by clicking on the View/Signal browser menu item.

Both browsers can be viewed and used simultaneously. When you click on a valid waterfall signal the browser will highlight the appropriate channel (if it is also tracking that signal).
This panel can be resized horizontally by dragging the interface between the browser and the Rx panel to the left and right. You can drag all the way to the left to close the panel (or use the menu button). The drag to the right is limited to prevent sizing the Rx/Tx panels below their allowable limits. The browser can decode up to 30 simultaneous signals. As each signal is acquired within a 100 Hz channel width it is printed on the associated line. The user can elect to have each line annotated with:

- a channel number,
- the waterfall audio frequency,
- the transceiver HF frequency + waterfall, or
- no annotation.

Left click on a line of received text to move the waterfall frequency and Rx panel tracking to that signal. The contents of the line of text is transferred to the Rx text widget, and the main signal processing loop begins to track and decode that signal. Right click on a line of received text and that line is cleared and the channel reset for a new detection. You may have to do this occasionally if the detector for that channel has locked on to a sideband of a PSK signal. This is most likely to occur when the received signal as a marginal or bad IMD. Use the **Clear** button to clear and reset all of the channels. Channel signal detection and processing should restart immediately.
**Keyboard Operation**

The transmit buffer for fldigi is type ahead which means that you can be typing text while the program is sending an earlier part of your transmitted message. Newly entered text appears in black and text which has been transmitted is changed to red. You can backspace into the red area. When you do and the modem in use supports the BS character it will be sent to the receiving station. If you monitor PSK and MFSK signals you will often find operators backspacing over previously sent text. It's probably just as easy to just send XXX and retype that part of the message, but we have gotten used to word processors, email, etc. that allow us to send perfect (right) text, so we expect our digital modems to do the same. Let's see, what was that prosign often used in CW for ooops.

All of the alpha numeric keys perform as you would expect, entering text into the transmit buffer. There is one very important exception:

the caret "^" symbol. This is used in the macro expansion routine and also used by the transmit buffer evaluator. A ^r puts fldigi into receive mode. So you can enter the ^r (caret followed by the r) at the end of your transmit buffer and when the sent character cursor (red chars) gets to that point the program will clear the text and return to the receive mode.

You can load the transmit buffer with any ASCII Text file of your choice. Merely right click in the buffer window and select from the pop-up menu. You can also short cut to the ^r from this popup.

Many ops (including me) do not like to be tied to a mouse. The fldigi text widget supports some short cuts to make your life easier:

- **Pause/Break** - a transmit / receive - pause button.
  - if you are in the receive mode and press the Pause/Break key the program will switch to the transmit mode. It will begin transmitting characters at the next point in the transmit buffer following the red (previously sent text). If the buffer only contains unsent text, then it will begin at the first character in the buffer. If the buffer is empty, the program will switch to transmit mode and depending on the mode of operation will send idle characters or nothing at all until characters are entered into the buffer.
  - if you are in the transmit mode and press the Pause/Break key the program will switch to the receive mode. There may be a slight delay for some modes like MFSK, PSK and others that require you to send a postamble at the end of a transmission. The transmit text buffer stays intact, ready for the Pause/Break key to return you to the transmit mode.
  - Think of the Pause/Break key as a software break-in capability.

- **Esc**
  - Abort transmission. - immediately returns the program to receive, sending the required postamble for those modes requiring it. The transmit buffer is cleared of all text.
  - Triple press on Esc - terminates the current transmission without sending a postamble - The PANIC button.

- **Ctrl-R** will append the ^r (return to receive) at the end of the current text buffer.
- **Ctrl-T** will start transmitting if there is text in the transmit text window.
- **Alt/Meta-R** will perform the same function as the Pause/Break key
- **Tab** moves the cursor to the end of the transmitted text (which also pauses tx). A tab press at that position moves the cursor to the character following the last one transmitted. CW operation
is slightly different, see the help for CW.

- **Ctrl + three digits** will insert the ASCII character designated by that entry.

**Function Keys**

Keys F1 through F12 are used to invoke the **macro F1 - F12**. You can also just click on the macro key button associated with that function key. There are 4 sets of 12 macros. If you press the numbered button on the macro button bar the next set of macros are referenced by the F1 - F12. A right click on the numbered button provides a reverse rotation through the 4 sets of macro keys. The respective macro set can be made available by pressing the Alt-1, Alt-2, Alt-3 or Alt-4 key combination. Note that this is not Alt-F1 etc.

**Mouse & Keyboard Shortcuts**

Fldigi has a bewildering number of keyboard and mouse shortcuts, some of which may help make your particular style of operation more efficient. You do not need to know them all to make effective use of the program!

- **Main window**
  - **Text input fields**
    Most text fields use a combination of CUA (PC) and Unix-style keybindings. Text can be marked, copied, pasted, saved to a file as well as transfer to other main panel controls. A right click on any text control will open a context sensitive menu for that particular control.

The received/transmitted text widgets use CUA key bindings with some modifications:

- **RX text**
  This widget is read-only and ignores shortcuts that would modify its contents. See logbook for details on the Rx right click popup menu system.

- **TX text**
  The text that has already been sent is protected, but can be deleted one character at a time with the Backspace key. Right clicking on the Tx text panel opens the following popup menu:
Select:
Transmit put the program into the transmit operation
Receive during a transmit or tune, end the transmit and restore receive operation
Abort during a transmit, receive without waiting for the modem to finish sending
Send image for MFSK only, send an image using MFSKpic mode
Clear clear all of the text
Cut delete the marked text (by left click drag over text)
Copy copy the marked text to the clipboard
Paste the clipboard text to the current text insertion point
Insert file select a file from file browser to insert in text at insertion point
Word wrap turn word wrap on/off

The Tx panel is fully drag and drop aware. That means you can add a file to the transmit text by simply opening up a file manager (different for different OS and choice of desktop). Select the file from the manager and then drag and drop it onto the Tx panel. The mouse pointer will move the cursor insert point for the drop.

A number of additional shortcuts can be found in the Keyboard Operation section.

- **Waterfall display**

  Most of fldigi's unusual shortcuts are specific to this widget.

  **Waterfall display - Keyboard**

  - Shift Left/Right move the b/w marker by 1 Hz
  - Ctrl Left/Right move the b/w marker by 10 Hz

  **Waterfall display - Mouse**

  - Left click/drag move the b/w marker to, and start decoding at the cursor frequency
  - Right click/drag as above, but return to previous position on release
  - Middle click toggle AFC
  - Ctrl-Left click replay audio history at b/w marker position
  - Ctrl-Right click replay at cursor frequency and return on button release
  - Ctrl-Middle click copy the frequency under the cursor to the currently selected (or first) channel in the PSK viewer, and select the next channel
  - Shift-Left click/drag same as unmodified left click; no signal search
  - Shift-Right click/drag likewise, with a return to the previous frequency when the button is released, no signal search
  - Shift-mouse wheel move the squelch slider
  - Scroll wheel usage is dependent upon the configuration (see ConfigWaterfall)
    - None - no mouse wheel activity in waterfall panel
    - AFC range or BW - adjust the AFC range/BW up/down
• Squelch level - adjust the squelch level up/down
• Modem carrier - adjust the audio tracking point +/- Hz increments
• Modem - select modem type from a full rotary of available modems
• Scroll - move the waterfall left/right in 100 Hz increments (for 2x, 4x expanded waterfall view)
• Ctrl-mouse wheel change the AFC search width in PSK modes, or the bandwidth in CW and FeldHell

• Waterfall "Store" button
  • Left click Add a new item for the current frequency and modem
  • Shift-Left click Delete all items
  • Middle click Select last item in menu
  • Right click Pop up menu
    • Left/right click Select item (and switch to that frequency/modem)
    • Shift-Left/right click Delete item
    • Middle click Update (replace) item

• Digiscope display
  • Mouse wheel Change AFC/BW, same as Ctrl-mouse wheel on the waterfall

• Rig control window

There are some shortcuts in addition to those described in the Rig Control

• Frequency display
  • Left/Right arrow key change the frequency by one 1 Hz
  • Up/Down arrow key change the frequency by 10 Hz

• Frequency list
  • Shift-Left click delete the line under the cursor
  • Middle click replace the line under the cursor with the current frequency/mode/modem

• PSK viewer window

• Besides the bindings mentioned in the Psk Viewer section, there are mouse shortcuts to change the nominal frequency of a viewer channel:
  • Middle click copy the current waterfall b/w marker frequency to the channel under the cursor, overwriting that channel's nominal frequency
  • Right click restore a channel's nominal frequency
  • Right click on Clear as above, for all channels
Operating Multiple Copies

There are times that you may need to simultaneously operate two or more instances of fldigi. Or you might simply want to maintain two different configurations based on hardware usage.

The screen shot shows how this is done on Vista, but the process is nearly the same on XP, Win7 and Linux. When you install fldigi it creates a desktop icon launcher. Most of the Linux window managers allow you to create a desktop launch icon. Make as many copies of the launcher as needed for your applications and rename them accordingly. Then change the executable target entry to include the command line switch ‘--config-dir’ followed by the full pathname of the folder that will hold that particular configuration. You do not need to create that folder as fldigi will do so the first time it is launched from that desktop icon.

If the various configurations all use independent hardware, i.e. sound cards and rig control ports, then you can have them operating simultaneously. Each instance will have it's own configuration files, status file, macros, and logbook. It is possible to have each instance use the same logbook, but then simultaneous operation is not possible as the logbook file is not currently designed to allow that type of sharing.

If each instance will be paired with a separate flarq (similarly set up for multiple operation) then you will also need to add the command line switch for arq-server address and port. The same is true for use with applications that talk to fldigi via its xml-rpc socket port. You change the address/port pairs on both the fldigi launcher and the paired application such as flarq or flrig.
### Macros

Macros are short text statements that contain imbedded references to text data used by the program fldigi. Macro definition files(s) are located in the $HOME/.fldigi/macros/ directory and all have the extension ".mdf". The default set of macros are contained in the file $HOME/.fldigi/macros/macros.mdf. Fldigi will create this file with a set of default macros on its first execution.

Fldigi supports up to 48 macro definitions in sets of 12. Macro definitions are not recursive, that is; a macro cannot reference another macro or itself.

The imbedded references are similar to those used by DigiPan and other fine modem programs. The imbedded reference is an uppercase plain text descriptor contained with the <> brackets.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;FREQ&gt;</td>
<td>my frequency</td>
<td>&lt;RX&gt;</td>
<td>receive, places ^r tag at end of expanded macro always the last macro tag executed</td>
</tr>
<tr>
<td>&lt;MODE&gt;</td>
<td>mode</td>
<td>&lt;TX&gt;</td>
<td>transmit</td>
</tr>
<tr>
<td>&lt;MYCALL&gt;</td>
<td>my call</td>
<td>&lt;TX/RX&gt;</td>
<td>toggle T/R</td>
</tr>
<tr>
<td>&lt;MYLOC&gt;</td>
<td>my locator</td>
<td>&lt;SRCHUP&gt;</td>
<td>search UP for signal</td>
</tr>
<tr>
<td>&lt;MYNAME&gt;</td>
<td>my name</td>
<td>&lt;SRCHDN&gt;</td>
<td>search DOWN for signal</td>
</tr>
<tr>
<td>&lt;MYQTH&gt;</td>
<td>my QTH</td>
<td>&lt;GOHOME&gt;</td>
<td>return to sweet spot</td>
</tr>
<tr>
<td>&lt;MYRST&gt;</td>
<td>my RST</td>
<td><a href="">GOFREQ:NNN</a></td>
<td>move to autod freq NNNN</td>
</tr>
<tr>
<td>&lt;VER&gt;</td>
<td>Fldigi version</td>
<td><a href="">FILE:</a></td>
<td>insert text file</td>
</tr>
<tr>
<td>&lt;CALL&gt;</td>
<td>other call</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;NAME&gt;</td>
<td>other name</td>
<td><a href="">IDLE:NN.nn</a></td>
<td>idle signal for NN.nn sec</td>
</tr>
<tr>
<td>&lt;QTH&gt;</td>
<td>other QTH</td>
<td><a href="">TIMER:NN</a></td>
<td>repeat every NN sec</td>
</tr>
<tr>
<td>&lt;LOC&gt;</td>
<td>other locator</td>
<td><a href="">TUNE:NN</a></td>
<td>tune signal for NN sec</td>
</tr>
<tr>
<td>&lt;RST&gt;</td>
<td>other RST</td>
<td><a href="">WAIT:NN</a></td>
<td>delay xmt for NN sec</td>
</tr>
<tr>
<td>&lt;INFO1&gt;</td>
<td>S/N etc. in status bar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;INFO2&gt;</td>
<td>IMD etc. in status bar</td>
<td>&lt;CWID&gt;</td>
<td>CW identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;ID&gt;</td>
<td>mode ID</td>
</tr>
<tr>
<td>&lt;CLRRX&gt;</td>
<td>clear RX pane</td>
<td>&lt;TEXT&gt;</td>
<td>video text</td>
</tr>
<tr>
<td>&lt;GET&gt;</td>
<td>text to NAME/QTH</td>
<td>&lt;TXRSID:on</td>
<td>off</td>
</tr>
<tr>
<td>Tag</td>
<td>Description</td>
<td>Example</td>
<td>Function</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>&lt;LOG&gt;</td>
<td>save QSO data immediately</td>
<td>&lt;RXRSID:on</td>
<td>off</td>
</tr>
<tr>
<td>&lt;LNW&gt;</td>
<td>insert ^L tag into TX text stream save log entries to logbook when ^L is reached.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;QSOTIME&gt;</td>
<td>QSO time (HHMM)</td>
<td><a href="">POST:+/-nn.n</a></td>
<td>CW QSK post-timing</td>
</tr>
<tr>
<td>&lt;ILDT&gt;</td>
<td>LDT in iso-8601 format</td>
<td><a href="">PRE:nn.n</a></td>
<td>CW QSK pre-timing</td>
</tr>
<tr>
<td>&lt;LDT&gt;</td>
<td>Local datetime</td>
<td><a href="">RISE:nn.n</a></td>
<td>CW rise time</td>
</tr>
<tr>
<td>&lt;IZDT&gt;</td>
<td>ZDT in iso-8601 format</td>
<td><a href="">WPM:NN</a></td>
<td>CW WPM</td>
</tr>
<tr>
<td>&lt;ZDT&gt;</td>
<td>UTC datetime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;LT&gt;</td>
<td>current local time as HHMM</td>
<td>&lt;ZT&gt;</td>
<td>current Zulu time as HHMMZ</td>
</tr>
<tr>
<td>&lt;LD&gt;</td>
<td>current local date as YYYY-MM-DD</td>
<td>&lt;ZD&gt;</td>
<td>current Zulu date as YYYY-MM-DD Z</td>
</tr>
<tr>
<td>&lt;AFC:on</td>
<td>off</td>
<td>t&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;CNTR&gt;</td>
<td>contest counter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;DECR&gt;</td>
<td>decrement counter</td>
<td>&lt;LOCK:on</td>
<td>off</td>
</tr>
<tr>
<td>&lt;INCR&gt;</td>
<td>increment counter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;XOUT&gt;</td>
<td>insert exchange out text from contest dialog</td>
<td><a href="">MACROS:</a></td>
<td>change macro defs file</td>
</tr>
<tr>
<td>&lt;XBEG&gt;</td>
<td>mark start of &quot;Exchange Out&quot; text</td>
<td>&lt;MAPIT&gt;</td>
<td>open default browser to maps.google.com trying Postal Address / Lat-Lon / Locator in order</td>
</tr>
<tr>
<td>&lt;XEND&gt;</td>
<td>mark end of &quot;Exchange Out&quot; text, save to field</td>
<td><a href="">MAPIT:adr/lat/loc</a></td>
<td>Map starting with Postal Address / Lat-Lon / Locator in order. <a href="">MAPIT:adr</a> is the same as &lt;MAPIT&gt;</td>
</tr>
<tr>
<td>&lt;SAVECHG&gt;</td>
<td>save entire expanded macro text to &quot;Exchange Out&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Macro tags are also assigned to each supported modem type and sub-modem type that is supported by fldigi:

<table>
<thead>
<tr>
<th>Modem Type</th>
<th>Mode</th>
<th>Modem Type</th>
<th>Mode</th>
<th>Modem Type</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>CW</td>
<td>MFSK-8</td>
<td>BPSK-31</td>
<td>OLIVIA</td>
<td>THOR4</td>
<td></td>
</tr>
<tr>
<td>DomEX 4</td>
<td>MFSK-16</td>
<td>BPSK-63</td>
<td>OLIVIA:2</td>
<td>THOR5</td>
<td></td>
</tr>
<tr>
<td>DomEX 5</td>
<td>MFSK-32</td>
<td>BPSK-63F</td>
<td>OLIVIA:5</td>
<td>THOR8</td>
<td></td>
</tr>
<tr>
<td>DomEX 8</td>
<td>MFSK-4</td>
<td>BPSK-125</td>
<td>OLIVIA:5</td>
<td>THOR1</td>
<td></td>
</tr>
<tr>
<td>DomX1 1</td>
<td>MFSK-11</td>
<td>BPSK-250</td>
<td>OLIVIA:1</td>
<td>THOR1</td>
<td></td>
</tr>
<tr>
<td>DomX1 6</td>
<td>MFSK-22</td>
<td>BPSK-500</td>
<td>OLIVIA:1</td>
<td>THOR2</td>
<td></td>
</tr>
<tr>
<td>DomX2 2</td>
<td>MFSK-31</td>
<td>QPSK-31</td>
<td>RTTY</td>
<td>THROB1</td>
<td></td>
</tr>
<tr>
<td>FeldHELL</td>
<td>MFSK-64</td>
<td>QPSK-63</td>
<td>THROB2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SlowHELL</td>
<td>MT63-500</td>
<td>QPSK-125</td>
<td>RTTY:170</td>
<td>THROB4</td>
<td></td>
</tr>
<tr>
<td>HellX 5</td>
<td>MT63-1XX</td>
<td>QPSK-250</td>
<td>RTTY:170</td>
<td>THRBX1</td>
<td></td>
</tr>
<tr>
<td>HellX 9</td>
<td>MT63-2XX</td>
<td>QPSK-500</td>
<td>RTTY:850</td>
<td>THRBX2</td>
<td></td>
</tr>
<tr>
<td>FSK-HELL</td>
<td>PSK125R</td>
<td>PSK250R</td>
<td>WWV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FSK-H105</td>
<td>PSK2-50R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hell80</td>
<td>PSK500R</td>
<td></td>
<td>ANALYSIS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Local references are specified during the program configuration and can be changed during program operation.

Remote references are all part of the qso log field definitions and are routinely changed from contact to contact.

Global references are for items like Greenwich Mean Time.
The macros.mdf file can be edited with any ascii text editor such as kedit, gedit, geany, nano etc. But it is much easier to use the built-in macro editor provided in the program.

Right click on any macro key (or the alternate set) and a macro editing dialog opens with the current copy of that macro and its label. This looks very similar to the DigiPan macro editor at the urging of Skip Teller, KH6TY.

![Macro editor - macros.mdf](image)

The Text box is a mini-editor with a very limited set of control functions. You can mark, bound and select text for deletion (ctrl-X), copy (ctrl-C), and paste (ctrl-V). Marked text can also be deleted with the delete or the backspace keys. Marked text modification can also be invoked by using the mouse right click after highlighting.

The macro reference in the pick list can be transfered to the current editing cursor location. Highlight the desired macro reference and then press the double << arrow key for each occurance of the reference to be put into the macro text. You can change the label name but any more than 8 characters may exceed the width of the button for the default sized main dialog.

The <TIMER:NN> and <IDLE:NN> macro tags should have the NN replaced with the time interval in seconds.

```
<TX><IDLE:5>CQ CQ de <MYCALL> <MYCALL> k<RX><TIMER:20>
```

- will enable the PTT
- cause 5 seconds of idle signal
- send the CQ CQ de W1HKJ W1HKJ k
- disable PTT
- and count down 20 seconds before repeating the macro
- after sending the text the count down timer button (upper right hand corner of main dialog) will display the current timer value in seconds. Press this button to disable the timer.
- the timer be disabled if the Escape key is pressed, the T/R is pressed, and macro key is pressed, or if a callsign is copied from the Rx text area to the callsign logbook entry.
- the time will be disabled if any mouse activity occurs in the waterfall control.

The label associated with each macro key can be individually annotated with a symbol. Here are the symbols that are recognized by the button label drawing routine:
The `@` sign may also be followed by the following optional "formatting" characters, in this order:

- `#' forces square scaling, rather than distortion to the widget's shape.
- `+[1-9]` or `-[1-9]` tweaks the scaling a little bigger or smaller.
- `'$' flips the symbol horizontally, `%' flips it vertically.
- `[0-9]` - rotates by a multiple of 45 degrees. '5' and '6' do no rotation while the others point in the direction of that key on a numeric keypad. '0', followed by four more digits rotates the symbol by that amount in degrees.

Thus, to show a very large arrow pointing downward you would use the label string "@+92->".

Here are my macro buttons suitably annotated:

There are 4 sets of 12 macro functions. You can move between the 4 sets using the keyboard and the mouse.

1. Left click on the "1" button to move to set #2. Right click on the "1" button to move to set #4.
2. Move the mouse to anywhere on the macro buttons. Use the scroll wheel to move forward & backward through the macro sets.
3. Press the Alt-1, Alt-2, Alt-3 or Alt-4 to immediately change to that macro set.

- The label for CQ is "CQ @>|", denoting that both <TX> and <RX> are present in the macro text.
- The label for QSO is "QSO @>>", denoting that only <TX> is present in the macro text.
- The label for KN is "KN @||", denoting that only <RX> is present in the macro text.

You could use any label that is symbolic to the function required. Refer to the [FLTK web site](http://www.w1hkj.com/) for a full list of label types.

If you modify the macros and do not save them ("Files/Save Macros" on the main window) fldigi will prompt you to save the macros when you exit the program if you have the "Nag me" option selected.

The `<EXEC>...</EXEC>` macro tag provides a way to create external shell scripts and programs that can interact with fldigi. See [Exec Macro](http://www.w1hkj.com/) to learn more about the `<EXEC>` macro tag.
**Exec Macro**

The `<EXEC>` ... `<EXEC>` macro is designed to be used on the Linux OS as it supports fully functional pipes. Windows’ version of file pipes is not fully POSIX compliant, but the function might work in the environment. Consider all that the following allows you to do from within fldigi and you might want to consider changing over to Linux.

The `<EXEC>` macro defines an external child process (or processes) that will be called by fldigi when the macro key is invoked.

**Exported variables**

Fldigi exports a set of variables to the child process and adds ~/.fldigi/scripts to the PATH variable before running the shell code. This is the directory location for all executable scripts and programs which you might want to call from within the macro. Some examples will be given later. Open the macro editor for an undefined macro key and enter the following:

- `<EXEC>env | grep FLDIGI</EXEC>`

Save the macro; call it ENV. Then press the newly defined macro key. All of the exported variables will be shown in the transmit window.

Here is an example of the results:

- `FLDIGI_RX_IPC_KEY=9876`
- `FLDIGI_LOG_LOCATOR=FM02BT`
- `FLDIGI_TX_IPC_KEY=6789`
- `FLDIGI_LOG_RST_IN=`
- `FLDIGI_LOG_FREQUENCY=3581.000`
- `FLDIGI_AZ=108`
- `FLDIGI_MY_CALL=W1HKJ`
- `FLDIGI_LOG_TIME=2113`
- `FLDIGI_MY_NAME=Dave`
- `FLDIGI_VERSION=3.0preG`
- `FLDIGI_LOG_NOTES=`
- `FLDIGI_LOG_QTH=Mt Pleasant, SC`
- `FLDIGI_MY_LOCATOR=EM64qv`
- `FLDIGI_DIAL_FREQUENCY=3580000`
- `FLDIGI_CONFIG_DIR=/home/dave/.fldigi/`
- `FLDIGI_LOG_RST_OUT=`
- `FLDIGI_MODEM=BPSK31`
- `FLDIGI_LOG_CALL=KH6TY`
- `FLDIGI_MODEM_LONG_NAME=BPSK-31`
- `FLDIGI_AUDIO_FREQUENCY=1000`
- `FLDIGI_LOG_NAME=Skip`
- `FLDIGI_PID=14600`
- `FLDIGI_FREQUENCY=3581000`

All of the above envelope variables can be referenced in a shell script that is called from within fldigi.
Detection of existing scripts

In anticipation of a collection of useful "fldigi scripts", the macro browser contains a macro line for each executable file found in the scripts directory.

The EXEC macro allows the text that is read from the child process to be parsed for more fldigi macros. For example, try this macro:

<EXEC>cat foo</EXEC>

where foo is a file that contains:

<MYCALL>

This may have some interesting uses but, if it is undesirable, it can be suppressed with an extra layer of redirection. Instead of <EXEC>command</EXEC>, you would use <EXEC>noexp command</EXEC> where noexp is the following very simple script:

```
#!/bin/bash
echo -n "<STOP>"
"$@" # run the command
r=$? # save its exit code
echo -n "<CONT>"
exit $?
```

There are three additional MACRO definitions that expand the capability of the <EXEC> command: <STOP>, <CONT> and <GET>.

The <STOP> and <CONT> macros stop and resume the expansion of all <MACRO> strings. For example, <STOP><MYCALL><CONT><MYCALL> would only expand the second <MYCALL>. By wrapping the command output in this way we can be sure that no text will be expanded. You might even use

"$@" | sed "s/<CONT>//g"

if you feel paranoid.

You can "fork and forget" with an exec macro defined as:<EXEC>exec command -args >/dev/null</EXEC>

Any of the text that appears between the <EXEC> and </EXEC> can reference an executable program or shell command found in the ~/.fldigi/scripts directory.

Any text output that is returned by the program or script program (or the result of the in-line command) is always returned to the transmit buffer and appears as appended to the transmit window.

Querying an external database

The <GET> command captures returned text from the external process and parses it for the following
content:
$NAME text_name $QTH text_qth

If either $NAME or $QTH is present the trailing text is transferred to the LOG_NAME or LOG_QTH widgets respectively. This means that you can create a script that accesses a local or net based database of callsign data and parse that data to form the above console output. Fldigi will accept that output, parse it and populate the associated log entries. Cool! Now for some examples. Here is a perl script that performs the above for the University of Arkansas on-line callsign database, ualr-telnet:
http://www.w1hkj.com/FldigiHelp-3.21/ualr-telnet.html

The matching macro key definition for the above is:
<EXEC>ualr-telnet.pl $FLDIGI_LOG_CALL</EXEC><GET>
which I named "ualr ?"

Google Earth Map

Here is a really cool perl script(http://www.w1hkj.com/FldigiHelp-3.21/map.html), Google Earth Mapping, that accepts the current "Loc" field in the logging area and generates a Google Earth map which is displayed in your default browser. The macro call is <EXEC>map.pl</EXEC>

Custom dates/times

You can use <EXEC> to create custom date/time entries. For example, BARTG contesters use %H%M, but in other circumstances a user might prefer %H:%M or %H.%M etc.

Create the following script file in the ~/.fldigi/scripts directory, call it mytime:

snip---------------------------------------
#!/bin/sh
date --utc "+%H:%M"

snip---------------------------------------

date calls strftime, the same C function used by fldigi for the ZDT/LDT expansion, so it has an equally vast number of format strings to choose from. Look for them in its manual page.

Give "mytime" execute permissions with a file manager or with chmod: chmod u+x ~/.fldigi/scripts/mytime.

Test it on the command line and make sure it works correctly: ~/.fldigi/scripts/mytime

Restart fldigi. The mytime script will now appear at the end of the list in the macro browser, and can be entered with the << button as usual. Test that macro and you will see that <EXEC>mytime</EXEC> inserts the datetime in the specified format.

Of course you could have entered:

<EXEC>date --utc "+%H:%M"</EXEC>

in the macro body text directly

Many other uses for the <EXEC>...</EXEC> macro pair can be imagined when used the with ENV
parameters. For example you could send Azimuth data to an automated antenna rotor. The exported variables should be sufficient for a script writer to create custom loggers and clients.
Operating fldigi

Operating CW Mode

Carrier Frequency

Fldigi generates CW by inserting a keyed tone at the current waterfall audio frequency. The CW carrier frequency is the USB carrier + the audio frequency, or the LSB carrier - the audio frequency. If fldigi is tracking and receiving a CW signal on the waterfall your transmitted signal will be exactly on the frequency of the other operator. You probably cannot use your transceivers CW filter unless that filter can be used with the SSB mode.

QSK and the TAB KEY

If you are operating QSK with a separate transmitter / receiver you can very quickly stop your transmit signal with the TAB key. In the CW mode only the TAB key causes the program to skip over the remaining text in the transmit text buffer. The text that is skipped will be color coded blue. The program remains in the transmit mode (PTT enabled), but since the buffer is now empty no A2 CW signal is generated. Code transmission will then restart with the very next keyboard closure of a valid CW character.

Pausing transmit

The Pause/Break momentarily key stops sending text. Pressing it again resumes transmission.

Aborting transmit

The Escape key is used to immediately stop text transmission. The Tx buffer is cleared.

WPM adjustment

In CW mode the status bar is changed to include a transmit WPM adjuster. Use the arrow buttons or the mouse. Mouse wheel up/down changes transmit WPM by +/- 1. Hold the shift and mouse mouse wheel changes transmit WPM by +/- 10. The "*" button immediately to the right of the WPM adjuster is used to toggle between the current and the default transmit WPM.

The transmit WPM can also be adjusted with three hot keys:
- Numeric keypad "+" increases the transmit WPM by 1
- Numeric keypad "+" decreases the transmit WPM by 1
- Numeric keypad "*" toggles between the selected transmit WPM and the default transmit WPM

The "Default" control on the CW tab sets that default value. If during a QSO you needed to slow down to give the other op a better chance to copy what you are sending, just hit the "*" on the numeric keypad and the cw code will immediately switch to sending cw at the set default value (18 wpm in this example). Press the "*" again to return to back to the cw speed that you were previously using.
Each time the transmit WPM is changed the receive decoder WPM tracking is reset to the new transmit WPM. This allows you to quickly force the decoder to a new WPM range.

The Rx and Tx WPM are shown in the status bar. * indicates that default WPM is selected

**CW configuration**

The CW configuration dialog is easily reached from the Config menu or by right clicking on the left most entry in the status bar (CW).

**Prosigns**

The prosigns available in the CW mode are:

<table>
<thead>
<tr>
<th>PROSIGN</th>
<th>KEYBOARD</th>
<th>DISPLAYED AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>~</td>
<td>&lt;AA&gt;</td>
</tr>
<tr>
<td>AR</td>
<td>}</td>
<td>&lt;AR&gt;</td>
</tr>
<tr>
<td>AS</td>
<td>%</td>
<td>&lt;AS&gt;</td>
</tr>
<tr>
<td>HM</td>
<td>^</td>
<td>&lt;HM&gt;</td>
</tr>
<tr>
<td>INT</td>
<td>&amp;</td>
<td>&lt;INT&gt;</td>
</tr>
<tr>
<td>SK</td>
<td>&gt;</td>
<td>&lt;SK&gt;</td>
</tr>
<tr>
<td>KN</td>
<td>&lt;</td>
<td>&lt;KN&gt;</td>
</tr>
<tr>
<td>VE</td>
<td>{</td>
<td>&lt;VE&gt;</td>
</tr>
</tbody>
</table>

**CW (Morse) Modes Description**

**General Description**

CW (Continuous Wave) is something of a misnomer, since the carrier of a CW transmission is anything but continuous! The name comes from the early days of radio where the alternative was Spark, where the transmission was of a damped wavetrain generated by non-electronic means.

CW is generally transmitted using on-off or 100% amplitude keying, i.e. ASK modulation. Coding is invariably Morse (Albert Vaile) varicode, using a wide range of abbreviations. Transmission can be manual or by electronic keyer or computer, and reception can be aural or by computer. Computer reception works best at high speed under noise-free conditions with computer sent transmissions. Reception is limited at lower speeds as noise between characters or during fades is frequently misinterpreted (use RF gain to reduce noise and limit AGC action).
A linear transmitter is not required, although the keyed elements must be well shaped to avoid clicks which considerably increase the bandwidth. A risetime of 4ms is normal, although 1ms is best at higher speeds.

Protocol

CW is a manually controlled message asynchronous simplex chat mode, used without Forward Error Correction. The default calling mode is typically about 15 WPM.

Coding and Character Set

The Morse varicode with a limited character set. More common characters are sent faster. Character synchronization is defined by element spacing - three element spaces between characters, seven between words. Modulation is on-off.

CW Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Duty Cycle</th>
<th>Bandwidth</th>
<th>ITU Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CW20</td>
<td>10 baud</td>
<td>~ 2 cps (20 wpm)</td>
<td>~ 44%</td>
<td>50 Hz</td>
<td>50H0A1A</td>
</tr>
<tr>
<td>CW40</td>
<td>20 baud</td>
<td>~ 4 cps (40 wpm)</td>
<td>~ 44%</td>
<td>100 Hz</td>
<td>100HA1A</td>
</tr>
<tr>
<td>CW100</td>
<td>50 baud</td>
<td>~ 10 cps (100 wpm)</td>
<td>~ 44%</td>
<td>200 Hz</td>
<td>200HA1B6</td>
</tr>
</tbody>
</table>

Notes:

1. WPM is based on typical word 'PARIS', plus word space (50 dot-length elements). Values are approximate because a variable length code is used.
2. Transmitter average power output relative to a constant carrier of the same PEP value. Duty cycle is calculated using the same standard word, but clearly duty cycle also depends on keying 'weight'. The word 'PARIS' contains 22 key-down elements.
3. This is the "Necessary Bandwidth" as defined by the ITU.

5. Calling mode is usually 15 - 20WPM.
6. Designator indicates reception by machine rather than by ear.
Operating Contestia

fldigi can operate on the following Contestia modes without special setup by the operator:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contestia 4-250</td>
<td>62.5 baud</td>
<td>~ 40 wpm</td>
<td>250 Hz</td>
</tr>
<tr>
<td>Contestia 8-250</td>
<td>31.25 baud</td>
<td>~ 30 wpm</td>
<td>250 Hz</td>
</tr>
<tr>
<td>Contestia 4-500</td>
<td>125 baud</td>
<td>~ 78 wpm</td>
<td>500 Hz</td>
</tr>
<tr>
<td>Contestia 8-500</td>
<td>62.5 baud</td>
<td>~ 60 wpm</td>
<td>500 Hz</td>
</tr>
<tr>
<td>Contestia 16-500</td>
<td>31.25 baud</td>
<td>~ 30 wpm</td>
<td>500 Hz</td>
</tr>
<tr>
<td>Contestia 8-1000</td>
<td>125 baud</td>
<td>~ 117 wpm</td>
<td>1000 Hz</td>
</tr>
<tr>
<td>Contestia 16-1000</td>
<td>62.5 baud</td>
<td>~ 78 wpm</td>
<td>1000 Hz</td>
</tr>
<tr>
<td>Contestia 32-1000</td>
<td>31.25 baud</td>
<td>~ 48 wpm</td>
<td>1000 Hz</td>
</tr>
</tbody>
</table>

Unusual combinations of symbol rate and bandwidth can be selected using the Contestia configuration tab.

Contestia is a digital mode directly derived from Olivia that is not quite as robust - but more of a compromise between speed and performance. It was developed by Nick Fedoseev, UT2UZ, in 2005. It sounds almost identical to Olivia, can be configured in as many ways, but has essentially twice the speed. Contestia has 40 formats just like Olivia - some of which are considered standard and they all have different characteristics. The formats vary in bandwidth (125, 250, 500, 1000, and 2000 Hz) and number of tones used (2, 4, 8, 16, 32, 64, 128, or 256). The standard Contestia formats (bandwidth/tones) are 125/4, 250/8, 500/16, 1000/32, and 2000/64. The most commonly used formats right now seem to be 250/8, 500/16, and 1000/32.

Contestia performs very well under weak signal conditions. It handles QRM, QRN, and QSB very well also. It decodes below the noise level but Olivia still outperforms it in this area by about 1.5 - 3 dB depending on configuration. It is twice as fast as Olivia per configuration. It is an excellent weak signal, ragchew, QRP, and DX digital mode. When ragchewing under fair or better conditions it can be more preferable to many hams than Olivia because of the faster speed. For contests it might also be a good mode IF the even faster configurations such as 1000/8 or 500/4 are used. Contestia get it's increased speed by using a smaller symbol block size (32) than Olivia (64) and by a using 6bit decimal character set rather than 7bit ASCII set that Olivia does. Therefore, it has a reduced character set and does not print out in both upper and lower case (like RTTY). Some traffic nets might not want to use this mode because it does not support upper and lower case characters and extended characters found in many documents and messages. For normal digital chats and ham communications that does not pose any problem.
Contestia Modes Description

General Description

Contestia is a family of MFSK modes with a high redundancy Forward Error Correction system similar to Olivia. The mode works well on poor HF paths and has good sensitivity. It is a mode designed to be used for contesting and keyboard to keyboard contacts.

The most widely used versions have a symbol rate of 31.25 baud. The typing speed is varied by changing the number of tones used, but can also be changed by change of baud rate. The MFSK is constant phase, and the transmission constant amplitude, so transmitter linearity is important. The modes are moderately tolerant of mistuning. The typing rate is very modest, given the high bandwidth used. The mode was designed by UT2UZ.

Protocol

These are unconnected, manually controlled message asynchronous symbol synchronous simplex chat modes, with full-time Forward Error Correction. The tones are spaced the same as the baud rate, for example 31.25Hz for the 31.25 baud modes. The default mode has 8 tones and is 500 Hz wide.

Coding and Character Set

The ASCII-128 user interface is converted to upper case and control codes are either dropped or mapped to a fixed set. This results in a smaller character block which is then mapped to a block of sequential MFSK tones via a Walsh-Hadamard function. Tones are deployed via a gray coder.

There are three popular modes, which have 8-FSK, 16-FSK and 32-FSK, thus having three, four or five bits per symbol. The combination of a reduced character set with a reduced number of bits per character provides approximately a 2 to 1 speed advantage over the equivalent set of tones in the mode Olivia.
Contestia Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed(^1)</th>
<th>Lowest S/N</th>
<th>Duty Cycle(^2)</th>
<th>Modulation</th>
<th>Bandwidth(^3)</th>
<th>ITU Designation(^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-250</td>
<td>62.5 baud</td>
<td>~40 wpm</td>
<td>-10 dB</td>
<td>100%</td>
<td>4-FSK</td>
<td>250 Hz</td>
<td>250HF1B</td>
</tr>
<tr>
<td>8-250</td>
<td>31.25 baud</td>
<td>~30 wpm</td>
<td>-13 dB</td>
<td>100%</td>
<td>8-FSK</td>
<td>250 Hz</td>
<td>250HF1B</td>
</tr>
<tr>
<td>4-500</td>
<td>125 baud</td>
<td>~ 78 wpm</td>
<td>-8 dB</td>
<td>100%</td>
<td>4-FSK</td>
<td>500 Hz</td>
<td>500HF1B</td>
</tr>
<tr>
<td>8-500</td>
<td>62.5 baud</td>
<td>~ 60 wpm</td>
<td>-10 dB</td>
<td>100%</td>
<td>8-FSK</td>
<td>500 Hz</td>
<td>500HF1B</td>
</tr>
<tr>
<td>16-500</td>
<td>31.25 baud</td>
<td>~ 30 wpm</td>
<td>-12 dB</td>
<td>100%</td>
<td>16-FSK</td>
<td>500 Hz</td>
<td>500HF1B</td>
</tr>
<tr>
<td>8-1000</td>
<td>125 baud</td>
<td>~ 117 wpm</td>
<td>-5 dB</td>
<td>100%</td>
<td>8-FSK</td>
<td>1000 Hz</td>
<td>1K00F1B</td>
</tr>
<tr>
<td>16-1000</td>
<td>62.5 baud</td>
<td>~ 78 wpm</td>
<td>-9 dB</td>
<td>100%</td>
<td>16-FSK</td>
<td>1000 Hz</td>
<td>1K00F1B</td>
</tr>
<tr>
<td>32-1000</td>
<td>31.25 baud</td>
<td>~ 48 wpm</td>
<td>-12 dB</td>
<td>100%</td>
<td>32-FSK</td>
<td>1000 Hz</td>
<td>1K00F1B</td>
</tr>
</tbody>
</table>

Notes:
1. WPM is based on an average 5 characters per word, plus word space.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the "Necessary Bandwidth" as defined by the ITU.
DominoEX - Operating

Fldigi can operate in the following DominoEX modes:

- dominoEX-4
- dominoEX-5
- dominoEX-8
- dominoEX-11
- dominoEX-16, and
- dominoEX-22

The sound card sampling rate is 8000 Hz for the 4, 8 and 16 modes. It is 11025 Hz for the 5, 11 and 22 modes. This change in sound card sampling rate will be seen in the drop rate on the waterfall. See DominoEX Technical Description.

The modem code for dominoEX uses a wide band multiple frequency detector that can lock on and detect the incoming signal even when badly mistuned. Frequency domain oversampling is used to allow proper tone detection with the need for AFC. The AFC control does not alter the decoder in any way.

The waterfall and digiscope will appear as:

![Waterfall and Digiscope](image)

The text displayed in the status area is the secondary text being sent by the transmitting station. When the keyboard buffer is empty the dominoEX modem transmits text from the secondary text buffer. Your secondary text buffer can be edited on the DominoEX configuration tab.

The digiscope display represents the tone pairs moving through the tone filters. You can also use an alternate digiscope display (left click on the digiscope display area).

![Alternate Digiscope](image)

In this display mode the red line represents the center of the multiple tone bins that are in the detector. The dots will be blurry if the AFC is not locked on and become very distinct when AFC lock has been achieved. The tone dots will move from bottom to top (opposite the direction of the waterfall). This is the same signal mistuned:
DominoEX Modes Description

General Description
DominoEX is a family of offset incremental multi-frequency shift keyed modes with low symbol rate. A single carrier of constant amplitude is stepped between 18 tone frequencies in a constant phase manner. As a result, no unwanted sidebands are generated, and no special amplifier linearity requirements are necessary. The tones change according to an offset algorithm which ensures that no sequential tones are the same or adjacent in frequency, considerably enhancing the inter-symbol interference resistance to multi-path and Doppler effects.

The mode is normally used without Forward Error Correction, as it is very robust. The default speed (11 baud) was designed for NVIS conditions (80m at night), and other speeds suit weak signal LF, and high speed HF use. The use of incremental keying gives the mode complete immunity to transmitter-receiver frequency offset, drift and excellent rejection of propagation induced Doppler.

Protocol
These are unconnected, manually controlled message asynchronous simplex chat modes, normally used without Forward Error Correction. The FEC option is rarely used. The default calling mode is DominoEX11.

Coding and Character Set
A nibble-based varicode with ASCII-256 user interface is used. Lower case characters are sent faster. An ASCII-128 secondary character set allows a fixed (typically ID) message to be sent whenever the transmitter is idle. Modulation is nibble-wise symbol synchronous, differential.

The FEC option uses a binary varicode with modified ASCII-128 user interface. Lower case characters are sent faster. Modulation uses two dibit pairs, symbol synchronous, differential.

The FEC option uses binary convolution to generate two dibits per varicode bit, and halves the corrected data rate for the same symbol rate. Rate R=1/2, Constraint length K=7, default Interleaver L=4 (16 bits). The default interleaver is too short for effective burst correction, and performs better with L=10 or more.
### DominoEX Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Duty Cycle</th>
<th>Bandwidth</th>
<th>ITU Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DominoEX4&lt;sup&gt;5&lt;/sup&gt;</td>
<td>3.90625 baud</td>
<td>29 wpm</td>
<td>100%</td>
<td>173 Hz</td>
<td>173HF1B</td>
</tr>
<tr>
<td>DominoEX5&lt;sup&gt;5&lt;/sup&gt;</td>
<td>5.3833 baud</td>
<td>44 wpm</td>
<td>100%</td>
<td>244 Hz</td>
<td>244HF1B</td>
</tr>
<tr>
<td>DominoEX8&lt;sup&gt;5&lt;/sup&gt;</td>
<td>7.8125 baud</td>
<td>58 wpm</td>
<td>100%</td>
<td>346 Hz</td>
<td>346HF1B</td>
</tr>
<tr>
<td>DominoEX11&lt;sup&gt;6&lt;/sup&gt;</td>
<td>10.766 baud</td>
<td>80 wpm</td>
<td>100%</td>
<td>262 Hz</td>
<td>262HF1B</td>
</tr>
<tr>
<td>DominoEX16</td>
<td>15.625 baud</td>
<td>115 wpm</td>
<td>100%</td>
<td>355 Hz</td>
<td>355HF1B</td>
</tr>
<tr>
<td>DominoEX22</td>
<td>21.533 baud</td>
<td>160 wpm</td>
<td>100%</td>
<td>524 Hz</td>
<td>524HF1B</td>
</tr>
</tbody>
</table>

**Notes:**

1. WPM is based on an average 5 characters per word, plus word space. Values based on sending 100 "paris " words.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the "Necessary Bandwidth" as defined by the ITU.
5. Double spaced mode.
6. Default and normal calling mode.

Copyright © M. Greenman 2008
Hell Modes

All Hellschreiber modes are based on character scanning, reproducing characters in a similar way to a dot-matrix printer. This technique uses a digital transmission, yet allows the received result to be interpreted by eye, a similar concept to the reception of Morse by ear. The character is scanned upwards, then left to right. There are typically 14 pixels (transmitted dot elements) per column (although single pixels are never transmitted) and up to seven columns per character including inter-character space.

These remarkably simple modes are easy to use, easy to tune, and although not especially sensitive, are entirely suited to HF/VHF since they use no sync and the eye can discern the text even in high levels of noise. fldigi can operate in the following Hellschreiber modes:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Duty Cycle</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feld-Hell</td>
<td>122.5 baud</td>
<td>~ 2.5 cps (25 wpm)</td>
<td>~ 22%</td>
<td>350 Hz</td>
</tr>
<tr>
<td>Slow Hell</td>
<td>14 baud</td>
<td>~ 0.28 cps (2.8 wpm)</td>
<td>~ 22%</td>
<td>40 Hz</td>
</tr>
<tr>
<td>Feld-Hell X5</td>
<td>612.5 baud</td>
<td>~ 12.5 cps (125 wpm)</td>
<td>~ 22%</td>
<td>1750 Hz</td>
</tr>
<tr>
<td>Feld-Hell X9</td>
<td>1102.5 baud</td>
<td>~ 22.5 cps (225 wpm)</td>
<td>~ 22%</td>
<td>3150 Hz</td>
</tr>
<tr>
<td>FSK-Hell</td>
<td>245 baud</td>
<td>~ 2.5 cps (25 wpm)</td>
<td>~ 80%</td>
<td>490 Hz</td>
</tr>
<tr>
<td>FSK-Hell 105</td>
<td>105 baud</td>
<td>~ 2.5 cps (25 wpm)</td>
<td>~ 80%</td>
<td>210 Hz</td>
</tr>
<tr>
<td>Hell 80</td>
<td>245 baud</td>
<td>~ 5.0 cps (50 wpm)</td>
<td>100%</td>
<td>800 Hz</td>
</tr>
</tbody>
</table>

Feld-Hell seems to be the most commonly used and use can usually be found on 80 and 40 meters at the high end of the digital sub bands. All of the Hellschreiber modes are described in detail with both waterfall and sound clips in the Sight and Sounds section. Feld-Hell look like this when being received by fldigi:

Feld-Hell, Slow Hell, Feld-Hell X5, and Feld-Hell X9 are all pulse modes. Extreme linearity is required in the transmit path in order to control the bandwidth of the transmitted signal. Feld-Hell X5, Feld-Hell X9 and Hell 80 should probably not be used in the US. They can be used on VHF and UHF.
Hellschreiber Modes Description

General Description
The name Hellschreiber means 'Hell's writer'. The technique was developed by Rudolf Hell in 1927, and was used as a successful radio mode first in 1937, well before RTTY. Use by Amateurs is relatively recent. The most popular mode is that used by the war-time Feld-Hellschreiber (field writer), which is on-off keyed, the same as Morse. In order to limit the bandwidth, the rise-time of dots is usually limited to about 1ms, and the best methods use raised cosine shaped dots.

FSK-Hell, of recent invention, uses differential Minimum Shift Keying (2-MSK) with a raised cosine envelope to minimize bandwidth. The technique is similar to PSK with one tone suppressed. These modes are much more sensitive than all other Hell modes, and have much better resistance to multi-path and Doppler. Of all Hell modes, only FSK-Hell 245 can correctly use a regular computer font. The FSK-Hell 105 version uses a special restricted resolution font to achieve very narrow bandwidth. Great for DX!

The Hell 80 mode is a Hell original from the 1970s. It is a rival to RTTY, with similar speed and bandwidth.

All Hellschreiber modes are based on character scanning, reproducing characters in a similar way to a dot-matrix printer. This technique uses a digital transmission, yet allows the received result to be interpreted by eye, a similar concept to the reception of Morse by ear. The character is scanned upwards, then left to right. There are typically 14 pixels (transmitted dot elements) per column (although single pixels are never transmitted) and up to seven columns per character including inter-character space.

These remarkably simple modes are easy to use, easy to tune, and although not especially sensitive, are entirely suited to HF since they use no sync and the eye can discern the text even in high levels of noise.

Protocol
These are unconnected, manual quasi-synchronous scanned image simplex chat modes, using high redundancy instead of error correction. The receiver presents a grey-scale view of the raw received signal. The default calling mode is Feld-Hell (122.5 baud).

Coding and Character Set
There is no coding. The character is scanned and transmitted as sequential pixels. An upper-case only character set is usually used, and a special font developed by Hell gives improved readability and cleverly limits the bandwidth. Most recent systems use proportional character spacing to improve speed. The modulation technique depends on the mode.
## Hellschreiber Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed(^1)</th>
<th>Duty Cycle(^2)</th>
<th>Modulation</th>
<th>Bandwidth(^3)</th>
<th>ITU Designation(^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feld-Hell(^5)</td>
<td>122.5 baud</td>
<td>~ 2.5 cps (25 wpm)</td>
<td>~ 22%</td>
<td>OOK ASK</td>
<td>350 Hz</td>
<td>350HA1B</td>
</tr>
<tr>
<td>Slow Hell</td>
<td>14 baud</td>
<td>~ 0.28 cps (2.8 wpm)</td>
<td>~ 22%</td>
<td>OOK ASK</td>
<td>40 Hz</td>
<td>40H0A1B</td>
</tr>
<tr>
<td>Feld-Hell X5</td>
<td>612.5 baud</td>
<td>~ 12.5 cps (125 wpm)</td>
<td>~ 22%</td>
<td>OOK ASK</td>
<td>1750 Hz</td>
<td>1K75A1B</td>
</tr>
<tr>
<td>Feld-Hell X9</td>
<td>1102.5 baud</td>
<td>~ 22.5 cps (225 wpm)</td>
<td>~ 22%</td>
<td>OOK ASK</td>
<td>3150 Hz</td>
<td>3K15A1B</td>
</tr>
<tr>
<td>FSK-Hell 245</td>
<td>245 baud</td>
<td>~ 2.5 cps (25 wpm)</td>
<td>~ 80%</td>
<td>2-MSK(^6)</td>
<td>490 Hz</td>
<td>490HF1B</td>
</tr>
<tr>
<td>FSK-Hell 105</td>
<td>105 baud</td>
<td>~ 2.5 cps (25 wpm)</td>
<td>~ 80%</td>
<td>2-MSK(^6)</td>
<td>210 Hz</td>
<td>210HF1B</td>
</tr>
<tr>
<td>Hell 80</td>
<td>245 baud</td>
<td>~ 5.0 cps (50 wpm)</td>
<td>100%</td>
<td>2-FSK (480Hz)</td>
<td>800 Hz</td>
<td>800HF1B</td>
</tr>
</tbody>
</table>

### Notes:

1. WPM is based on an average 5 characters per word, plus word space. Values are approximate because a proportional or small font increases speed.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the "Necessary Bandwidth" as defined by the ITU.
5. Default and normal calling mode.
6. Differential Minimum Shift Keying, includes raised cosine ASK.

---

Copyright © M. Greenman 2008
MFSK

MFSK16 and MFSK8 are multi-frequency shift keyed (MFSK) modes with low symbol rate. A single carrier of constant amplitude is stepped (between 16 or 32 tone frequencies respectively) in a constant phase manner. As a result, no unwanted sidebands are generated, and no special amplifier linearity requirements are necessary. The tones selected are set by the transmitted (4 or 5 bit) bit pattern and a gray-code table.

The mode has full-time Forward Error Correction, so it is very robust. Tuning must be very accurate, and the software will not tolerate differences between transmit and receive frequency. The mode was designed for long path HF DX, and due to its great sensitivity is one of the best for long distance QSOs and skeds. MFSK8 has improved sensitivity, but is very difficult to tune, and suffers more from Doppler. It is useful as the band fades out.

MFSK-32 and MFSK-64 are high baud rate and wide bandwidth modes designed for use on VHF and UHF. These are very useful for send large documents or files when some transmission errors are can be tolerated.

This is an example of properly tuned MFSK16 signal with a s/n of approximately 9 dB.

The same signal viewed with the waterfall expanded to the x2 factor.

MFSK Picture Mode

Fldigi can send and receive images using all MFSK baud rates. When operating with other modem programs you should limit sending pictures to the MFSK-16 baud rate. The program can send and receive MFSK images in both black and white and in 24 bit color. The transmission mode for MFSKpic is similar to FAX.

Reception of an MFSKpic transmission is fully automatic. The MFSKpic transmission has a preamble sent which will be visible on the text screen. The preamble reads as "Pic:WWWxHHH;" or "Pic:WWWxHHHC;" for b/w or color respectively. The WWW and HHH are numbers specifying the width and height of the picture in pixels.

The successful reception of a MFSKpic is highly dependent on s/n conditions. The data is transmitted
as an FM modulated signal and is subject to burst and phase noise on the transmission path. It can provide excellent photo transmission on a really good path.

This is an example of a photo received on a bench test. The received image is an exact replica of the transmitted image. The color depth is a full 24 bits.

Images should be carefully selected for size before beginning a transmission. To calculate the transmit time for an image use the following formula:

\[
\text{Time(sec)} = \frac{W \times H}{1000} \text{ for black and white}
\]

\[
\text{Time(sec)} = \frac{W \times H \times 3}{1000} \text{ for color}
\]

where the W and H are the dimensions of the photo in pixels. A 200 x 200 image will take 120 seconds to transmit in color and 40 seconds to transmit in b/w. The symbol rate for this mode is 1000 data bytes per second. The color image consists of 3 bytes; red, blue and green for each pixel.

This is an example of a picture received live on 80 meters (thanks K0OG)

Received images are saved in the default folder $HOME/.fldigi/images (Linux) or <defaultpath>/fldigi.files/images (Windows).
Transmitting an Image

You can only transmit an image while in the MFSK-16 mode. The image can be prepared for transmission while in the receive mode. Right click in the transmit text box and select "Send Image" from the popup menu. This will open up the transmit image dialog which will be blank to start.

Press the "Load" button and a file selection dialog will allow you to select a suitable image for transmit. The file selection dialog also has a preview capability so you will see what the image looks like.

You may also open a window manager file browser and drag and drop an image to the center part of the Send image dialog.

The "X1" button is a three-way toggle that allows you to transmit an image file in
- X1 - normal and compatible with other modem programs
- X2 - double speed, and
- X4 - quadruple speed. X2 and X4 are fldigi specific image modes.

The Send image dialog after the image was drag and dropped onto the dialog.

The properties box said this image was 120 x 119 24 bit color. So it should take 42.8 seconds to transmit in full color. You can send a color or a b/w image in either color mode or b/w mode. If you transmit a color image in b/w the program will convert the image before transmitting. If you transmit a b/w image as full color you are in effect transmitting redundant information, but it can be done. I selected the "XmtClr" button for a trial run. Pressing either the "XmtClr" or "XmtGry" will put the program and the transceiver into the transmit mode if it was in the receive mode. The image is cleared and then repainted as the transmission proceeds. You see the same image progression that the receiving station should see. The main display also displays the % completion on the status bar.
Hold the mouse over either the XmtClr or the XmtGry button and the tooltip will tell you the transmit time for this image.

You may abort the transmission at any time by pressing the "Abort Xmt" button. That will return you to the text mode for MFSK. You will then have to toggle the T/R button if you want to return to receive.

The receiving program decodes the "Pic:110x119C;" as a color picture 110 wide by 119 high. Here is shown being received on a computer running Vista Home Premium.

This is what the waterfall will look like during the reception of an MFSK-16 image.

The actual spectrum signature will vary with the image bytes being transmitted. The waterfall scale is in the x4 mode and the above photo was being transmitted in 24 bit color for this screenshot. The waterfall clearly shows that the image transmission is within the bandwidth occupied by MFSK-16.

**Picture with a slant**

If either the send, receive or both ends of the transmission are using an uncalibrated sound card whose sampling rate is not an exact multiple of 8000 Hz the resulting picture at the receive end will appear slanted. The degree of slant is directly related to the accumulation of the frequency error at both ends of the transfer. Stations wishing to send and receive MFSKpic's should calibrate their sound card. The WWV calibration mode is used to measure and set the parts per million (ppm) correction factor for the sound card.

Your sound system may be fully corrected, but the sending station may have an uncorrected sound card. You can usually correct for small errors in the following way. After the full picture is received move the mouse to bottom left or right corner of the slanted images (the corner that clearly visible). Then left click on that corner. The program will correct for the slant. The correction will not be perfect but it may help to make the image more viewable.
MFSK Modes Description

General Description

MFSK16 and MFSK8 are multi-frequency shift keyed (MFSK) modes with low symbol rate. A single carrier of constant amplitude is stepped (between 16 or 32 tone frequencies respectively) in a constant phase manner. As a result, no unwanted sidebands are generated, and no special amplifier linearity requirements are necessary. The tones selected are set by the transmitted (4 or 5 bit) bit pattern and a gray-code table.

The mode has full-time Forward Error Correction, so it is very robust. Tuning must be very accurate, and the software will not tolerate differences between transmit and receive frequency. The mode was designed for long path HF DX, and due to its great sensitivity is one of the best for long distance QSOs and skeds. MFSK8 has improved sensitivity, but is very difficult to tune, and suffers more from Doppler. It is useful as the band fades out.

MFSK16 and MFSK8 were developed by Murray ZL1BPU and Nino IZ8BLY.

Protocol

These are unconnected, manually controlled message asynchronous symbol synchronous simplex chat modes, with full-time Forward Error Correction. MFSK tone spacing is in both cases equal to the symbol rate (15.625 Hz and 7.8125 Hz). In order to maintain sync during idle periods, every few seconds a few non-printing characters is sent. The default calling mode is MFSK16.

MFSK16 also has an image transfer mode. This is controlled and triggered from MFSK16, and will transmit B&W or Colour pictures of any size and shape, although smaller is better, as transmission is only 1000 pixels/sec. This image transmission is an analog mode without sync. The image transmission is FSK of the same bandwidth as MFSK16, and again transmitter linearity is unimportant.

Coding and Character Set

A nibble-based varicode with ASCII-256 user interface is used. Lower case characters are sent faster. Modulation is four or five bit symbol synchronous, and in MFSK16 the dibit order is determined automatically. MFSK8 has five bits per symbol and uses an extra trial Viterbi decoder to determine the correct bit order.

The FEC uses binary convolution to generate two dibits per varicode bit. Rate R=1/2, Constraint Length K=7, Interleaver L=10 (40 bits). A matrix interleaver is used. This is a standard NASA design.
# MFSK Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed(^1)</th>
<th>Duty Cycle(^2)</th>
<th>Modulation</th>
<th>Bandwidth(^3)</th>
<th>ITU Designation(^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFSK4</td>
<td>3.906 baud</td>
<td>18 wpm</td>
<td>100%</td>
<td>32-FSK</td>
<td>154 Hz</td>
<td>154HF1B</td>
</tr>
<tr>
<td>MFSK8(^5)</td>
<td>7.8125 baud</td>
<td>36 wpm</td>
<td>100%</td>
<td>32-FSK</td>
<td>316 Hz</td>
<td>316HF1B</td>
</tr>
<tr>
<td>MFSK11</td>
<td>10.767 baud</td>
<td>40 wpm</td>
<td>100%</td>
<td>16-FSK</td>
<td>218 Hz</td>
<td>218HF1B</td>
</tr>
<tr>
<td>MFSK16</td>
<td>15.625 baud</td>
<td>58 wpm</td>
<td>100%</td>
<td>16-FSK</td>
<td>316 Hz</td>
<td>316HF1B</td>
</tr>
<tr>
<td>MFSK22</td>
<td>21.533 baud</td>
<td>80 wpm</td>
<td>100%</td>
<td>16-FSK</td>
<td>435 Hz</td>
<td>435HF1B</td>
</tr>
<tr>
<td>MFSK31</td>
<td>31.250 baud</td>
<td>55 wpm</td>
<td>100%</td>
<td>8-FSK</td>
<td>330 Hz</td>
<td>330HF1B</td>
</tr>
<tr>
<td>MFSK32</td>
<td>31.250 baud</td>
<td>120 wpm</td>
<td>100%</td>
<td>16-FSK</td>
<td>630 Hz</td>
<td>630HF1B</td>
</tr>
<tr>
<td>MFSK64</td>
<td>62.500 baud</td>
<td>240 wpm</td>
<td>100%</td>
<td>16-FSK</td>
<td>1260 Hz</td>
<td>1260HF1B</td>
</tr>
<tr>
<td>MFSK Image</td>
<td>1000 px/sec</td>
<td>128x128 B&amp;W 16 sec</td>
<td>100%</td>
<td>Analog FSK</td>
<td>316 Hz</td>
<td>316HF1C</td>
</tr>
</tbody>
</table>

**Notes:**
1. WPM is based on an average 5 characters per word, plus word space. Values based on sending 100 "paris " words.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the "Necessary Bandwidth" as defined by the ITU.
5. Double spaced mode.
6. Default and normal calling mode.
MT63

MT63 is an Orthogonal Frequency Division Multiplexed mode consisting of 64 parallel carriers each carrying part of the transmitted signal. The tones are differential BPSK modulated. MT63 employs a unique highly redundant Forward Error Correction system which contributes to its robustness in the face of interference and facing. The tones have synchronous symbols, and are raised cosine modulated. This mode requires a very linear transmitter. Over-driving leads to excessive bandwidth and poorer reception.

The mode is very tolerant of tuning and fldigi will handle as much as 100 Hz of mistuning. This is very important since MT63 is often used in very low Signal to Noise ratios. There are three standard modes:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT63-500</td>
<td>5.0 baud</td>
<td>5.0 cps (50 wpm)</td>
<td>500 Hz</td>
</tr>
<tr>
<td>MT63-1000</td>
<td>10.0 baud</td>
<td>10.0 cps (100 wpm)</td>
<td>1000 Hz</td>
</tr>
<tr>
<td>MT63-2000</td>
<td>20 baud</td>
<td>20.0 cps (200 wpm)</td>
<td>2000 Hz</td>
</tr>
</tbody>
</table>

In addition there are two interleaver options (short and long) which can be set on the MT63 configuration tab. The default calling mode is MT63-1000. If the short interleaver is used then one can expect some compromise in robustness. The long interleaver results in somewhat excessive latency (delay between overs) for keyboard chatting. MT63-1000 with the long interleaver has a latency of 12.8 seconds.

You can change from receive to transmit immediately upon seeing the other station's signal disappear from the waterfall. You do not need to wait until the receive text completes. Any remaining data in the interleaver will be flushed and the associated receive text printed quickly to the Rx pane. Tx will commence right after the buffer is flushed.

MT63 may be operated in the default fixed audio frequency mode. In this mode you are not allowed to randomly place the signal on the waterfall. Your transmit signal, and also the received signal should be centered at 750 Hz for MT63-500, 1000 Hz for MT63-1000, and 1500 Hz for MT63-2000. If you click on the waterfall to move the tracking point it will be restored to the required position.

The default mode, MT63-1000, looks like this on fldigi's waterfall.
You can also elect to operate the MT63 modem in a "manual tune" mode (see MT63 configuration tab). The manual tune allows you to place both the Rx and the Tx signal to be anywhere within the confines of your SSB bandwidth. This screen shot shows this capability:

This view also demonstrates how immune MT63 is to interference. The multiple PSK31 signals that appear on top of the MT63 signal did not degrade the decoder. MT63 is usually used above 14073 MHz to avoid the possibility of this type of mode conflict.

Edited excerpts from Pawel Jalocha's official mt63 code release

The MT63 modem is intended for amateur radio as a conversation (RTTY like) mode where one station transmits and one or more other stations can listen. In short, the modem transmits 64 tones in its baudrate specific bandwidth. The differential bipolar phase modulation is used to encode 10 bits of information per second on each tone. The user data in the form of 7-bit ASCII characters is encoded as a set of 64-point Walsh functions. The bits are interleaved over 32 symbols (3.2 seconds) to provide resistance against both pulse and frequency selective noise or fading. The character rate equals to the symbols rate thus the modem can transmit 10 7-bit characters per second.

This modem can as well run in two other modes obtained by simple time scaling, the possible modes are summarized here:

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>Symbol Rate</th>
<th>Character Rate</th>
<th>Interleave / Char.</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 Hz</td>
<td>5 baud</td>
<td>5 char / sec</td>
<td>6.4 or 12.8 sec</td>
</tr>
<tr>
<td>1000 Hz</td>
<td>10 baud</td>
<td>10 char / sec</td>
<td>3.2 or 6.4 sec</td>
</tr>
<tr>
<td>2000 Hz</td>
<td>20 baud</td>
<td>20 char / sec</td>
<td>1.6 or 3.2 sec</td>
</tr>
</tbody>
</table>

For each mode the interleave factor can be doubled thus each character becomes spread over twice as long period of time.
The MT63 modem is made for single sideband operation. The audio generated by the modem (sound card output) is applied to the SSB modulator. On the receiver side, the output of the SSB demodulator is put into the sound card input. The envelope of the MT63 signal is not constant as in other multi-tone systems - it is rather noise-like. One must be careful not to overdrive the transmitter.

The receiver of the MT63 is self-tuning and self-synchronizing thus the radio operator is only required to tune into the signal with +/- 100 Hz accuracy. The modem will tell the actual frequency offset after it is synchronized. The operator should not try to correct this offset unless he is able to tune the radio receiver very slowly, because MT63 as a low rate phase modulated system cannot tolerate sudden frequency changes.

The MT63 is a synchronous system and it relies on the sampling rate to be the same at the receiver and the transmitter. At least the sampling rates should not be different by more that $10^{-4}$.

If you have calibrated your sound card to WWV then you will meet this requirement.

**MT63 Modes Description**

**General Description**

MT63 is an Orthogonal Frequency Division Multiplexed (OFDM) mode consisting of 64 parallel carriers each carrying part of the transmitted signal. The tones are differential BPSK modulated. MT63 employs a unique highly redundant Forward Error Correction system which contributes to its legendary robustness in the face of interference and fading. The tones have synchronous symbols, and are raised cosine modulated. The mode requires a very linear transmitter. Overdriving leads to excessive bandwidth and poorer reception. The mode is unpopular with many operators because of the excessive bandwidth used.

The mode is very tolerant of tuning, as most software will handle 100Hz mistuning. The mode was designed by Pawel SP9VRC.

**Protocol**

These are unconnected, manually controlled message asynchronous symbol synchronous simplex chat modes, with full-time Forward Error Correction. The 64 PSK tones are spaced at 1/64th of the bandwidth, for example 15.625Hz for the default MT63-1000 (long) mode.

**Coding and Character Set**

The ASCII-128 user interface is mapped to 64 tones via a Walsh-Hadamard function (rate 7/64), and uses an interleaver to spread each character across 32 or 64 symbols. Character mapping is direct (via the coding function and interleaver, there being no scrambler or gray coder).

There are three modes, which are scaled in all dimensions (baud rate, spacing, typing speed), and two interleaver options (short and long). One shortcoming of the mode is that robustness is compromised with the short interleaver, and latency (delay between overs) is unacceptably long with the long interleaver. For example, with the default mode the latency is 12.8 seconds.
## MT63 Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Duty Cycle</th>
<th>Modulation</th>
<th>Bandwidth</th>
<th>ITU Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT63-500</td>
<td>5.0 baud</td>
<td>5.0 cps (50 wpm)</td>
<td>80%</td>
<td>64 x 2-PSK</td>
<td>500 Hz</td>
<td>500HJ2DEN</td>
</tr>
<tr>
<td>MT63-1000</td>
<td>10.0 baud</td>
<td>10.0 cps (100 wpm)</td>
<td>80%</td>
<td>64 x 2-PSK</td>
<td>1000 Hz</td>
<td>1K00J2DEN</td>
</tr>
<tr>
<td>MT63-2000</td>
<td>20 baud</td>
<td>20.0 cps (200 wpm)</td>
<td>80%</td>
<td>64 x 2-PSK</td>
<td>2000 Hz</td>
<td>2K00J2DEN</td>
</tr>
</tbody>
</table>

**Notes:**

1. WPM is based on an average 5 characters per word, plus word space.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the “Necessary Bandwidth” as defined by the ITU.
5. Default and normal calling mode (long interleaver).

---

Copyright © M. Greenman 2008
Olivia

fldigi can operate on the following Olivia modes without special setup by the operator:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Olivia 8-250</td>
<td>31.25 baud</td>
<td>1.46 cps (14.6 wpm)</td>
<td>250 Hz</td>
</tr>
<tr>
<td>Olivia 8-500</td>
<td>62.5 baud</td>
<td>2.92 cps (29.2 wpm)</td>
<td>500 Hz</td>
</tr>
<tr>
<td>Olivia 16-500</td>
<td>31.25 baud</td>
<td>1.95 cps (19.5 wpm)</td>
<td>500 Hz</td>
</tr>
<tr>
<td>Olivia 32-1000</td>
<td>31.25 baud</td>
<td>2.44 cps (24.4 wpm)</td>
<td>1000 Hz</td>
</tr>
</tbody>
</table>

Unusual combinations of symbol rate and bandwidth can be selected using the Olivia configuration tab.

These are unconnected, simplex chat modes with full time Forward Error Correction. Olivia is a very robust mode with low error rates, but the penalty can be an annoyingly slow transfer of information. If you are a one finger typist then Olivia is your cup of tea. The tones are spaced the same as the baud rate, for example 31.25 Hz for the default baud rates. The default calling mode is 32-1000. It has the following appearance on fldigi's waterfall:

Excerpts from the web pages of Gary, WB8ROL http://www.oliviamode.com/

Oliva Mode is a little different than PSK, RTTY, and many other digital modes. Below are tips on how to maximize your use of this mode.

Disable your software squelch or turn it down as low as you can
Generally turn your squelch setting in your software off or set it as low as it will go. You will see some “garbage” letters get printed out if there is NO Olivia signal present but it doesn't harm anything. When an Olivia signal is there it will start decoding it and print out the text without garbage at that time. ItFldigiHelp-3.21/Modes/index.htm doesn't do much good to use a digital mode like Olivia that can
decode signals -14 db below the noise level IF you squelch it AT the noise level! It would be like getting a pair of high power binoculars and using them only in a 10x10 room with no windows.

**Be Patient!**
When you call CQ on this mode be patient and wait at least 45-60 seconds before you put out another call. When the other person who hears your CQ clicks on the waterfall it may take 4-20 seconds or even longer before they might actually start decoding your signal. That varies a lot depending on the software they are using AND value they have their Sync Integration Period set to.

The Sync Integration Period setting determines how "deep" the Olivia decoding algorithm searches in the noise to get the signal. A higher setting takes longer BUT usually decodes with more accuracy - at least to a point. However, a higher setting (since it does more work and takes longer) will increase the delay factor. So, when you finish your CQ and your transmitter switches to receive - the station listening to you (depending on his Sync Integration Period setting) MAY NOT finish decoding your CQ for another 4-20 seconds. The same applies during a QSO when you pass it back to the other guy -- be patient if he doesn't come back right away because his software may still be decoding your signal long after you stopped transmitting.

It DOES NOT PAY to be impatient on this mode and send SHORT CQ's or NOT wait at least 45-60 seconds between CQ's. Generally a a 2x2 CQ sent at least 2 or 3 times is going to work much better for you than a short one. Below is the normal CQ I use though on real fast Olivia formats (like 500/4) I will do a 3x3 and send it 3 times.

CQ CQ de WB8ROL WB8ROL
CQ CQ de WB8ROL WB8ROL
CQ CQ de WB8ROL WB8ROL pse K

**Don't set your Sync Integration Period setting TOO high**

If you set your Sync Integration Period too high it MAY take minutes before your software will start decoding a signal AND there is no or little benefit to doing that past a certain point. I usually set mine so that the delay factor is abut 15-20 seconds. I can time this delay factor by sending a very short test and then when it is done and the software switches back to receive - time the number of seconds before you see random garbage start appearing on the screen (assuming you have your SQUELCH OFF). For the standard Olivia modes like 2000/64, 1000/32, 500/16, 250/8, and 125/4 that usually means my Sync Integration Period is set between 3-5 most of the time. If I use the faster formats I set it higher often between 6-10. As long as my delay factor is approx 15-20 seconds. Any higher than that and I don't see any real improvement in the quality of the decoding. But play with your own settings and see what does best for you. If you leave it always on one setting, though, and use standard and non standard formats of Olivia you are short changing yourself.

**Generally keep your Search (Tune Margin) setting to about 8**

The setting of 8 is usually good for most situations and this setting is usually not all that critical. However, under a few band conditions it might (or might not) help to temporarily adjust this. If you
find other Olivia signals very very close to you - almost adjacent or even overlapping it might help to reduce this setting to 4 or even 2. This setting determines how far, either side of your center frequency, Olivia will search for a signal to decode. If you reduce this when another Olivia signal is close or overlapping it may keep it from locking onto the other signal instead of yours. Also ... if you are trying to decode an extremely weak signal and can't even tell exactly WHERE to click on the waterfall because the trace is too faint or non existent then it might help to increase this setting to 16 or 32 temporarily. Then it would perhaps decode the signal even if you were OFF his center frequency by a large margin.

If the slow speed of Olivia bothers you some ...

If you find yourself wanting things to go a little faster then start using more (ham) common abbreviations like "hw" for how and "ur" for your. Don't waste time sending words like "the" and "and" all the time. An example : The weather here is nice and sunny today and the high will get to 85 degrees --- instead send : Wx nice + sunny - high 85 deg -- No need to spell out everything and use superfluous words like the, and, many others. And why use words like HERE and TODAY in the above context when the other station already knows you are telling the weather for YOUR QTH for TODAY. You aren't writing a novel, an article, or in a spelling bee. Also after you establish the QSO don't send BOTH calls all the time at the beginning and end of every transmission. After the QSO is in progress come back to the station like this : .. de WB8ROL -- instead of : W9ZZZ de WB8ROL -- and when you sent it back to the other guy send : BTU - de WB8ROL KN -- That will help speed things up too. You don't need to send the other stations call sign continually to fulfill your legal obligation to indentify your own station.

Don't be afraid to switch to a NON standard Olivia format if conditions warrant it.

If signals are real strong and you prefer to be sending and receiving at a faster speed - don't be afraid to ask the other station if they would like to speed things up and switch to another Olivia format - even a non-standard one. If you, for instance, were talking to me on 500/16 Olivia format and we both had very strong signals and not much QRM, QRN, etc. then ask me if I would like to go to 500/8 format or even 500/4 format. 500/16 format is approximately 20wpm while 500/8 is close to 30wpm and 500/4 close to 40wpm. If you do end up switching to the faster modes you may also want to increase your Sync Integration Period setting substantially too - to maintain the best quality decoding. If not, you might get more errors in the decoded text. And if the band conditions become worse - go back to the original format AND remember to reset your Sync Integration Period setting or the delay in decoding will be way too long! Also, if the band starts getting real crowded and say, for example, you were on 500/16 mode - you might suggest to the other station to switch to 250/4 mode (increase Sync Integration Period setting too) to save space and be a "good neighbor" to all the other operators nearby. 250/4 is the SAME speed as 500/16 and nearly as sensitive with the correct settings.

Olivia Modes Description

General Description

Olivia is a family of MFSK modes with a high redundancy Forward Error Correction system similar to MT63. The family is very large, with 40 or more different options, which can make it very difficult to work out which is which. The mode works well on poor HF paths and has good sensitivity, and is best
used for fixed skeds.

The most widely used versions have a symbol rate of 31.25 baud. The typing speed is varied by changing the number of tones used, but can also be changed by change of baud rate. RTTYM and Contestia are variants of Olivia. The MFSK is constant phase, and the transmission constant amplitude, so transmitter linearity is inimportant. The modes are moderately tolerant of mis-tuning. The typing rate is very modest, given the high bandwidth used.

The mode was designed by Pawel SP9VRC.

Protocol

These are unconnected, manually controlled message asynchronous symbol synchronous simplex chat modes, with full-time Forward Error Correction. The tones are spaced the same as the baud rate, for example 31.25Hz for the 31.25 baud modes. The default mode has 32 tones and is 1kHz wide.

Coding and Character Set

The ASCII-128 user interface is mapped to a block of sequential MFSK tones via a Walsh-Hadamard function (rate 7/64). Tones are deployed via a gray coder.

There are three popular modes, which have 8-FSK, 16-FSK and 32-FSK, thus having three, four or five bits per symbol. The modes have two serious shortcomings - excessive bandwidth combined with slow typing rate, and excessive latency (delay between overs).

Olivia Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed1</th>
<th>Lowest S/N</th>
<th>Duty Cycle2</th>
<th>Modulation</th>
<th>Bandwidth3</th>
<th>ITU Designation4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-250</td>
<td>62.5 baud</td>
<td>~ 20 wpm</td>
<td>-12 dB</td>
<td>100%</td>
<td>4-FSK</td>
<td>250 Hz</td>
<td>250HF1B</td>
</tr>
<tr>
<td>8-250</td>
<td>31.25 baud</td>
<td>~ 15 wpm</td>
<td>-14 dB</td>
<td>100%</td>
<td>8-FSK</td>
<td>250 Hz</td>
<td>250HF1B</td>
</tr>
<tr>
<td>4-500</td>
<td>125 baud</td>
<td>~ 40 wpm</td>
<td>-10 dB</td>
<td>100%</td>
<td>4-FSK</td>
<td>500 Hz</td>
<td>500HF1B</td>
</tr>
<tr>
<td>8-500</td>
<td>62.5 baud</td>
<td>~ 30 wpm</td>
<td>-11 dB</td>
<td>100%</td>
<td>8-FSK</td>
<td>500 Hz</td>
<td>500HF1B</td>
</tr>
<tr>
<td>16-500</td>
<td>31.25 baud</td>
<td>~ 20 wpm</td>
<td>-13 dB</td>
<td>100%</td>
<td>16-FSK</td>
<td>500 Hz</td>
<td>500HF1B</td>
</tr>
<tr>
<td>8-1000</td>
<td>125 baud</td>
<td>~ 58 wpm</td>
<td>-7 dB</td>
<td>100%</td>
<td>8-FSK</td>
<td>1000 Hz</td>
<td>1K00F1B</td>
</tr>
<tr>
<td>16-1000</td>
<td>62.5 baud</td>
<td>~ 40 wpm</td>
<td>-10 dB</td>
<td>100%</td>
<td>16-FSK</td>
<td>1000 Hz</td>
<td>1K00F1B</td>
</tr>
<tr>
<td>32-1000</td>
<td>31.25 baud</td>
<td>~ 24 wpm</td>
<td>-12 dB</td>
<td>100%</td>
<td>32-FSK</td>
<td>1000 Hz</td>
<td>1K00F1B</td>
</tr>
</tbody>
</table>

Notes:
1. WPM is based on an average 5 characters per word, plus word space.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the "Necessary Bandwidth" as defined by the ITU.
## PSK

Fldigi supports the following Phase Shift Keying formats:

### Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed&lt;sup&gt;1&lt;/sup&gt;</th>
<th>Duty Cycle&lt;sup&gt;2&lt;/sup&gt;</th>
<th>Bandwidth&lt;sup&gt;3&lt;/sup&gt;</th>
<th>ITU Designation&lt;sup&gt;4&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK31&lt;sup&gt;5&lt;/sup&gt;</td>
<td>31.25 baud</td>
<td>50 wpm</td>
<td>~ 80%</td>
<td>62.5 Hz</td>
<td>63H0G1B</td>
</tr>
<tr>
<td>BPSK63</td>
<td>62.5 baud</td>
<td>100 wpm</td>
<td>~ 80%</td>
<td>125 Hz</td>
<td>125HG1B</td>
</tr>
<tr>
<td>BPSK125</td>
<td>125 baud</td>
<td>200 wpm</td>
<td>~ 80%</td>
<td>250 Hz</td>
<td>250HG1B</td>
</tr>
<tr>
<td>BPSK250</td>
<td>250 baud</td>
<td>400 wpm</td>
<td>~ 80%</td>
<td>500 Hz</td>
<td>500HG1B</td>
</tr>
<tr>
<td>BPSK500</td>
<td>500 baud</td>
<td>800 wpm</td>
<td>~ 80%</td>
<td>1000 Hz</td>
<td>1000HG1B</td>
</tr>
<tr>
<td>QPSK31</td>
<td>31.25 baud</td>
<td>50 wpm</td>
<td>~ 80%</td>
<td>62.5 Hz</td>
<td>63H0G1B</td>
</tr>
<tr>
<td>QPSK63</td>
<td>62.5 baud</td>
<td>100 wpm</td>
<td>~ 80%</td>
<td>125 Hz</td>
<td>125HG1B</td>
</tr>
<tr>
<td>QPSK125</td>
<td>125 baud</td>
<td>200 wpm</td>
<td>~ 80%</td>
<td>250 Hz</td>
<td>250HG1B</td>
</tr>
<tr>
<td>QPSK250</td>
<td>250 baud</td>
<td>400 wpm</td>
<td>~ 80%</td>
<td>500 Hz</td>
<td>500HG1B</td>
</tr>
<tr>
<td>QPSK500</td>
<td>500 baud</td>
<td>800 wpm</td>
<td>~ 80%</td>
<td>1000 Hz</td>
<td>1000HG1B</td>
</tr>
<tr>
<td>PSK63FEC</td>
<td>62.5 baud</td>
<td>55 wpm</td>
<td>~ 80%</td>
<td>125 Hz</td>
<td>125HG1B</td>
</tr>
<tr>
<td>PSK125R</td>
<td>125 baud</td>
<td>110 wpm</td>
<td>~ 80%</td>
<td>250 Hz</td>
<td>250HG1B</td>
</tr>
<tr>
<td>PSK250R</td>
<td>250 baud</td>
<td>220 wpm</td>
<td>~ 80%</td>
<td>500 Hz</td>
<td>500HG1B</td>
</tr>
<tr>
<td>PSK500R</td>
<td>500 baud</td>
<td>440 wpm</td>
<td>~ 80%</td>
<td>1000 Hz</td>
<td>1000HG1B</td>
</tr>
</tbody>
</table>

These are narrow band low symbol rate modes using single carrier differential Binary Phase Shift Keying, BPSK, or Quadrature Phase Shift Keying, QPSK.

PSK63FEC and the PSKxxxR modes are forward error correcting modes. PSK63FEC is compatible with the MultiPsk mode of the same name. The PSKxxxR, or robust, modes use both forward error correction and interleaving to achieve about 4 dB s/n improvement over standard PSK. These modes are use primarily by the PskMail user community. They are the invention of John Douyere, VK2ETA, a member of the fldigi development team.

In addition to the binary phase shift keying the signal is 100% raised-cosine amplitude modulated at the symbol rate. This reduces the power to zero at the phase change. Because of this amplitude modulation, the signal bandwidth is relatively narrow. Synchronization at the receiver is straightforward because it can be recovered from the amplitude information. Differential PSK is used to provide continuous phase changes when idle (to maintain sync), and by allowing the receiver to measure phase difference from symbol to symbol, to reduce the effects of ionospheric Doppler phase changes which modulate the signal. The slower modes are more affected by Doppler, and the QPSK modes are particularly affected.

With no interleaver and limited coding length, the QPSK mode Forward Error Correction coding gain is limited, and under burst noise conditions on HF the performance is usually worse than the BPSK.
option at the same baud rate. In general the narrow-band BPSK modes work well on a quiet single-hop path, but give poor performance in most other conditions.

PSK63 signal transmitting text data - oscilloscope / waterfall views

QPSK63 signal transmitting text data - oscilloscope / waterfall views

The two oscilloscope views above clearly show the combined phase and amplitude modulation of these modes.

With these modes, a very linear transmitter is required. Over-driven operation results in excessive bandwidth, poorer reception and difficult tuning. Overdrive usually occurs by having the audio signal much too large. The Sights & Sounds Page (http://www.w1hkj.com/FldigiHelp-3.20/Modes/index.htm) has demonstrations of overdriven PSK signals. These are very sensitive modes and usually very little power is required. QRP operation of 80, 40, 30 and 20 meters can provide nearly 100% copy over multi-hop paths. In many instances PSK can provide better decoding than CW.

Setting up for a good clean on air signal that will receive the accolades of your qso partners is easy. Follow the instructions on using the tune button and you will have a clean on signal.

Good reception of PSK signals requires that the demodulator be phase locked to the incoming signal. Fldigi has both a fast acquire / slow tracking AFC system. Place the red bandwidth bar (see above) so that it overlies the desired signal and then press the left mouse button. The signal should quickly lock on a decoding should commence immediately. It is almost impossible to visually tell whether a BPSK or QPSK signal is being received. Under very high s/n you might be able to hear the difference, but that is even difficult for most operators. If you are not able to decode a signal that looks like a BPSK and the bandwidth of the signal matches the baud rate then it might be a QPSK signal. Just change mode a try reacquiring the signal.

PSK Modes Description

General Description

Narrow band modes such as PSK31 are low symbol rate, single carrier differential Binary PSK (called 2-PSK or BPSK) or Quadrature PSK (4-PSK or QPSK). With digital phase modulation the phase changes abruptly, and without additional measures wide sidebands would be created. To prevent this, all these modes also include 100% raised-cosine amplitude modulation (ASK) at the symbol rate,
which reduces the power to zero at the phase change.

Because of this amplitude modulation, the signal bandwidth is relatively narrow. Synchronization at the receiver is straightforward because it can be recovered from the amplitude information. Differential PSK is used to provide continuous phase changes when idle (to maintain sync), and by allowing the receiver to measure phase difference from symbol to symbol, to reduce the effects of ionospheric Doppler phase changes which modulate the signal. The slower modes are more affected by Doppler, and the QPSK modes are particularly affected.

With no interleaver and limited coding length, the QPSK mode Forward Error Correction coding gain is limited, and under burst noise conditions (HF) the performance is usually worse than the BPSK option at the same baud rate. In general the narrow-band BPSK modes work well on a quiet single-hop path, but give poor performance in most other conditions.

To counter this lack of robustness under adverse conditions, the PSKR ("R" for "robust"), series of modes has been developed. Using a similar design as the MFSK modes with a convolutional encoder and an interleaver these modes provide a much more robust link at the expense of the data speed which is divided by half when compared to the standard BPSK mode. Soft bits decoding was also added to maximize the probabilities of decoding the right sequence.

With these modes, a very linear transmitter is required. Over-driven operation results in excessive bandwidth, poorer reception and difficult tuning. However, the sensitivity is such that very little power is usually required.

**Protocol**

The BPSK modes are unconnected, manually controlled message asynchronous simplex chat modes, used without Forward Error Correction. The default calling mode is BPSK31. PSK63F is a mode ideally suited for keyboard to keyboard chat. The PSK-R modes are designed for use with data transfer applications such as pskmail, flarq or other automated-repeat-request applications.

**Coding and Character Set**

A binary varicode with ASCII-256 user interface is used. Lower case characters are sent faster. Modulation is bit-wise symbol synchronous, differential.

The QPSK modes use binary convolution to generate two dibits per varicode bit at the same symbol rate. Rate R=1/2, Constraint length K=5. No interleaver is used. Two-bit quadrature modulation is based on a differential code table.

The PSKR modes use binary convolution to generate two bits per varicode bit at the same symbol rate. These modes use Rate R=1/2, Constraint length K=7 with diagonal interleavers, variable in size to keep the timing between redundant information constant. The number of interleavers for each baud rate is 40, 80 and 160 for PSK125R, PSK250R and PSK500R respectively. PSKR modes use the MFSK varicode which provides a small increase in average speed of about 13%.

The PSK63FEC (also PSK63F in short) is identical to the PSKR modes but for the absence of an interleaver.
### PSK Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed(^1)</th>
<th>Duty Cycle(^2)</th>
<th>Bandwidth(^3)</th>
<th>ITU Designation(^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK3(^5)</td>
<td>31.25 baud</td>
<td>50 wpm</td>
<td>~ 80%</td>
<td>62.5 Hz</td>
<td>63H0G1B</td>
</tr>
<tr>
<td>BPSK63</td>
<td>62.5 baud</td>
<td>100 wpm</td>
<td>~ 80%</td>
<td>125 Hz</td>
<td>125HG1B</td>
</tr>
<tr>
<td>BPSK125</td>
<td>125 baud</td>
<td>200 wpm</td>
<td>~ 80%</td>
<td>250 Hz</td>
<td>250HG1B</td>
</tr>
<tr>
<td>BPSK250</td>
<td>250 baud</td>
<td>400 wpm</td>
<td>~ 80%</td>
<td>500 Hz</td>
<td>500HG1B</td>
</tr>
<tr>
<td>BPSK500</td>
<td>500 baud</td>
<td>800 wpm</td>
<td>~ 80%</td>
<td>1000 Hz</td>
<td>1000HG1B</td>
</tr>
<tr>
<td>QPSK31</td>
<td>31.25 baud</td>
<td>50 wpm</td>
<td>~ 80%</td>
<td>62.5 Hz</td>
<td>63H0G1B</td>
</tr>
<tr>
<td>QPSK63</td>
<td>62.5 baud</td>
<td>100 wpm</td>
<td>~ 80%</td>
<td>125 Hz</td>
<td>125HG1B</td>
</tr>
<tr>
<td>QPSK125</td>
<td>125 baud</td>
<td>200 wpm</td>
<td>~ 80%</td>
<td>250 Hz</td>
<td>250HG1B</td>
</tr>
<tr>
<td>QPSK250</td>
<td>250 baud</td>
<td>400 wpm</td>
<td>~ 80%</td>
<td>500 Hz</td>
<td>500HG1B</td>
</tr>
<tr>
<td>QPSK500</td>
<td>500 baud</td>
<td>800 wpm</td>
<td>~ 80%</td>
<td>1000 Hz</td>
<td>1000HG1B</td>
</tr>
<tr>
<td>PSK63FEC</td>
<td>62.5 baud</td>
<td>55 wpm</td>
<td>~ 80%</td>
<td>125 Hz</td>
<td>125HG1B</td>
</tr>
<tr>
<td>PSK125R</td>
<td>125 baud</td>
<td>110 wpm</td>
<td>~ 80%</td>
<td>250 Hz</td>
<td>250HG1B</td>
</tr>
<tr>
<td>PSK250R</td>
<td>250 baud</td>
<td>220 wpm</td>
<td>~ 80%</td>
<td>500 Hz</td>
<td>500HG1B</td>
</tr>
<tr>
<td>PSK500R</td>
<td>500 baud</td>
<td>440 wpm</td>
<td>~ 80%</td>
<td>1000 Hz</td>
<td>1000HG1B</td>
</tr>
</tbody>
</table>

**Notes:**

1. WPM is based on an average 5 characters per word, plus word space. Values based on sending 100 "paris " words.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the "Necessary Bandwidth" as defined by the ITU.
4. A summary of the ITU Designation system can be found at
5. Default and normal calling mode.

---

*Copyright © M. Greenman 2008, J. Douyere 2009*
**RTTY**

fldigi can operate on a wide range of RTTY symbol rates and bandwidths. The selection of symbol rate and bandwidth is made on the RTTY configuration tab. The three most common in amateur radio use can be selected from the mode menu. These are

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTTY 45</td>
<td>45.45 baud</td>
<td>6.0 cps (60 wpm)</td>
<td>270 Hz</td>
</tr>
<tr>
<td>RTTY 50</td>
<td>50.0 baud</td>
<td>6.6 cps (66 wpm)</td>
<td>270 Hz</td>
</tr>
<tr>
<td>RTTY 75</td>
<td>75.0 baud</td>
<td>10.0 cps (100 wpm)</td>
<td>370 Hz</td>
</tr>
</tbody>
</table>

These modes were a result of mechanical and electrical designs of the early TTY machines. The 45.45 baud and 75 baud machines were for the US / Canadian market and used 60 Hz synchronous motors. The 50 baud machines were for the European market and used 50 Hz synchronous motors.

fldigi can encode and decode many other symbol rates and bandwidths. "Custom" combinations are set up on the RTTY configuration tab. You probably will never have to do that unless you like experimenting with unusual RTTY modes.

**AFSK is not FSK**

All of the modem signals that fldigi produces are audio signals. That includes the RTTY signal. fldigi can encode and decode an RTTY signal that is anywhere within the passband of the sideband transceiver. It is not limited to the traditional tone pairs around 2100 Hz. The following screen captures clearly show three side-by-side RTTY-45 signals with the middle one being tracked correctly. These signals were generated using fldigi's ability to save audio waveforms. The three RTTY signals were combined and then white noise added to create the three signals with s/n of approximately 10 dB.

The decoding on any of the three signals was equal and very near 100% print. The decoder uses a hysterisis detector to help in noise burst rejection. It also uses AFC to track the signal. If a nearby CWI or RTTY signal drags the AFC you can disable it with the AFC button on the fldigi main dialog. Fldigi uses a DSP bandpass filter to reduce interference. The width of the filter can be set on the rtty configuration tab or by positioning mouse pointer in the waterfall; pressing the control key; and rotating the mouse-wheel.

The slider and the red bar in the frequency scale correspond.

To start decoding a signal simply left click on the signal and the AFC should lock on to the signal.
DSP filtering is applied to the audio before decoding takes place. That process helps to reject nearby interference, other RTTY signals or CWI. When a baud rate / shift combination is selected the optimum filter bandwidth is computed and the Receive filter bandwidth is set to that value. You might find a different value gives better performance, especially if you are using a narrow band transceiver filter. You will need to reset the DSP filter each time you reselect the RTTY modem.

The digiscope display will extinguish when the Rx signal level falls below the squelch setting.

You must operate your transceiver in the USB mode for the RTTY signal to be the correct polarity. You must also observe the requirement to maintain linearity in the transmit path.

It is possible to use fldigi to generate the keying waveform for use with an FSK type of transmitter. See Pseudo FSK for a description of how this can be accomplished.

**RTTY Modes Description**

**General Description**

RTTY has been used by radio amateurs since the 1950s. Initially an electromechanical system designed for use on telephone wires, it was not conceived as a radio system, and could not be used by radio until the development of the Ratio Detector during the 1939-1945 war. RTTY (the name means simply Radio Teletype) uses FSK to avoid noise on the transmission path, but requires high power and is still prone to propagation effects, especially selective fading and multi-path timing.

Early RTTY equipment used separate oscillators for each of the tones, and so could produce very wide key clicks, requiring extra filters. Modern software uses phase coherent switching between tones, which somewhat improves the signal bandwidth.

With no error correction, and a start-stop system that is prone to false starts on noise, RTTY is not the best mode for amateur use. However, it is easy to use, easy to tune, fast, tolerant of drift, and is widely used for contesting for these reasons alone. A linear transmitter is not required.

**Protocol**

RTTY is an unconnected, manually controlled message asynchronous character asynchronous simplex chat mode, used without Forward Error Correction. The most widely used and default calling mode is RTTY 45 (45.45 baud). Other less common speeds are 50 baud and 75 baud. The shift is usually 170 Hz, with the upper tone used for idle condition (MARK). Commercial systems operate 425 or 850Hz shift.

**Coding and Character Set**

The ITA-2 character set is used. This has two 'shifts', one for letters, the other for figures and punctuation, a total of 60 characters. There is no lower case. Modulation is direct 2-FSK, one data bit per symbol. Each character is transmitted serially, preceded by an equal length start bit and followed by a stop bit of the opposite sense at least 1.5 data bits long. Receiver synchronism is from the leading edge of the start bit and independently timed from there.

**RTT Operating Parameters**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Duty Cycle</th>
<th>Bandwidth</th>
<th>ITU Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTTY 45</td>
<td>45.45 baud</td>
<td>6.0 cps (60 wpm)</td>
<td>100%</td>
<td>270 Hz</td>
<td>270HF1B</td>
</tr>
<tr>
<td>RTTY 50</td>
<td>50.0 baud</td>
<td>6.6 cps (66 wpm)</td>
<td>100%</td>
<td>270 Hz</td>
<td>270HF1B</td>
</tr>
<tr>
<td>RTTY 75</td>
<td>75.0 baud</td>
<td>10.0 cps (100 wpm)</td>
<td>100%</td>
<td>370 Hz</td>
<td>370HF1B</td>
</tr>
</tbody>
</table>
Notes:
1. WPM is based on an average 5 characters per word, plus word space.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the "Necessary Bandwidth" as defined by the ITU.
5. Default and normal calling mode.
Thor

Thor is a new forward error correcting incremental frequency shift keyed communications mode. It was developed specifically to meet the needs of ARQ transfers in the HF spectrum. It is particularly well suited under conditions of atmospheric static noise. Thor borrows from two current modem technologies, MFSK and DominoEX. Fldigi can operate in the following Thor modes:

- Thor-4 - double spaced mode
- Thor-5 - double spaced mode
- Thor-8 - double spaced mode
- Thor-11 - single spaced mode
- Thor-16 - single spaced mode and
- Thor-22 - single spaced mode

The sound card sampling rate is 8000 Hz for the 4, 8 and 16 modes. It is 11025 Hz for the 5, 11 and 22 modes. This change in sound card sampling rate will be seen in the drop rate on the waterfall. See: Thor Technical Description. Thor emits a distinctive double rising tone sequence at the beginning of each transmission. It is used to flush the receive decoder and also provides a visual and audibal clue to its being used.

The modem code for Thor uses a wide band multiple frequency detector that can lock on and detect the incoming signal even when badly mistuned. Frequency domain oversampling is used to allow proper tone detection with the need for AFC. The AFC control does not alter the decoder in any way.

The waterfall and digiscope will appear as:

![Waterfall and Digiscope](image)

The text displayed in the status area is the secondary text being sent by the transmitting station. When the keyboard buffer is empty the Thor modem transmits text from the secondary text buffer. Your secondary text buffer can be edited on the Thor configuration tab.

The digiscope display is similar to the DominoEX display and represents the tone pairs moving through the tone filters. You can also use an alternate digiscope display (left click on the digiscope display area).
In this display mode the red line represents the center of the multiple tone bins that are in the detector. The dots will be blurry if the AFC is not locked on and become very distinct when AFC lock has been achieved. The tone dots will move from bottom to top (opposite the direction of the waterfall).

This is the same signal mistuned:

![Image of mistuned signal]

and with the signal badly mistuned:

![Image of badly mistuned signal]

**THOR Modes Description**

**General Description**

THOR is a family of offset incremental multi-frequency shift keyed modes with low symbol rate, closely related to DominoEX. A single carrier of constant amplitude is stepped between 18 tone frequencies in a constant phase manner. As a result, no unwanted sidebands are generated, and no special amplifier linearity requirements are necessary. The tones change according to an offset algorithm which ensures that no sequential tones are the same or adjacent in frequency, considerably enhancing the inter-symbol interference resistance to multi-path and Doppler effects.

The mode has full-time Forward Error Correction, so is extremely robust. The default speed (11 baud) was designed for NVIS conditions (80m at night), and other speeds suit weak signal LF, and high speed HF use. The use of incremental keying gives the mode complete immunity to transmitter-receiver frequency offset, drift and excellent rejection of propagation induced Doppler.

**Protocol**

These are unconnected, manually controlled message asynchronous simplex chat modes, using binary convolutional Forward Error Correction. The default calling mode is THOR11.
Coding and Character Set

A binary varicode with ASCII-256 user interface (same as MFSK16) is used. Lower case characters are sent faster. An ASCII-128 secondary character set extension allows a fixed (typically ID) message to be sent whenever the transmitter is idle. Modulation uses two dibit pairs, symbol synchronous, differential.

The FEC system uses binary convolution to generate two dibits per varicode bit, and halves the corrected data rate compared to the equivalent DominoEX mode. Rate R=1/2, Constraint length K=7, Interleaver L=10 (40 bits).

THOR Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed(^1)</th>
<th>Duty Cycle(^2)</th>
<th>Bandwidth(^3)</th>
<th>ITU Designation(^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>THOR4(^5)</td>
<td>3.90625 baud</td>
<td>14 wpm</td>
<td>100%</td>
<td>173 Hz</td>
<td>173HF1B</td>
</tr>
<tr>
<td>THOR5(^5)</td>
<td>5.3833 baud</td>
<td>22 wpm</td>
<td>100%</td>
<td>244 Hz</td>
<td>244HF1B</td>
</tr>
<tr>
<td>THOR8(^5)</td>
<td>7.8125 baud</td>
<td>28 wpm</td>
<td>100%</td>
<td>346 Hz</td>
<td>346HF1B</td>
</tr>
<tr>
<td>THOR11(^6)</td>
<td>10.766 baud</td>
<td>40 wpm</td>
<td>100%</td>
<td>262 Hz</td>
<td>262HF1B</td>
</tr>
<tr>
<td>THOR16</td>
<td>15.625 baud</td>
<td>58 wpm</td>
<td>100%</td>
<td>355 Hz</td>
<td>355HF1B</td>
</tr>
<tr>
<td>THOR22</td>
<td>21.533 baud</td>
<td>78 wpm</td>
<td>100%</td>
<td>524 Hz</td>
<td>524HF1B</td>
</tr>
</tbody>
</table>

Notes:

1. WPM is based on an average 5 characters per word, plus word space. Values based on sending 100 "paris" words.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the "Necessary Bandwidth" as defined by the ITU.
4. A summary of the ITU Designation system can be found at

5. Double spaced mode.
6. Default and normal calling mode.
Throb

The THROB family of modes use two tones at a time. These tones are also amplitude modulated and can be a single tone for some symbols.

The mode has no Forward Error Correction, and is difficult to tune. It is fairly sensitive and moderately robust. Keyboard-to-keyboard QSOs are reasonably fast. Tuning must be very accurate, and the software will not tolerate differences between transmit and receive frequency.

The amplitude modulation component of THROB is a raised cosine AM modulation of each symbol. This combined with two tones transmitted at the same time, means that a very linear transmitter is required. It also gives the mode its very unique sound. You will never mistake Throb for any other mode.

For THROB, nine tones are used, spaced 8 or 16 Hz. For THROBX, 11 tones are used, spaced 7.8125 or 15.625 Hz.

Fldigi supports the following Throb baud rates and tone spacings:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>THROB1</td>
<td>1.0 baud</td>
<td>1.0 cps (10 wpm)</td>
<td>72 Hz</td>
</tr>
<tr>
<td>THROB2</td>
<td>2.0 baud</td>
<td>2.0 cps (20 wpm)</td>
<td>72 Hz</td>
</tr>
<tr>
<td>THROB4</td>
<td>4.0 baud</td>
<td>4.0 cps (40 wpm)</td>
<td>144 Hz</td>
</tr>
<tr>
<td>THROBX1</td>
<td>1.0 baud</td>
<td>1.0 cps (10 wpm)</td>
<td>94 Hz</td>
</tr>
<tr>
<td>THROBX2</td>
<td>2.0 baud</td>
<td>2.0 cps (20 wpm)</td>
<td>94 Hz</td>
</tr>
<tr>
<td>THROBX4</td>
<td>4.0 baud</td>
<td>4.0 cps (40 wpm)</td>
<td>188 Hz</td>
</tr>
</tbody>
</table>

THROB Modes

General Description

The THROB family of modes are MFSK in nature, but are unusual in that (like DTMF) they use two tones at a time. Unique among other MFSK modes, the THROB family also uses amplitude modulation and sometimes sends just one tone!

The mode has no Forward Error Correction, is difficult to tune, but reasonably sensitive and moderately robust. Because of the limited character set however, the typing speed is reasonably good, one character per symbol. Tuning must be very accurate, and the software will not tolerate differences between transmit and receive frequency. THROB is essentially a curiosity mode.

Another unusual feature (which led to the name), THROB also includes raised cosine AM modulation of each symbol. This combined with two tones transmitted at the same time, means that a very linear transmitter is required.

For THROB, nine tones are used, spaced 8 or 16 Hz. For THROBX, 11 tones are used, spaced 7.8125 or 15.625 Hz. The THROB family was developed by Lionel G3PPT.
Protocol

These are unconnected, manually controlled message asynchronous symbol synchronous simplex chat modes, with no Forward Error Correction. The most frequent calling mode is THROB2. Reception is non-coherent, and sync is recovered from the amplitude modulation.

Coding and Character Set

A limited 44 character user interface is used, one directly mapped tone combination (symbol) per character, which contributes to the reasonable typing speed. There is no lower case. Modulation is symbol synchronous, and with 45 possible one- or two-tone combinations available, the remaining combination is used for the idle condition. (ThrobX has 53 characters, 11 tones, and never sends single tones).

THROB Operating Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol Rate</th>
<th>Typing Speed1</th>
<th>Duty Cycle2</th>
<th>Modulation</th>
<th>Bandwidth3</th>
<th>ITU Designation4</th>
</tr>
</thead>
<tbody>
<tr>
<td>THROB1</td>
<td>1.0 baud</td>
<td>1.0 cps (10 wpm)</td>
<td>80%</td>
<td>1/2 of 9-FSK</td>
<td>72 Hz</td>
<td>72H0F1B</td>
</tr>
<tr>
<td>THROB25</td>
<td>2.0 baud</td>
<td>2.0 cps (20 wpm)</td>
<td>80%</td>
<td>1/2 of 9-FSK</td>
<td>72 Hz</td>
<td>72H0F1B</td>
</tr>
<tr>
<td>THROB4</td>
<td>4.0 baud</td>
<td>4.0 cps (40 wpm)</td>
<td>80%</td>
<td>1/2 of 9-FSK</td>
<td>144 Hz</td>
<td>144HF1B</td>
</tr>
<tr>
<td>THROBX1</td>
<td>1.0 baud</td>
<td>1.0 cps (10 wpm)</td>
<td>80%</td>
<td>2 of 11-FSK</td>
<td>94 Hz</td>
<td>94H0F1B</td>
</tr>
<tr>
<td>THROBX2</td>
<td>2.0 baud</td>
<td>2.0 cps (20 wpm)</td>
<td>80%</td>
<td>2 of 11-FSK</td>
<td>94 Hz</td>
<td>94H0F1B</td>
</tr>
<tr>
<td>THROBX4</td>
<td>4.0 baud</td>
<td>4.0 cps (40 wpm)</td>
<td>80%</td>
<td>2 of 11-FSK</td>
<td>188 Hz</td>
<td>188HF1B</td>
</tr>
</tbody>
</table>

Notes:

1. WPM is based on an average 5 characters per word, plus word space. Values are approximate because a variable length code is used.
2. Transmitter average power output relative to a constant carrier of the same PEP value.
3. This is the "Necessary Bandwidth" as defined by the ITU.
4. A summary of the ITU Designation system can be found at
5. Default and normal calling mode.

Copyright © M. Greenman 2008
Digital Mode Comparison

Introduction

The performance of different digital modes can be compared in numerous ways. The comparison most expected by uninformed users is 'sensitivity', i.e. performance in the presence of white noise, which can be measured with reasonable repeatability using an ionospheric simulator.

However, this measurement is generally meaningless as so many other factors come into play when considering HF performance - things like the susceptibility to ionospheric Doppler effects, sensitivity to impulse or burst noise, and sensitivity to multi-path reception, where there can be selective fading and timing differences between paths. However, these can also be measured with a simulator, provided care is taken in choosing appropriate simulations.

Several other comparisons prove interesting and certainly less controversial, since the performance of each mode can be calculated. The factors which are easily compared are bandwidth, bandwidth efficiency and coding efficiency.

Propagation Performance

Rather than providing a series of measurements that would be difficult to understand, the information shown below was condensed from a detailed study made of representative popular modes, using the standard CCIR ionospheric simulation known as 'Mid Latitude Disturbed NVIS'. This is a near approximation of poor multi-path conditions, without static or other interference, but with white noise reducing the signal to 0dB S/N in 2.4kHz bandwidth.

It should be stressed that simulation results can be highly subjective, and are usually presented as comparative text results for the observer to assess. A method has been evolved which involves counting the number of correctly received complete words and the total number of words, and giving the ratio the value '% Copy'. This technique has proved to be repeatable. From practical experience a QSO is viable if '% Copy' is on average greater than 90%, and impossible below 70%.

From this graph it is clear that under these conditions (typical of 80m at night) PSK31 gives poor
results, and RTTY is not much better (remember this is for this specific simulation - for others the results will be different). The error corrected modes all perform well. Not how well DominoEX8 performs - with no error correction! This mode was designed for these conditions.

The horizontal axis of this graph is typing speed, and you can see that MT63 is the clear winner. However, the graph tells only part of the story - MT63 also consumes much more bandwidth than most other modes. Olivia is the same bandwidth as MT63, but look at the low typing speed!

![Mode Error Rate vs Channel Efficiency](chart.png)

'Error Rate' as shown in this next graph is simply 100 - '% Copy'. Thus an Error Rate below 10% will give comfortable copy during a QSO. The data has been rearranged to show (on the horizontal axis) a measure of how effectively the mode manages its bandwidth. Clearly PSK31 uses the least bandwidth for its typing speed, while Olivia is the least efficient, but at least has a low error rate.

Again, it must be stressed that the measurements are for 'Mid Latitude Disturbed NVIS' and the modes which perform poorly may well perform better under other conditions. Anyone with two computers and a copy of the AE4JY 'Pathsim' silulator can repeat these measurements for any appropriate simulation conditions.

**Bandwidth**

There is no controversy, subjectivity or choice of measurement conditions involved in comparison of bandwidths of different modes - it all comes down to mathematics and modulation theory. The 'Necessary Bandwidth' is defined by the CCIR using a set of mode-related formulae.
Clearly the narrowest bandwidth mode considered here is good old CW - Morse code, with PSK31 a close second. The widest modes are MT63 and Olivia, both at 1kHz. However, this graph doesn't tell us everything, it is purely arranged in order of bandwidth. Let's look at this bandwidth in relation to the typing speed of each mode.

The modes at the bottom left are the best performers, having a low bandwidth for their typing speeds. MT63 doesn't look so bad in this comparison, as the typing speed is high (but who can type at 100 WPM?). Olivia is the least bandwidth efficient by a wide margin. THROB achieves high efficiency because it has a very small character set, while the MFSK varicoded modes also have good bandwidth efficiency even (in the case of THOR and MFSK16) with FEC halving the typing speed.
Coding Efficiency

The last graph gives an indication of how well each mode transfers data with each transmitted symbol. This measurement is a combination of the size and efficiency of the alphabet used, the type of modulation, and the rate of any FEC used. Clearly modes with low FEC rates have the highest redundancy, and so should have the best error correction performance - but at the cost of data transfer speed.

![Digital Mode Coding Gain](image)

The coding gain term used here is simply a measure of the typing speed compared with the symbol rate. You can see that the mode with the most redundancy is Feld-Hell, effectively 49 symbols per character. MFSK16 and THOR occupy the middle ground, as they have FEC, while DominoEX, with no FEC has twice the coding gain of THOR. The clear winners are THROB and MT63, but for completely different reasons - THROB has an extremely limited character set, and can achieve one character per symbol (the only mode to do so), while MT63 has the advantage of 64 carriers, and so despite the FEC rate of 7/64, achieves a high coding efficiency, of course at the cost of bandwidth.
WEFAX

This modem is able to receive and transmit HF-Fax images, traditionally used for weather reports. You can refer to the wikipedia article Radiofax [http://en.wikipedia.org/wiki/Radiofax](http://en.wikipedia.org/wiki/Radiofax) for more detailed information.

Two modes are implemented IOC=576 or 288. The focus is made on black-and-white images, color mode is still experimental.


Transmitting an Image

To open the transmit window, you must of course select one Wefax modem, and then right-click on the transmit (blue) window:
Now, to start the transmission, you just need to click "Xmt Grey" for black-and-white images, etc... During transmission, image reception is paused. The window will display each image line as it is sent. Please be patient, this may take a while. You might note that FlDigi status line displays the estimation transmit time, and the current stage (Start, phasing etc...)

Then, the transmit window just opens. This is the same logic as sending MFSK images.

Then, you must open an image file using the button "Load". The image is then displayed, for example like that:
Receiving an image

To go into reception mode, one must click the "View" menu tab, and select "Weather Fax Image".

At this time, the reception window opens. A big blank picture is visible, this is where the received image will be displayed. Several controls are available:

- **Save**: This allows to save the current image as a PNG file at any moment.
- **Manual mode/Abort**: At startup, the modem goes into automatic mode, and the text "Abort" is displayed. When clicking "Abort", this blanks the image and resets the APT detection. When "Manual mode" is clicked, no APT detection is done. The Automatic/Manual mode is displayed in the reception window label.
- **Pause/Resume**: At any moment, the image reception can be paused and resumed (State is displayed in the reception window label).
- **Zoom**: This allows to zoom in/out the image.
- **FIR**: This allows to select an input FIR (Finite Impulse Response) filter. Practically, the narrow filter (Default value) gives the best results. The selected value is saved in the configuration file.
- **Skip APT/Skip phasing**: When in automatic mode, this allows to skip detection steps. This is often necessary when the signal is not very good.
When receiving an image, either in manual or automatic mode (APT control), other controls are displayed:

- Max lines: This is the maximum number of lines admissible for a fax. When this number is reached, an image file is automatically saved and the line counter returns to one. Typical fax have about 1300 lines.
- Line: The number of the line currently received.
- Width: The image width in pixels. This is usually 1809, if LPM is 120.
- LPM: Lines per minute: Typically 120, can be 60, 90, 120 or 240. This is detected in automatic mode, but can be manually adjusted.
- Slant: This is used to adjust the slant of the image due to a clock inaccuracy. This value is saved in the configuration file, so it is not needed to reenter it each time.
- Center: This is used to manually adjust the horizontal center of the image, if it could not be detected in the phasing step.
- Auto: When this button is set, the image will be automatically centered. This process starts after some hundredth of lines are received, by shifting left and right the image. It takes some time to converge.
APT control reception mode
This mode uses the APT start and STOP frequencies to detect the beginning and end of an image. Additionally, it tries to detect the phasing signal - a wide black band - to detect the center of the image. This method is helped by the estimation of the signal power of these frequencies.

Manual reception mode
In this mode, the image is continuously read and displayed. When the maximum number of lines is reached, the image is saved and blanked, and the line counter returns to one.

Centering an image
If the phasing was not automatically detected, the modem could not deduce the beginning of an image. The result is an image which is horizontally shifted. To correct this, one can use the "Center" slider.

Picture with a slant
If either the send, receive or both ends of the transmission are using an uncalibrated sound card whose sampling rate is not an exact multiple of the sample rate the resulting picture at the receive end will appear slanted. The degree of slant is directly related to the accumulation of the frequency error at both ends of the transfer. Stations wishing to receive Weather fax pictures should calibrate their sound card. The WWV calibration mode is used to measure and set the parts per million (ppm) correction factor for the sound card.

Your sound system may be fully corrected, but the sending station may have an uncorrected sound card. You can usually correct for small errors during reception by using the slant slider. Its value (Typically between - 0.005 and 0.005) will be stored in fldigi configuration parameters.

Automatic centering.
If the phasing signal could not be used for centering the image, the program waits for a string image signal anyway to go into reception mode, but it sets an internal flag allowing to automatically center the image. This feature can be freely enabled and disabled at any moment. It works by detecting a wide vertical band of about hundred pixels, where the sum of the contrast as the lowest among the complete image width.

That is: It computes for each row and each pixel, the absolute value of the horizontal derivate. It then sums these derivativatives pixel-wise, row by row. Then, it computes an average of about hundred pixels along this single row. The column which has the lowest averaged contrast is considered to be the image margin, which is then shifted on the left of the window.

This method takes some time to stabilize, because at the beginning, there are many areas of the image, without details. It gets stable at the end, when only the margin stays with few constraed details.
Image detection based on signal power.

The APT control - inherited from the Hamfaw signal, does not work very well when the image is noised. On the other hand, fldigi provides ways to evaluate the signal power on a given bandwidth. This is used because APT control relies on the emission on specific frequencies.

Therefore, in the APT start and phasing loops, when check for the presence of strong signals associated to APT control. This information is used to take a decision when the traditionnaly method does not detect anything.

These two methods are interchangeable but used together for better detection.

Saving an image

Received images are saved in the default folder $HOME/.fldigi/images (Linux) or <defaultpath>/fldigi.files/images (Windows).

Additionally, they can be saved manually, at any time, using the button 'Save'. The PNG images received some extra text comments which can be displayed, for example, with GIMP-GNU Image Manipulation Program http://www.gimp.org/

SSB Mode Receive Only mode.

From ChangeLog

...  
commit 0d77c1bedf2d8e4ea5ecbd1dd3b6a49b97aafcae  
Author: David Freese <w1hkj@w1hkj.com>  
Date: Thu Apr 8 12:25:46 2010 -0500

    SSB modem

        Receive only SSB modem added to allow occassional logging of a voice contact :>)

commit 586e3d5ea265853f3676d599d06e204dccaee6c23  
Author: David Freese <w1hkj@w1hkj.com>  
Date: Thu Apr 8 12:22:27 2010 -0500

    QPSK dcd postamble

        Restored CW postamble to QPSK signal encoder

commit 1aae11e61cb3ab535630cf53529d85d916619058  
Author: David Freese <w1hkj@w1hkj.com>  
Date: Thu Apr 8 12:22:06 2010 -0500

    RX_ONLY Modem type

        Added rx_only_ boolean to allow creation of modems that are receive only, such as WWV, Analysis. Default modem type is to allow both rx and tx. ...
Digiscope Display - WWV mode

The WWV mode is used to measure the offset of the sound card oscillator. It does this by comparing the timing loop for the sound card measurements against the clock tick signal that is transmitted by WWV and WWVH. The sampling rate for the sound card should be set to "native". The sound card samples the signal and returns the values in 512 blocks. This block sampling is what sets the basic timing mechanism for the thread that reads the sound card, sends data to the waterfall, and sends data to the modem signal processing functions. A process of filtering is used that simultaneously reduces the sampling rate. Most modern soundcards will use 44100 or 48000 as the native sampling rate. That sample rate in down converted to 1000 using a decimation in time type FIR. The resulting signal is then power detected and further filtered with a filter called a moving average filter. The moving average is very good at detecting the edge of a pulse such as the 1 second tick transmitted by WWV. This output is then displayed in a manner very similar to a FAX signal. Each scan line represents the received signal over a 1 second interval. The bright white line is the time tick. You can see a very slight slope from left to right as the signal goes from top to bottom of the display.

Open the configure dialog box to the "SndCrd" tab. You are going to be adjusting the "Rx corr Rate" while you observe the effect of this control on the slope of the time tick line.

Tune in WWV or WWVH on 2.5, 5.0, 10.0 or 15.0 MHz in the AM mode. This seems to give the best signal view. Select the WWV modem and allow the data to begin to accumulate in the digiscope display. When you can clearly see the bright tick line, move the cursor to the bottom of the line and left click at that position. That will resync the digiscope display and put the ensuing tick marks at the center line red graticule.

Then right click anywhere in the digiscope display. That changes the zoom level to show more detail regarding the slope of the time tick line. The zoom level increases by a factor of 5. Right clicking again restores the original zoom level. I recommend making the adjustments to the Rx corr Rate control in the x5 zoom level.

If the slope of the time tick line is positive you will need to apply a negative value to the Rx corr Rate. If it is negative then a positive correction is needed.

Start with a correction of 0 ppm and observe the slope. Try a value of 1000 ppm and observe the slope. Again, try a -1000 ppm correction and observe the slope. The following are some observations made on 10 MHz WWV, DCF-77 and RWM under less than ideal conditions.
WWV corrected
20 minute trace
5x scale

-1000 ppm WWV
5x scale

0 ppm WWV
5x scale

+1000 ppm WWV
5x scale

+120 ppm WWV
5x scale

0 ppm DCF-77
1x scale

0 ppm DCF-77
5x scale

+1000 ppm DCF-77
1x scale

+65 ppm DCF-77
5x scale

RWM uncorrected
1x scale

RWM
+25361 ppm
1x scale

RWM
+25361 ppm
5x scale
You can see that my sound card requires a positive correction since the slope is negative with a 0 ppm entry. The required correction of +120 ppm was determined by guessing the needed correction to be close to 1/10 of the -1000 ppm slope and then adjusting for a steady track along the red graticule. The DCF-77 images were provided by Walter, DL8FCL. The RWM images were provided by Andy G3TDJ.

You can left click on the tick line anytime you want to recenter the signal. That will aid in making your visual observation.

When you are finished, the Rx corr Rate entry is the correct one for your sound card. Save the configuration for future fldigi use.

Andy also provided information on the RWM transmissions:

RWM details extracted from http://www.irkutsk.com/radio/tis.htm

Station RWM - Main characteristics

Location: Russia, Moscow 55 degr. 44' North, 38 degr. 12' East
Standard frequencies: 4996, 9996 and 14996 kHz
Radiated power: 5kW on 4996 and 9996 kHz; 8kW on 14996 kHz

Period of operation: 24 hours per day, except 08.00-16.00 msk for maintenance as below:
- on 4996 kHz: 1st Wednesday of the 1st month of quarter;
- on 9996 kHz: 2nd Wednesday of the 1st month of the quarter;
- on 14996 kHz: 3rd Wednesday of each odd month;

Coverage: 20 degr. - 120 degr. East
35 degr. - 75 degr. North

Time signals A1X are given every second of 100 ms duration with a frequency of 1 Hz. Minute pip is extended to 500 ms.

Hourly transmission schedule
m s m s
00:00 - 07:55 -- MON signals (no modulation)
08:00 - 09:00 -- transmitter is signed off
09:00 - 10:00 -- station's identification is sent by Morse Code
10:00 - 19:55 -- A1X signals and identification of DUT1+dUT1
20:00 - 29:55 -- DXXXW signals
30:00 - 37:55 -- N0N signals (no modulation)
38:00 - 39:00 -- transmitter is signed off
39:00 - 40:00 -- station's identification is sent by Morse Code
40:00 - 49:55 -- A1X signals and identification of DUT1+dUT1
50:00 - 59:55 -- DXXXW signals
**Frequency Analyzer**

Fldigi can be used to accurately measure the frequency of a remote signal that is transmitting a steady carrier.

I have set the sound card up using the WWV modem and have it adjusted for the proper PPM offset on receive. I followed the frequency calibration procedure that Icom recommends for the IC-746PRO, adjusting WWV at 10 MHz for a zero beat.

Then fldigi was used in the "Freq Analysis" mode to track the WWV carrier at 10 MHz. In this mode the decoder is merely a very narrow band AFC tracking filter. The filter bandwidth is set to 2 Hz and the tracking time constants to about 5 seconds. Future releases will probably make both of these user adjustable. When the signal is being tracked the digiscope (right hand display) will be a horizontal line. If the signal is very noisy and tracking difficult the digiscope will jump and become wavy. You can see from the above image that I am tracking about 0.22 Hz high on WWV. I have repeated this measurement at various times during the day and on various days with nearly the same result. So I am comfortable with knowing that my local oscillator is just a little low (that is why the Frequency reads high).

ARRL frequently announces a frequency measurement test (FMT) which takes place on 160, 80 and 40 meters. This is a chance to test your skills in frequency measurement. You should be able to make a submission to the FMT using this technique. Make corrections to the FMT transmission based upon your WWV measurement. You may have to adjust for other local oscillator effects as well. If you have some good ways to measure and correct for these I would be glad to share them with the other fldigi users.
TUNE Mode

Too often you see an overdriven signals on the digital sub-bands; multiple audio sidebands on PSK, splatter from overdriven MFSK and RTTY. There is absolutely no reason for a transceiver driven by fldigi to exhibit this type of performance. You can set up your computer / transceiver for good solid performance without excessive drive.

The "TUNE" button generates a continuous single frequency audio signal at the exact frequency to which the waterfall cursor has been set. The peak amplitude of this signal is the peak amplitude of every modem signal generated by fldigi. None will exceed this value, even the simultaneous multi-tone modes like Throb. Every modern SSB transmitter uses some automatic level control ALC for preventing overdrive for SSB voice. A little overdrive on a voice channel can be tolerated to a degree. In fact, that is what an analog RF compressor does, overdrive and then subsequent filtering. But you absolutely cannot tolerate that with the digital modes. Here is the way to set up your transceiver for a clean signal. I recommend starting out with a dummy load, but an "off hour" for a band might work just as well if you do not have a dummy load.

- For Windows users
  - Set your sound card output level to the minimum on the Windows mixer
- For Linux users
  - Set your PCM level to about 80%
  - Set your Transmit Levelcontrol for minimum output level.
- Enable the "Tune" mode in fldigi ... you do have CAT or PTT set up ...right?
- Make sure your transceiver's speech compression control is OFF
- Slowly bring up the Mixer audio out until your rig's ALC just starts to function (a light blinking or a meter showing this condition).
- Reduce the Mixer audio output until the ALC is disabled.
- You are now transmitting at maximum output power without distortion.

You can use any level below this and be assured that your output signal will be clean. All digital signals that fldigi generates will be limited to this peak-to-peak voltage. You should always use the minimum power necessary to maintain good comms, remember that even if you are clean at 100 W you signal will be so strong among the QRP signals that it will overpower the AGC on many receivers that are working another digital station within the same SSB bandwidth that you are on. You will appreciate this the first time that you are working a weak PSK DX station and someone blasts through and captures your AGC.

You should try the the above adjustments at different audio frequencies. Transceivers that achieve the SSB filtering with crystal or mechanical filters will have a considerable amount of variation across the passband of the filter. This will show up as a varying amount of ALC that is dependent on the audio frequency. Once you are comfortable with the process you can very quickly repeat the "Tune" and set the power for the frequency to which the waterfall is set.
### QSO Logbook

Fldigi maintains a large set of QSO logbook fields that will probably be sufficient for casual operating, contesting and some certificate logging. All of the fields that are captured in the logbook are maintained in an ADIF database that can be read by any logbook program that can read the ADIF text format.

The complete set of logbook fields are:

<table>
<thead>
<tr>
<th>ADIF FIELD</th>
<th>USE</th>
<th>ADIF FIELD</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BAND</strong></td>
<td>QSO band (computed from frequency)</td>
<td><strong>STATE</strong></td>
<td>* contacted stations state</td>
</tr>
<tr>
<td><strong>CALL</strong></td>
<td>* contacted stations call sign</td>
<td><strong>QSLRDATE</strong></td>
<td>QSL received date</td>
</tr>
<tr>
<td><strong>COMMENT</strong></td>
<td>* comment field for QSO</td>
<td><strong>QSLSDATE</strong></td>
<td>QSL sent date</td>
</tr>
<tr>
<td><strong>COUNTRY</strong></td>
<td>* contacted stations DXCC entity name</td>
<td><strong>STX</strong></td>
<td>* QSO transmitted serial number</td>
</tr>
<tr>
<td><strong>FREQ</strong></td>
<td>* QSO frequency in MHz</td>
<td><strong>SRX</strong></td>
<td>* QSO received serial number</td>
</tr>
<tr>
<td><strong>GRIDSQUARE</strong></td>
<td>* contacted stations Maidenhead Grid Square (Loc)</td>
<td><strong>TIME_OFF</strong></td>
<td>end time of QSO in HHMM format</td>
</tr>
<tr>
<td><strong>MODE</strong></td>
<td>QSO mode</td>
<td><strong>TIME_ON</strong></td>
<td>* start time of QSO in HHMM format</td>
</tr>
<tr>
<td><strong>NAME</strong></td>
<td>* contacted operators name</td>
<td><strong>TX_PWR</strong></td>
<td>* power transmitted by this station</td>
</tr>
<tr>
<td><strong>QSO_DATE</strong></td>
<td>* QSO data at start of contact</td>
<td><strong>IOTA</strong></td>
<td>Islands On The Air</td>
</tr>
<tr>
<td><strong>RST_RCVD</strong></td>
<td>* received signal report</td>
<td><strong>VE_PROV</strong></td>
<td>2 letter abbreviation for Canadian Province</td>
</tr>
<tr>
<td><strong>RST_SENT</strong></td>
<td>* sent signal report</td>
<td><strong>XCHG1</strong></td>
<td>* received contest exchange</td>
</tr>
<tr>
<td><strong>QTH</strong></td>
<td>* contacted stations city</td>
<td><strong>MYXCHG</strong></td>
<td>sent contest exchange</td>
</tr>
<tr>
<td><strong>DXCC</strong></td>
<td>contacted stations DXCC country code</td>
<td><strong>ITUZ</strong></td>
<td>ITU zone</td>
</tr>
<tr>
<td><strong>IOTA</strong></td>
<td>Islands-On-The-Air designator</td>
<td><strong>CQZ</strong></td>
<td>CQ zone</td>
</tr>
</tbody>
</table>

* - these fields are either captured on the main dialog, computed from internal values, or determined by configuration

The data in the fldigi logbook can be exported to external text files; ADIF, text, and CSV (comma separated value). The ADIF can be read by any ADIF compatible logbook program. The text output is suitable for use in a wordprocessor and for printing. The CSV can be read into many spreadsheet programs such as Excel, Open Office or Gnumeric.
Digital modes signal reports

Fldigi does not enforce any rules on signal reporting. It could very well do so for many of the modes in which signal quality is inherently measured as a part of the decoder. Learning how to evaluate a signal, to properly report it, and then help in correcting deficiencies should be the goal of every amateur operator.

RST

Is the traditional Readability, Strength, Tone reporting system used for CW operations for nearly as long as amateurs have enjoyed the airwaves.

READABILITY

1. Unreadable
2. Barely readable, occasional words distinguishable
3. Readable with considerable difficulty
4. Readable with practically no difficulty
5. Perfectly readable (that is 100% print in today's jargon)

SIGNAL STRENGTH

1. Faint signals, barely perceptible
2. Very weak signals
3. Weak signals
4. Fair signals
5. Fairly good signals
6. Good signals
7. Moderately strong signals
8. Strong signals
9. Extremely strong signals

TONE

1. Sixty cycle ac or less, very rough and broad
2. Very rough ac, very harsh and broad
3. Rough ac tone, rectified but not filtered
4. Rough note, some trace of filtering
5. Filtered rectified ac, but strongly ripple modulated
6. Filtered tone, definite trace of ripple modulation
7. Near pure tone, trace of ripple modulation
8. Near perfect tone, slight trace of modulation
9. Perfect tone, no trace of ripple, or modulation of any kind
RSQ
Give the report as RSQ for digital modes, but especially BPSK and QPSK; see: [http://www.psb-info.net/RSQ-Reporting-Table.html](http://www.psb-info.net/RSQ-Reporting-Table.html)

**READABILITY**
1. 0% undecipherable
2. 20% occasional words distinguishable
3. 40% considerable difficulty, many missed characters
4. 80% practically no difficulty, occasional missed characters
5. 95%+ perfectly readable

**STRENGTH**
1. Barely perceptible trace
2. Weak trace
3. Moderate trace
4. Strong trace
5. Very strong trace

**QUALITY**
1. Splatter over much of the visible waterfall
3. Multiple visible pairs
5. One easily visible pair
7. One barely visible pair
9. Clean signal - no visible unwanted sidebar pairs

### Capturing QSO data
Fldigi supports two QSO capture panels. The first for casual QSO logging

<table>
<thead>
<tr>
<th>Rig Not Specified</th>
<th>QSO Freq</th>
<th>On</th>
<th>Off</th>
<th>Call</th>
<th>Name</th>
<th>In</th>
<th>Out</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3580.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1033</td>
<td></td>
<td>1034</td>
<td>WA1DAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Donald</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

and the second for contest fields

<table>
<thead>
<tr>
<th>Rig Not Specified</th>
<th>QSO Freq</th>
<th>On</th>
<th>Off</th>
<th>Call</th>
<th>Name</th>
<th>In</th>
<th>Out</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10140.000</td>
<td></td>
<td></td>
<td></td>
<td>2346</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You might prefer a more minimal view of the logging fields. You can select to completely suppress the log panel or to use a single line view as in either of these two:

- `10140.000`
- `10140.000`
These are selectable from the View menu:

![View menu](image)

The frequency, Off (time off), and #Out are filled by the program. All the others can be populated by manual keyboard entry or by selection from the Rx panel. The time off, Off, is continuously update with the current GMT. The time on, On, will be filled in when the Call is updated, but can be modified later by the operator.

A right click on the Rx panel brings up a context sensitive menu that will reflect which of the two QSO capture views you have open.

![Menu views](image)

If you highlight text in the Rx pane then the menu selection will operate on that text. If you simply point to a word of text and right click then the menu selection will operate on the single word.

Certain fields may also be populated with automatic parsing, Call, Name, Qth and Loc. You point to the Rx pane word and then either double-left-click or hold a shift key down and left click. The program
will attempt to parse the word as a regular expression to populate the Call, Name, Qth, and Loc fields in that order. It may place some non standard calls into the Loc field if they qualify as a proper Maidenhead Grid Square, such as MM55CQ. That may be a special event station, but it also looks like a grid square locator value. You need to decide when that occurs and use the pop up menu for those special cases. The first non-Call non-Loc word will fill the Name field and subsequent qualify words will go into the Qth field.

A highlighted section of text, can always be copied to the clipboard for subsequent pasting elsewhere. The Copy menu item will be active when text in the Rx pane has been highlighted. That text can also be saved to a file. Use the "Save as..." menu item for that purpose. All data fields in fldigi share a common set of keyboard shortcuts. Linux users will recognize these as familiar Emacs shortcuts. There is also a small popup menu that can be opened for each field by right clicking the contents with the mouse:

Highlighted text will be overwritten when a paste is selected. Otherwise the clipboard will be pasted at the current cursor position.

You can query on-line and local CD based data base systems for data regarding a Call. Set up your query using the Callsign Db configuration tab. You make the query by either clicking on the globe button, or selecting "Look up call" from the menu. The latter will also move the call to the Call field and make the query.

If you have previously worked a station the logbook will be searched for the most recent qso and fill the Name, Qth and other fields from the logbook. If the logbook dialog is open that last qso will be selected for viewing in the logbook.

You open the logbook by selecting from the View menu; View/Logbook. The logbook title bar will show you which logbook you currently have open. fldigi can maintain an unlimited (except for disk space) number of logbooks.
You can resize the dialog to suit your screen size and operating needs. Fldigi will remember the placement and size for subsequent use.

You can create new entries, update existing entries, and delete entries using this dialog. You can also search for an entry by callsign. The browser can be sorted by Date, Callsign, Frequency or Mode. The sort can be forward or backward with the most recent being the default selected entry after each sort. You execute the sort by clicking on the column button at the top of the column to be sorted. Each click causes the sort to reverse. I like to view my log with the most recent at the top. You might want to view it with the most recent on the bottom.

There are no frills such as keeping track of DXCC worked, fancy printouts etc. Fldigi's logbook is primarily a capture function. You can export your data for use with an external database or for uploading to LOTW or eQSL. Data from those sources can also be used for importing into the logbook.
Exporting Logbook Data

Fldigi provides automatic export of log records as they are recorded. On Linux the data is forwarded to Xlog compatible programs using the SysV message queue system. On Windows the records are exported via a temporary file structure and are accepted by Logger32.

The user may also export all or selected records consisting of all or selected fields. Access to this export function of available from the menu "File/Log/Export ADIF", "File/Log/Export Text", and "File/Log/Export CSV".

Export ADIF

Selecting the Export ADIF menu item opens the following dialog:

If you want to export every record press the "Check All" in the left panel. You can also select and deselect individual records. Choose which fields you want to export with the right panel controls. Press the OK button to continue or Cancel to abort the operation. A file chooser dialog will open which allows you to specify the name and location of the exported file. Use the extension ".adi" on Windows and ".adif" on the other OS's.

Export Text / CSV

The same Export Setup dialog is used for Text and CSV exports.

The Text export produces a simple space delimited file with columns set at locations dictated by the field size for each field that is exported. It is suitable for use with a word processing program or for printing a hardcopy of your activities.

The CSV is a "Character Separated Value" file with the TAB character used as the field separator. This type of file can be imported into nearly all spreadsheet programs such as Gnumeric, Open Office or MS Excel.
Contest How To

Fldigi supports a basic contesting format. Select the menu item View/Contest fields to see how the qso entry entry fields change for contest data. You will see that fldigi has fields to support received and sent contest numbers as well as generic contest exchange information.

The serial number out (#Out) is automatically initialized and updated by the built-in serial number generator. You can enter the appropriate exchange information via the keyboard or mouse. Text in the Rx pane can be selected by the usual left-click-swipe of highlighting. Then right click anywhere after highlighting the desired text and a popup menu will appear allowing you to select the destination QSO field. Make your selection and the info is placed in the correct text box. Note that the popup menu changes with the QSO logging view and also with a change in "Quick entry". A full description is found in the description of operating the Logbook. The important thing to note for contest operation is that the Call and Serial # are single word captures. The Xchg capture can be either single word or multiple word (mark / right click). If the Xchg field has text contents then the new capture is appended to end of the current text in that field. That means you can point to the word representing the field, right click and select from the menu. You do not need to highlight the text for the word capture. You can very rapidly fill in the serial number and the exchange data (even if multi value) by simply pointing and right clicking on the desired word.

To set up fldigi for contesting you will need to open configure contest. the 1st row contains what info you want sent with the appropriate macro tag. ie...if the contest requires rst and name you would fill in the Exchange Out box with your name. The contents of this field are accessed from a macro with the <XOUT> tag. You will also need to check the rst always 599 box as this is the de-facto signal report in contests.

If you are
participating in a cw contest you may want to select the "Send CW cut numbers", cut numbers is the norm for a CW contest. The cut numbers will send N for 9 and T for zero.

The next box contains the needed requirements to use serial numbers for a contest. you will always want to use leading zeros, start with 1 and use 3 digits. Press reset to initialize the #Out QSO field to the Start number.

Check the appropriate fields for determining if this is a duplicate call. If a duplicate is detected the Call entry will be highlighted as shown in the "Dup Color" button. Pressing this button opens a color selector so you may customize the color. There are many choices to alert you to a duplicate contact. The duplicate is based on the logical AND of all of the fields to be checked. The DUPE is cleared when you press the clear QSO log button (the brush icon).

After you have filled in all the required information, make sure you save and close.

Remember YOU MUST click the Reset button in the Serial number panel for the serial number counter to be initialized. You should also press the QSO clear button (broom) in the qso entry widget for the other changes to take effect.

It would be best to create a new log for each contest. You create a new log by selecting the menu item File/Logs/New logbook. The default new log name will be newlog.adif on Linux and newlog.adi on Windows. You can rename the new log file now or later by using the system file manager or when you save the log. The import/export feature of fldigi will allow you to export the log into your everyday logging software or the built-in fldigi logbook.

**Restarting a contest session**

You might have closed down fldigi in the middle of a contest, everyone needs a break now and then. You then start fldigi and want to continue the contest. Here are the steps to insure that you continue operations with no glitches.

• Load your macro file that contains your contest macros (more on that below)
• Select the menu item View/Contest fields
• Select the menu item View/Logbook
• Make sure you have the contest logbook open ... if not then this is the time to open that logbook database.
  Select the menu item "File/Logs/Open logbook..." and find your log data file.
• Look at the last record and check the serial number sent. Enter that number plus one in the Start entry on the config contest tab (see above).
• Press the Reset button in that panel.

You are ready to keep on contesting
**Remembering a contact**

If you are copying a potential contact but you are not being heard you can save fldigi's modem state using one of two methods

1. double click the signal on the waterfall
2. right click on the Rx panel and select "Insert divider"

A line of text will be inserted at the end of the Rx text buffer. It will appear similar to this:

```text
<<2008-12-30T10:06Z BPSK-31 @ 3580000+0781>>
```

The date-time, the mode, the transceiver operating frequency and the audio offset will be recorded. The text line is in blue and behaves in a way that you might expect a url reference to behave in a web browser window. Work a few more contacts (even on a different band or frequency) and then scroll the Rx pane to that special divider. Left click on the line of text and fldigi will restore the transceiver to its frequency, change the mode to the saved mode and put the waterfall cursor at the audio offset frequency. Changing the transceiver frequency will only work if you are using CAT control of your transceiver. If you are not using CAT control the mode and waterfall cursor will still be restored.

There is no limit to the number of divider lines that can be inserted into the Rx pane. They will all be removed when the Rx pane is cleared.

**Saving the entire session**

Select the menu item "File/Logs/Log all RX/TX text". If this toggle menu is checked your entire session of received and sent text will be saved to a file in the fldigi default files folder. It will be given a name synonymous with the date and time is is started, ie: fldigi20081230.log. You can review this log by selecting the menu item "File/Show config" which will open your OS default file explorer to the fldigi files folder. The file is an ASCII text file.

The format of the daily log is shown in Working Logs.

**Contesting Macro Tips**

OK, now we have fldigi setup for basic contesting, lets move on to some ideas on macros to use. I tend to make generic one size fits all macros. I recommend that you make a new macro file, mine is named contest.mdf, this will give you 48 macros to use based on the type of contest you are entering. Take a good look at the examples I have listed, you will notice there are no commas, hyphens or other extraneous items. I have seen just about every example of a poorly thought out macro there is or has ever been dreamed up. Classic examples are:

- w3nr you are 599 in Alabama your serial number is 001-001-001 how copy ??
- hello ed thanks for the call you are 599-599-001-001 qth Alabama back to you

The list goes on and on. Just think, you have to try and capture the exchange, try it and you will see what I mean.
When you enter a contest you have to decide whether you are going to sit on one frequency and call CQ (Run) or are you going to tune the band looking for stations to work (S&P). So lets set up some macros that should cover both cases.

Several new macro tags have been created to facilitate contesting, these include the following tags.

- `<LOG>` add QSO data to the logbook & clear the QSO data fields
- `<CNTR>` insert current contest serial number into the text stream
- `<INCR>` increment contest serial number
- `<DECR>` decrement contest serial number
- `<XOUT>` contest exchange
- `<QSOTIME>` current log time in Zulu HHMM format
- `<LDT>` local date time
- `<ILDT>` LDT in iso-8601 format
- `<ZDT>` Zulu date time
- `<IZDT>` ZDT in iso-8601 format
- `<QSOTIME>` actual time of execution of the macro ... useful where exact times are used to match contest log submissions
- `<SAVEXCHG>` save entire contents of the expanded macro text to the "Exchange Out" field in the logbook
- `<XBEG>` mark the beginning of a text string that is to be saved to the "Exchange Out" field in the logbook
- `<XEND>` mark the end of the text string that is to be saved to the "Exchange Out" field in the logbook

Note: `<SAVEXCHG>` and the `<XBEG>...<XEND>` macro tags are mutually exclusive. `<XBEG>...<XEND>` is given priority if both all three are specified in a single macro.

**RUN Macros**

We need just a few, starting with a CQ macro - Put this in the F1 key definition

```
<TX>
cq test de <MYCALL> <MYCALL> cq k
```

Notice that I left 2 spaces between my call and 3 spaces at the end before the k. This will make it easier for a station to grab my call and the k on the end eliminates garbage characters before my macro finishes. The tx/rx are on separate lines as I want to be sure my macro is on a line by itself and not mixed in with screen garbage.

Now the exchange macro - Put this in the F2 key definition

```
<TX>
<CALL> 599 <CNTR> <CNTR> <X1> <X1> <CALL> k
```
Why do I have his call at the beginning as well as the end, to make sure I have copied his call correctly. You will also see that I have not as yet logged the contact, why, well are you sure he does not need to correct his call or ask for a repeat.

You are asked to repeat the exchange, you can just re-send the exchange macro, this verifies all of the information. Now he sends you his info and if you have copied it correctly you need a TU macro. - Put this in the F3 key definition

```
<TX>
qsl tu qrz test <MYCALL> k
<RX> <LOG> <INCR>
```

Here we have done all the necessary items to complete the exchange. Notice that I did not log the contact until after everything was correct. I have fldigi set to clear on save, so when the <LOG> part of the macro executes the QSO area is cleared.

That's the end of my RUN macro setup, told you it was rather simplistic and generic.

**S&P Macros**

I rarely if ever use S&P, but there are times I need to, especially if my QSO rate drops while running. Again the macros are very generic with only the needed info. If band conditions warrant you may want to send your call 3 times. Put this in the F5 key definition

```
<TX>
<MYCALL> <MYCALL> k
<RX>
```

Why just my call ?? Well I assume the other guy already knows his call!

The exchange macro is basically the same as the RUN macro. Put this one in the F6 key definition

```
<TX>
599 <CNTR> <CNTR> <X1> <X1> k
<RX>
```

As you see I have not as yet logged the QSO or incremented the serial number. This is the final S&P macro. Put this one in the F7 key definition

```
<LOG> <INCR>
```

Now this is the most important macro you will ever need......trust me. Put it where you won't fail to find it. How about F9 ?

```
<TX>
```
You will see that it is used many times during a contest, especially with weak stations and heavy QRN/QRM.

<QSOTIME>

- time sent in Tx stream
- repeat execution of <QSOTIME> before a <LOG> macro or a save to log button press will resend the original time
- <LOG> macro or a save-to-log button press appends the QSOTIME to the STX_STRING field in the adif log record and clears the QSOTIME.

<XBEG>

- use at end of a contest exchange to save the entire exchange string in STX_STRING
- usurps QSOTIME if both are contained in same macro text, ie: "<RST> <CNTR> <QSOTIME> <SAVEXCHG>" will send an exchange as 599 024 1125 if RST = 599, Counter = 024 and time of execution is 1125
- repeats the same as <QSOTIME>
- <LOG> macro or a save-to-log button press saves the associated macro text (after expansion). QSOTIME and the saved exchange text are cleared after the save occurs.

An example of the SAVEXCHG macro tag

```xml
<RST> <CNTR> <XOUT> <QSOTIME> <SAVEXCHG>
```

where RST = 599, CNTR = 0125, XOUT = AL, QSOTIME = 1433

will save this string to the Exchange Out field in the logbook: "599 0125 AL 1433"

Please note that you should not include any text or macro tags that are not to be a part of Exchange Out. If your macro had this:

```xml
<TX> <CALL> UR <RST> <CNTR> <XOUT> <QSOTIME> de <MYCALL> k <RX> <SAVEXCHG>
```

where CALL = W3NR, MYCALL = W1HKJ

the saved Exchange Out field would contain: "W3NR UR 599 0125 AL 1433 de W1HKJ k"

Probably not what you want. Use separate function keys for the "<TX>CALL ..." and the "de <MYCALL> k <RX>" or use the next set of macro tags

```xml
<XBEG>...<XEND>
```

These two macro tags are delimiters for capturing the transmitted exchange data, for example:
<TX><CALL> de <MYCALL> QSL <XBEG><RST> <CNTR> <QSOTIME><XEND> K<RX>

will place the expanded <RST> <CNTR> <QSOTIME> into the *Exchange Out* field of the logbook when the contact is saved. This is much better illustrated with a screen shot. This one shows the macro editor contents, the logbook entry in *Exchange Out*, and the transmit text buffer.
Creating a Cabrillo Report

Fldigi can generate a basic Cabrillo report that meets most contest needs.

Selecting the "File/Log/Cabrillo report" menu item opens the following dialog:

You must then open the file with a plain text editor and modify the appropriate entries. Check with each contest sponsor to see what their requirements are.

Here is an example of a generated cabrillo report format before being edited:
START-OF-LOG: 3.0
CREATED-BY: fldigi 3.11

# The callsign used during the contest.
CALLSIGN: W1HKJ

# ASSISTED or NON-ASSISTED
CATEGORY-ASSISTED:

# Band: ALL, 160M, 80M, 40M, 20M, 15M, 10M, 6M, 2M, 222, 432, 902, 1.2G
CATEGORY-BAND:

# Mode: SSB, CW, RTTY, MIXED
CATEGORY-MODE:

# Operator: SINGLE-OP, MULTI-OP, CHECKLOG
CATEGORY-OPERATOR:

# Power: HIGH, LOW, QRP
CATEGORY-POWER:

# Station: FIXED, MOBILE, PORTABLE, ROVER, EXPEDITION, HQ, SCHOOL
CATEGORY-STATION:

# Time: 6-HOURS, 12-HOURS, 24-HOURS
CATEGORY-TIME:

# Transmitter: ONE, TWO, LIMITED, UNLIMITED, SWL
CATEGORY-TRANSMITTER:

# Overlay: ROOKIE, TB-WIRES, NOVICE-TECH, OVER-50
CATEGORY-OVERLAY:

# Integer number
CLAIMED-SCORE:

# Name of the radio club with which the score should be aggregated.
CLUB:

# Contest: AP-SPRINT, ARRL-10, ARRL-160, ARRL-DX-CW, ARRL-DX-SSB, ARRL-SS-CW,
# ARRL-SS-SSB, ARRL-UHF-AUG, ARRL-VHF-JAN, ARRL-VHF-JUN, ARRL-VHF-SEP,
# ARRL-RTTY, BARTG-RTTY, CQ-160-CW, CQ-160-SSB, CQ-WPX-CW, CQ-WPX-RTTY,
CONTEST: ARRL-RTTY

# Optional email address
EMAIL:

LOCATION:

# Operator name
NAME:

# Maximum 4 address lines.
ADDRESS:
ADDRESS:
ADDRESS:
ADDRESS:

# A space-delimited list of operator callsign(s).
OPERATORS:

# Offtime yyyy-mm-dd nnnn yyyy-mm-dd nnnn
# OFFTIME:

# Soapbox comments.
SOAPBOX:
SOAPBOX:
SOAPBOX:

QSO: 14095 RY 2009-01-04 1952 W1HKJ 599 GA 12345 ND2T 599 CA 67890
QSO: 14098 RY 2009-01-04 1949 W1HKJ 599 GA W0SD 599 SD
QSO: 14099 RY 2009-01-04 1948 W1HKJ 599 1234567890 KB7Q 599 1234567890
QSO: 14100 RY 2009-01-04 1948 W1HKJ 599 GA N6WS 599 CA
QSO: 14103 RY 2009-01-04 1946 W1HKJ 599 GA VE6AO 599 AB
END-OF-LOG:
DXCC List

Fldigi uses several data files that are not included with the distribution. These must be downloaded from the list maintenance web sites for the most current data. These lists include:

<table>
<thead>
<tr>
<th>List Data</th>
<th>List Name</th>
<th>Web source</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOTW</td>
<td>lotw1.txt</td>
<td><a href="http://www.hb9bza.net/lotw/lotw1.txt">http://www.hb9bza.net/lotw/lotw1.txt</a></td>
</tr>
<tr>
<td>EQSL</td>
<td>AGMemberList.txt</td>
<td><a href="http://www.eqsl.cc/QSLcard/DownloadedFiles/AGMemberList.txt">http://www.eqsl.cc/QSLcard/DownloadedFiles/AGMemberList.txt</a></td>
</tr>
</tbody>
</table>

These files should be downloaded and placed in the fldigi files directory. The most convenient way to open the fldigi files directory is via the menu item "File / Show config".

The DXCC list browser is shown by selecting the menu item "View / Countries".

You can sort the list by Country, Continent, ITu or CQ zone by clicking on the various column headers.
Working Logs

Fldigi maintains a number of working log files that are found in its default folder. The default folder is easy to find, simply select the menu item "File/Show config" and your OS default files explorer will be opened to that location.

Rx/Tx Capture File

Every time you start or stop fldigi that event is recorded in a daily log file. The daily log is name as:

fldigiYYYYMMDD.log

where YYYYMMDD is the current GMT date. This log will also contain your entire session of Rx and Tx data annotated as to activity and timestamped. Here is a small example of the daily log:

--- Logging started at Tue Dec 30 11:37:21 2008 UTC ---
RX (2008-12-30 11:37Z): our property. No pwr even for a day is rough.
TX (2008-12-30 11:39Z):
TX (2008-12-30 11:39Z): CQ CQ CQ de W1HKJ W1HKJ W1HKJ
TX (2008-12-30 11:40Z): CQ CQ CQ de W1HKJ W1HKJ W1HKJ pse k
RX (2008-12-30 11:40Z): mG sk
--- Logging stopped at Tue Dec 30 11:48:11 2008 UTC ---

This log is appended to with each start and stop. That means that no data is ever overwritten.

Status log

A log of errors, warnings and status reports is written for each session. This file is overwritten each time the program is opened and subsequently closed. Its format is also ASCII text and will contain data such as:

Q: main: fldigi 3.04BV log started on Tue Dec 30 05:47:10 2008
W: dxcc_open: Could not read contest country file "/home/dave/.fldigi/cty.dat"

This data is identical to that which can be viewed with the event log dialog which is opened using the menu item "Help/Event log"
There are six levels of event logging with increasing depth of reports:

- Quiet
- Info
- Verbose
- Warning
- Error
- Debug

The default level for logging events is "warning."

The event log shown above was captured during a period of psk-reporting. Fldigi was set up to monitor and report all detected signals that satisfied the requirements of the pskreporter web site. The "spotted" signals were then automatically sent to the web site. A complete report of the recorded events was obtained by a right click in the text pane. Select-all and Save as was chosen. The report can be read here.

At the Debug level you will probably see more events than you need. You can select which events to suppress using the "Log sources" menu button. It defaults to all enabled.
PSK Reporter

The PSK reporter can generate reception reports from three different sources:

- The decoded text
- The log data
- Data entered manually

The configuration for the PSK reporter in Misc/Spotting. PSKR needs the following fields from the Oper tab to be non-empty:

1. Callsign (freeform because it's impossible to verify, and because we need to support SWLs without callsigns)
2. Locator (standard 6 character format)
3. Antenna info (freeform, should be kept reasonably short)

Sources (1) and (2) are configurable from Misc/Spotting configuration tab, while (3) is always enabled. To keep the code sane, changing the PSKR options (or the above station info) does not take immediate effect. Instead, the Initialize button changes colour to indicate that the changes have not been applied. Clicking on the button will do so (or display an error) for the current and future sessions. This is similar to the Initialize buttons in the rig control configuration.

Here are the options in some more detail:

**Automatically spot callsigns in decoded text**

The text that is sent to the main window or the PSK browser is continuously searched for callsigns. If this option is enabled, the main window gets a "Spot" light button that can toggle the auto-spotter on and off. It is automatically turned off when playback is selected in the Files menu. The main window text is not searched if the viewer is active, i.e., if it is displayed and the current modem is PSK.

**Send reception report when logging a QSO**

A reception report is queued for every QSO as soon as it's logged

**Report QRG (etc.)**

This makes the reception reports include the current rig frequency, adjusted for modem audio frequency and rig sideband. It does not need a click on "Initialize" to take effect. This needs to be an option because it is impossible to tell whether the user has real or "fake" rig control with 100% certainty. Besides that, users may want to run a dedicated spotter for a narrow modes subband, and in that case they won't have to synchronise fldigi's frequency display with the rig all that often.
Host and port

With the port set to 14739 the reports will not be entered in the main database, but instead will be analysed and displayed here:

http://pskreporter.info/cgi-bin/psk-analysis.pl

Probably of no interest to anyone who is not hacking on a PSKR client but may be useful for debugging. The PSKR protocol uses UDP with no acknowledgements or retransmissions, so don't be surprised if the occasional report never makes it to the server. There should be enough coverage overlap to make packet loss irrelevant (and save a lot of bandwidth and CPU cycles).

The spotter needs to see a repeated callsign within a short search window, but stations do not always repeat their callsigns. In addition, some operators like to be creative with their macros, and as a result some signals will decode 100% but the callsign will never be auto-captured. Such callsigns can be reported manually.

The manual spotting is done by right-clicking the QRZ "globe" icon. This will generate a report for whatever is in the Call & Loc fields, so make sure that those are correct! You should also verify the frequency (e.g. by placing the waterfall marker on the signal being spotted).

There is a confirmation popup that will open when you right click the "globe" button. The aim of course is to avoid accidentally sending rubbish reports to the PSK reporter database.

Reception reports are filtered for duplicates among all data sources: a report is queued only once every hour for each callsign and frequency band. The queue is flushed every five minutes. You can see what the spotter is doing in the Event Log window or on the terminal if you set the log level to "Info". "Debug" will show all the gory details.

A button and popup text field on the rig control frame give access to the most recent receptions reports in your geographic area. The area is determined by the contents of the field to the right of the button, or by the locator text on the operator tab if the mini field is empty. The first two characters of the locator are used. If the locator is not set, the pskreporter.info uses the current IP geolocation to approximate it.

A popup is displayed when the user clicks the button or presses the Enter key from within the field. The popup shows the frequencies by measure of activity that gives more weight to transmissions. If rig control is active, the user can click on one of the lines to go to that band. Clicking on the 18100000 (1 report) line would immediately QSY the transceiver to 18.1 MHz.

The data is retrieved from http://pskreporter.info/cgi-bin/psk-freq.pl or with a filled text field, http://pskreporter.info/cgi-bin/psk-freq.pl?grid=TEXT

There is a link to the pskreporter.info map page in the Help menu.
PSKmail Configuration

Fldigi can act as both a server and a client for PskMail, a separate application from fldigi. The PSKmail specific configuration parameters are all located on the Misc/Pskmail tab of the configuration dialog.

Instructions on setting these parameters are a part of the pskmail installation.
Notifier

This dialog available is used to specify search patterns and alerts that are triggered when the decoded Rx text matches those patterns. This only happens when the Spot button on the main window is activated, as with the PSK Reporter client.

First, here's how it works in general. You specify a regular expression (RE) that contains one or more parenthesised capturing groups. Fldigi's spotter matches it against the incoming text (main window or Signal Browser, so it works with both PSK and RTTY) and, if the RE matches, it performs one or more of the following:

- Displays an alert window with some text and a "go to that frequency" button.
- Enters some arbitrary text into the Transmit pane. The text may contain <MACRO>s and these will be expanded as usual.
- Runs a program (Unix/Linux only for now).

The text described by the capturing group(s) can be used in all of the above. There is an example of this at the end of this page.

Not everyone is at ease in writing regular expressions for the notifier to act upon. So a few "canned" searches are coded into the notifier and are selected from the event chooser at the upper left of the dialog.

1) My Callsign de CALL. Can be used to alert you when CALL calls you.

2) Station heard twice. Pretty much the same search that the PSK reporter client does.

3) Custom text search. This reveals an input field where you type your own RE.

Both (1) and (2) are special cases of (3), but with some extra processing available because in each case fldigi knows what it has just found.

The Filter pane is available for the first two event types only, i.e. not the custom text search. In this pane you can specify some properties that the spotted callsign must have for the actions to take place:

a) The Callsign radio button reveals a text field when selected. If you enter something in that field, the event will be accepted only if the text matches the spotted callsign (I may change this to a RE match).
b) The "DXCC entity" radio button reveals a button that brings up a list of DXCC entities. Select entities by clicking or dragging. If you select any at all, the spotted callsign's country will have to be one of those or the event will be ignored. Having no entities selected is the same as selecting all of them, i.e. any country, but is a more efficient.

The entity list can be sorted by clicking on the row headers, and there is a right-click context menu that can (de)select by continent and CQ zone. The buttons and search fields at the bottom behave as you'd expect.

The list is also available with the menu item "View / Countries" in the main window

You need cty.dat for all this to work

c) The "Not worked before" check button asserts that, if you have selected (a) above, the callsign must not be in your logbook. Same with (b), but now you must not have had any QSOs with stations from that country in the log.

d) The "LotW user" and "eQSL user" buttons specify that the callsign must be on one of these two lists (the documentation explains where to get the user lists from and where to put the files).

The Action pane is where you choose how fldigi will alert you when an event matches the filter bits.

a) The text in the "Show alert text" box, if not empty, is shown in a pop-up window. The alert window has a timer and dismisses itself after a configurable time interval (the "Hide after" control). The user can click anywhere inside the window to stop the timer.

The button next to the text box enters the default alert text for the event you have selected. There are a few variables that are substituted when the window is displayed:

For all three event types: $MODEM (modem name), $DF_HZ (dial frequency), $RF_HZ (actual receive frequency), $RF_KHZ, $AF_HZ (modem audio frequency)

For the 1st event type (my call): $CALLSIGN, $TEXT (all matched text).

For the 2nd event type (station): $CALLSIGN, $TEXT, $COUNTRY.

For the 3rd event type (custom): you're on your own here, but fldigi will helpfully list all
the possible substrings found in your RE.

The whole text is passed through strftime(3) so you can customize the date. Here's a reference for the % characters:

http://www.opengroup.org/onlinepubs/007908799/xsh/strftime.html

b) The "Append to TX text" box -- self explanatory. The same variable substitutions apply, as well as macro expansion. The nearby button shows the macro editor. The appended rx text is clickable. Clicking it will move the waterfall frequency (and transceiver if under CAT) to the detected signal and change to the indicated mode.

<<2009-07-18T19:21Z BPSK-31 @ 3580000+1589>>

c) The "Run program" field and browse button are only available on Unix systems. Field contents are passed to the shell ("/bin/sh -c"), as with system(3). No variable or \backref substitution is done for this field, but all substrings are exported as environment variables, such as FLDIGI_NOTIFY_STR_1. The usual <EXEC> macro variables are also there and your ~/.fldigi/scripts directory will be in the shell's path. Try it out with a test script for the full list of variables.

d) The trigger limit box specifies how much time must pass between subsequent invocations of whatever actions you have specified.

The Duplicates pane has a check button that displays the rest of that group when checked. If you enable this, fldigi will remember what it has seen and ignore the event if it is a duplicate. The other controls in that pane determine what constitutes a duplicate:

a) The menu tells fldigi what to look at. For the first two event types, the menu will display "Callsign", and for the custom search it will contain a list of \X references for the RE.

b) The time box is also essential; it determines how close the events must be in time to be considered duplicates.

c) The Band and Mode check boxes further restrict the comparison.

An example:

You are looking at callsigns, with a dup time of 600s, and both Band and Mode checked. A callsign is found once and fldigi alerts you. Now if this callsign is spotted again, less than 600s later and in the same band and mode, it is a duplicate and will be ignored. With (say) Band and Mode unchecked, it is a duplicate regardless of frequency band or mode as long as it's heard before the 600s elapse.

Three of the four buttons at the bottom left are pretty much self-explanatory. Add to the list an event you have just specified, or select an event from the list and Remove it, or change some of its parameters and Update it.
The Test... button allows you to test an event with some text of your choice. This is particularly useful with the custom text search, as it's too easy to enter a RE that will never match. The dialog will show you the default test string for the two fixed event types. Careful: the "Station heard twice" event type expects a non-alphanumeric character at the end of its input. The default test string has a space at the end.

If nothing happens, it may be because you have not specified any actions, or because the event's filter does not match, or because the trigger limit or dup handling are preventing the actions from happening. In the latter case, updating an event will reset its dup data. But it's better to add the dup and trigger limits at the end, after you've tested the event.

The list at the bottom of the window shows the events you have added. All contents are saved in the file ~/.fldigi/notify.prefs.

The list has a context menu for quick access to Update, Remove, and Toggle. The first two have the same effect as clicking on the button of the same name.

The Toggle item lets you flip the "Enabled" status of an event: this is like selecting an event, clicking on the "Enabled" button in the Event pane to (de)activate it, and then clicking "Update". Disabled events are kept on the list but are not registered with the spotter and so they are never triggered.

If you disable all the events and there is nothing else using the spotter (e.g. PSK Reporter), the Spot button will disappear from the main window.

A 2nd example:
Here's how to do the "my call" event using the custom text search:

a) Select "Custom text search" in the event pane

b) In the RE box, enter (without the quotes or leading white space):

"<YOUR_CALL>\.+de\[[:space:]\]+(\[[:alnum:]\]?\[[:alpha:]/\]+\[[:digit:]]+\[[:alnum:]/\]+)"

and remember to replace <YOUR_CALL> with your callsign.

c) In the actions pane you can now use \0 for the whole text matched by the above RE, and \1 for the first capturing group (the callsign).

d) Select "\1" in the duplicates menu if you want dup filtering.

e) Test with "<YOUR_CALL> de <SOME_OTHER_CALL>" and you should see the alert window with the text you specified.
**Addtional examples:**

Add a "My callsign de CALL" event with a script that will do something to get your attention when someone calls you.

Add a "Station heard twice" with the DXCC filter and the "Not worked before" option. Also set the LotW or eQSL options if desired.

Add a "Station heard twice" with no callsign/dxcc/etc. filter but with duplicate filtering. Write a script that sends the data to a DX cluster or similar.

Here is a simple Perl script that uses notify-send (in the package libnotify-bin on Debian) to display desktop notification "bubbles". A better version would use the libnotify bindings for Perl or Python directly.

```perl
#!/usr/bin/perl
exec("notify-send", "-t", "5000", "-i", "/usr/share/pixmaps/fldigi.xpm",
"Heard " . $ENV{"FLDIGI_NOTIFY_CALLSIGN"} . " ($ENV{FLDIGI_NOTIFY_COUNTRY})",
$ENV{"FLDIGI_NOTIFY_STR_0"});
```

---snip---
XmlRpc Control

As of version 3.0, various aspects of Fldigi's operation can be controlled remotely using XML-RPC. XML-RPC data is transported via simple HTTP and client implementations exist for most programming languages. A Perl client that can be used as a control script is included in the source tarball as scripts/fldigi-shell. This control method is currently used by several external programs including flrig, logger32 and Xlog.

The following command line arguments become available when XML-RPC support is compiled into fldigi, as described in the build instructions:

--xmlrpc-server-address HOSTNAME
Set the XML-RPC server address. The default is 127.0.0.1.

--xmlrpc-server-port PORT
Set the XML-RPC server port. The default is 7362.

--xmlrpc-allow REGEX
Allow only the methods whose names match REGEX

--xmlrpc-deny REGEX
Allow only the methods whose names don't match REGEX

--xmlrpc-list
List all available methods

The --xmlrpc-deny and --xmlrpc-allow switches can be used as a simple access control mechanism. REGEX specifies a POSIX extended regular expression. This invocation disables the methods that may cause fldigi to transmit:

--xmlrpc-deny 'main\.\(tx|tune|run_macro\)'

By default all methods are allowed.

The --xmlrpc-list switch outputs the method list and exits the program. If preceded by --xmlrpc-deny or --xmlrpc-allow, it shows the list of methods as filtered by those switches.

The methods are listed below. The three columns are method name, signature (return_type:argument_types), and description. Refer to the XML-RPC specification for the meaning of the signature characters (briefly: n=nil, b=boolean, i=integer, d=double, s=string, 6=bytes, A=array, S=struct).

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fldigi.list</td>
<td>A:n</td>
<td>Returns the list of methods</td>
</tr>
<tr>
<td>fldigi.name</td>
<td>s:n</td>
<td>Returns the program name</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>fldigi.version_struct</td>
<td>Returns the program version as a struct</td>
<td></td>
</tr>
<tr>
<td>fldigi.version</td>
<td>Returns the program version as a string</td>
<td></td>
</tr>
<tr>
<td>fldigi.name_version</td>
<td>Returns the program name and version</td>
<td></td>
</tr>
<tr>
<td>fldigi.config_dir</td>
<td>Returns the name of the configuration directory</td>
<td></td>
</tr>
<tr>
<td>fldigi.terminate</td>
<td>Terminates fldigi. &quot;i&quot; is bitmask specifying data to save: 0=options; 1=log; 2=macros</td>
<td></td>
</tr>
<tr>
<td>modem.get_name</td>
<td>Returns the name of the current modem</td>
<td></td>
</tr>
<tr>
<td>modem.get_names</td>
<td>Returns all modem names</td>
<td></td>
</tr>
<tr>
<td>modem.get_id</td>
<td>Returns the ID of the current modem</td>
<td></td>
</tr>
<tr>
<td>modem.get_max_id</td>
<td>Returns the maximum modem ID number</td>
<td></td>
</tr>
<tr>
<td>modem.set_by_name</td>
<td>Sets the current modem. Returns old name</td>
<td></td>
</tr>
<tr>
<td>modem.set_by_id</td>
<td>Sets the current modem. Returns old ID</td>
<td></td>
</tr>
<tr>
<td>modem.set_carrier</td>
<td>Sets modem carrier. Returns old carrier</td>
<td></td>
</tr>
<tr>
<td>modem.inc_carrier</td>
<td>Increments the modem carrier frequency. Returns the new carrier</td>
<td></td>
</tr>
<tr>
<td>modem.get_carrier</td>
<td>Returns the modem carrier frequency</td>
<td></td>
</tr>
<tr>
<td>modem.get_afc_search_range</td>
<td>Returns the modem AFC search range</td>
<td></td>
</tr>
<tr>
<td>modem.set_afc_search_range</td>
<td>Sets the modem AFC search range. Returns the old value</td>
<td></td>
</tr>
<tr>
<td>modem.inc_afc_search_range</td>
<td>Increments the modem AFC search range. Returns the new value</td>
<td></td>
</tr>
<tr>
<td>modem.get_bandwidth</td>
<td>Returns the modem bandwidth</td>
<td></td>
</tr>
<tr>
<td>modem.set_bandwidth</td>
<td>Sets the modem bandwidth. Returns the old value</td>
<td></td>
</tr>
<tr>
<td>modem.inc_bandwidth</td>
<td>Increments the modem bandwidth. Returns the new value</td>
<td></td>
</tr>
<tr>
<td>modem.get_quality</td>
<td>Returns the modem signal quality in the range [0:100]</td>
<td></td>
</tr>
<tr>
<td>modem.search_up</td>
<td>Searches upward in frequency</td>
<td></td>
</tr>
<tr>
<td>modem.search_down</td>
<td>Searches downward in frequency</td>
<td></td>
</tr>
<tr>
<td>modem.olivia.set_bandwidth</td>
<td>Sets the Olivia bandwidth</td>
<td></td>
</tr>
<tr>
<td>modem.olivia.get_bandwidth</td>
<td>Returns the Olivia bandwidth</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Type 1</td>
<td>Type 2</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>modem.olivia.set_tones</td>
<td>n:i</td>
<td></td>
</tr>
<tr>
<td>modem.olivia.get_tones</td>
<td>i:n</td>
<td></td>
</tr>
<tr>
<td>main.get_status1</td>
<td>s:n</td>
<td></td>
</tr>
<tr>
<td>main.get_status2</td>
<td>s:n</td>
<td></td>
</tr>
<tr>
<td>main.get_wf_sideband</td>
<td>s:n</td>
<td></td>
</tr>
<tr>
<td>main.set_wf_sideband</td>
<td>n:s</td>
<td></td>
</tr>
<tr>
<td>main.get_frequency</td>
<td>d:n</td>
<td></td>
</tr>
<tr>
<td>main.set_frequency</td>
<td>d:d</td>
<td></td>
</tr>
<tr>
<td>main.inc_frequency</td>
<td>d:d</td>
<td></td>
</tr>
<tr>
<td>main.get_afc</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.set_afc</td>
<td>b:b</td>
<td></td>
</tr>
<tr>
<td>main.toggle_afc</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.get_squelch</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.set_squelch</td>
<td>b:b</td>
<td></td>
</tr>
<tr>
<td>main.toggle_squelch</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.get_squelch_level</td>
<td>d:n</td>
<td></td>
</tr>
<tr>
<td>main.set_squelch_level</td>
<td>d:d</td>
<td></td>
</tr>
<tr>
<td>main.inc_squelch_level</td>
<td>d:d</td>
<td></td>
</tr>
<tr>
<td>main.get_reverse</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.set_reverse</td>
<td>b:b</td>
<td></td>
</tr>
<tr>
<td>main.toggle_reverse</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.get_lock</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.set_lock</td>
<td>b:b</td>
<td></td>
</tr>
<tr>
<td>main.toggle_lock</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.get_rsid</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.set_rsid</td>
<td>b:b</td>
<td></td>
</tr>
<tr>
<td>main.toggle_rsid</td>
<td>b:n</td>
<td></td>
</tr>
<tr>
<td>main.get_trx_status</td>
<td>s:n</td>
<td></td>
</tr>
<tr>
<td>main.rx</td>
<td>n:n</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>main.tx</td>
<td>n:n</td>
<td>Transmits</td>
</tr>
<tr>
<td>main.tune</td>
<td>n:n</td>
<td>Tunes</td>
</tr>
<tr>
<td>main.abort</td>
<td>n:n</td>
<td>Aborts a transmit or tune</td>
</tr>
<tr>
<td>main.get_trx_state</td>
<td>s:n</td>
<td>Returns T/R state</td>
</tr>
<tr>
<td>main.run_macro</td>
<td>n:i</td>
<td>Runs a macro</td>
</tr>
<tr>
<td>main.get_max_macro_id</td>
<td>i:n</td>
<td>Returns the maximum macro ID number</td>
</tr>
<tr>
<td>rig.set_name</td>
<td>n:s</td>
<td>Sets the rig name for xmlrpc rig</td>
</tr>
<tr>
<td>rig.get_name</td>
<td>s:n</td>
<td>Returns the rig name previously set via rig.set_name</td>
</tr>
<tr>
<td>rig.set_frequency</td>
<td>d:d</td>
<td>Sets the RF carrier frequency. Returns the old value</td>
</tr>
<tr>
<td>rig.set_modes</td>
<td>n:A</td>
<td>Sets the list of available rig modes</td>
</tr>
<tr>
<td>rig.set_mode</td>
<td>n:s</td>
<td>Selects a mode previously added by rig.set_modes</td>
</tr>
<tr>
<td>rig.get_modes</td>
<td>A:n</td>
<td>Returns the list of available rig modes</td>
</tr>
<tr>
<td>rig.get_mode</td>
<td>s:n</td>
<td>Returns the name of the current transceiver mode</td>
</tr>
<tr>
<td>rig.set_bandwidths</td>
<td>n:A</td>
<td>Sets the list of available rig bandwidths</td>
</tr>
<tr>
<td>rig.set_bandwidth</td>
<td>n:s</td>
<td>Selects a bandwidth previously added by rig.set_bandwidths</td>
</tr>
<tr>
<td>rig.get_bandwidth</td>
<td>s:n</td>
<td>Returns the name of the current transceiver bandwidth</td>
</tr>
<tr>
<td>rig.get_bandwidths</td>
<td>A:n</td>
<td>Returns the list of available rig bandwidths</td>
</tr>
<tr>
<td>rig.take_control</td>
<td>n:n</td>
<td>Switches rig control to XML-RPC</td>
</tr>
<tr>
<td>rig.release_control</td>
<td>n:n</td>
<td>Switches rig control to previous setting</td>
</tr>
<tr>
<td>log.get_frequency</td>
<td>s:n</td>
<td>Returns the Frequency field contents</td>
</tr>
<tr>
<td>log.get_time_on</td>
<td>s:n</td>
<td>Returns the Time-On field contents</td>
</tr>
<tr>
<td>log.get_time_off</td>
<td>s:n</td>
<td>Returns the Time-Off field contents</td>
</tr>
<tr>
<td>log.get_call</td>
<td>s:n</td>
<td>Returns the Call field contents</td>
</tr>
<tr>
<td>log.get_name</td>
<td>s:n</td>
<td>Returns the Name field contents</td>
</tr>
<tr>
<td>log.get_rst_in</td>
<td>s:n</td>
<td>Returns the RST(r) field contents</td>
</tr>
<tr>
<td>log.get_rst_out</td>
<td>s:n</td>
<td>Returns the RST(s) field contents</td>
</tr>
<tr>
<td>log.get_serial_number</td>
<td>s:n</td>
<td>Returns the serial number field contents</td>
</tr>
<tr>
<td>log.set_serial_number</td>
<td>n:s</td>
<td>Sets the serial number field contents</td>
</tr>
<tr>
<td>log.get_serial_number_sen</td>
<td>s:n</td>
<td>Returns the serial number (sent) field contents</td>
</tr>
<tr>
<td>t</td>
<td>log.get_exchange</td>
<td>s:n</td>
</tr>
<tr>
<td></td>
<td>log.set_exchange</td>
<td>n:s</td>
</tr>
<tr>
<td></td>
<td>log.get_state</td>
<td>s:n</td>
</tr>
<tr>
<td></td>
<td>log.get_province</td>
<td>s:n</td>
</tr>
<tr>
<td></td>
<td>log.get_country</td>
<td>s:n</td>
</tr>
<tr>
<td></td>
<td>log.get_qth</td>
<td>s:n</td>
</tr>
<tr>
<td></td>
<td>log.get_band</td>
<td>s:n</td>
</tr>
<tr>
<td></td>
<td>log.get_notes</td>
<td>s:n</td>
</tr>
<tr>
<td></td>
<td>log.get_locator</td>
<td>s:n</td>
</tr>
<tr>
<td></td>
<td>log.get_az</td>
<td>s:n</td>
</tr>
<tr>
<td></td>
<td>log.clear</td>
<td>n:n</td>
</tr>
<tr>
<td></td>
<td>log.set_call</td>
<td>n:s</td>
</tr>
<tr>
<td></td>
<td>log.set_name</td>
<td>n:s</td>
</tr>
<tr>
<td></td>
<td>log.set_qth</td>
<td>n:s</td>
</tr>
<tr>
<td></td>
<td>log.set_locator</td>
<td>n:s</td>
</tr>
<tr>
<td></td>
<td>text.get_rx_length</td>
<td>i:n</td>
</tr>
<tr>
<td></td>
<td>text.get_rx</td>
<td>6:ii</td>
</tr>
<tr>
<td></td>
<td>text.clear_rx</td>
<td>n:n</td>
</tr>
<tr>
<td></td>
<td>text.add_tx</td>
<td>n:s</td>
</tr>
<tr>
<td></td>
<td>text.add_tx_bytes</td>
<td>n:6</td>
</tr>
<tr>
<td></td>
<td>text.clear_tx</td>
<td>n:n</td>
</tr>
<tr>
<td></td>
<td>spot.get_auto</td>
<td>b:n</td>
</tr>
<tr>
<td></td>
<td>spot.set_auto</td>
<td>n:b</td>
</tr>
<tr>
<td></td>
<td>spot.toggle_auto</td>
<td>n:b</td>
</tr>
<tr>
<td></td>
<td>spot.pskrep.get_count</td>
<td>i:n</td>
</tr>
</tbody>
</table>
### Deprecated methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Replacement Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>main.get_sideband</td>
<td>Use main.get_wf_sideband and/or rig.get_mode</td>
<td></td>
</tr>
<tr>
<td>main.set_sideband</td>
<td>Use main.set_wf_sideband and/or rig.set_mode</td>
<td></td>
</tr>
<tr>
<td>main.rsid</td>
<td>Use main.{get, set, toggle}_rsid</td>
<td></td>
</tr>
<tr>
<td>main.set_rig_name</td>
<td>Use rig.set_name</td>
<td></td>
</tr>
<tr>
<td>main.set_rig_frequency</td>
<td>Use rig.set_frequency</td>
<td></td>
</tr>
<tr>
<td>main.set_rig_modes</td>
<td>Use rig.set_modes</td>
<td></td>
</tr>
<tr>
<td>main.set_rig_mode</td>
<td>Use rig.set_mode</td>
<td></td>
</tr>
<tr>
<td>main.get_rig_mode</td>
<td>Use rig.get_mode</td>
<td></td>
</tr>
<tr>
<td>main.set_rig_bandwidth</td>
<td>Use rig.set_bandwidth</td>
<td></td>
</tr>
<tr>
<td>main.set_rig_bandwidth</td>
<td>Use rig.set_bandwidth</td>
<td></td>
</tr>
<tr>
<td>main.get_rig_bandwidth</td>
<td>Use rig.get_bandwidth</td>
<td></td>
</tr>
<tr>
<td>log.get_sideband</td>
<td>Use main.get_wf_sideband</td>
<td></td>
</tr>
</tbody>
</table>

If your external control program duplicates some of the fldigi controls such as the Rx and Tx pane you can run fldigi in a fully minimized mode. Fldigi then only provides the controls necessary for signal acquisition and waterfall management. Minimization is accomplished by setting the command line switch ( --wo). The user interface then has this appearance:

![Fldigi minimized mode](image)

The documentation for the external control program will provide additional information if this user interface is desired.
fldigi Command Line Switches

Usage:
fldigi [option...]

fldigi options:

--config-dir DIRECTORY
Look for configuration files in DIRECTORY
The default is: /home/dave/.fldigi/

--rx-ipc-key KEY
Set the receive message queue key
May be given in hex if prefixed with "0x"
The default is: 9876 or 0x2694

--tx-ipc-key KEY
Set the transmit message queue key
May be given in hex if prefixed with "0x"
The default is: 6789 or 0x1a85

--arq-server-address HOSTNAME
Set the ARQ TCP server address
The default is: 127.0.0.1

--arq-server-port PORT
Set the ARQ TCP server port
The default is: 7322

--xmlrpc-server-address HOSTNAME
Set the XML-RPC server address
The default is: 127.0.0.1

--xmlrpc-server-port PORT
Set the XML-RPC server port
The default is: 7362

--xmlrpc-allow REGEX
Allow only the methods whose names match REGEX

--xmlrpc-deny REGEX
Allow only the methods whose names don't match REGEX

--xmlrpc-list
List all available methods

--wo
hide all controls but the waterfall

--debug-level LEVEL
Set the event log verbosity

--noise
unhide controls for noise tests

--version
Print version information

--build-info
Print build information

--help
Print this option help

Standard FLTK options:

-bg COLOR, -background COLOR
Set the background color

-bg2 COLOR, -background2 COLOR
Set the secondary (text) background color

-di DISPLAY, -display DISPLAY
Set the X display to use DISPLAY,
format is "host:n.n"

-dn, -dnd or -nodn, -nodnd
Enable or disable drag and drop copy and paste in text fields

-fg COLOR, -foreground COLOR
Set the foreground color

-g GEOMETRY, -geometry GEOMETRY
Set the initial window size and position
GEOMETRY format is "WxH+X+Y"
** fldigi may override this setting **

-i, -iconic
Start fldigi in iconified state

-k, -kbd or -nok, -nokbd
Enable or disable visible keyboard focus in non-text widgets
-na CLASSNAME, -name CLASSNAME
Set the window class to CLASSNAME

-ti WINDOWTITLE, -title WINDOWTITLE
Set the window title

Additional UI options:

--font FONT[:SIZE]
Set the widget font and (optionally) size
The default is: sans:12
flarq-fast light automatic repeat request

Fast light automatic repeat request is a file transfer application that is based on the ARQ specification developed by Paul Schmidt, K9PS. It is capable of transmitting and receiving frames of ARQ data via either fldigi or MultiPsk on Windows, or fldigi on Linux. The interaction between flarq and fldigi requires no operator intervention. Program data exchange between flarq and fldigi is accomplished using a localhost socket interface. The socket interface requires that one program act as the server and the other the client. Flarq is a client program and fldigi is a server program.

Flarq will not execute unless either fldigi (preferred) or MultiPsk is already running on the host computer. If MultiPsk is used it must be set to the "socket" mode before executing flarq. If you attempt to run flarq without fldigi already running you will view an information window asking you to first start fldigi.

The ARQ transfer must take place between two systems both of which are running the flarq / fldigi. The ARQ specification and the source code for flarq are GPL licensed. Other developers wishing to duplicate or expand upon the flarq ARQ implementation may freely do so.

Flarq can be used with the following digital modem as the transport layer:

- BPSK all baud rates
- QPSK all baud rates
- MFSK all baud rates
- DOMINOEX 11 or faster (do not use FEC)
- THOR all baud rates, 11 or faster recommended
- MT63 - all baud rates, flarq timing should be increased to accommodate mt63 latency

Please note that Olivia IS NOT compatible with flarq transmissions. Olivia does not allow the transmission of the control codes required for flarq.

The main screen dialog for flarq is:

ARQ data is sent in data frames which clearly delineate from whom the data is being sent, it's purpose or type, the actual data, and a checksum value. The "Beacon" button will cause the transmission of the ARQ equivalent of a "CQ" frame which can be easily recognized by a receiving station. Upon receipt of the beacon frame a monitoring flarq will automatically insert the sending station's callsign into the edit box to the right of the "Connect" button. The receiving station can then press the "Connect" button and the connect process
begins.

Diamond Indicator
WHITE Not Connected
YELLOW Connecting
GREEN Connected - Receiving
RED Connected - Transmitting
BLACK Timed Out

The state of the connection also appears in plain text to the right of the diamond indicator.

Pressing the "Beacon" button sends both the text and T/R commands to the modem program. The beacon will repeat with a wait time between transmissions set by the Beacon interval in seconds on the Configure dialog. During the silent period between beacon transmissions the Beacon button will show the count down timer. You can stop the beacon at any time it is in the count down mode by simply pressing that button or by pressing the Abort button. The default beacon interval is 60 seconds. The minimum is 15 seconds and the maximum 300 seconds. If you find that stations are having difficulty connecting to your beacon call then you probably need to increase the time between beacons to avoid ARQ collisions similar to doubling on voice. When flarq is in the Beacon mode the words BEACON ON, and the green light on the Beacon button will be lit.

The status bar at the bottom of the main dialog contains a status text message area to the left and a progress bar to the right. During a file transfer you will be notified of actions and also see the transfer percent completion in the progress bar.

Configuring flarq

Before using flarq (and upon its initial execution) you will have to enter some configuration parameters: The highlighted field, "My Call" is the only one that you must fill in to start using the application. The folder locations for Text/Binary Files and also for the Mail Client files should all be OK as preconfigured for the operating system in use.

Read through the ARQ specification for additional detail. The Exponent is a $2^N$ factor which delineates the size of the text data block that is transmitted in a data frame. $2^5$ is 32 and should be
satisfactory for most s/n conditions. If you are experiencing many repeats you can lower the Exponent value. If the path between rx and tx stations is very good you could increase its value. flarq allows the following range of values:

<table>
<thead>
<tr>
<th>Exponent</th>
<th>Block size</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
</tr>
</tbody>
</table>

Retries specifies how many times a repeat request should be made before the link is declared DOWN.

Wait time is the time between retries.

Timeout is the time period during which NO RECEPTION of frames has occurred before the link is declared DOWN.

The Tx delay is the time between the end of the Rx and the beginning of the Tx cycle for an ARQ exchange. One-half second should be more than sufficient for most transceivers. You might be able to lower the value for your rig. Older rigs may need the value increased.

Beacon text - will be transmitted with each beacon transmission.

Sylpheed Mail Client - Sylpheed stores the message files using numeric file names without extension. Check this box to ensure that flarq stores the files in sequential numeric order without creating duplicates. This is primarily for Linux users. Windows users should leave this box unchecked unless they are using the Sylpheed Mail Client.

**Connecting**

The normal connection process is that the station with traffic will request the connecting station to send a beacon. This tests the ability of the connecting station to forward. The station with traffic will then be the one to press the Connect button. The receiving station will see the beacon message displayed in the flarq text area and also see the beacon station's callsign appear in the callsign box. The connecting station presses the connect button and after a few automatic exchanges the diamond indicator to the right of the callsign turns green indicating that they are successfully connected. A connect may take a few retries if the transmission path s/n is marginal. During the connect process the diamond indicator will be yellow.

After the two stations are CONNECTED either operator may effect a file transfer. Who goes first may be negotiated using fldigi in a plain text mode or using the flarq "Plain Talk" facility, once connected. The fldigi T/R functions can be used in parallel with any ARQ transmissions. It might be best to conclude those negotiations before establishing the ARQ connection.

Either station may initiate a DISCONNECT process.
This is what the outgoing beacon will look like on fldigi:

```
<soh> FLARQ Beacon >>< e W1HKJ-1
<soh>00b WIHKJ-1:72 10DD<eot>
```

The <soh> and <eot> are control codes that surround every ARQ frame. Fldigi recognizes that it is connected to flarq and shows the control codes as ASCII named equivalents since they are not normally printable.

The text will appear very similar on the receiving end of the exchange.

**Transferring files**

You may transfer several different file types:

- Email - created using Flarq's composer, Outlook Express, Outlook, Thunderbird, Sylpheed or any other email client, or received via the internet and handled by the email client.
- Text - any ASCII file which does not contain non-printable text
- Image - any image file, jpeg, png, bmp, etc.
- Binary - any file containing arbitrary data where each byte is anything from 00 hex to FF hex.

**Transferring Text, Images or Binary Files**

If you select Text, Image or Binary file for transfer a regular file picker dialog is opened. You can navigate anywhere in the file system to pick a file. The default location for the files are unique in Windows and Linux. In Linux the default location is in $HOME/ARQsend and in Windows it is c:\NBEMS\ARQsend. Move files to that location to make finding the target file easy. Use the file manager or move the file using command line processing in a terminal window. Image and binary files will be converted into ASCII text files using base64 conversion. This basically is the same type of conversion that an email client would perform on an image or binary attachment. The file is encoded using base64 coding at the sending end and then decoded back to its original form at the receiving end. At the conclusion of a satisfactory ARQ transfer the two files will be identical, including name and size. The target directory for received files is $HOME/ARQrcvd in Linux, and c:\NBEMS\ARQrcvd in Windows. The receiving station opens the c:\NBEMS\Mail\ARQin folder and drags the incoming message placed there by flarq over to the Outlook Express email client, or just double-clicks on the
.eml message to open it in the default email client. It is the reciprocal process from that which the sending station uses.

During the transfer the sending station transmits blocks of data. Each block has a header, data, checksum and trailing component. The receiving station acknowledges which blocks have been correctly received and which need retransmission. Missing blocks sometimes occur in the middle of the set of acknowledged blocks. The text in the flarq text window will only update as contiguous blocks are available. So you might see the update occur in what appears to be random intervals. As the sending and receiving stations go from receive-to-transmit-to-receive the diamond indicator will toggle from green to red and back to green.

**Composing Email traffic without an Email client**

Flarq has a built-in email composer that creates files with a minimum of email overhead. It is a text only composer with no attachment or other niceties associated with a normal email client. It does produce very small email files which is a benefit when transmitting files over slower modem baud rates. Click on the "Compose" menu item. That will open the following dialog:

![Flarq Mail Composer dialog](image)

Enter the email address of the destination addressee, the subject and the email body text. Then press "Save" to save the file for later transmission.

You can create a template for later use and then save the template by pressing SHIFT while clicking on the Save button. Here is an example of a template in preparation:
This file will be saved as "wxstatus.tpl". Flarq recognizes the tpl extension as the template file. When you later click the Template button the new "wxstatus.tpl" template will be available to load and use in creating the wx status report. After opening a template you can either modify it and save as a new or overwrite the existing template, or you can save the filled-in template as a regular file for subsequent transmission.
Using Email client - Outlook Express

You can use Outlook Express for the email client to create outgoing ARQ traffic. Just remember that you should create all email traffic as ASCII text and not HTML text to reduce the size of the message body. You are going to send this via an RF link and not over the internet with a high speed connection.

Create your email just as you would for transfer over the internet and then save it in the Drafts folder. In Outlook Express, click Create Email and use the format, name@phonenumber, such as information@8005551212, for the address if it is to be delivered by phone, and is not an email. Save each message in the Drafts folder by clicking File, and then Save. Exit the composition window. Open the ARQout folder (located at c:\NBEMS\Mail\ARQout) on the desktop along with Outlook Express as shown below. Then drag the
message from the "To ... Subject" area of Outlook Express and drop it on the ARQout folder. This places them in a folder that flarq can locate. When you select the flarq menu item "Send / Email" a dialog will open that shows the contents of the messages that are in the ARQout folder:

<table>
<thead>
<tr>
<th>Date</th>
<th>To</th>
<th>Subj</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 Dec 2007 18:26:01</td>
<td>&lt;kh6ty</td>
<td>Test Message #1 for ARQout</td>
</tr>
<tr>
<td>9 Dec 2007 18:26:28</td>
<td>&quot;kh6ty</td>
<td>Test Message #2 for ARQout</td>
</tr>
</tbody>
</table>

Multiple email entries would appear on separate lines with scroll bars as appropriate. You highlight the desired file and then press "Send" or the Enter key to commence the file transfer. The email may contain attachments (which may be images) or be just plain text. Remember that this is a fairly slow transfer process so small is beautiful. If the email has images then it will be in html and base-64 format. That adds a lot of overhead to the email. "Cancel" aborts the email transfer process. After a successful transfer from sending to receiving station the email is automatically moved from the ARQout to the ARQsent folder. If you are using Outlook Express you can open the c:\NBEMS\Mail\ARQsent folder the same way that the ARQout folder can be opened. At the receiving end the email will be placed in the Sylpheed ARQin folder. Sylpheed does not need to be executing for this to occur at either end.

**Using Email client - Sylpheed**

flarq has been optimally designed to interoperate with Sylpheed as its email client for emergency communications of email traffic. When you install Sylpheed you will be asked to choose a default directory for the mail store. On Linux this should be the default $HOME/Mail. On Windows you should choose c:\NBEMS\Mail. Three additional folders are used for transferring files between the flarq application and Sylpheed. These are:

- ARQin
- ARQout, and
- ARQsent

You can create these folders from within the Sylpheed application.

The above image shows the folders already in place. If they were not present they could be created by right clicking on the "Mailbox (MH)" icon and selecting "Create new folder". Name each new folder as specified above and shown in the image. These folders are required for flarq to be able to work with the Sylpheed email messages. Each message in Sylpheed is a separate file. These are usually numbered sequentially in each of the Sylpheed folders. flarq manages the correct sequential naming of files as they are transferred in, out and moved between these three folders. If you run the flarq application
before Sylpheed then the c:\NBEMS\Mail and the c:\NBEMS\Mail\ARQin, ARQout, and ARQsent folders will be created by that application and will appear in the Sylpheed folder system.

To create a new email traffic you press the "Compose" button. Fill out the email as usual and then press the "Draft" button from within the composer. The new message for transfer via flarq is now in the Drafts folder shown above. Open that folder by clicking on it. Select the desired draft message and drag and drop in onto the ARQout folder icon. That's it! The message is now ready for flarq to perform the ARQ transfer.

Upon completion of the transfer flarq will move the message to the ARQsent folder. Sylpheed will not immediately recognize that the change has occurred. That is easily accomplished by either changing to another folder and then back again or by right clicking on the ARQout (or ARQin, or ARQsent) folder icon and selecting "Update summary". Sylpheed will re-read the folder contents and adjust its views accordingly.

Incoming traffic will be placed in the ARQin folder. You may have to refresh the folder as described above.

flarq can find and parse the newly created email document that has been moved or copied to the ARQout folder. If you select to send email a picker dialog will appear that lists all of the outgoing email traffic that is contained in the Sylpheed ARQout folder.

Other Email clients

If you use an email client other than Sylpheed or Outlook Express you can transfer emails as above. Just be sure that the emails have file names with the extension "eml" as in "mytest_message.eml".

Aborting a transmission

The transmission may be aborted by either the sending or the receiving station at any time during the file transfer. When Connected and transferring a file, the Connect button is re-labeled Abort. Since data is sent in multiple blocks the actual abort will take place at the conclusion of the current group transmission. Abort will cause the transfer to be interrupted. The connection will be maintained and a new transfer can be initiated if required.

Plain Talk

After a connection is established the two stations can exchange text using the "Plain Talk" entry control at the bottom of the main dialog. Enter up to 80 characters and then press the Enter key to transmit the text. This text is sent UNPROTO, which means that NO repeat request will be made. The block is sent without any acknowledgement from the receiving station. This is not an ACK/NACK system, but meant only as a way of allowing quick operator to operator exchanges without having to disconnect and use the digital modem program keyboard entry. "Plain Talk" can be interspersed with normal ARQ file transfer blocks.
flwrap - 1.3.1

Lets suppose you want to send a critical message that must be received exactly as it was sent, or maybe you want to send a data file where it's highly unlikely that even a trained operator could tell that there was an error in receiving the data. Or perhaps you need to broadcast a weather bulletin or situation report to multiple stations and allow each station to verify that the message was received exactly as it was sent.

Flwrap is an application that is designed to take care of each of these situations. Flwrap allows you to transmit a text message, image, or binary file to either single or multiple stations and allow each receiving station to verify that the transmission was received without error. In the discussion which follows the application name is Flwrap, the file encapsulation process is called wrap, wrapping and the encapsulated file is said to be wrapped.

The flwrap executable is a small desktop application that encapsulates a text file, an image file, or a binary file within a set of identifier blocks. These blocks include a 16 bit checksum that is used to test the encapsulated file for integrity. Flwrap is designed to be used to best advantage with fldigi but can be used with any digital modem program. Fldigi can recognize an wrapped transmission in the Rx data stream and automatically save the file. Fldigi can process multiple wrapped files that are sent in rapid succession. Fldigi does this without operator intervention. The conversion of files to and from the wrapped format do still require an operator.

A text file is encapsulated without changes to the text. Images and other binary files are first converted to a base-64 format. The converted file is then encapsulated. The header blocks identify the type of file and whether or not it has been converted to base-64. Files with any of the following extensions will be treated as binary data files and converted to the base-64 format:

.jpg, .jpeg,.png,.gif,.bmp,.ico,.zip,.gz,.tgz,.and.bz2

Using flwrap is very simple. Install the executable file and then create a short cut to the executable on the desktop.

Windows users may find it a bit confusing to install a program that does not have an installation wizard, but it really is very easy to do. The flwrap.exe program is contained in the zipped fldigi archive that also contains fldigi.exe and flarq.exe. If you have already installed fldigi from this archive then you already have flwrap.exe installed. The installer creates a desktop icon for the application. Additional shortcuts to flwrap may be dragged into any convenient folder where files to be wrapped are located.

Left click (or double click) to launch the application and you will see a brief description of usage.
You may wrap/unwrap a file using three different techniques.

1) Drag and drop the file to be wrapped onto the desktop icon

2) Drag and drop the file into the drop box in the flwrap dialog

3) Invoke the application from a command line

flwrap FILENAME

All three can be used on Linux and Windows, but (1) and (2) are the most user friendly.

(2) and (3) can be used on OS X, but (2) is prefered. OS X users should not expect technique (1) to work. This is a limitation of the OS X desktop manager and not flwrap.

Either technique will yield the following when successful and you will see the following results:
!! FLWRAPPED FILES ARE ALWAYS PLACED IN THE SAME FOLDER AS THE SOURCE FILE !!

The file "Scottie_small.jpg" has been wrapped by the drag and drop technique. The results of the flwrap appear in the flwrap info dialog and you can see the new file in the original source directory (all on a Linux computer). The original file was an image and so it was converted to ASCII format using base-64 encoding before it was wrapped.

[WRAP:beg][WRAP:lf][WRAP:fn Scottie_small.jpg]
[b64:start]AUxaTUEAABzXXQAAAQAf7Yb7fCEQFjjkd4QJ1hhjKFp2u0ecgqwnuNw3ZIVRxMRPng1Ghkt9zOJ117omOU0L/w9j2ace8X3+DAO/4C54Tr6Xuru1g4tiuMU2oaA2+R2tLXaNEkTeUyKgWG... removed data for this example
GwFcGQIpP8wA6WQTbzm5ggV7Qj524JWO2cEcS2nRFTG6EeoQdvmGsLwXwM1eQA=[b64:end][WRAP:chksum 8710][WRAP:end]

• [WRAP:beg] ... [WRAP:end] bracket the contents of the file
• [WRAP:lf] specifies that the single line feed character delineates lines
• [WRAP:fn Scottie_small.jpg] specifies that the source (and destination) filename is "Scottie_small.jpg"
• [b64:start] ... [b64:end] bracket the base-64 encoded contents
• [WRAP:chksum 8710] is the checksum associated with the file and it's contents

At the receiving end the file might be named "wrap-090318-001.wrap." Drag and drop it onto the flwrap launch icon to unwrap the file.

If the transfer had errors then the unwrap process would fail. The message dialog would so indicate:

![Checksum failed 8710 in file 482A computed]

You can continue to drop files into the drop box and each new file will be treated independently.
flwrap with compression

flwrap also compresses and expands files in addition to encapsulating them. Binary files such as a MS Word file or an executable must be converted to base 64 (http://en.wikipedia.org/wiki/Base64) before being wrapped. Those files are first compressed before the base64 encoding is applied. A Word document recipe for Pfefferseuse cookies is 11 kB. When wrapped by flwrap it is 3.1 kB in size.

The wrapped file is an ASCII text file suitable for transmission using any of the fldigi modes other than CW, RTTY or the fuzzy modes such as FeldHell.

The compression method used in flwrap is called LZMA, Lempel–Ziv–Markov chain algorithm, a form of dictionary compression. There are times that LZMA cannot compress the data. The data set might be too small or have zero redundancy. Under those circumstances you will warned with a appropriate message:

![Flwrap 1.3.1](http://www.w1hkj.com/Flwrap/pfefferneusse.doc)

Test files used for this help documentation:

http://www.w1hkj.com/Flwrap/pfefferneusse.doc
http://www.w1hkj.com/Flwrap/Scottie_small.jpg
http://www.w1hkj.com/Flwrap/SMPTE_Color_Bars.jpg
flmsg - quick guide

flmsg is a simple forms management editor for the amateur radio supported standard message formats. These current include:

- ICS-203 - Organization Assignment List
- ICS-205 - Incident Radio Communications Plan
- ICS-205A - Comms List - special USCG Plan
- ICS-206 - Medical Plan
- ICS-213 - emergency management report
- ICS-214 - Unit log
- ICS-216 - Radio Requirements Worksheet
- Radiogram - NTS message
- Plaintext - generic message format
- Blank - very simple text format with no preset fields
- Drag and Drop - target control (widget) that accepts either a data file (.203 etc), a wrapped data file (.wrap), or the text associated with a data file. The later may be a copy and paste from another application such as fldigi or a text editor.

It's data files are pure ASCII text that can be sent from point to point using the internet, amateur radio, or other electronic link. The data files are designed to minimize the transfer size. This is particularly important on amateur HF. The data file and the transfer file are one in the same, but can be further encapsulated using either flarq or wrap for the purpose of confirming the received file integrity.

flmsg Menu's

File:
- New - clear all fields and name the default file "new.f2s" (new.m2s for radiogram)
- Open - open an existing file flmsg data files have the extension
  - ".203" for ICS-203 forms
  - ".205" for ICS-205 forms
  - ".25A" for ICS-206A forms
  - ".206" for ICS-206 forms
  - ".213" for ICS-213 forms
  - ".214" for ICS-214 forms
  - ".216" for ICS-216 forms
  - ".m2s", read as "message 2 send" for radiogram forms
  - ".p2s" for plain text, generic forms
  - ".b2s" for blank forms
- Save - save the current file to the name in the "file:" display box
- Save As - save using a new filename that the user provides
View- write the data to specified type of file

- **Html delivery** - viewed in default browser, contains only those elements sent to final recipient
- **Html file copy** - viewed in default browser, contains ALL fields including record keeping
- **Text** - viewed in default text editor - suitable for CW / Voice transmission

Q-forms

- **Import** - Import the data fields from a Qforms eXtended Markup Language (xml) file
- **Export** - Export the data fields to a Qforms compatible xml file

Wrap (Import / Export / **AutoSend**)

- **Import** the data fields from a Wrapped data file. If the data file is corrupt you will be given the opportunity to either allow flmsg to recover as many fields as possible or to view the file using the default text editor.
- **Export** the data fields to a Wrapped data file
- **Create** a wrapped datafile and save in the NBEMS.files/WRAP/auto directory. If running, fldigi will find and automatically transmit the file.

Template:

- **Load** - load an existing template file - the default extension for the supported files are:
  - ICS-203 template- ".203T"
  - ICS-205 template - ".205T"
  - ICS-205A template - ".25T"
  - ICS-206 template - ".206T"
  - ICS-213 template - ".213T"
  - ICS-214 template - ".214T"
  - ICS-216 template - ".216T"
  - Radiogram template - ".m2t"
  - Plain text template - ".p2t"
  - Blank text template - ".b2t"

- **Save** - save the current form as a template file, using the default (or current) filename
Save As - save the current form as a template file, user provides the filename.

Config:

Open the configuration dialog

Note: data files and template files for ICS213 are identical. The only difference is their location in the computer directory structure and their extension. Message files and template files maintain their uniqueness by virtue of their file name. If you reuse a filename the old data will be lost.

File locations:

On XP: C:\Documents and Settings\<username>\NBEMS.files
On Vista: C:\Users\<username>\NBEMS.files
On Linux: /home/<username>/.nbems
On Puppy: /root/.nbems
On OS X: /home/<username>/.nbems

Data files are located in the sub directory "ICS/messages"
Template files are located in the sub directory "ICS/templates"
View files (.rtf, .html, .txt) are located in the "ICS/" subdirectory.

Configuring flmsg

The configuration items for flmsg are just a few and can be changed on the configuration dialog:

- Date format - select the date format that will be used for both ICS213 and Radiogram
- Time format - select the time format that will be used for both ICS213 and Radiogram
- Radiogram format -
- Fill in the data relating to the originating amateur station
- # words per line to be used when formatting the radiogram message text
• Wrap - automatically open the target folder (directory) when the file is exported
• Naming files - automatic file name generation as:
  CALLSIGN-YYYYMMDD-HHMMSS(Z)-NNNN where
  CALLSIGN is the operators callsign
  YYYYMMDD is year, day, month
  HHMMSS is hours, minutes, seconds local or Zulu at time file is created
  NNNN is an auto incremented serial number
You can elect to use any or none of the autogeneration components
Filename extensions are f2s for ICS-213 data, f2t for ICS-213 templates and
m2s for radiogram data.
  • Radiogram serial numbers can be auto-incremented. The auto-increment number is also used
    for the file name. In the example shown the next Radiogram will be number 104 and the
    associated file sill be W1HKJ-104.m2s

Qform data file
flmsg can read and write Qform ICS213 data files. The Qform data file is larger by virtue of the xml
structure. The file size ratio is less when the content is larger.

Viewing the data in a printable format
The program can produce a viewable document in ASCII text and Hypertext Markup Language (html)
file formats. After creating the document flmsg will request the file to be opened by the default viewer /
editor for that type of document.

Html Text Format
The html Text Format file that the File/Write menu item produces can be opened with any web browser
program. Use that software for printing the report.

Note ICS Forms Source:
http://training.fema.gov/EMIWeb/IS/ICSResource/ICSResCntr_Forms.htm
Guide to ICS Forms:
http://www.fs.fed.us/r1/fire/incident/ics-forms/icsguide.pdf

File: /home/w1npp.org/ares/manuals/fldigi-3.21_Manual.odt