

# FLDIGI Users Manual

3.23

Generated by Doxygen 1.8.10

Thu Jan 28 2016 10:01:04



# Contents

- 1 FLDIGI Users Manual - Version 3.23** **1**
- 1.1 Fldigi Configuration and Operational Instructions . . . . . 1
  
- 2 Configuration** **3**
- 2.1 User Interface configuration . . . . . 3
- 2.2 Windows Specific Install / Config . . . . . 3
- 2.3 Other Configuration options . . . . . 3
- 2.4 Command Line Switches . . . . . 4
- 2.5 Modem Configuration Options . . . . . 4
- 2.6 Configure Operator . . . . . 5
- 2.7 Sound Card Configuration . . . . . 6
  - 2.7.1 Right Channel Audio Output . . . . . 9
  - 2.7.2 WAV File Sample Rate . . . . . 10
  - 2.7.3 Multiple sound cards . . . . . 10
- 2.8 Rig Control . . . . . 12
  - 2.8.1 Rig Configuration . . . . . 13
  - 2.8.2 RigCAT control . . . . . 15
  - 2.8.3 Hamlib CAT control . . . . . 16
  - 2.8.4 Xml-Rpc CAT . . . . . 16
- 2.9 RX/TX Audio Adjustment . . . . . 17
  - 2.9.1 Receive audio . . . . . 17
  - 2.9.2 Windows Audio Properties . . . . . 18
  - 2.9.3 Transmit audio . . . . . 21
- 2.10 New Installation . . . . . 22
- 2.11 Configure ARQ/KISS I/O . . . . . 25
  - 2.11.1 I/O Configuration . . . . . 26
    - 2.11.1.1 ARQ I/O . . . . . 26
    - 2.11.1.2 KISS I/O . . . . . 27
    - 2.11.1.3 XMLRPC I/O . . . . . 28
- 2.12 Installing Fldigi on Windows . . . . . 29
  - 2.12.1 Special Note for Vista/Win7 from user! . . . . . 30

2.13	New Install Wizard	30
2.14	Callsign DB Configuration	34
2.14.1	QRZ	34
2.14.2	EQSL	35
2.14.3	FLdigi Logging	35
2.15	Colors and Fonts	36
2.15.1	Text Ctrlis	37
2.15.2	Character Set Selection	37
2.15.3	Frequency Display	39
2.15.4	Logging controls	39
2.15.5	Func keys	40
2.15.6	Tab Colors	40
2.15.7	Light Buttons Colors	41
2.15.8	Signal Level	41
2.16	Configure FFT Scan	42
2.16.1	FFT Scan	43
2.17	RSID / Video / ID Configuration	44
2.17.1	Reed Solomon Identifier	44
2.17.2	Video Text	46
2.17.3	CW Postamble	48
2.18	PSKmail Configuration	50
2.19	Configure Touch Screen	51
2.19.1	Touch Screen	52
2.20	User Interface Configuration - Browser	53
2.21	User Interface Configuration - Contest	55
2.22	User Interface Configuration - General	56
2.22.1	Visible Modes	58
2.22.2	Limit Modes	59
2.23	User Interface Configuration - Logging	59
2.23.1	Capturing Log data	60
2.23.2	Pop ups	61
2.23.3	Country Files	61
2.23.4	Internal Logbook	61
2.23.5	Logbook Network Address	62
2.24	User Interface Configuration - Macros	63
2.25	User Interface Configuration - WF Controls	65
2.26	Waterfall Configuration	66
2.26.1	FFT Waterfall	68
2.26.2	Waterfall Mouse Behavior	69
2.26.2.1	Waterfall Mouse Behavior Tailoring	69

2.27	WEFAX	69
2.27.1	Configuration	70
2.27.1.1	Embedded vs floating display mode	71
2.27.1.2	Hide transmission window	72
2.27.1.3	Logging messages to ADIF file	72
2.27.1.4	Frequency shift adjustment	72
2.27.1.5	Maximum rows number	73
2.27.1.6	Destination directory for saved images	73
2.27.1.7	Monochrome images	73
2.27.2	Transmitting an Image	73
2.27.3	Receiving an image	74
2.27.3.1	APT control reception mode	76
2.27.3.2	Manual reception mode (Non-Stop)	76
2.27.3.3	Input FIR filters	77
2.27.3.4	Centering an image	78
2.27.3.5	Picture with a slant	78
2.27.3.6	Automatic centering	78
2.27.3.7	Image detection based on signal power	79
2.27.3.8	AFC: Automatic Frequency Control	79
2.27.3.9	Noise elimination	79
2.27.3.10	Binary images	79
2.27.3.11	Displaying received files	79
2.28	Working Logs	80
2.29	Contestia Configuration	82
2.30	CW Configuration	83
2.30.1	Timing	84
2.30.2	QSK	88
2.30.3	PROSIGNS	89
2.31	DominoEX Configuration	90
2.32	Feld Hell Configuration	91
2.33	FSQ Configuration	92
2.33.1	Rx Parameters	92
2.33.2	Tx Parameters	93
2.33.3	Logging	93
2.34	IFKP Configuration	94
2.34.1	Tx Parameters	94
2.34.2	Logging	94
2.35	RSID / Video / ID Configuration	95
2.35.1	Reed Solomon Identifier	95
2.35.2	Video Text	97

2.35.3	CW Postamble	99
2.36	Miscellaneous Configuration	101
2.36.1	Sweet spot	101
2.36.2	Callsign spotting	102
2.36.3	CPU performance	103
2.36.4	Text i/o	103
2.36.5	NBEMS (flmsg / flwrap) interface	104
2.36.6	DTMF decoding	105
2.36.7	WX rss-xml queries	105
2.37	MT63 Configuration	107
2.38	Olivia Configuration	108
2.39	PSK Configuration	109
2.40	RTTY / FSK Configuration	110
2.41	Thor Configuration	114
<b>3</b>	<b>Logging</b>	<b>115</b>
3.1	Cabrillo Report	115
3.2	Exporting Logbook Data	117
3.2.1	Export ADIF	117
3.2.2	Export Text / CSV	118
3.3	QSO Logbook	119
3.3.1	List of Log Fields	119
3.3.2	Digital Modes Signal Reports	120
3.3.3	Capturing QSO Data	120
3.4	User Interface Configuration - Logging	125
3.4.1	Capturing Log data	125
3.4.2	Pop ups	126
3.4.3	Country Files	126
3.4.4	Internal Logbook	127
3.4.5	Logbook Network Address	127
<b>4</b>	<b>Macros</b>	<b>129</b>
4.1	Macros	129
4.1.1	Macro tags	130
4.1.2	Modem macro tags	134
4.1.3	Other Modems	135
4.1.4	Macro Editor	135
4.1.5	Macro Display Symbols	137
4.1.6	Contest macro tags	138
4.2	Inline Macros	138
4.2.1	CW Code Practice Transmission	138

4.2.2	QSY:ffff.f:aaaa test	139
4.2.3	Advanced QSY operations	140
4.2.4	ARRL style broadcast in multiple modes	140
4.2.5	Appearance of tx buffer	140
4.3	Delayed Execution Macro Tags	141
4.3.1	Check in to a net with video ID	141
4.4	Advanced QSY operations	142
4.4.1	Distinct frequencies	142
4.4.2	Frequencies with increments	143
4.4.3	Combination with <TIMER> macro tag.	143
4.4.4	Execution errors.	143
4.4.5	Invalid frequency range	143
4.4.6	Increment must be positive	143
4.4.7	Frequency not positive	143
4.4.8	Frequencies must be increasing	144
4.4.9	Inline Macro Tags	144
4.4.10	Delayed Execution Macro Tags	144
4.5	Exec Macro	144
4.5.1	Exported variables	144
4.5.2	Detection of existing scripts	145
4.5.3	Querying an external database	146
4.5.4	Google Earth Map	146
4.5.5	Custom dates/times	146
<b>5</b>	<b>Modems</b>	<b>149</b>
5.1	Contestia	149
5.2	CW	150
5.2.1	Carrier Frequency	150
5.2.2	QSK and the TAB KEY	150
5.2.3	Pausing transmit	151
5.2.4	Aborting transmit	151
5.2.5	WPM adjustment	151
5.2.6	Farnsworth keying	151
5.2.7	CW configuration	152
5.2.8	Prosigns	152
5.3	DominoEX	152
5.4	Hellschreiber	154
5.4.1	Hellschreiber modem	154
5.4.2	Hellschreiber modes	154
5.4.3	Hellschreiber Waterfall	154

5.5	FSQ	155
5.5.1	Modulation Scheme	157
5.5.2	Transmit Functions	159
5.5.3	FSQ Protocol	160
5.5.4	Directed Mode	160
5.5.5	Triggers	162
5.5.6	Commands	163
5.5.7	Relaying messages	164
5.5.8	Special keys	165
5.5.9	Heard List	165
5.5.10	Image Mode	166
5.5.11	Disable RxID	170
5.6	IFKP	170
5.6.1	Incremental Frequency Keying Plus	170
5.6.2	IFKP heard list	171
5.6.3	IFKP specification	172
5.6.4	IFKP Varicode	172
5.6.5	IFKP xmt speed	174
5.6.6	IFKP Image Mode	174
5.6.7	Avatars	177
5.7	MFSK	179
5.7.1	MFSK Picture Mode	179
5.7.2	Transmitting an Image	181
5.7.3	Picture with a slant	183
5.8	MT63	183
5.9	NAVTEX and SITOR-B	187
5.9.1	Transmitting a text	188
5.9.2	Receiving a text	188
5.9.3	AFC: Automatic Frequency Control	189
5.9.4	Logging Navtex data to ADIF files	189
5.9.5	Configuration	190
5.9.6	ADIF journal file	191
5.9.7	Stations file	192
5.9.8	XML/RPC functions	192
5.10	Olivia	193
5.11	PSK Single and Multi-Channel Modems	195
5.11.1	BPSK, QPSK, 8PSK modems	195
5.11.2	Multi-Channel BPSK modems	197
5.11.3	8PSK modems	197
5.12	RTTY	200



5.12.1	RTTY modulator	201
5.12.2	RTTY demodulator	206
5.13	Thor	207
5.13.1	Thor Image Mode	208
5.13.2	Avatars	211
5.14	Throb	212
5.14.1	Throb baud rates and tone spacings	212
5.15	WEFAX	213
5.15.1	Configuration	213
5.15.1.1	Embedded vs floating display mode	214
5.15.1.2	Hide transmission window	215
5.15.1.3	Logging messages to ADIF file	215
5.15.1.4	Frequency shift adjustment	215
5.15.1.5	Maximum rows number	216
5.15.1.6	Destination directory for saved images	216
5.15.1.7	Monochrome images	216
5.15.2	Transmitting an Image	216
5.15.3	Receiving an image	217
5.15.3.1	APT control reception mode	219
5.15.3.2	Manual reception mode (Non-Stop)	219
5.15.3.3	Input FIR filters	220
5.15.3.4	Centering an image	221
5.15.3.5	Picture with a slant	221
5.15.3.6	Automatic centering	221
5.15.3.7	Image detection based on signal power	222
5.15.3.8	AFC: Automatic Frequency Control	222
5.15.3.9	Noise elimination	222
5.15.3.10	Binary images	222
5.15.3.11	Displaying received files	222
5.16	WWV transmit mode	223
5.17	Mode Table	224
5.17.1	PSK	224
5.17.2	PSKR / PSK-FEC	224
5.17.3	QPSK	226
5.17.4	8PSK	227
5.17.5	Contestia	227
5.17.6	DominoEX	228
5.17.7	MFSK	228
5.17.8	MT-63	228
5.17.9	Olivia	229

5.17.10 RTTY . . . . .	229
5.17.11 THOR . . . . .	229
5.17.12 THROB . . . . .	230
<b>6 Operating</b>	<b>231</b>
6.1 Contest - How To . . . . .	232
6.1.1 Contest Fields . . . . .	232
6.1.2 Restarting a contest session . . . . .	233
6.1.3 Remembering a contact . . . . .	233
6.1.4 Saving the entire session . . . . .	234
6.1.5 Contesting Macro Tips . . . . .	234
6.1.6 RUN Macros . . . . .	235
6.1.7 S & P Macros . . . . .	236
6.2 CW Keying . . . . .	239
6.3 Digiscope Display . . . . .	240
6.3.1 CW . . . . .	241
6.3.2 DominoEX / Thor . . . . .	241
6.3.3 MFSK . . . . .	242
6.3.4 PSK . . . . .	242
6.3.5 RTTY . . . . .	243
6.4 Digiscope Display - WWV mode . . . . .	244
6.5 DXCC List etal . . . . .	247
6.6 Configure FFT Scan . . . . .	249
6.6.1 FFT Scan . . . . .	249
6.7 Frequency Analyzer . . . . .	250
6.8 Keyboard Operation . . . . .	251
6.8.1 Function Keys . . . . .	253
6.9 Extended Characters . . . . .	253
6.10 KML . . . . .	254
6.10.1 KML generation from Navtex messages . . . . .	255
6.10.2 KML generation from SYNOP reports . . . . .	256
6.10.3 KML files structure . . . . .	256
6.10.4 Extended data . . . . .	256
6.10.5 Parameters . . . . .	257
6.10.6 Destination directory . . . . .	258
6.10.7 KML root file . . . . .	259
6.10.8 KML refresh interval . . . . .	259
6.10.9 Cleanup on startup . . . . .	259
6.10.10 Merging distance . . . . .	259
6.10.11 KML balloon display type . . . . .	259

6.10.11.1 Plain text . . . . .	259
6.10.11.2 HTML tables . . . . .	260
6.10.11.3 Distinct HTML matrix . . . . .	260
6.10.12 Data Retention Time . . . . .	261
6.10.13 Command run on KML creation . . . . .	261
6.10.13.1 Example of commands . . . . .	261
6.11 Menus . . . . .	263
6.11.1 Menu Hierarchy . . . . .	263
6.11.1.1 Files . . . . .	264
6.11.1.2 Configure Scripts . . . . .	266
6.11.1.3 Op Mode . . . . .	266
6.11.1.4 Configure . . . . .	275
6.11.1.5 View . . . . .	275
6.11.1.6 Logbook . . . . .	277
6.11.1.7 Help . . . . .	278
6.11.1.8 Spot, Tx RSID, Rx RSID, and Tune Buttons . . . . .	279
6.12 Mouse and Keyboard Shortcuts . . . . .	279
6.12.1 Main window . . . . .	279
6.12.1.1 Text input fields . . . . .	279
6.12.1.2 Waterfall display . . . . .	281
6.13 Notifier . . . . .	283
6.14 Operating Controls and Displays . . . . .	288
6.14.1 Mode Status Indicators . . . . .	292
6.14.2 Transmit level attenuator . . . . .	292
6.14.3 Operating split with fldigi / flrig . . . . .	293
6.15 Operating Multiple Copies . . . . .	294
6.16 PSK Reporter . . . . .	295
6.16.1 Automatically spot callsigns in decoded text . . . . .	295
6.16.2 Send reception report when logging a QSO . . . . .	295
6.16.3 Report QRG (etc.) . . . . .	296
6.16.4 Host and port . . . . .	296
6.17 Rig Control . . . . .	297
6.17.1 Rig Configuration . . . . .	298
6.17.2 RigCAT control . . . . .	300
6.17.3 Hamlib CAT control . . . . .	301
6.17.4 Xml-Rpc CAT . . . . .	302
6.18 RST and RSQ Reporting . . . . .	303
6.18.1 RST . . . . .	303
6.18.1.1 READABILITY . . . . .	303
6.18.1.2 SIGNAL STRENGTH . . . . .	303

6.18.1.3	TONE	303
6.18.2	RSQ	304
6.18.2.1	READABILITY	304
6.18.2.2	STRENGTH	304
6.18.2.3	QUALITY	304
6.19	Signal Browser	305
6.20	Supporting Data Files Acquisition	308
6.21	SYNOP decoding	309
6.21.1	What is Synop	309
6.21.1.1	Weather data	309
6.21.1.2	Synop reports broadcasts	309
6.21.2	Parameters	311
6.21.3	Reception window	311
6.21.4	Data files	313
6.21.4.1	Outputs	314
6.21.4.2	Output formats	315
6.21.5	Command-line SYNOP decoder program : synop_tool	315
6.22	UTF-8 support	316
6.23	Using the FLDigi Rt. Channel for transmitter PTT	317
6.24	Working Logs	318
6.25	WWV transmit mode	320
<b>7</b>	<b>Developers</b>	<b>323</b>
7.1	Build Information	323
7.1.1	Build Info	323
7.2	Google Maps	324
7.3	Installing from Source	326
7.4	FLDIGI KISS Hardware Commands	326
7.4.1	Set Commands	326
7.4.2	Query Commands	327
7.4.2.1	Query Commands (FLDIGI to HOST)	328
7.4.3	Broadcast Status Messages (FLDIGI to HOST)	328
7.4.4	Format of Hardware Commands	329
7.4.5	Format of RAW Data	329
7.5	Parse UALR	329
7.6	Pseudo FSK	331
7.7	Rig Xml How to	332
7.8	ualr telnet	338
7.9	Xmlrpc Control	339
7.9.1	XML Command Symbol Interpretation	340

7.9.2	Table of XML Commands	340
7.9.3	Minimized WF Window	345
<b>8</b>	<b>Installing fldigi on a thumb drive</b>	<b>347</b>
<b>9</b>	<b>Configure Scripting</b>	<b>349</b>
9.1	Configure Scripting	349
9.1.1	Executing Scripts	350
9.1.2	Generating Scripts	350
9.2	Configure Script Commands	351
9.3	Scripting Syntax	352
9.3.1	Structured Commands	352
9.3.2	Dot Notation Commands	353
9.3.3	Parameters	353
9.3.4	Script Files	353
9.4	FLDigi Commands	354
9.4.1	Frequency Control	354
9.4.2	Mode	354
9.4.3	Waterfall Offset (Hz)	354
9.4.4	RxID	354
9.4.5	TxID	354
9.4.6	Spot	354
9.4.7	Rev	355
9.4.8	AFC	355
9.4.9	Lk	355
9.4.10	SQL	355
9.4.11	KPSQL	355
9.4.12	MODEM Control	355
9.4.13	Examples	355
9.5	Operator Commands	356
9.5.1	Callsign:	356
9.5.2	Name:	356
9.5.3	QTH:	356
9.5.4	Locator:	356
9.5.5	Operators Antenna Description	357
9.5.6	Examples	357
9.6	Audio Device Commands	357
9.6.1	OSS	357
9.6.2	Device:	357
9.6.3	PortAudio	357
9.6.4	Playback:	358

9.6.5	Capture:	358
9.6.6	PulseAudio	358
9.6.7	Server string:	358
9.6.8	Examples	358
9.7	Audio Settings	359
9.7.1	Capture Audio Sample Rate	359
9.7.2	Playback Audio Sample Rate	359
9.7.3	Converter	359
9.7.4	Receive Audio PPM Correction	359
9.7.5	Transmit Audio PPM Correction	359
9.7.6	Transmit Audio Hertz Offset Correction	359
9.7.7	Examples	360
9.8	Audio Right Channel Commands	360
9.8.1	Mono audio output	360
9.8.2	Modem signal on left and right channels	360
9.8.3	Reverse Left/Right channels	360
9.8.4	PTT tone on right audio channel	360
9.8.5	CW QSK signal on right channel	361
9.8.6	Pseudo-FSK on right audio channel	361
9.8.7	Examples	361
9.9	Wav write sample rate	361
9.10	Hardware PTT Option Commands	361
9.10.1	PTT tone on right audio channel	362
9.10.2	Use separate serial port PTT	362
9.10.3	Device:	362
9.10.4	Use RTS	362
9.10.5	Use DTR	362
9.10.6	RTS = +V	362
9.10.7	DTR = +V	362
9.10.8	PTT delays valid for all CAT/PTT types	362
9.10.9	PTT delays valid for all CAT/PTT types	362
9.10.10	Use uHRouter PTT	362
9.10.11	Use parallel port PTT	362
9.10.12	Initialize Changes	362
9.10.13	Examples	363
9.11	RIGCAT Commands	363
9.11.1	Use RigCAT	363
9.11.2	Rig description file:	363
9.11.3	Device:	364
9.11.4	Retries	364

9.11.5	Retry interval (ms)	364
9.11.6	Write delay (ms)	364
9.11.7	Init delay (ms)	364
9.11.8	Baud rate:	364
9.11.9	Stopbits	364
9.11.10	Commands are echoed	364
9.11.11	Toggle RTS for PTT	364
9.11.12	Toggle DTR for PTT	364
9.11.13	Restore Settings on Close	364
9.11.14	CAT command for PTT	364
9.11.15	RTS +12 v	365
9.11.16	DTR +12 v	365
9.11.17	RTS/CTS flow control	365
9.11.18	VSP Enable	365
9.11.19	Initialize Changes	365
9.11.20	Examples	365
9.12	HAMLIB Commands	366
9.12.1	Use Hamlib	366
9.12.2	Rig:	366
9.12.3	Device:	367
9.12.4	Retries	367
9.12.5	Retry Interval (ms)	367
9.12.6	Write delay (ms)	367
9.12.7	Post write delay (ms)	367
9.12.8	Baud rate:	367
9.12.9	Stopbits	367
9.12.10	Sideband:	367
9.12.11	PTT via Hamlib command	367
9.12.12	DTR +12	367
9.12.13	RTS +12	367
9.12.14	RTS/CTS flow control	368
9.12.15	XON/XOFF flow control	368
9.12.16	Advanced configuration:	368
9.12.17	Initialize Changes	368
9.12.18	Examples	368
9.13	XMLRPC Rig Control Commands	369
9.13.1	Use XML-RPC program	369
9.13.2	Mode/BW delay	369
9.13.3	Initialize Changes	369
9.13.4	Examples	369

9.14 IO	370
9.14.1 Lock	370
9.14.2 Enable ARQ or KISS IO Access	370
9.14.3 AX25 Decode	370
9.14.4 Enable CSMA	370
9.14.5 KISS Sub Command	370
9.14.5.1 IP Address	370
9.14.5.2 I/O	371
9.14.5.3 O	371
9.14.5.4 Dual Port	371
9.14.5.5 Enable Busy Channel	371
9.14.5.6 Continue After (sec)	371
9.14.5.7 KPSQL Attenuation	371
9.14.6 ARQ Sub Command	371
9.14.6.1 IP Address	371
9.14.6.2 Port	371
9.14.7 XMLRPC Sub Command	372
9.14.7.1 IP Address	372
9.14.7.2 Port	372
9.14.8 Examples	372
9.15 Misc NBEMS Commands	373
9.15.1 Enable	373
9.15.2 Open message folder	374
9.15.3 Open with flmsg	374
9.15.4 FLMsg executable path	374
9.15.5 Open in browser	374
9.15.6 Timeout (secs)	374
9.15.7 Examples	374
9.16 ID	374
9.16.1 RsID	374
9.16.1.1 Notify only	375
9.16.1.2 Searches passband	375
9.16.1.3 Mark prev freq/mode	375
9.16.1.4 Disables detector	375
9.16.1.5 Disable alert dialog	375
9.16.1.6 Retain tx freq lock	375
9.16.1.7 Disable freq change	375
9.16.1.8 Allow errors	375
9.16.1.9 Squelch open (sec)	375
9.16.1.10 Pre-Signal Tone Seconds	376



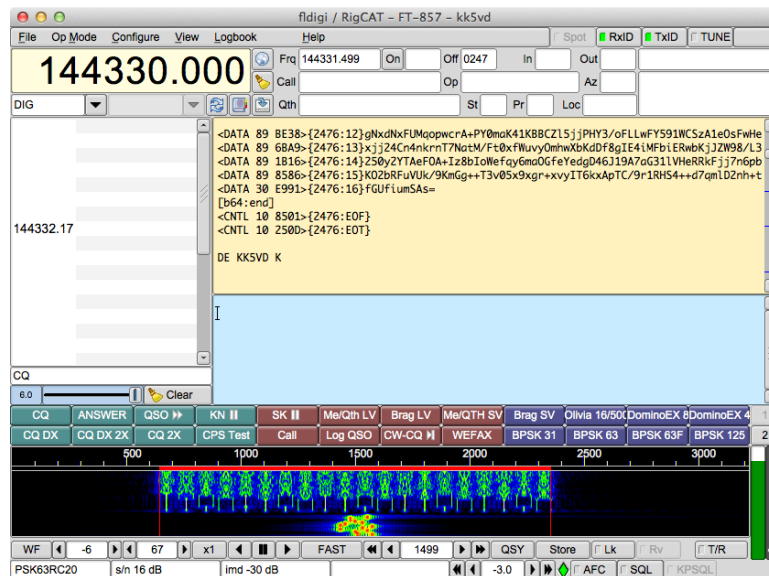
---

9.16.1.11 End of xmt ID . . . . .	376
9.16.2 Video . . . . .	376
9.16.2.1 Transmit mode ID . . . . .	376
9.16.2.2 Transmit video text . . . . .	376
9.16.2.3 Custom Video Text Input . . . . .	376
9.16.2.4 Use small font . . . . .	376
9.16.2.5 500 Hz limit . . . . .	376
9.16.2.6 Mode width limit . . . . .	376
9.16.2.7 Chars/Row: . . . . .	376
9.16.3 CW . . . . .	377
9.16.3.1 Transmit callsign . . . . .	377
9.16.3.2 Speed (WPM): . . . . .	377
9.16.4 Examples . . . . .	377
9.17 Load Macro Command . . . . .	379
9.18 Example Script . . . . .	380
<b>10 Command Line Switches</b>	<b>383</b>
10.1 Linux Terminal . . . . .	383
10.2 Windows Icon Properties . . . . .	385
10.3 Mint Launcher Properties . . . . .	386
<b>11 License(s)</b>	<b>389</b>
11.1 Copyright . . . . .	389
<b>12 Recognitions</b>	<b>391</b>
<b>13 Using BPQ32 with FLDIGI</b>	<b>395</b>
13.1 General Information, BPQ32 . . . . .	395



# Chapter 1

## FLDIGI Users Manual - Version 3.23



### 1.1 Fldigi Configuration and Operational Instructions

- [Configuration](#)
- [Modems](#)
- [Operating](#)
- [Logging](#)
- [Macros](#)
- [Configure Scripting](#)
- [Developers](#)
- [License\(s\)](#)
- [Recognitions](#)



## Chapter 2

# Configuration

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.

### 2.1 User Interface configuration

You should initially configure the following:

- [Configure Operator](#)
- [Sound Card Configuration](#)
- [Rig Control](#)
- [RX/TX Audio Adjustment](#)
- [New Installation](#)
- [Configure ARQ/KISS I/O](#)

### 2.2 Windows Specific Install / Config

- [Installing Fldigi on Windows](#)
- [New Install Wizard](#)

### 2.3 Other Configuration options

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

- [Callsign DB Configuration](#)

- [Colors and Fonts](#)
- [Configure FFT Scan](#)
- [RSID / Video / ID Configuration](#)
- [PSKmail Configuration](#)
- [Configure Touch Screen](#)
- [User Interface Configuration - Browser](#)
- [User Interface Configuration - Contest](#)
- [User Interface Configuration - General](#)
- [User Interface Configuration - Logging](#)
- [User Interface Configuration - Macros](#)
- [User Interface Configuration - WF Controls](#)
- [Waterfall Configuration](#)
- [WEFAX](#)
- [Working Logs](#)

## 2.4 Command Line Switches

Additional configurational items are available from the command line.

See [Command Line Switches](#) for details.

## 2.5 Modem Configuration Options

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

- [Contestia Configuration](#)
- [CW Configuration](#)
- [DominoEX Configuration](#)
- [Feld Hell Configuration](#)
- [FSQ Configuration](#)
- [IFKP Configuration](#)
- [RSID / Video / ID Configuration](#)
- [Miscellaneous Configuration](#)
- [MT63 Configuration](#)
- [Olivia Configuration](#)
- [PSK Configuration](#)
- [RTTY / FSK Configuration](#)

- [Thor Configuration](#)

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`.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.6 Configure Operator

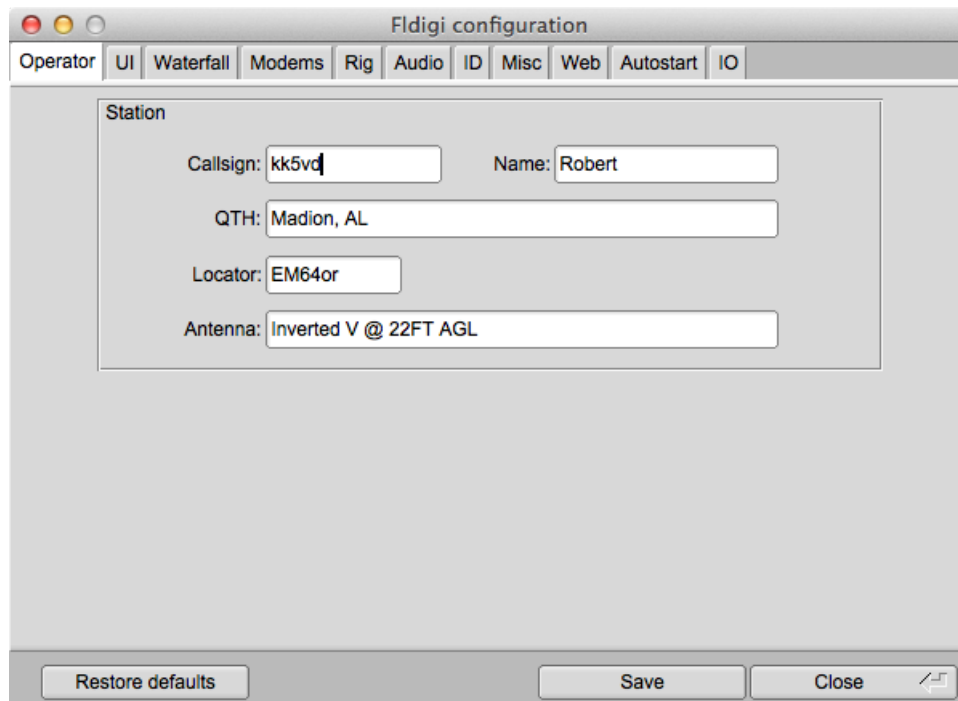


Figure 2.1: Operator

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

### NOTE:

Thor and DominoEX secondary text field are initially set to the operator callsign. On subsequent changes to the callsign, these fields are not updated. See [Thor Configuration](#) and [DominoEX Configuration](#) for the location of the secondary text entry fields.

The antenna information is required if you elect to report to the spotting web site, [PSK reporter](#).

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.

[Return to Top of Page](#)  
[Return to Main Page](#)

## 2.7 Sound Card Configuration

A few words about sound I/O on the PC. "You are in a maze of twisty little passages, all alike".

PortAudio, PulseAudio and OSS are different ways in which fldigi can access your sound card through the various sound systems.

OSS was the first audio backend in fldigi. It works with the Linux sound system of the same name, which has now been replaced by ALSA but is still supported via an emulation layer. Its only advantage, as an audio backend, is that it's simple and doesn't require any external libraries.

The PortAudio backend was written subsequently to support OSS on Linux and FreeBSD, ALSA and JACK on Linux, CoreAudio on OS X, and also the various sound APIs on Windows – all through the same PortAudio library.

PulseAudio is more than an audio hardware access layer; refer to its website for a summary of what it does. Fldigi supports it mainly because many Linux distributions are now integrating it with their desktops, but also because it has a few interesting features:

- it can take care of the resampling and volume control for us,
- it can stream audio over the network, and
- it makes it easier to run multiple fldigi instances (all accessing the same sound card).
- it provides mixer controls for input and output audio streams
- it remembers which hardware is used for each application it serves, and it remembers the mixer levels associated with that application

In the future it might be possible to replace all of these with a single backend, without any loss of functionality, performance, sound system or platform support. That'll be the day! Until then:

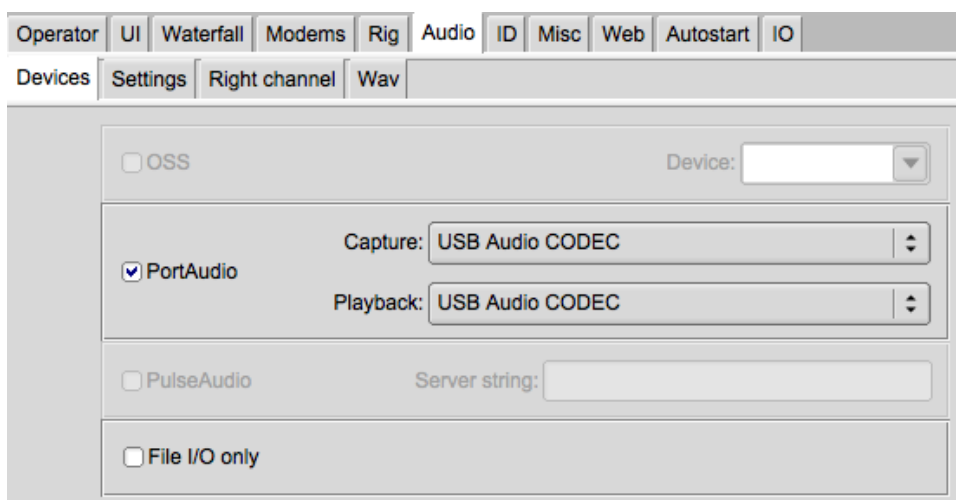


Figure 2.2: Audio Devices



**On Linux:**

- Use PulseAudio if your Linux distro ships it, and you already have the pulseaudio daemon running (this is the case with Fedora 8/9 and Ubuntu 8.04, probably also with openSUSE 11.0). Or if you want networked audio, etc. etc.
- Otherwise, use PortAudio and select a device from the list(s). PortAudio is also the best way to access JACK, through which you can use other programs as audio sources/sinks – particularly useful with SDR software. As with PulseAudio, you can select different capture and playback audio devices.
- The OSS backend should be used only as a last resort. Note that it has not been updated to support user-configurable sample rates.

**On Windows:**

Use the PortAudio and select the device from the list(s).

Select the *SndCrd* tab on the configuration dialog.

On Linux Fldigi can interface to the sound card using either the OSS, the Portaudio, or the PulseAudio. Each of the appropriate libraries must be present on the computer to use that particular sound i/o.

On Windows Fldigi uses the Portaudio sound driver only.

It is also possible to configure Fldigi with File I/O only, which is useful for testing the application without an interface to the sound card. In the File I/O only configuration you can record and playback audio files in a number of different formats including the "wav" format associated with the Windows operating system.

The program will find all active sound cards and the associated drivers for both. Select the sound card and driver type that will be used with the program. I recommend using the Pulseaudio device driver if that is available on your Linux distribution.

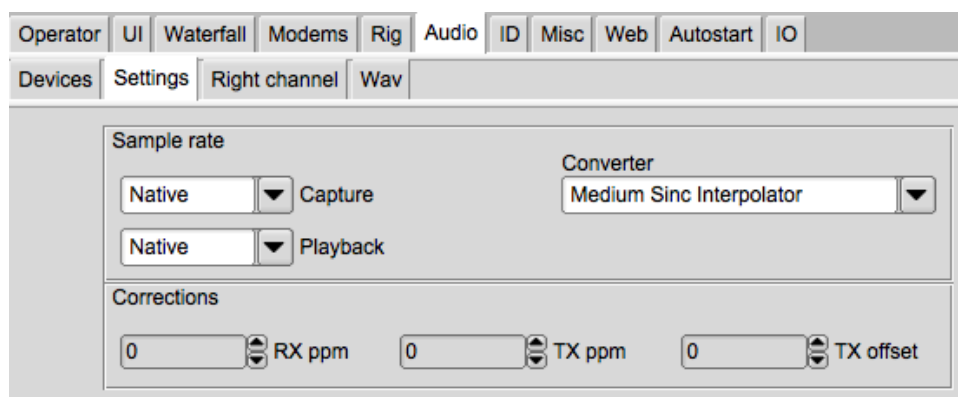


Figure 2.3: Config Audio Settings

If Pulse or Port audio is selected then you can either allow the program to use the auto detect to determine the best sound card sampling rate, or you can pick from the drop down list. If you know your RX and TX sound card oscillator correction factors you can enter them now. If not you can determine the RX rate correction using a special WWV modem built into *Fldigi*. The decoder and encoder logic for each of the various modems require a specific

sound card sample rate which may not be the the actual sound card sample rate. The conversion between the modem sample rate and the sound card sample rate is accomplished by one of a set of sample rate converters.

Sound card oscillators may have a slight error in frequency that causes their sampling rate to not be the value specified. This error is usually small enough to be measured in a parts per million. Fldigi uses a technique called rate conversion to correct the sampled waveform for this error. The error can be measured and the correction factor determined by using the [WWV calibration](#) modem. The supporting library used for the converter provides several different levels of conversion, Best, Medium, Fastest and Linear Interpolator. The default, Medium interpolator, will be satisfactory for most sound cards. If you are running fldigi on a computer with limited cpu power you might find it necessary to select one of the more cpu efficient converters, either Fastest or Linear. Each gives progressively poorer performance but use fewer cpu cycles to perform the frequency conversion. You should also be sure that the cpu type is set to [Slow cpu](#) on the miscellaneous configuration tab.

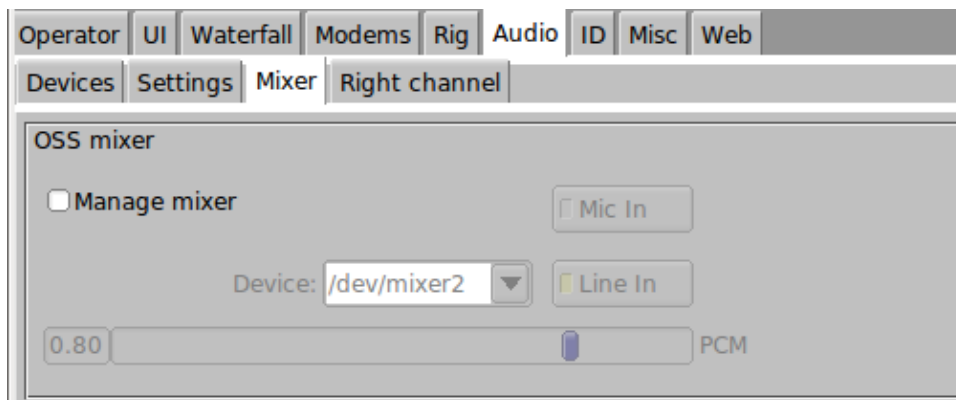


Figure 2.4: Audio Mixer

Mixer controls are only active on Linux using OSS, ALSA backends and if the distribution provides application layer mixer support. Select whether you will be using Line-In or Mic-In for the audio connection from the receiver output. Fldigi ALWAYS expects to use the Line-Out for driving the transmitter audio. Set the PCM level for your sound card. If you check "Manage mixer" then the Tx and Rx "volume" controls on the main fldigi dialog will be active.

## 2.7.1 Right Channel Audio Output

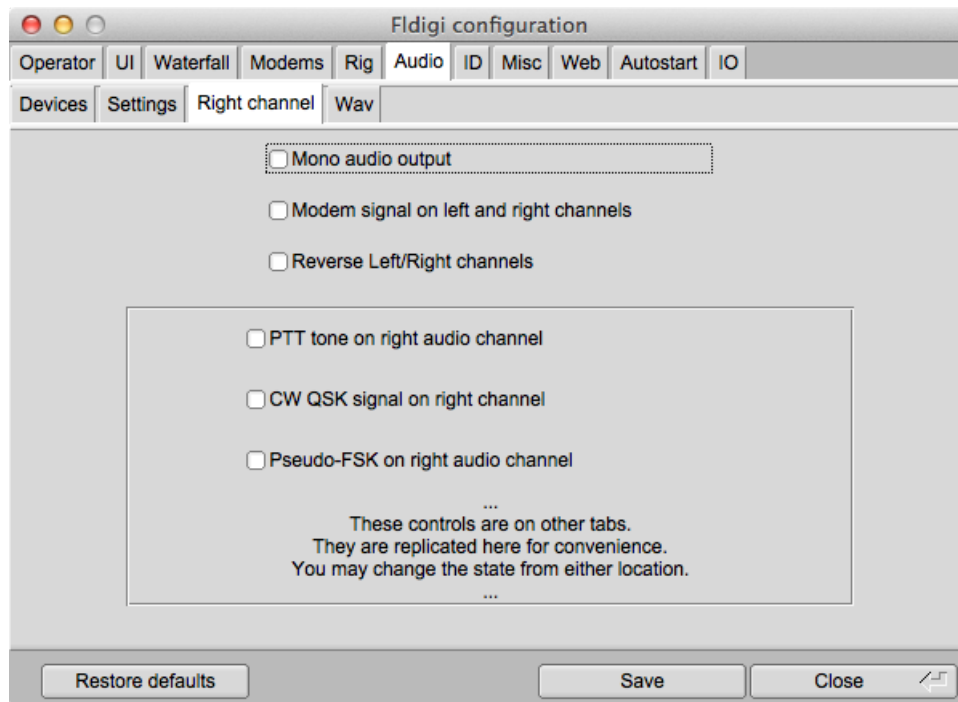


Figure 2.5: Right Audio Channel

You may elect to make the right and left channels both contain the modem signal. Or, you may reverse the right and left channel audio signals. These two controls only change the function of the audio output.

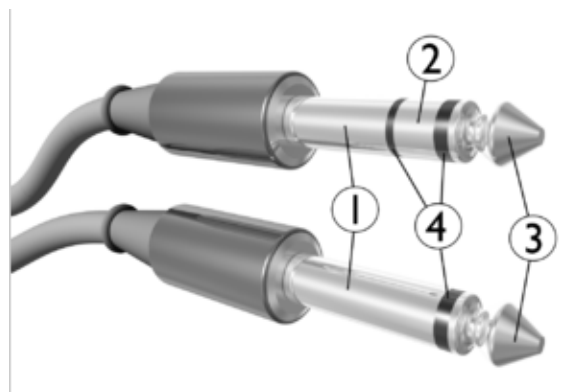


Figure 2.6: Stereo Plug

1. Sleeve: usually ground
2. Ring: Right-hand channel for stereo signals, negative polarity for balanced mono signals. Fldigi uses this channel for special controls signals.

3. Tip: Left-hand channel for stereo signals, positive polarity for balanced mono signals. Fldigi uses this channel for Rx and Tx audio.

The PTT, CW QSK and Pseudo-FSK items are found on their respective configuration tabs. They are replicated here for your convenience and these controls may be changed on this or the other tabs.

## 2.7.2 WAV File Sample Rate

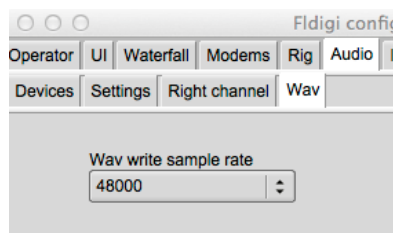


Figure 2.7: Sample Rate of Saved WAV Files

User selectable WAV file sample rate.

## 2.7.3 Multiple sound cards

In systems with multiple sound cards they will not always be in the correct sort order on boot. This may cause problems not only with fldigi, but other apps that depend on a certain sound card. The work around is not that difficult and will reliably place the preferred sound card in the correct slot. Sound cards are numbered from 0 on to however many cards you have in your computer. Usually its only 2.

The 1st step is to determine the correct id of the cards in your system. Open a terminal and issue the following ::  
 aplay -l

Here is an example of what you may see.

```
**** List of PLAYBACK Hardware Devices ****
card 0: ICH6 [Intel ICH6], device 0: Intel ICH [Intel ICH6]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: ICH6 [Intel ICH6], device 4: Intel ICH - IEC958 [Intel ICH6 -
IEC958]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: AudioPCI [Ensoniq AudioPCI], device 0: ES1371/1 [ES1371
DAC2/ADC]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: AudioPCI [Ensoniq AudioPCI], device 1: ES1371/2 [ES1371 DAC1]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

If it does not already exist you will need to create the following file:

```
/etc/modprobe.d/alsa-options
```

You will have to root privileges to create the file, so a system such as Ubuntu that depends on sudo you would enter

```
$ sudo touch /etc/modprobe.d/alsa-options
```

Using the above example, you edit that file so that it's contents contain

```
# Set preferred order of the sound cards
```

```
options snd-ICH6 index=0
options snd-AudioPCI index=1
```

The simplest terminal editor that is on just about every distribution is "nano" so to edit the file you just created you can again use sudo

```
$ sudo nano /etc/modprobe.d/alsa-options
```

If you want to test this new file, reboot and open a terminal and again issue the command :: aplay -l

You will normally find that your internet browser and/or Flash want to use card 0, so you then may want to consider using card 1 for fldigi

Here is another example with 3 sound systems; mother board ALC1200, a thumb-drive audio codec, and a Signa↔ Link USB:

```
**** List of PLAYBACK Hardware Devices ****
card 0: NVidia [HDA NVidia], device 0: ALC1200 Analog [ALC1200 Analog]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: Set [USB Headphone Set], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: default [USB Audio CODEC ], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

The /etc/modprobe.d/alsa-options contains these lines

```
options snd-NVidia index=0
options snd-Set index=1
options snd-CODEC index=2
```

Notice that each line is uniquely related to the aplay -l report

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.8 Rig Control

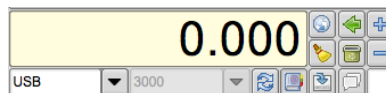


Figure 2.8: Initial Rig Setup

CAT not enabled

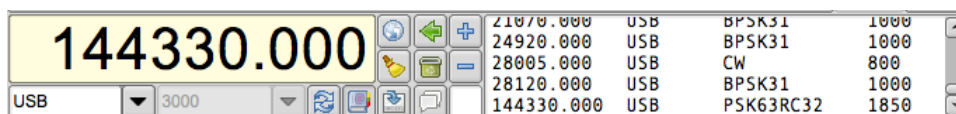


Figure 2.9: Docked Rig Control

CAT not enabled, Manual Entry of transceiver frequency

**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 [psk-reporter/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 increment by clicking/pressing on the upper half of the digit and decrement by clicking/pressing the lower half of the digit. 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 hovering the mouse pointer over the frequency display and entering the desired frequency (in Khz) via the keyboard. Pressing <return> or <enter> on completion sets the frequency to an active state, effecting rig control and future log entries. Don't forget the decimal point if you are entering a fractional KHz value.

The Smeter / Power-meter display is only active if fldigi is configured for xmlrpc control with flrig. The transceiver must support the smeter and power out CAT query. fldigi does not try to estimate the smeter reading based on the audio stream. You can toggle viewing the Smeter / Power-meter display using the button just to the right of the shared space used by the "Smeter/Power-meter" and the "Mode/Bandwidth" controls. Be sure to save the configuration for this to persist the next time fldigi is executed.

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.

### 2.8.1 Rig Configuration

#### Hardware PTT control

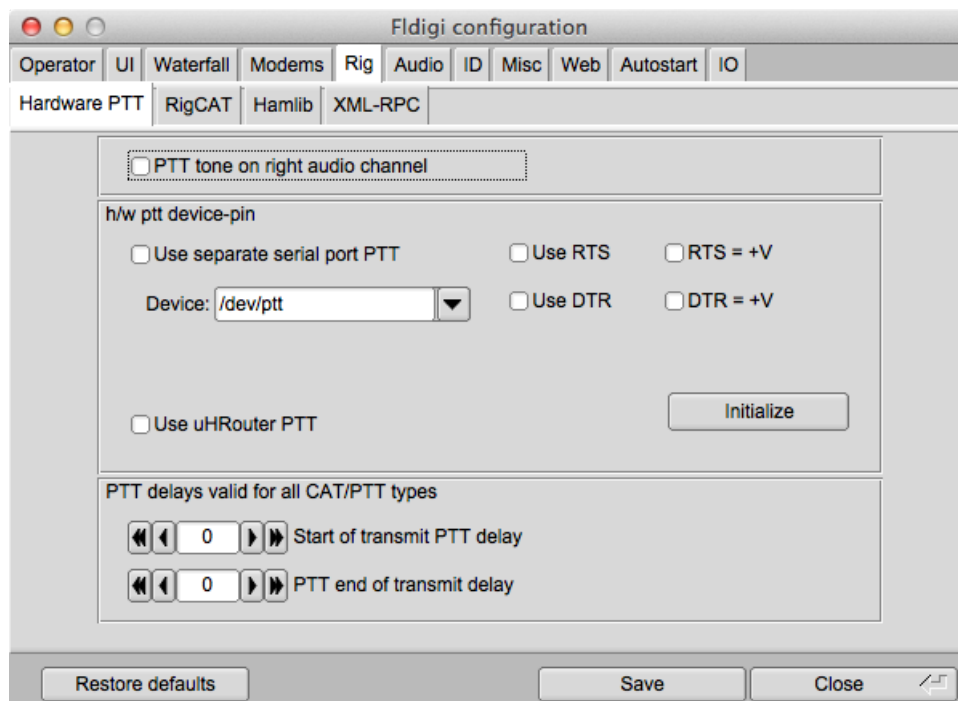


Figure 2.10: Hardware PTT control

### 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. Jim, W5ZIT, has provided details on building an [interface](#) for this type of PTT control.

### 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 your 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)

Fldigi 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.

### μH Router (MacOS X)

Similar functionality can be achieved on the Macintosh operating system using 'μH Router' by Kok Chen, W7AY. See [μH Router Website](#) for specific details and requirements. A selectable (check box) option will be available on the Rig->Hardware PTT Configuration panel.

### PTT delays

You can accommodate delays in transceiver switching between receive and transmit by adjusting the PTT delays. The control values are in milliseconds. These controls have no effect on external PTT circuits such as those implemented in the Signalink interfaces. They rely on detecting the audio data stream. You can use a combination of macro tags in a macro key definition to achieve a resolution. For example try a macro definition similar to this to insure that the RSID is sent via a slow FM xcvr (or via a VHF repeater)

```
<TX><MODEM:NULL><IDLE:2.5>  
<!MODEM:MT63-500>  
<TXRSID:on>
```

Change the idle time value (in fractional seconds) to suit your needs.



## 2.8.2 RigCAT control

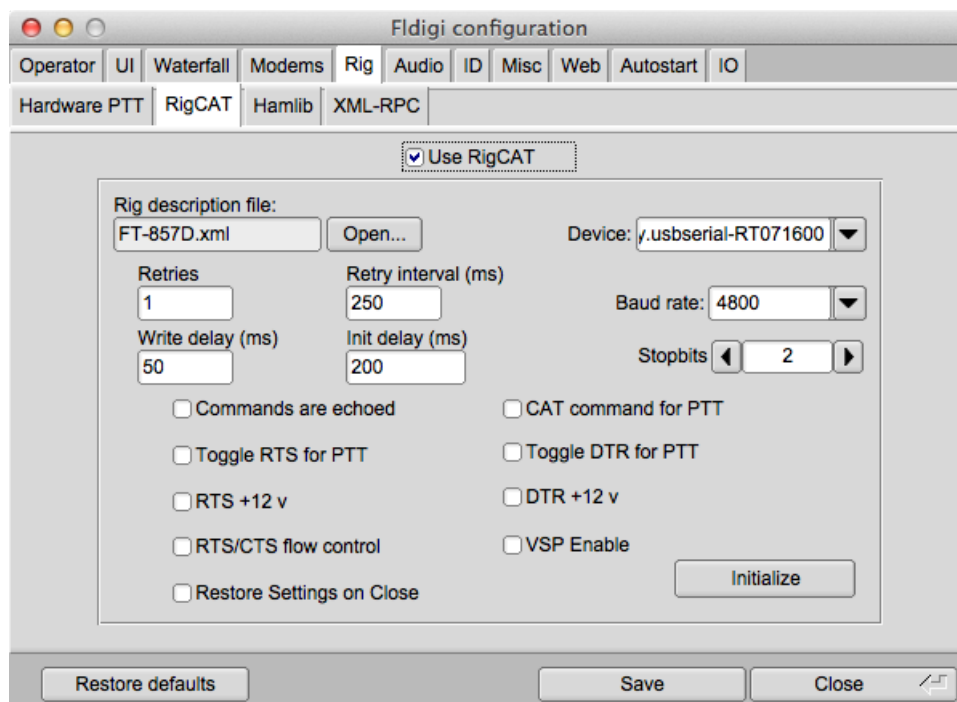


Figure 2.11: RigCAT control

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 specific xml 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 [web page](#). 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.

### 2.8.3 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.

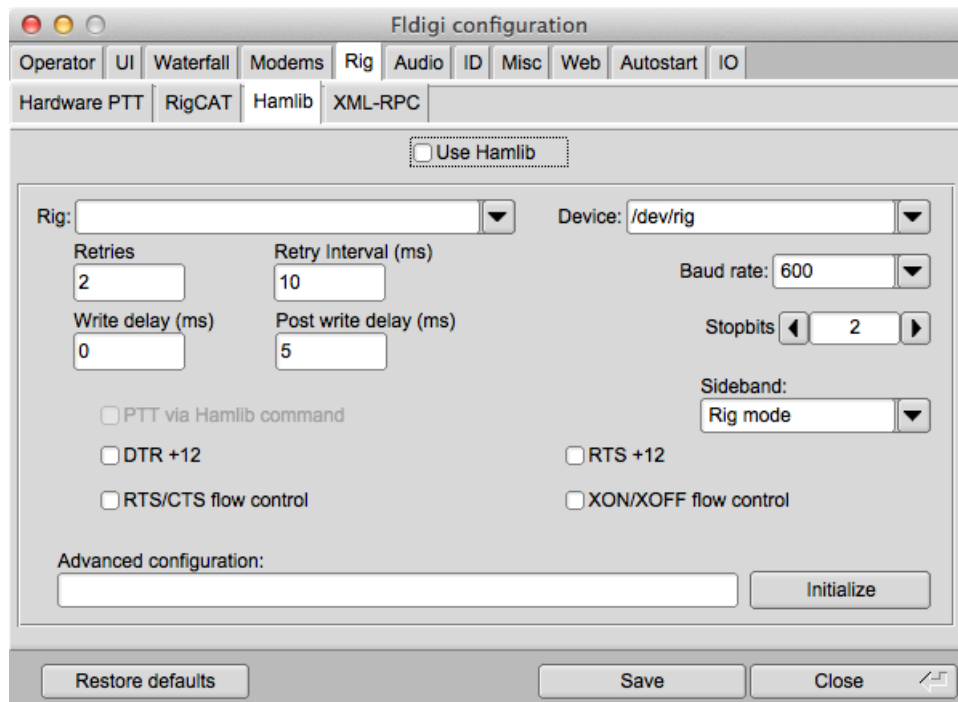


Figure 2.12: Hamlib CAT Control

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 annunciating them on the rig control frequency display.

### 2.8.4 Xml-Rpc CAT

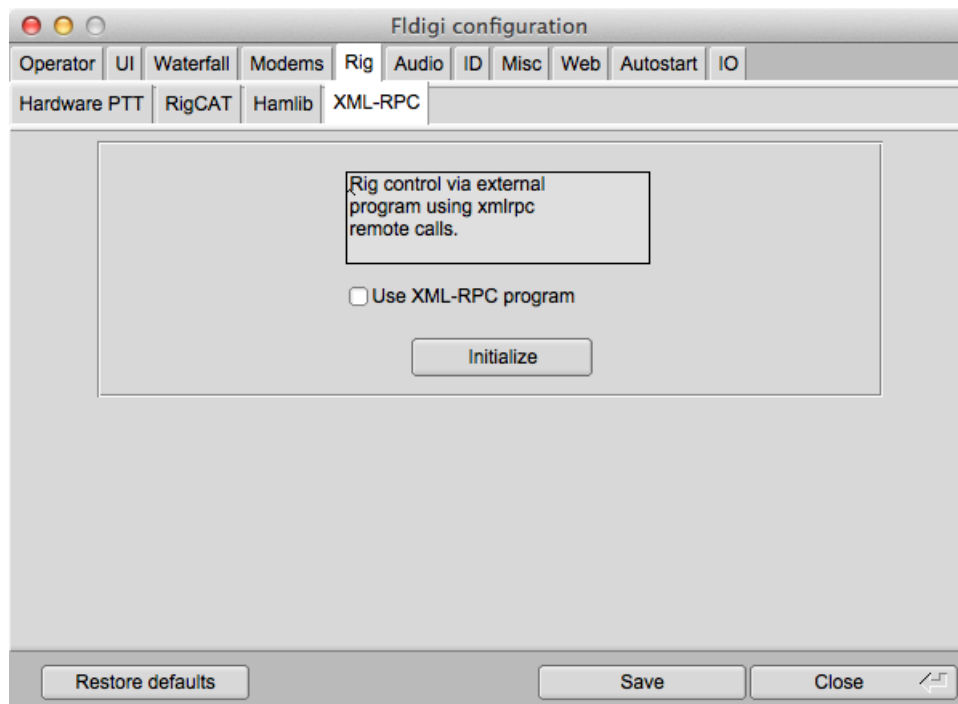


Figure 2.13: Xml-Rpc CAT Control

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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.9 RX/TX Audio Adjustment

### 2.9.1 Receive audio

Setting the correct hardware, operating system, and fldigi received audio levels is not difficult, but it is the one setup procedure most often done incorrectly. The most commonly used sound card devices contain either a 16 or 24 bit analog to digital (a/d) converter. A 16 bit a/d can provide approximately 90 db of signal conversion. For the 16 bit converter, if the peak audio signal that the a/d can handle is +/- 1 volt then the minimum discernable signal (1 bit) will be +/- 30.5 microvolts. If more than a +/- 1 volt signal is applied to the a/d input then either one of two things may occur, (1) the audio is clipped, or (2) the audio is wrapped, large positive signals wrap to large negative signals and vice versa.

The objective in adjusting the Rx audio is to use the full dynamic range of the a/d without incurring overdrive. fldigi provides a number of display controls to assist in setting up the Rx audio. There are two controls that ONLY adjust the visual appearance of the waterfall and DO NOT effect the a/d or the signal decoders. These controls are below and to the left of the waterfall, "Upper signal level (dB)", and "Signal range (dB)".



Figure 2.14: Waterfall level/range controls

Change the waterfall display to the "Scope" view. Do this by pressing the "WF" button twice if it is currently displaying the waterfall. You can also right click once on that button. That button acts as a rotary and is left/right click sensitive. The display should show what looks like an oscilloscope view of the received audio. This is the entire audio signal and not just the signal that is currently decoded.

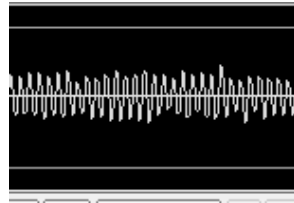


Figure 2.15: Scope view

If your sound card a/d (also referred to as codec) is functioning correctly there should be no signal offset and the signal will be centered vertically about the y-axis. Set your transceiver to a portion of the band with digital signals, 14.070 MHz is a good spot. Now adjust the transceiver audio output and the operating system mixer controls so that the peak signals do not exceed the upper/lower gray lines on the display. The diamond indicator to the left of the fldigi AFC button will be colored as follows:

- BLACK - no signal, or insufficient Rx audio
- GREEN - signals are in the correct range
- YELLOW - signals are exceed 75% of maximum, but are less than 90% of maximum
- RED - signals exceed 90% of maximum a/d capability - WARNING WILL ROBINSON!

You can then proceed to adjust the waterfall/fft appearance when you have the audio Rx level adjusted correctly.

The three signal views should be similar to this with the level/range controls set to 0 and 60 respectively. The waterfall palette used is "default.pal", and the received signal is about 20 dB above the noise floor.

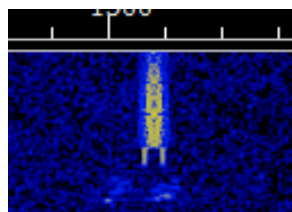


Figure 2.16: Waterfal

## 2.9.2 Windows Audio Properties

A combination of the Windows operating system and certain audio codecs needs to be accommodated. The codecs used by devices like the SignalLink-USB, the TS590S, and recent Icom transceivers are identified by the Windows

operating system as "microphone" devices. The Windows mixer controls for that device are then set to insert about 30 dB of gain in the signal path. To compensate most users simply lower the transceiver audio output. The result is very poor decoding. The correct procedure is to force the Windows mixer controls to remove that 30 dB of gain. This procedure is for Windows 10, but is similar on Windows 7, and Windows 8.

Right click on the task bar speaker icon and select the "Recording devices" menu item. This will open the Sound devices dialog:

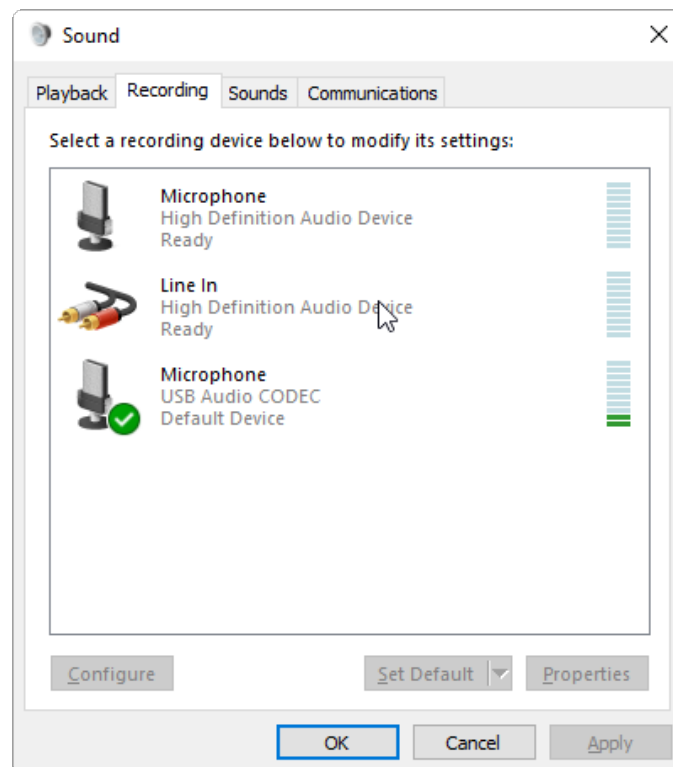


Figure 2.19: Sound Devices

This is the dialog on my Windows 10 test computer with a Signalink-USB connected to a USB hub. The Signalink-USB shows up as a **Microphone** USB Audio CODEC. Right click on the device entry and select Properties. This opens up the Microphone Properties dialog:

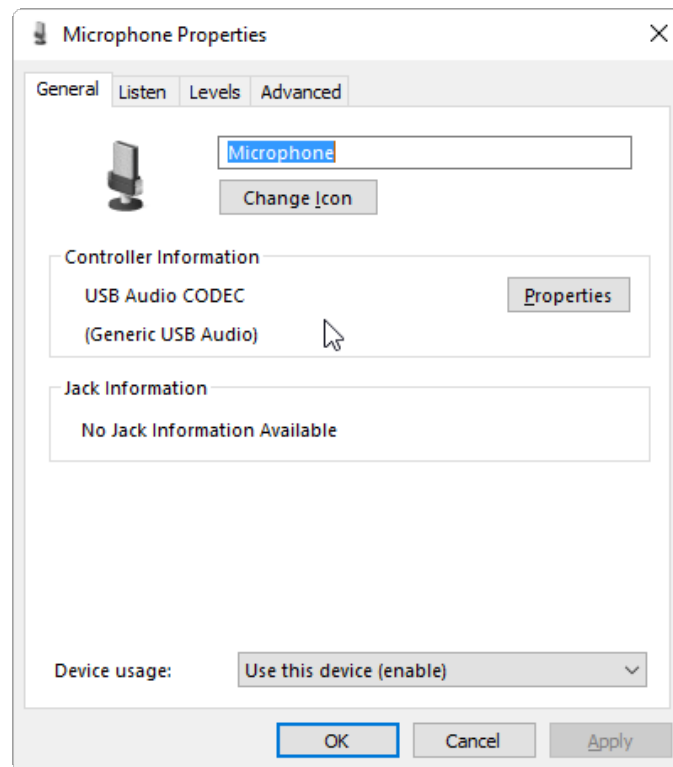


Figure 2.20: Microphone Properties

Select the "Levels" tab and set the level to as close to 0 dB as is available:

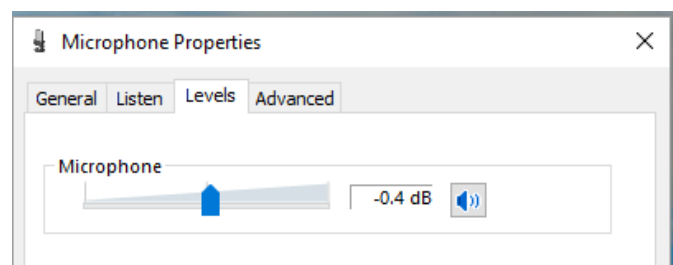


Figure 2.21: Microphone Level

Then select the "Advanced" tab and change the type of converter.

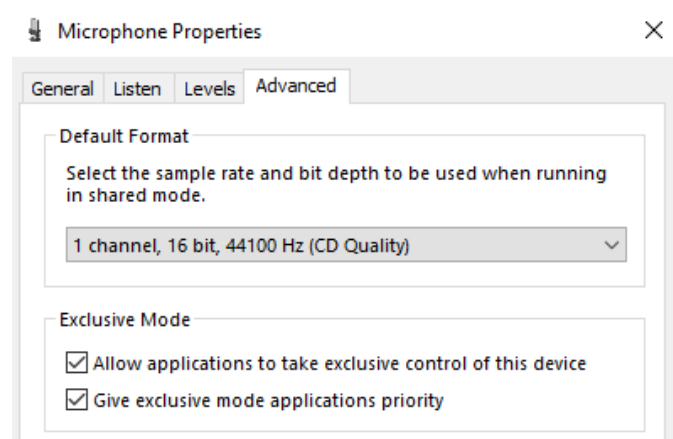


Figure 2.22: Microphone Advanced Tab

Apply and save the changes. The last step is usually needed to insure that the settings are not lost when the operating system is closed and restarted.

### 2.9.3 Transmit audio

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 Level control for minimum output level.
- Set the fldigi transmit attenuator to -3 dB (control to the left of the AFC button)
- Set the waterfall cursor to 1000 Hz
- 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 the fldigi transmit attenuator to make small adjustments in the output power to compensate for variations in the SSB transmit filter.

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 your 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. The fldigi transmit attenuator control should have sufficient range to compensate for transmit SSB filter variation. You do not want to transmit (or receive) near the cutoff frequencies of the SSB filter. The transceiver filter will introduce both phase and amplitude distortion over the signal bandwidth.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.10 New Installation

fldigi's opening screen looks like the following when starting fldigi for the first time or when setting up a second or subsequent instance using the `-config-dir` command line switch. The Wizard has been completed and the callsign, W1HKJ, entered.

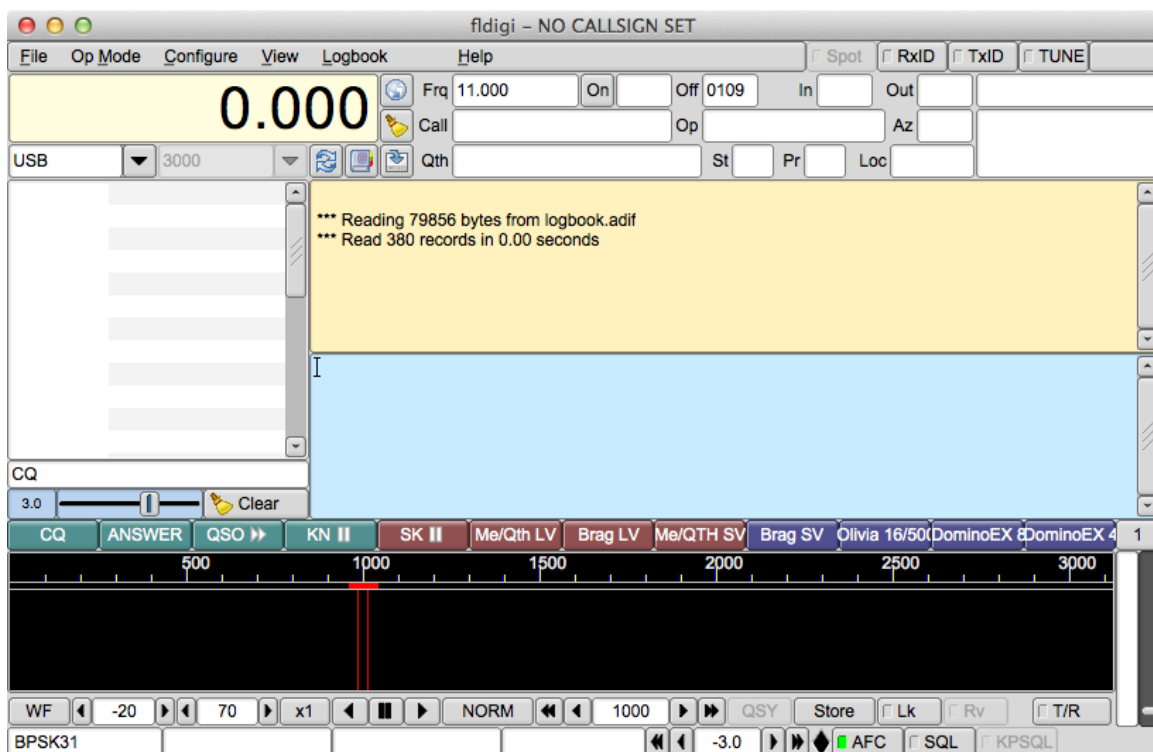


Figure 2.23: Main Dialog New Install



Fldigi will create a working files folder, multiple sub folders and also populate them with a set of default files. The working files folder is different on the different OS.

Operating System	Folder/Directory
Windows	C:\Documents and Settings\ <username>\fldigi.files</username>
Vista	C:\Users\ <username>\fldigi.files</username>
Linux	/home/<username>/fldigi
Macintosh	/Users/User_Login_Name/fldigi

After closing the application the working folder will contain the following folders and files:

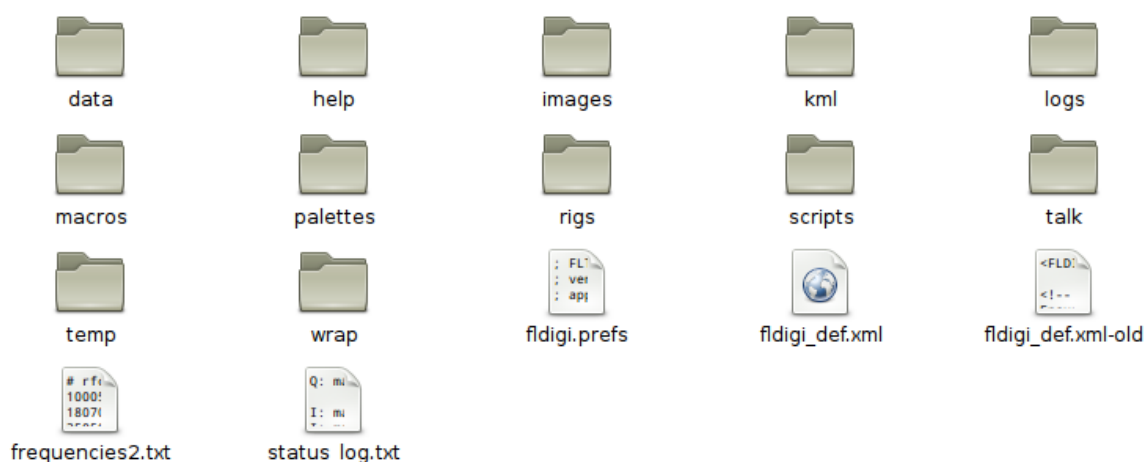


Figure 2.24: Working Folder

The help, images, logs, scripts and temp folders will be empty. They will contain program created files as you use the program or you may post files in those folders for use by fldigi. Images to be sent with the MFSK pic mode should be placed in images. Your logbook database will appear in logs. If you are running on Linux then you can use various scripts to enhance the macro language that fldigi supports. The temp directory holds files that are transitory and you can safely delete those files between sessions. The 5 files that appear initially are:

fldigi.prefs	contains variables that describe the status of fldigi when last used. This is an ASCII text file that you can safely read. You should not edit or change this file.
fldigiYYYYMMDD.log	this is an historical log of all the received and transmitted text during the day for which the log refers
fldigi_def.xml	contains variables that relate to all of fldigi's configurable items. This is an ASCII text file that conforms with the XML specification. You can safely read this file but should not edit or change it.

frequencies2.txt	an ASCII text file that contains the default (and / or modified) entries for fldigi's rig control process
<a href="#">status_log.txt</a>	a log of events for the most current fldigi execution. This file will contain information relative to any errors that may occur and is important for debugging purposes.

The macros folder contains a single file: macros.mdf. This is an ASCII text file that contains the default macro definitions. After running fldigi for a while and creating your own sets of macro definitions there will be additional \*.mdf files located here.

The palettes folder contains the following files:

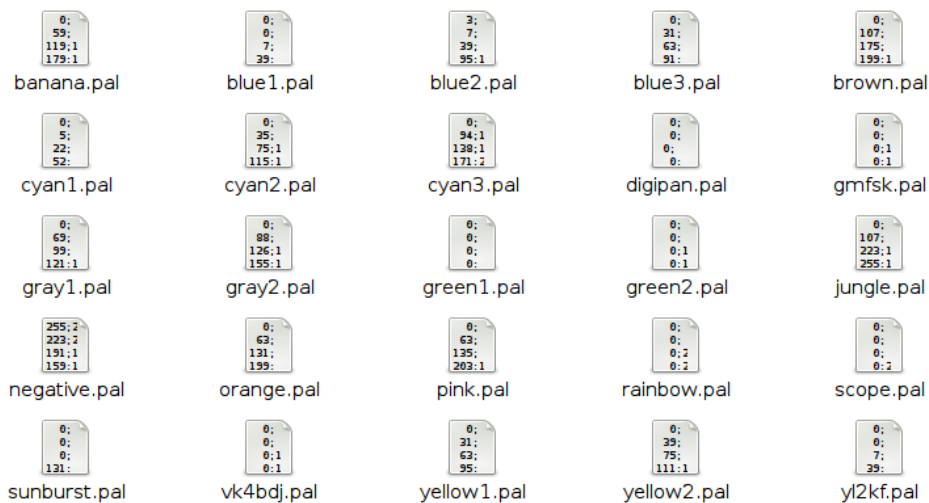


Figure 2.25: Folder Palettes

Each of these is a palette definition file that is used to modify the appearance of the waterfall. Fldigi has a palette editor that enables you to modify these default files or to create your own. The file format of these files is identical to the palette files used by DigiPan. The final color rendition might be a little different as a result of using different painting functions. The file digipan.pal contains:

```
0; 0; 0
0; 0; 62
0; 0;126
0; 0;214
145;142; 96
181;184; 48
223;226;105
254;254; 4
```

Don't bother trying to modify these using an editor. The palette editor is much easier to use and will keep you from

wrecking havoc with the program.

The easiest way to find the working files folder is to start fldigi and then select the menu item File/Show config.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.11 Configure ARQ/KISS I/O

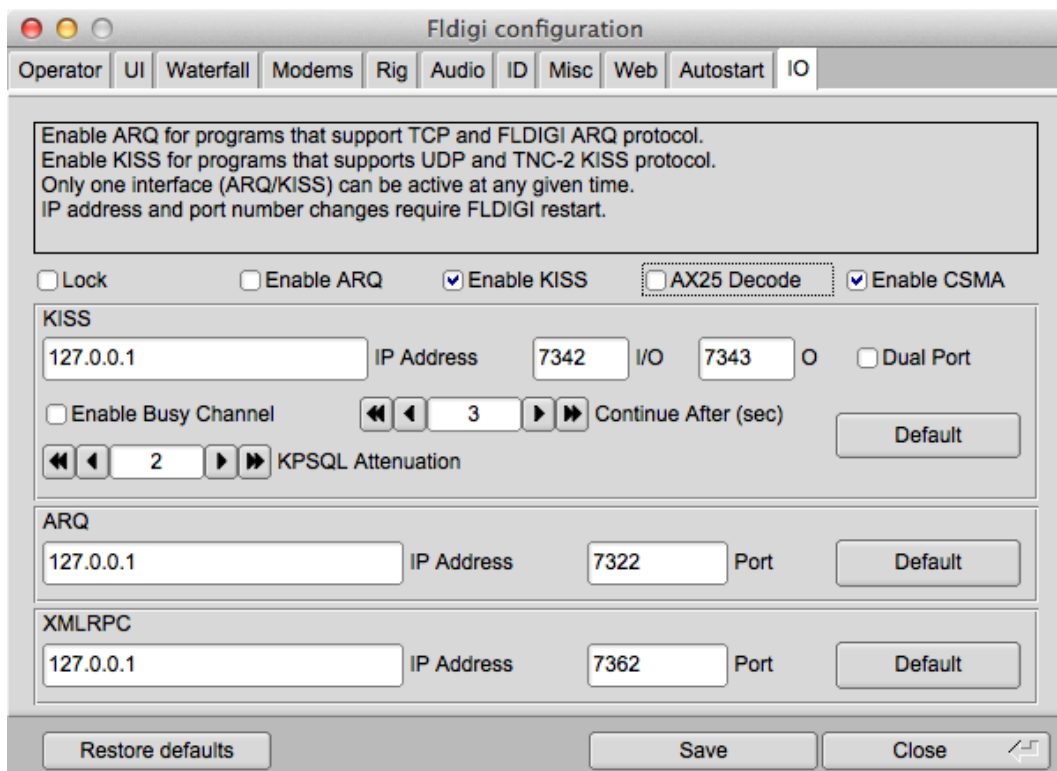


Figure 2.26: ARQ/KISS IO Configuration Panel

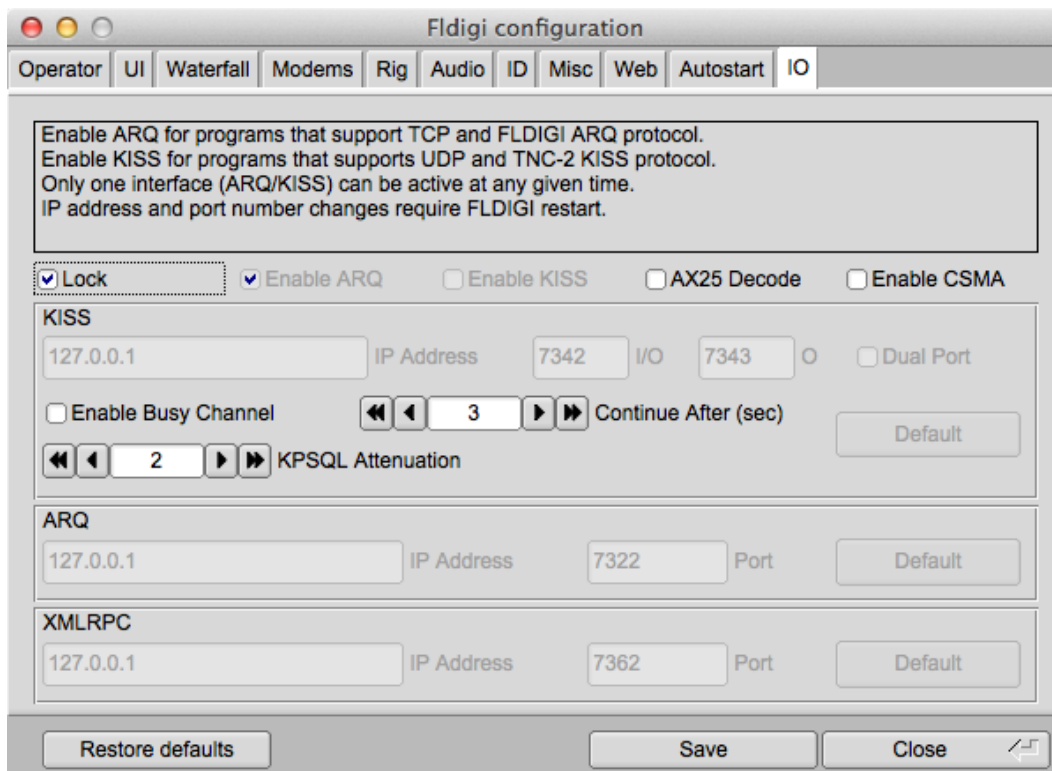


Figure 2.27: Locked IO Configuration Panel

### 2.11.1 I/O Configuration

I/O configuration panel as seen above allows the user to select one of the IO interfaces. Each of the two interface options are used for different purpose(s). ARQ is selected when the user requires the use of the FLDIGI suite of programs (FLARQ, FLMSG, FLAMP, etc). The KISS interface is used with Ax25/ARQ networking software. Currently the only software tested and verified to work with FLDIGI's KISS interface is BPQ32. See [Using BPQ32 with FLDIGI](#)

#### Note:

Any changes to the IP Address and/or port numbers requires FLDIGI restart.

#### 2.11.1.1 ARQ I/O

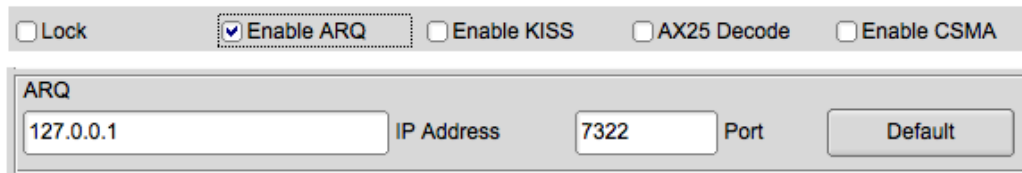


Figure 2.28: ARQ IO Panel Options

To select ARQ I/O, mouse click on the [ ] Enable ARQ check box. If the checkbox is grayed out, this indicated the panel is locked. Deselect the [ ] Lock checkbox to gain access.

The ARQ section of the configuration panel allow the user to set the IP address and port number. Pressing the [Default] button set the fields to their startup values. In order for communications to occur both FLDIGI and the FLDIGI suite of program must be assign the same IP address and port number if the programs are used on the same computer. If the program are running on different machines then the IP Address must be set accordingly while maintaining the same port number. The IP Address entered is the destination Address for the recipient program/host computer.

#### 2.11.1.2 KISS I/O

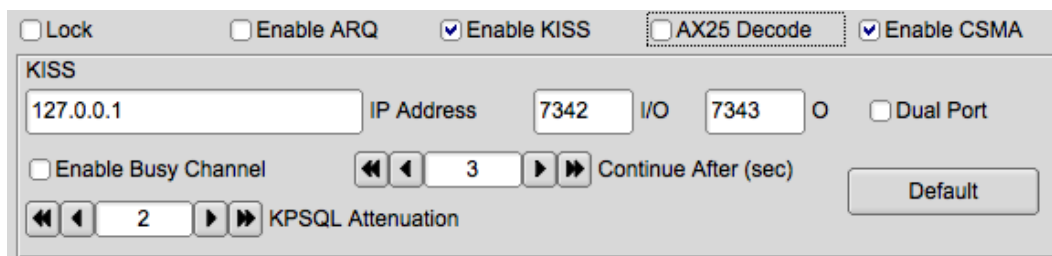


Figure 2.29: KISS IO Panel Options

To select KISS I/O, mouse click on the [ ] Enable KISS check box.

Setting and configuring IP address and ports numbers are described in the ARQ section.

##### 2.11.1.2.1 Dual Port

Since the data exchange protocol is based on UDP/IP (connectionless) loopback issues can arise if the two programs are hosted on the same machine. By allowing more then one port number. The loopback condition can be avoided. Both programs must support dual port for proper communications.

##### 2.11.1.2.2 KISS I/O CSMA

Selecting this option enables Carrier Sense Multiple Access. Basically in conjunction with KPSQL it monitors for busy signals. When the frequency is clear a small variable delay is initiated preventing the transmitter from transmitting. This variance allows other CSMA stations time to monitor the frequency in an attempt to prevent/limit packet collisions. Only available when KISS I/O selected.

### 2.11.1.2.3 Ax25 Decoding

If the data received is using Ax25 protocol enabling this option translates the protocol into human readable form. Only available when KISS I/O selected.

#### NOTE:

When activated, only valid ax25 data will be displayed to the receive panel. No un-squelched random characters will be seen.

### 2.11.1.2.4 Busy Channel

Busy Channel when enabled and with the properly set threshold of the KPSQL squelch. Frequencies that are intermittently present within the bandwidth of the modem will inhibit the transmitter for the specified duration. This is a useful feature to prevent your station from capitalizing the frequency, allowing casual users to continue without interruptions from "BBS/Email" type operation.

To prevent your station from transmitting between the gap of the RSID and the data stream. Enable Busy Channel with a 1~2 second inhibit period.

### 2.11.1.2.5 KPSQL Attenuation.

Reduces the sensitivity of the KPSQL squelch. Useful when a direct audio card to audio card connection.

KPSQL monitors for differences in the signal strength within the bandpass of the modem. Making it tolerant to AGC variations from the radio.

### 2.11.1.2.6 Modem Selection Kiss I/O

Use of the KISS I/O places a limit on the number of available modems. These modems support 8 bit transfers. The slowest of these modems have been omitted and generally comprise of the BPSK, 8PSK, MFSK, THOR, Contestia, Olivia and MT63 at various baud rates. Any attempt to select a modem not supported FLDIGI will ignore the request. If you are unsure, open the event log under the Help menu and note a message indicating that particular modem is not supported.

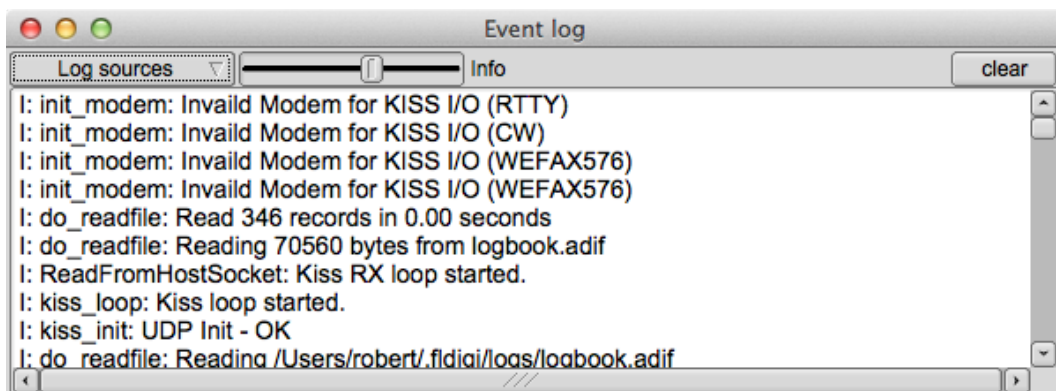


Figure 2.30: Invalid Modem for KISS I/O

### 2.11.1.3 XMLRPC I/O



Figure 2.31: XMLRPC IO Panel Options

Setting and configuring IP address and ports numbers are described in the ARQ section.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.12 Installing Fldigi on Windows

The port of fldigi to the Windows operating system is built using cross-compilers on Linux. The cross-compilation environment is created using mingw32.

Installing fldigi on windows is very simple. Simply execute the installer program and both fldigi and flarq will be installed in the default programs directory structure for the specific Microsoft version that is being targeted. Desktop icons and desktop menu items will be created. An uninstaller link will be created on the desktop menu.

Click on the desktop icon to start the application. If this is a new installation you will be guided through the initial configuration with a [new install wizard](#).

Resize the main dialog to suit your screen. Adjust the Rx/Tx divider to your liking. Then set up the operator and sound card configuration items; [configuring fldigi](#). When you have fldigi receiving and decoding signals you can exit the application which will allow you to save your configuration settings.

Now open up the following folder using your windows file explorer if you do not have a login name and password:

On XP	C:\Documents and Settings\<urlogin>\fldigi.files"
On W2K	C:\Documents and Settings\<urlogin>\fldigi.files"
On Vista/Win7	C:\User\<urlogin>\fldigi.files

where <urlogin> is the name with which you log onto the computer.

All of these files were generated by fldigi when it first started. The files with the extension pal are palette definition files. The file "macros.mdf" contains the macro definitions which you can change using the macro editor. fldigi.prefs and fldigi\_def.xml are used for storing the application state and configuration items respectively. With the exception of the location of this folder the operation of fldigi on windows is identical to linux. In all instances where the help files make reference to \$HOME/.fldigi you should be substitute the appropriate directory for XP or Vista/Win7.

Please take the time to read and reread the on-line help file. Better yet download the Adobe Reader file so that you can view the help locally without needing access to the internet. Fldigi is a large complex program with many ways for the user to customize its operation to his or her hardware environment.

### 2.12.1 Special Note for Vista/Win7 from user!

I am trying to install the latest and greatest version of FLDIGI and FLARQ on a new Toshiba laptop computer that runs Vista/Win7 / Win7. The main screen of FLDIGI comes up ok and it says that no call sign has been set and down in the bottom center right there is an **error message about there not being the sound card that it thinks it should see**. I understand the messages as the program needs to be configured. The problem is that when I click on 'Configuration' the main screen grays out and it seems to go off into never never land and not comeback or put up the configuration window. At that point the only thing that works on the main screen is the close button. What am I missing?

I solved the mystery! Cockpit error, kind of! I tried installing MULTIPSK and got the "no sound card error" also. Went into the Control Panel to see what was going on. When I looked at sound input it said there was NO microphone plugged in. Then the light came on and I remembered that you **MUST** have a microphone plugged in when using the Vista OS or it doesn't think there is a sound card in the computer!!! Well this is my first encounter with Vista and how am I suppose to know or remember that little quark on an OS I've never used before??

Once I plugged a mic into the sound card input both programs worked just fine.

**Vista/Win7 requires** either a **microphone** or a **line-in** device actually plugged into the 8 mm audio jack before the sound driver reports that there is an audio capture device.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.13 New Install Wizard

### New Install Wizard

The new installation wizard borrows from the normal configuration dialogs. You will find information on setting each of the wizard dialog pages on the associated configuration link.



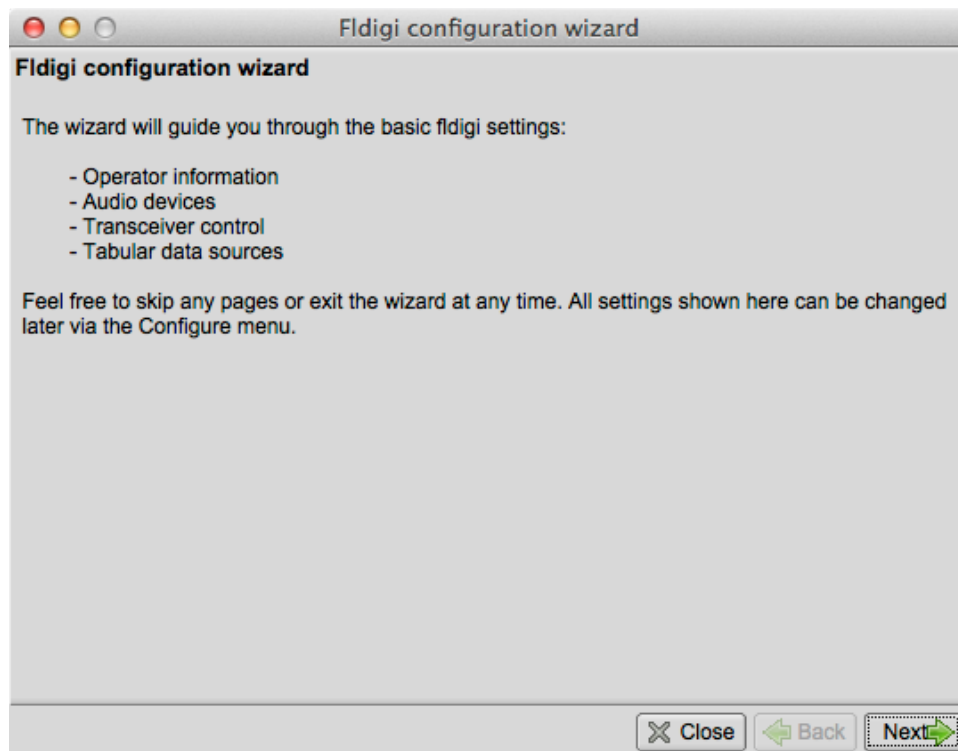


Figure 2.32: Wizard Configuration

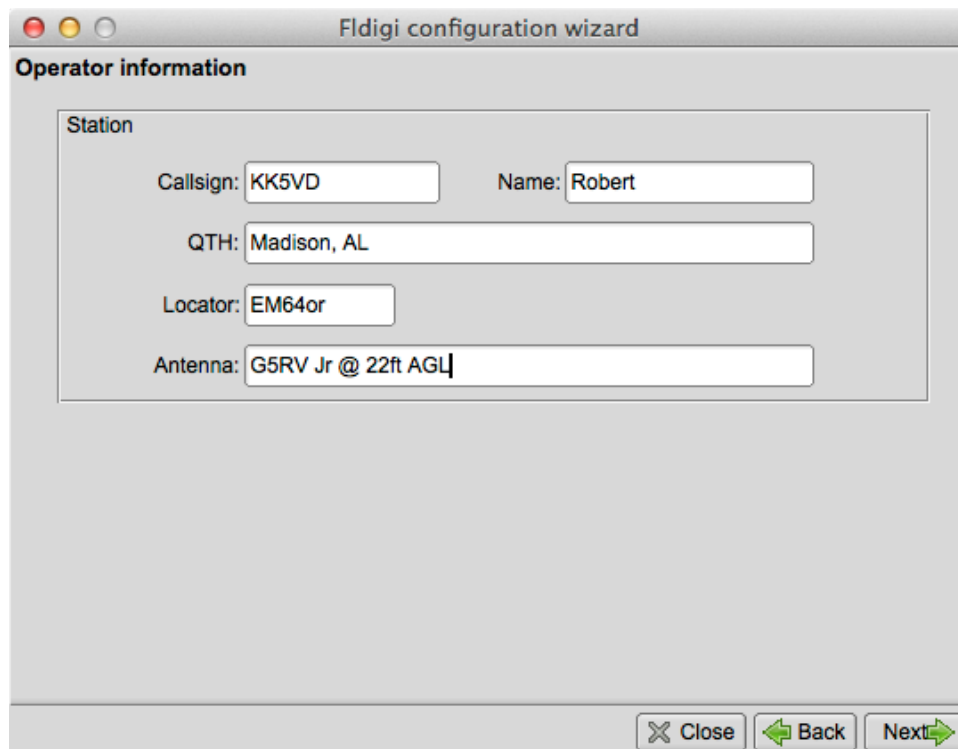


Figure 2.33: Operator Configuration

See [Operator Configuration](#)

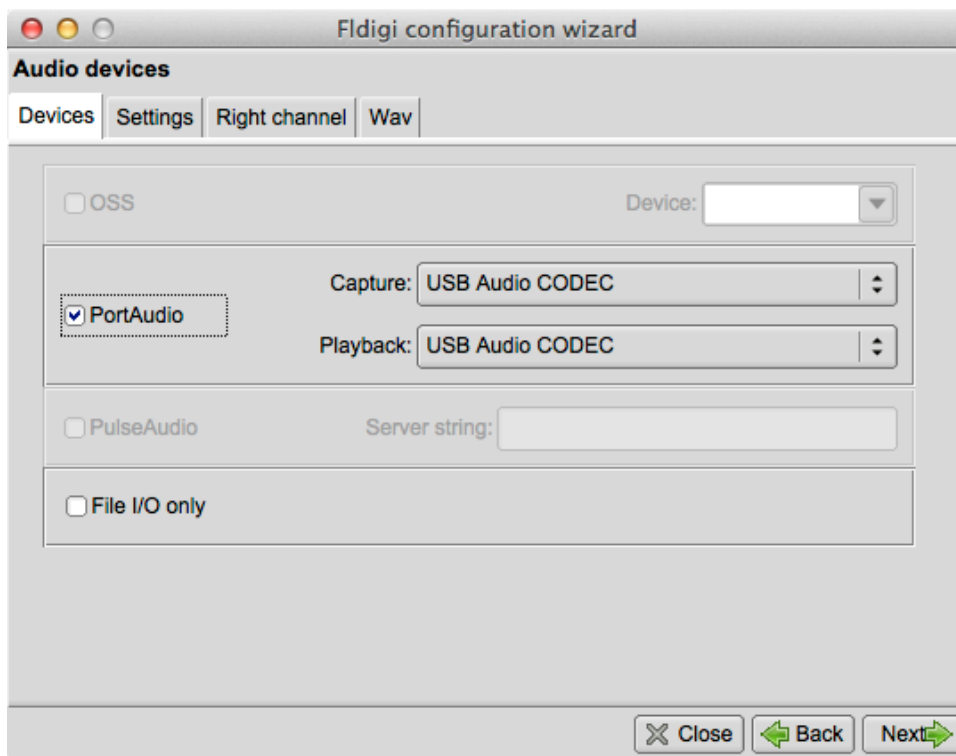


Figure 2.34: Sound Card Configuration

See [Sound Card Configuration](#)

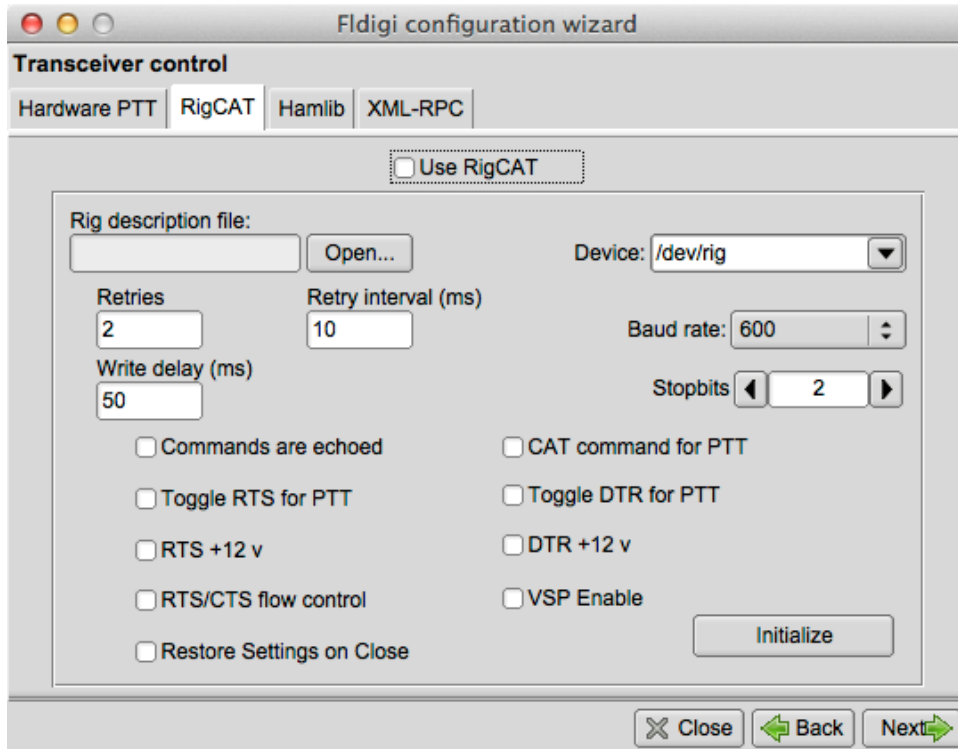


Figure 2.35: Transceiver Configuration

See [Transceiver Configuration](#)

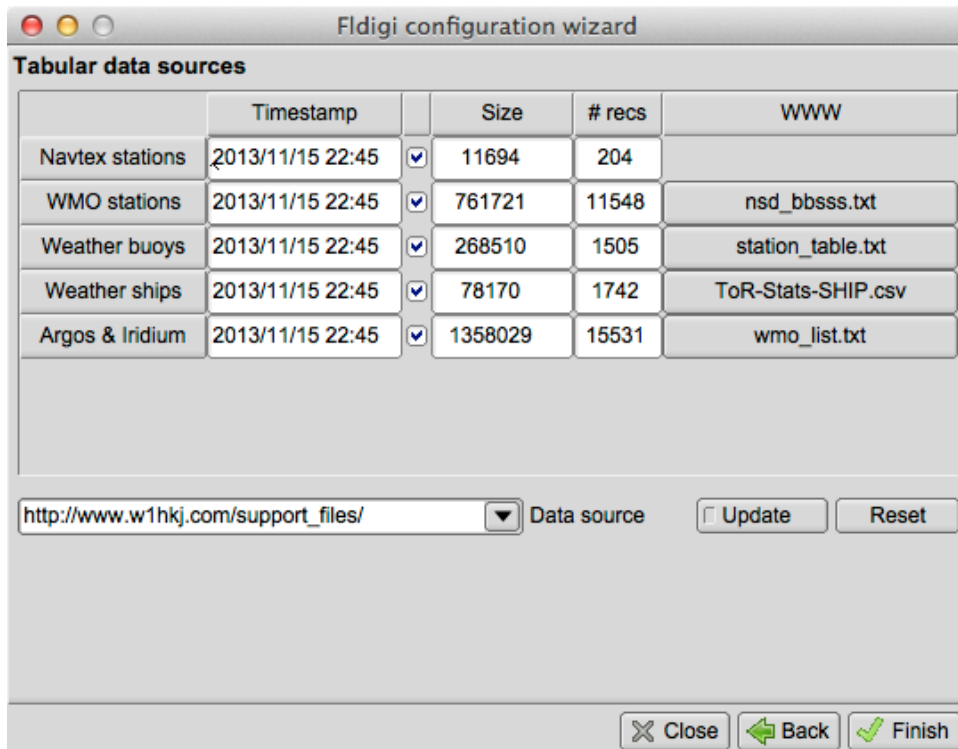


Figure 2.36: Tabular Configuration

See [Tabular Data](#)

[Return to Top of Page](#)  
[Return to Main Page](#)

## 2.14 Callsign DB Configuration

### 2.14.1 QRZ

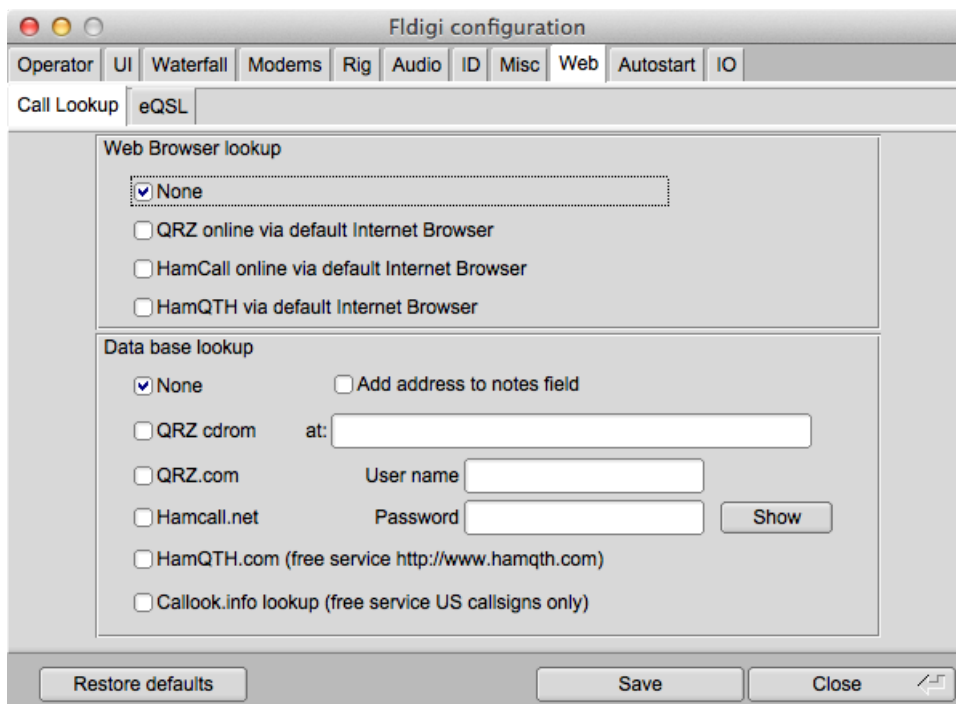


Figure 2.37: QRZ

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.

OK2CQR provides a very nice callsign lookup service on his hamqth.com web site. There is no subscriber fee to use this service. Simply register and then use your registered user name and password for access. You might want to provide monetary support to Petr if this service meets your needs.

### 2.14.2 EQSL

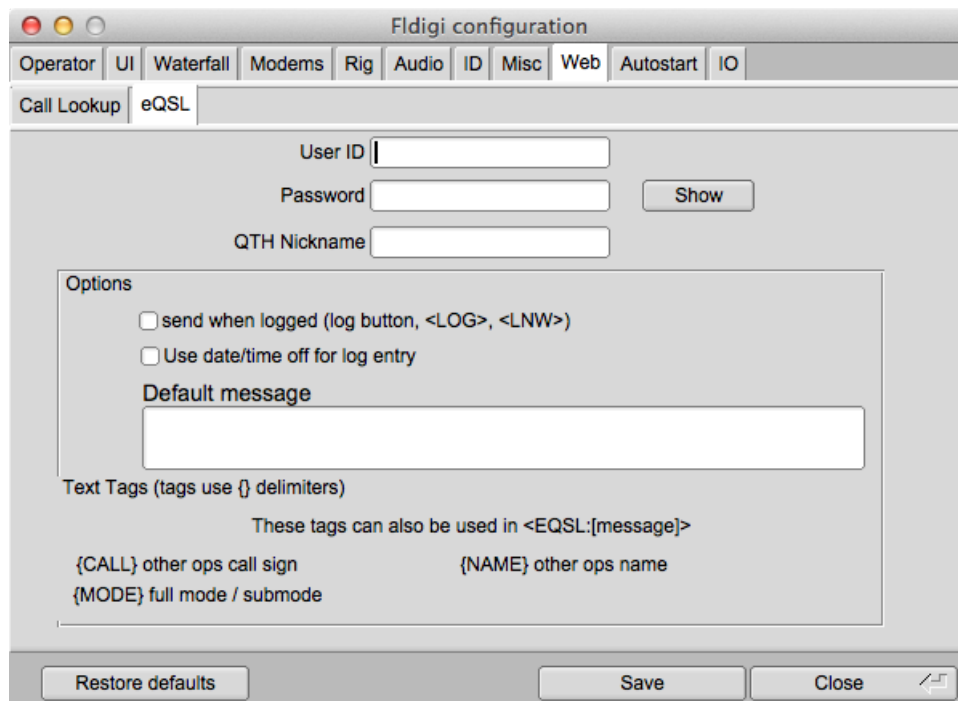


Figure 2.38: EQSL

Configure the eQSL log entry upload with your user ID, password, and QTH nickname. See [Macros](#) for additional information.

### 2.14.3 Fldigi Logging



Figure 2.39: Log It

If you check the "send when logged" button then the logged record will be sent to eQSL when either the log it button is pressed, or when the macro tag <LOG> or <LNW> is executed.

You may elect to use the default of date-time ON or date-time OFF for the eQSL submission.

You may send a message with the log record. If you fill in the default message text then it will be used for that purpose. You might want to use something like:

Tks for {MODE} qso {NAME}. 73, pse eQSL

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.15 Colors and 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.

## 2.15.1 Text Ctrls

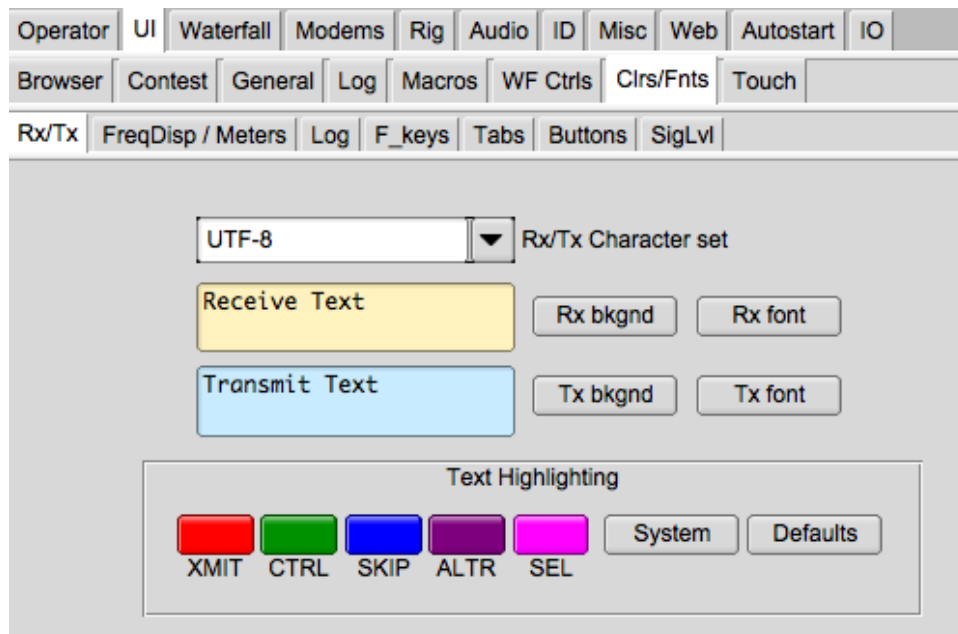


Figure 2.40: Colors Fonts RX / TX

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

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" or "Consolas". Both are excellent fonts for this use.

## 2.15.2 Character Set Selection



Figure 2.41: Character set

This setting affects how fldigi encodes the text that it transmits and how it interprets what it receives. This mainly matters if you intend to transmit and/or receive text containing characters that are not in the English alphabet (for example letters with diacritics, Cyrillic letters, Japanese script and similar). Successful transmission and reception of such symbols is only possible if both your and the correspondent's program are set to use the same encoding. Seeing "strange" characters instead of what you would expect for example "&ccaron;" always turning up as "è" indicates that there is probably a mismatch between your encoding and the correspondent's.

Leaving this control set to UTF-8 (the default) is strongly recommended as this is a cover-all encoding that enables communication in almost any language and script imaginable. However, there are three cases in which you might want to switch encodings (at least temporarily):

- You never expect to transmit or view any character except for the lower 128 characters (ASCII) of the ANSI data set. This might be true for English to English communications in which you do not want to have decoded noise appear as UTF-8 character renditions.
- You want to communicate using non-English characters, but the correspondent's program does not support UTF-8. In such a case, you should find out what encoding the correspondent is using and change your setting accordingly.
- You expect to exchange a lot of traffic that mostly consists of non-English characters (communication in Cyrillic script being a notable example). UTF-8 encodes each non-English character into a symbol two to four bytes long. Such characters require more time to transmit and thus reduce the effective transmission speed. Choosing another encoding that requires less bytes to be sent might be beneficial if the transmission speed is crucial. For example, a pair of operators wanting to communicate in Cyrillic script might want to choose CP1251 to retain the maximum transmission speed.

Warning: transmission of non-English characters, regardless of the encoding used, requires that the digital mode used be capable of handling 8-bit traffic. The following modes suit this requirement:

- DominoEX
- MFSK
- MT63 (8 bit extended characters must be enabled)
- Olivia (8 bit extended characters must be enabled)
- PSK (all variants)
- THOR



### 2.15.3 Frequency Display

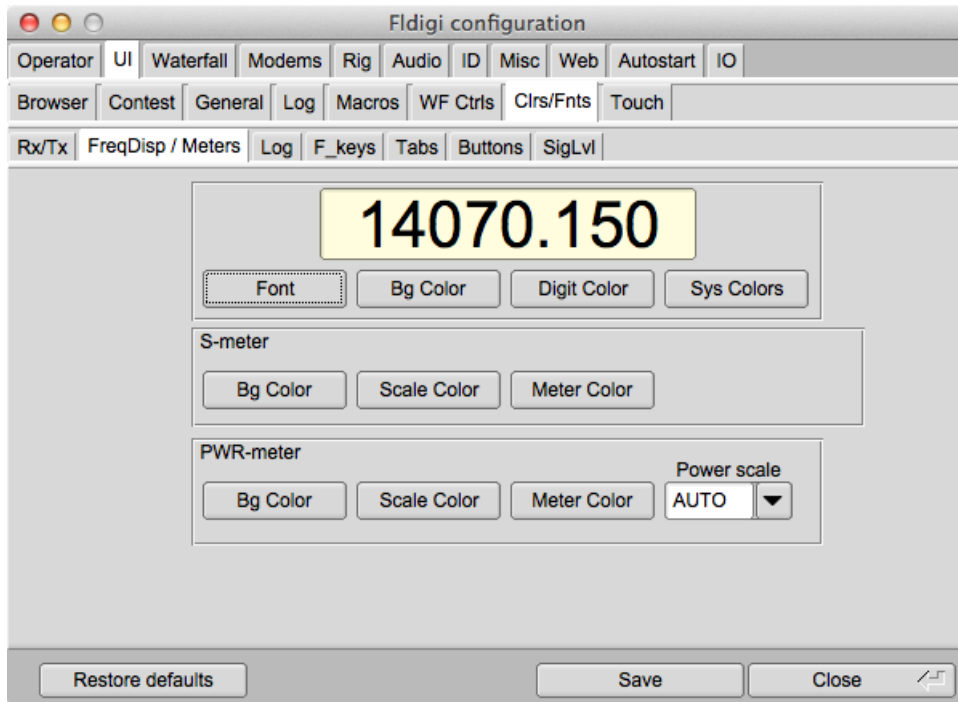


Figure 2.42: Frequency Display

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.

### 2.15.4 Logging controls

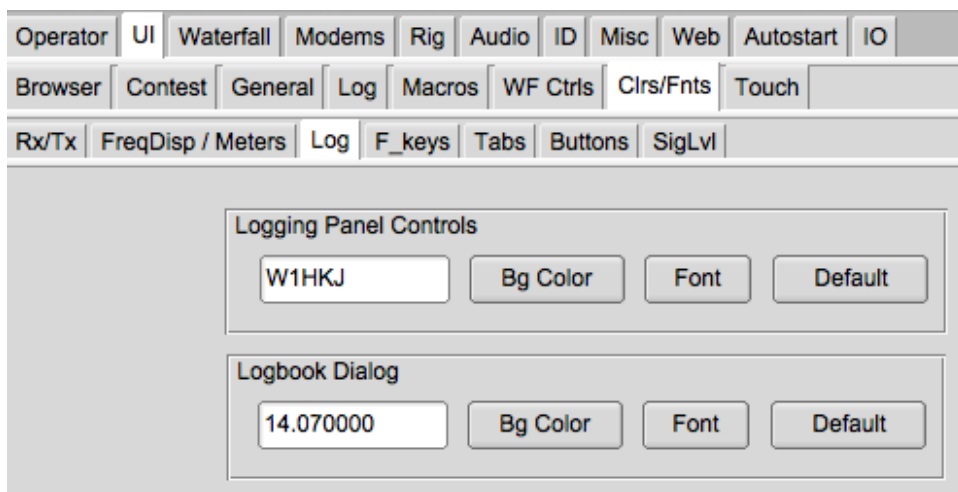


Figure 2.43: Logging controls

You can select the background color, the font, the font color and the font size for both the logging controls on the main dialog and the separate logbook dialog.

### 2.15.5 Func keys

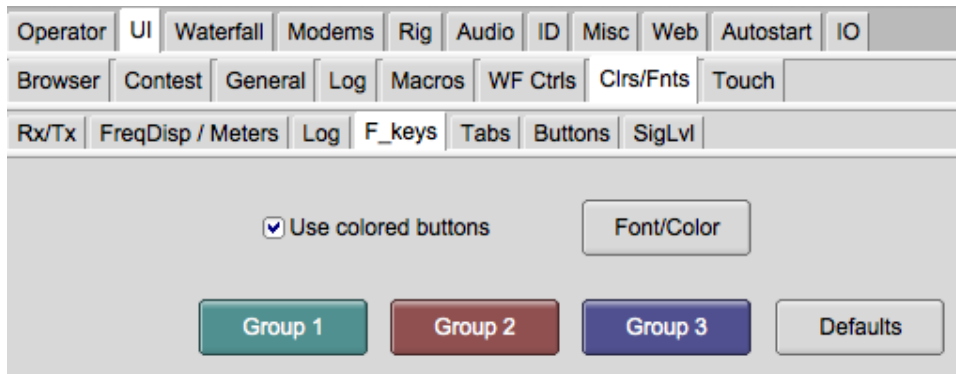


Figure 2.44: Colors Fonts FKeys

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.

### 2.15.6 Tab Colors



Figure 2.45: Colors Fonts Tabs

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

## 2.15.7 Light Buttons Colors

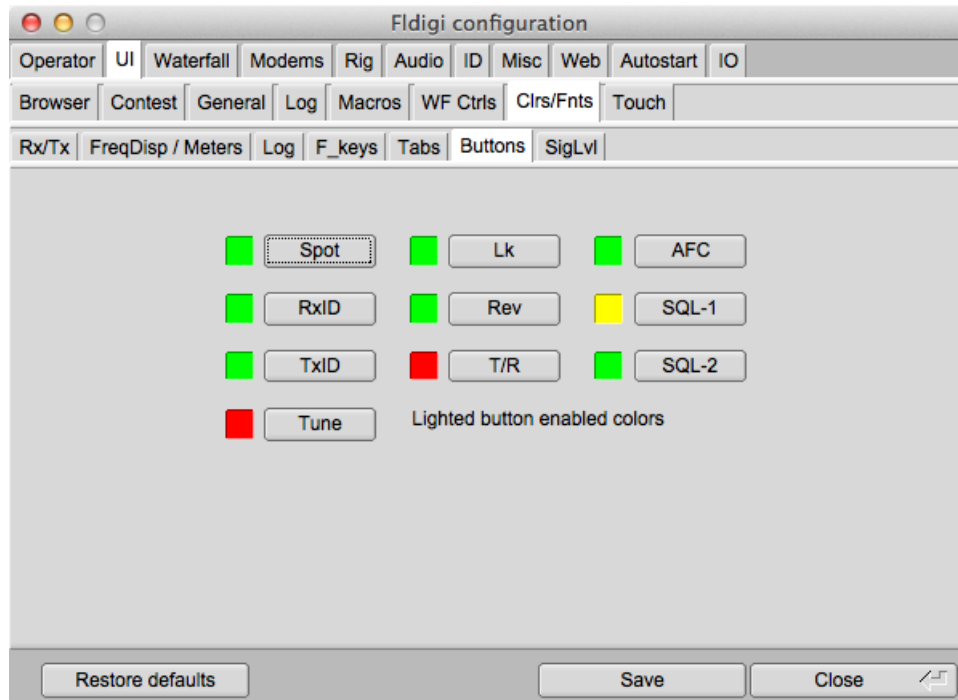


Figure 2.46: Colors Fonts Buttons

- 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

## 2.15.8 Signal Level

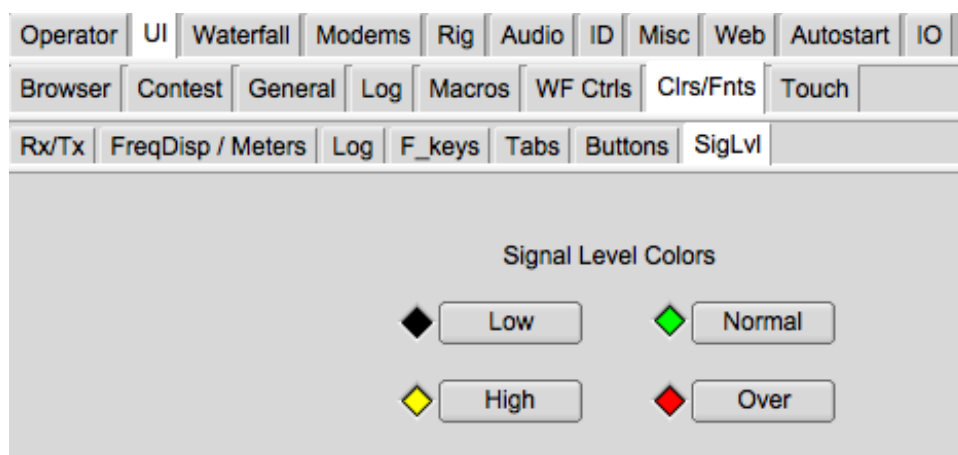


Figure 2.47: Colors Fonts Signal Level

You can select the low/normal/high/over-drive colors for the diamond signal level in the lower right hand corner of the main dialog.



Figure 2.48: Level Indicator

Your audio Rx level should be set to allow the loudest signals to drive the indicator into the yellow and never into the red zone.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.16 Configure FFT Scan

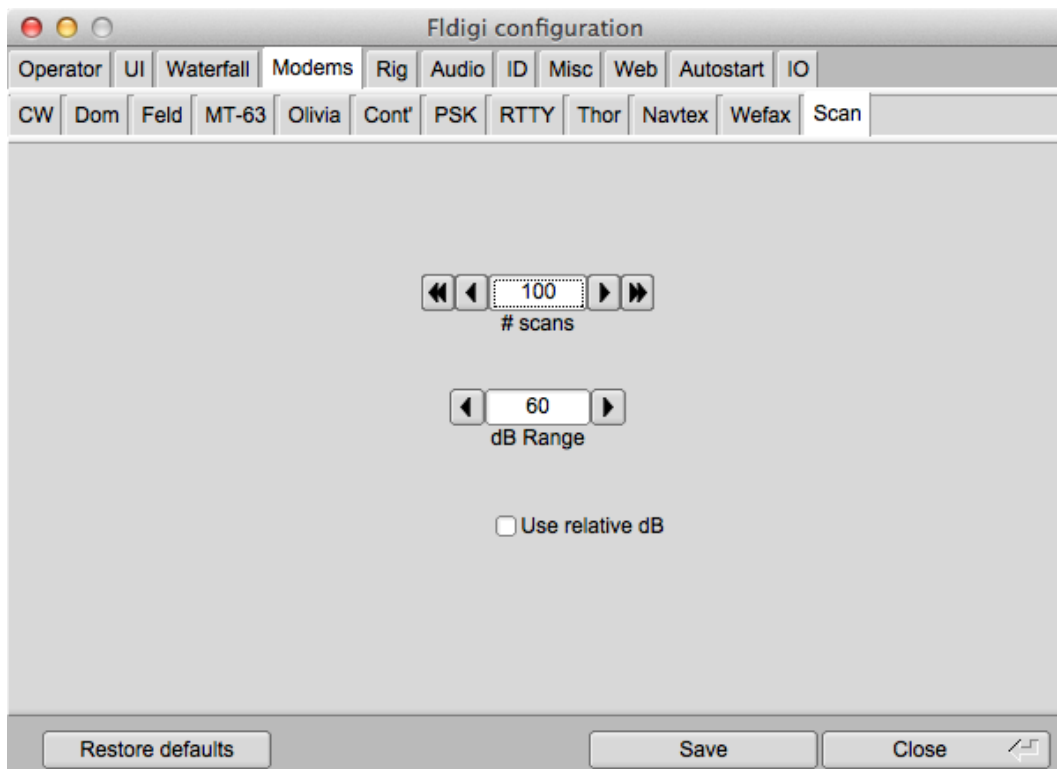


Figure 2.49: Scan Configuration Panel

# scans: The number of scans to perform (averaging).

db Range: The dynamic range in dB (vertical scale).

### 2.16.1 FFT Scan

Analysis tool used to evaluate Rx/Codec signal path. Can be used to measure noise floor of the codec; view the transceiver passband characteristics, Rx signal spectrum, etc.

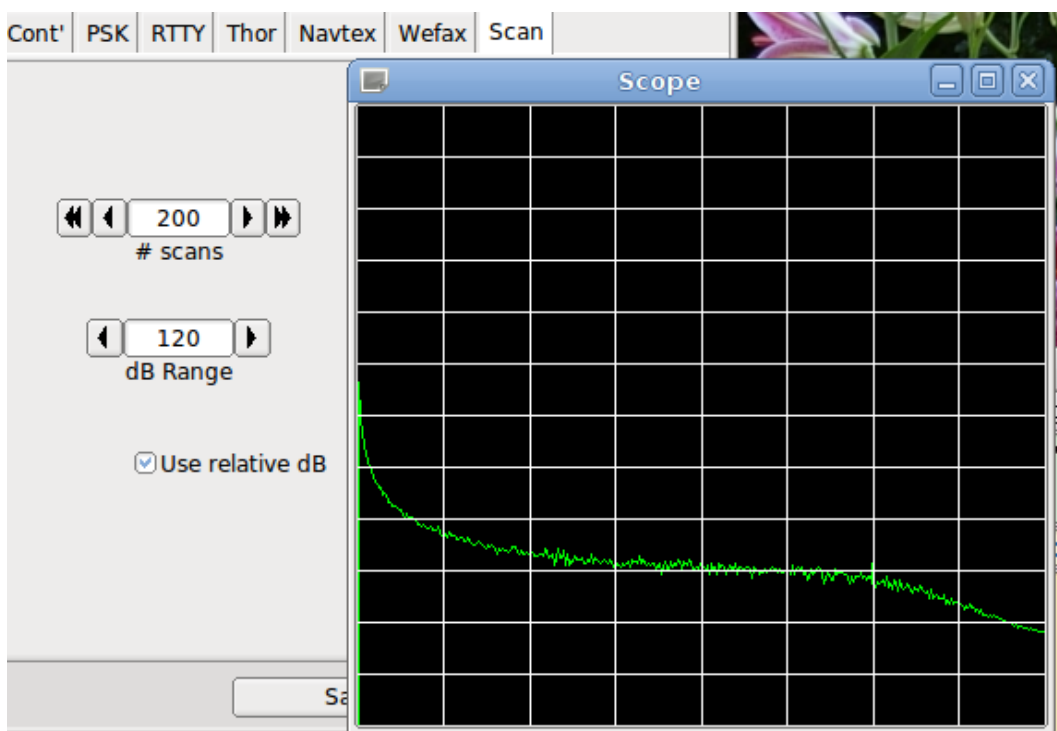


Figure 2.50: Scan results

The scope shows the results, both final and during the scan averaging. The # of scans specifies how many scans are averaged for the display. The dB range specifies the vertical scale as seen above. "Use relative dB" removes the vertical offset so that a wider range can be displayed with more detail.

Left clicking anywhere on the waterfall restarts the scan. You can observe the effect on the scope display. A comma separated file is also generated; `~/fldigi/fftscan.csv` which can be opened with any spreadsheet program.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.17 RSID / Video / ID Configuration

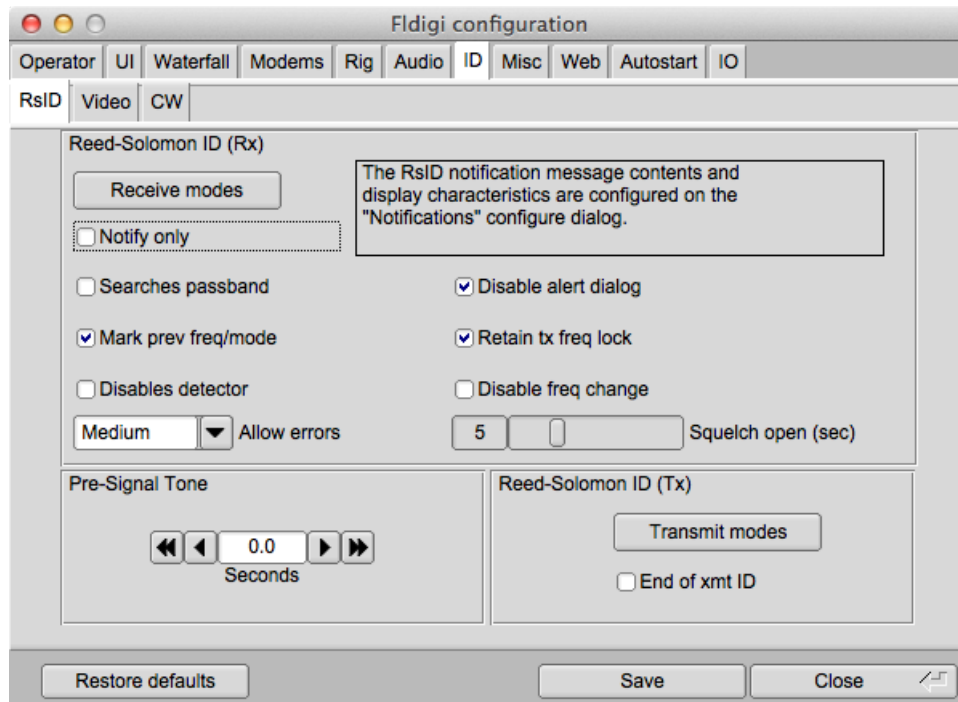


Figure 2.51: RSID Config Panel

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](#).

### 2.17.1 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 RSID 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.

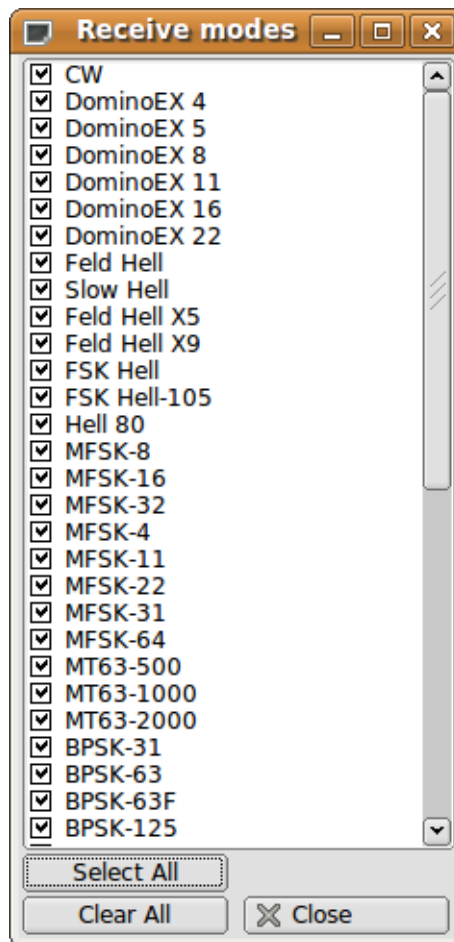


Figure 2.52: Receive Modes

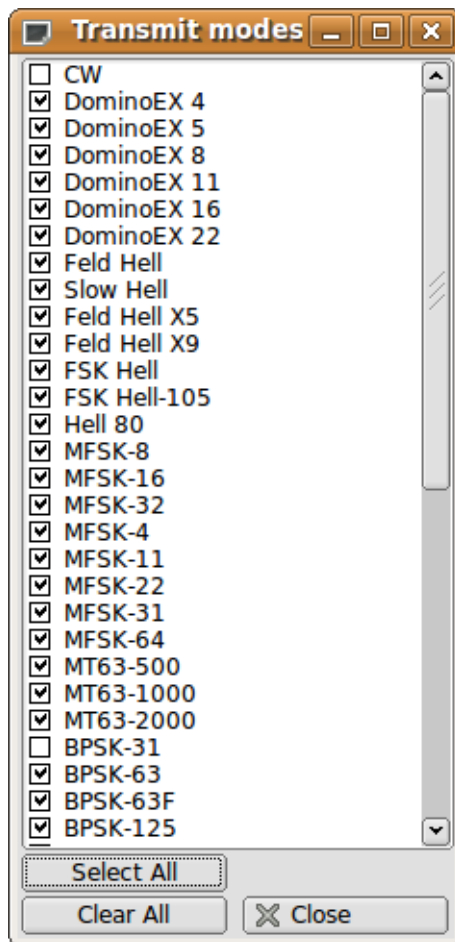


Figure 2.53: Transmit Modes

### 2.17.2 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.





Figure 2.54: Small Font



Figure 2.55: 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.

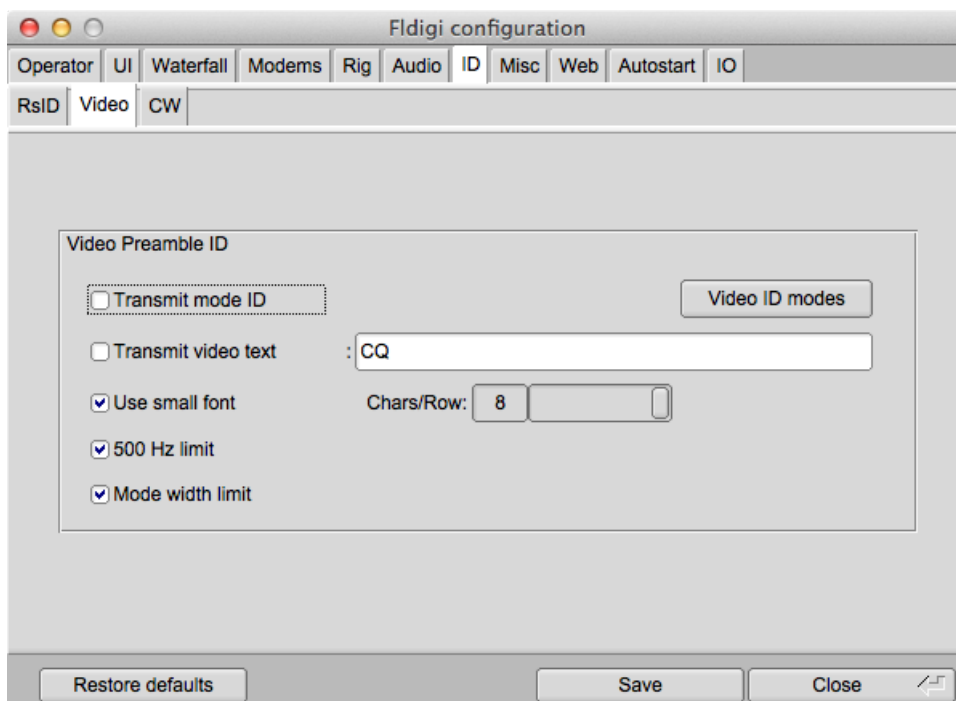


Figure 2.56: Video ID Config Panel

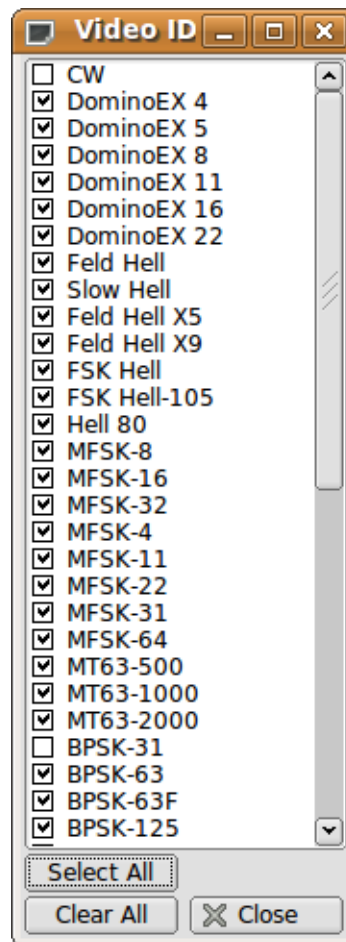


Figure 2.57: Video ID Modes

### 2.17.3 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.

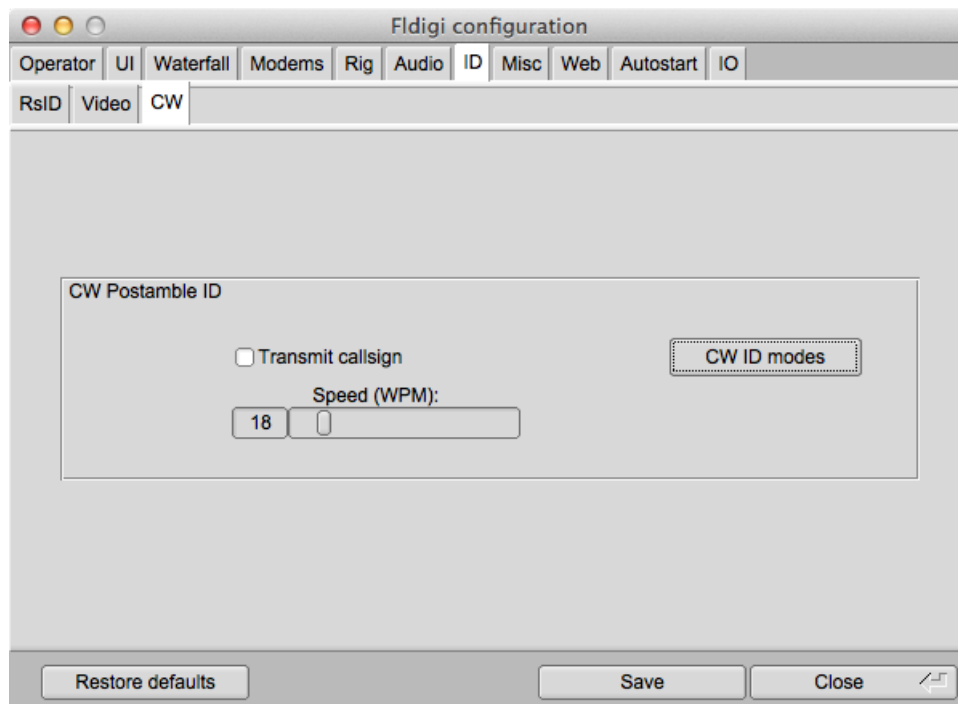


Figure 2.58: CW Postamble ID Config Panel

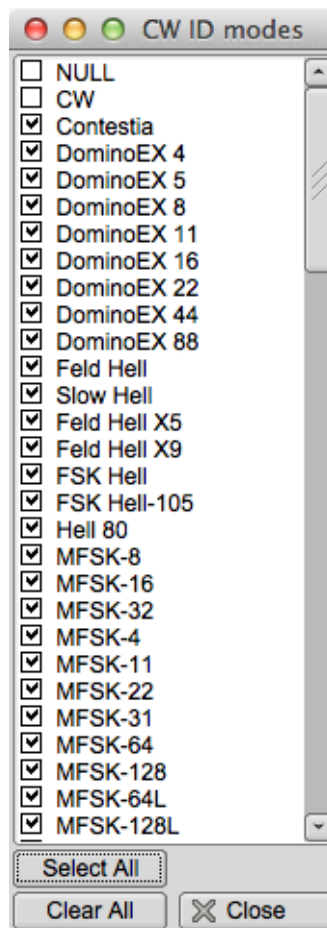


Figure 2.59: CW ID Modes

[Return to Top of Page](#)  
[Return to Main Page](#)

## 2.18 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.

Operator	UI	Waterfall	Modems	Rig	Audio	ID	Misc	Web	Autostart	IO
CPU	NBEMS	Psksmail	Spotting	Sweet Spot	Text i/o	DTMF	WX	KML		

Mail Server Attributes

Carrier frequency (Hz)  Reset to Carrier

Search range (Hz)

Acquisition S/N (dB)

AFC range (Hz)

General

Report ARQ frames average S/N

Figure 2.60: PSK Mail Configuration

[Return to Top of Page](#)  
[Return to Main Page](#)

## 2.19 Configure Touch Screen

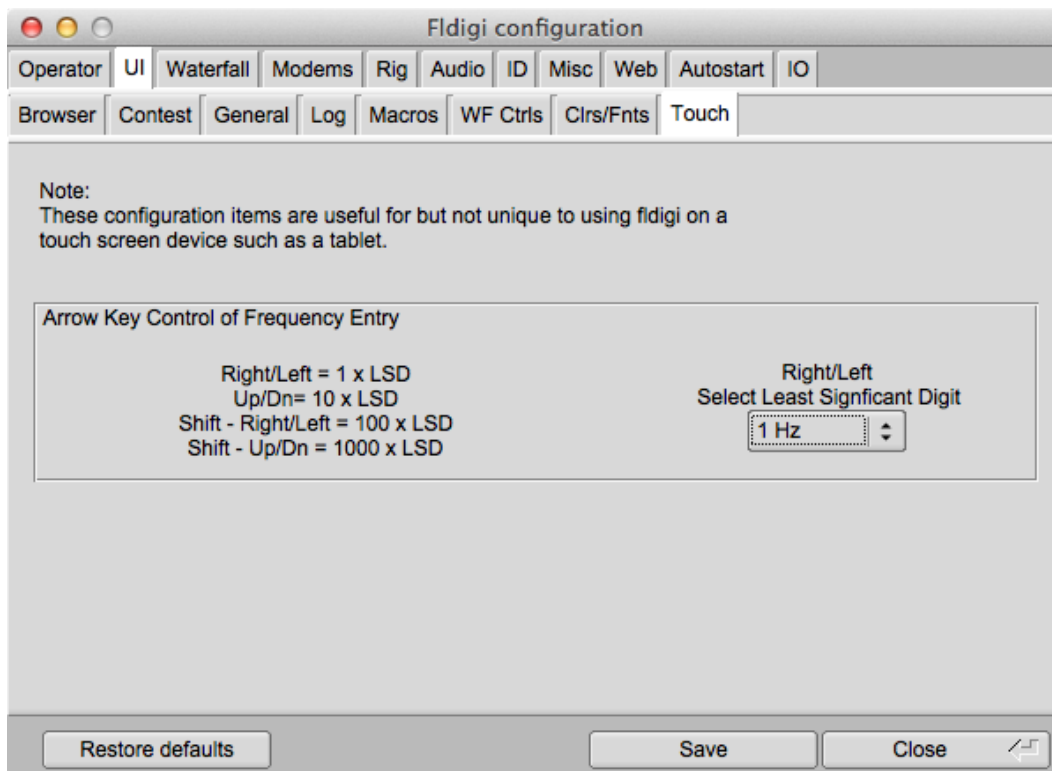


Figure 2.61: Touch Screen Configuration Panel

### 2.19.1 Touch Screen

Provides a few UI configurable items that improve using fldigi on a touch screen device, such as a Windows tablet, or an Android running a remote desktop with fldigi as the application. These items are enhanced by the addition of a macro tag that allows the operator to change focus to the frequency input widget by pressing that macro key:

<CLRRX>	clear RX pane
<CLRFX>	clear TX pane
<FOCUS>	rig freq has kbd focus
<GET>	text to NAME/QTH
<CLRLOG>	clear log fields
<LOG>	save QSO data

Figure 2.62: Macro Command

#### <FOCUS>

Which then enables keyboard entry into the frequency widget, but more important also allows the cursor keys, and Shift-cursor keys to modify the frequency entry.

Tablets usually put the on-screen keyboard at the bottom of the display which covers up the fldigi Tx panel. Typing

in the blind is not very satisfactory, so the "Tx above Rx" allows the the Rx and Tx panels to be swapped. The functions keys are too small for most adult male fingers on a touch screen. The "one Macro Row above Rx/Tx" and the size adjustment provides relief for bumbling fingers and also moves the function key bar from beneath that immovable tablet on screen keyboard.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.20 User Interface Configuration - Browser

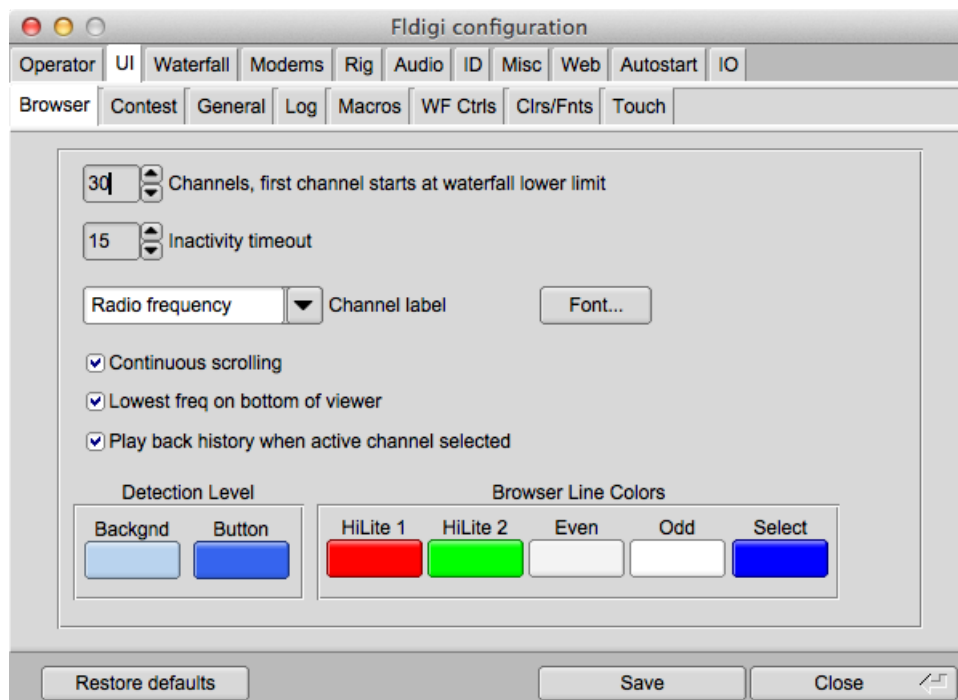


Figure 2.63: UI Browser Configuration

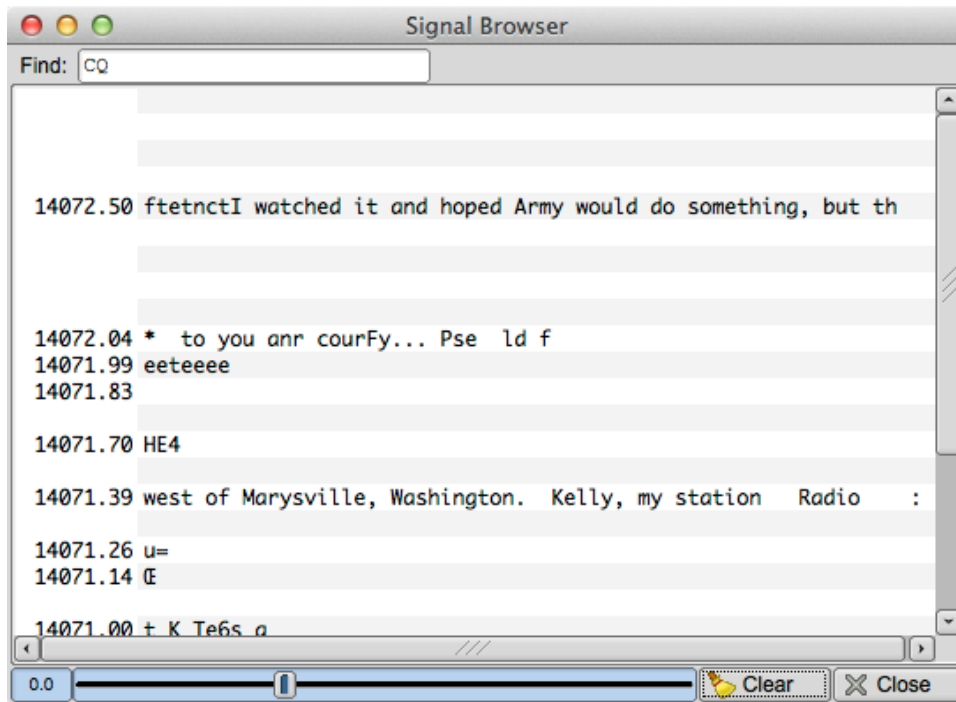


Figure 2.64: Browser Panel

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 snippet 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:

```
>.[{() \*+?|^$</span></big><br>
```



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 highlight colors can be selected for the signal browser detection level control. The default colors are shown in these images.
- 11) You can set the 2 levels of text hi-lighting that is used in the browser lines

- HiLite 1 - Text color when the regular expression evaluator finds the target text

- HiLite 2 - Text color when your call sign appears in that receive channel

- 12) You can set the background colors for the odd/even lines and the line selection color

[Return to Top of Page](#)  
[Return to Main Page](#)

## 2.21 User Interface Configuration - Contest



Figure 2.65: UI Tabs

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.

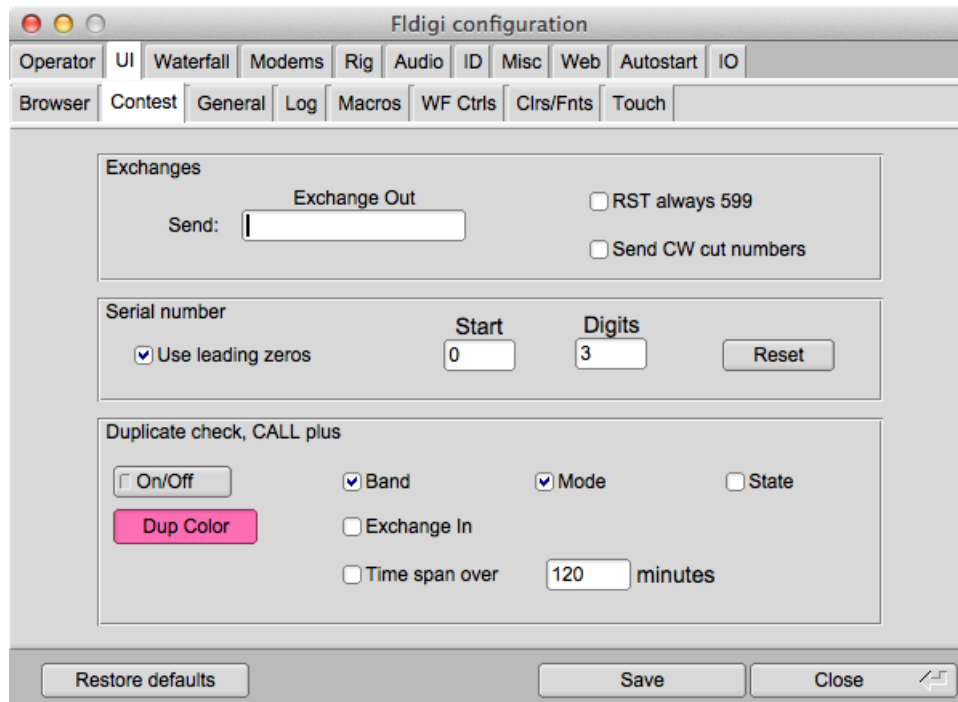


Figure 2.66: UI Contest

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. See [Contest How To](#) for more info.

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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.22 User Interface Configuration - General

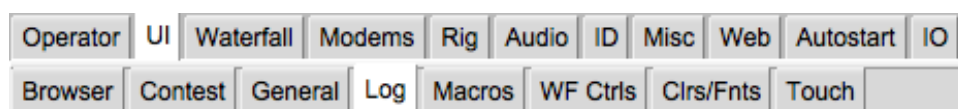


Figure 2.67: UI Tabs

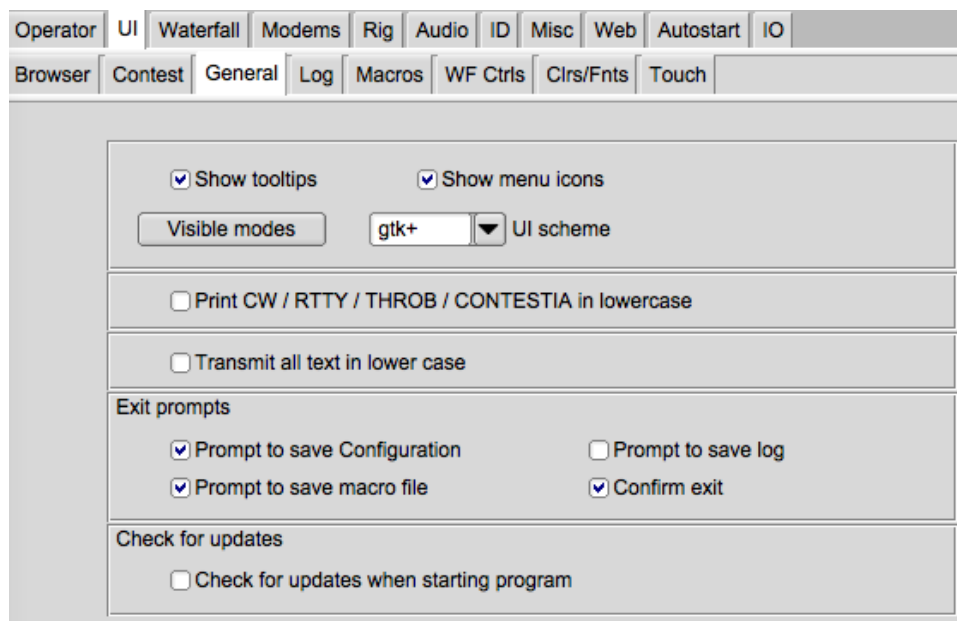


Figure 2.68: General UI Config Panel

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 its 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.

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

Select the configuration items to allow prompting when exiting the program. Note that the prompts are only active if the menu item File/Exit is used.

You can elect to have the program check for updates every time it is started. You can also manually check for updates from the "Help / Check for updates" menu item.

### 2.22.1 Visible Modes

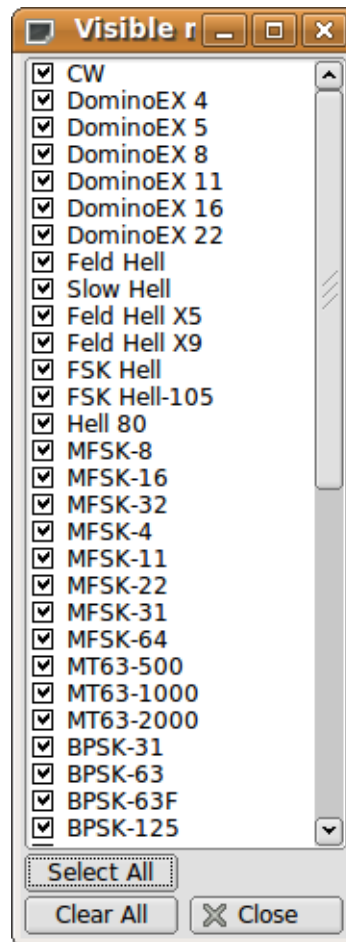


Figure 2.69: Visible Modes

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.

## 2.22.2 Limit Modes

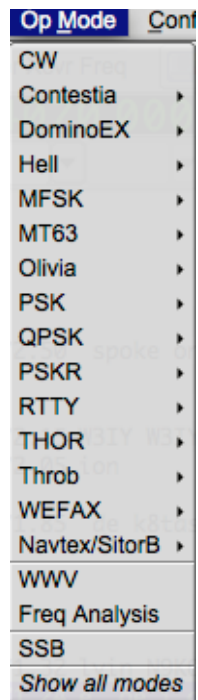


Figure 2.70: Limit Modes

By deselecting all but CW, PSK31, PSK63 and RTTY the Op\_Mode menu is uncluttered 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."

[Return to Top of Page](#)  
[Return to Main Page](#)

## 2.23 User Interface Configuration - Logging

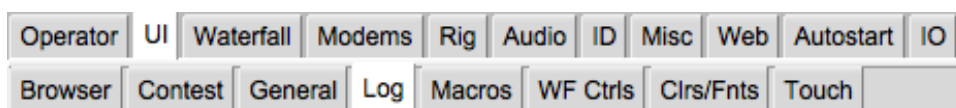


Figure 2.71: UI Tabs

The screenshot shows the Fldigi configuration window with the following sections:

- QSO logging:**
  - Prompt to save log on exit
  - Clear on save
  - Convert callsign to upper case
  - Auto-fill Country and Azimuth
  - Sort by Date/Time OFF
  - Date time ON == OFF
  - Default RST in to 599
  - Default RST out to 599
  - cty.dat pathname:
  - Buttons: Browse, Default, Reload
  - Transmit Power:
- Rx Text Capture:**
  - Word delimiters:
  - Single-click to capture
  - callsign tooltips in received text
- Client/Server Logbook:**
  - Address:
  - Port:
  - Reconnect button

Figure 2.72: QSO Logging

### 2.23.1 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 "cty.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 elect to use a single left click method if you prefer. The word pointed to by the cursor is parsed to determine it's intended field use; call, RSQ, operator name, and QTH. The call will be overwritten, but the other log fields need to be blank for the data transfer to be accomplished. 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. Clicking on the 599 NJ and Bozo would treat each as a separate word. You may also right click on a word (or a phrase that has been highlighted by dragging the cursor over the text). That will open a pop-up menu with selections to transfer the data into a number of different log fields.

You can elect to have the RST in/out preset to 599 after you clear the QSO entry fields.

If you check the "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:

### 2.23.2 Pop ups

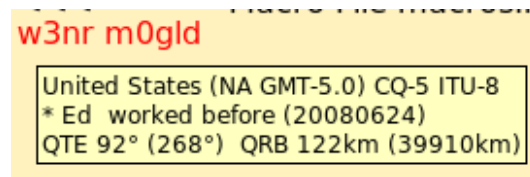


Figure 2.73: Pop ups

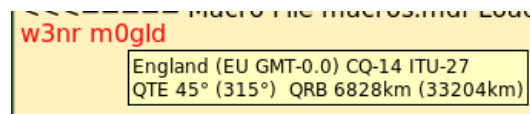


Figure 2.74: Pop ups 2

### 2.23.3 Country Files

The data is derived by parsing the callsign and referral to both the logbook and the "cty.dat" file. 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/>

The default location for this file is in the fldigi default files folder. You have the option of changing that file location by either entering the new folder pathname in the edit control, or by using the "Browse" or "Default" button. If you change the contents of cty.dat while fldigi is running you should force fldigi to reload the data from the file. This data is normally only read when the program starts.

### 2.23.4 Internal Logbook

Fldigi has an internal logbook. That log should be used for single operator operations. There may be times that you need to share a log, either between programs on a single computer, or with other operators running fldigi on other computers on a LAN (or even WAN).

In lieu of the internal logbook you can elect to use a common logbook server. This logbook is maintained by a separate logbook program, `fllog`. `fllog` provides access to read, query and update records via an xmlrpc socket interface. `fllog` provides the server function and connecting applications are clients.

You need to specify both the socket address and socket port. The defaults are as shown and are for the instance when both `fllog` and `fldigi` are on a single computer. Unless configured otherwise, `fllog` will always use the port

address 8421.

### 2.23.5 Logbook Network Address






<b>IP Address / Name</b>	
 192.168.1.89 / HPBA49A1	
 192.168.1.92 / linux-dev	
 192.168.1.93 / HelensDell	
 192.168.1.94 / fl-mac	
 192.168.1.96 / dave-vista	

Figure 2.75: Network Address

My home network has IP address assignments as shown. If fllog were running on the mini-mac (fl-mac) and fldigi running on the linux-dev machine. I would enter the server address 192.168.1.94 into the fldigi configuration for the Client/Server Logbook.

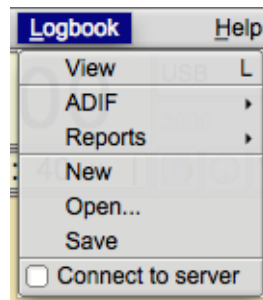


Figure 2.76: Connect to Server

The "Logbook/Connect to server" menu item allows you to connect to the remote logbook. If successful then the toggle remains checked and the menu items for accessing the internal logbook are disabled.

[Return to Top of Page](#)  
[Return to Main Page](#)



## 2.24 User Interface Configuration - Macros



Figure 2.77: UI Tabs

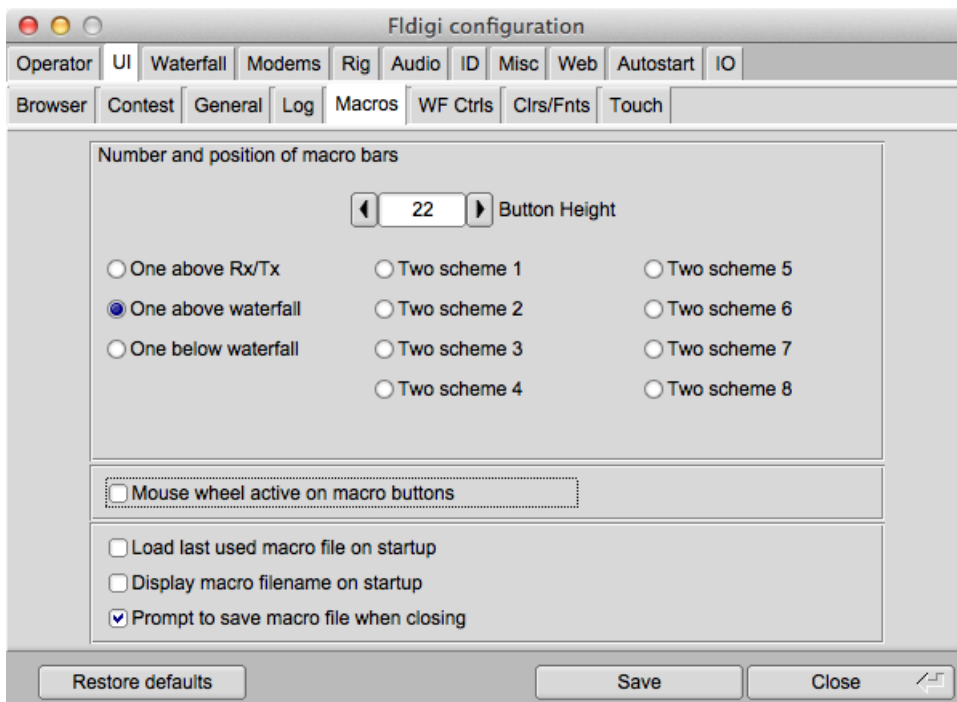


Figure 2.78: UI Configure - Macros

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.

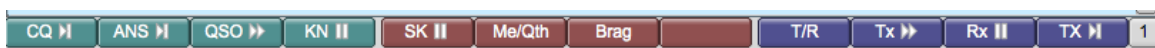


Figure 2.79: Macro Buttons

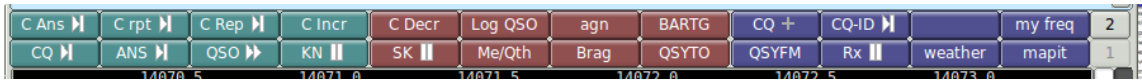


Figure 2.80: Macro Buttons - Scheme 4

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.

All 48 macro buttons can be displayed in a matrix.

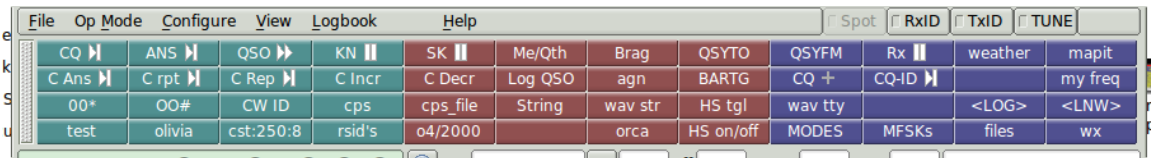
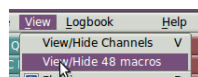


Figure 2.81: Macro Buttons Matrix

The matrix display is toggled on and off using the menu item



The matrix display can be dragged off and restored to it's position below the main menu. The drag button is the knurled vertical button on the left edge of the matrix. Left click on the drag button and drag the matrix with the mouse.

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.

[Return to Top of Page](#)  
[Return to Main Page](#)

## 2.25 User Interface Configuration - WF Controls



Figure 2.82: UI Tabs

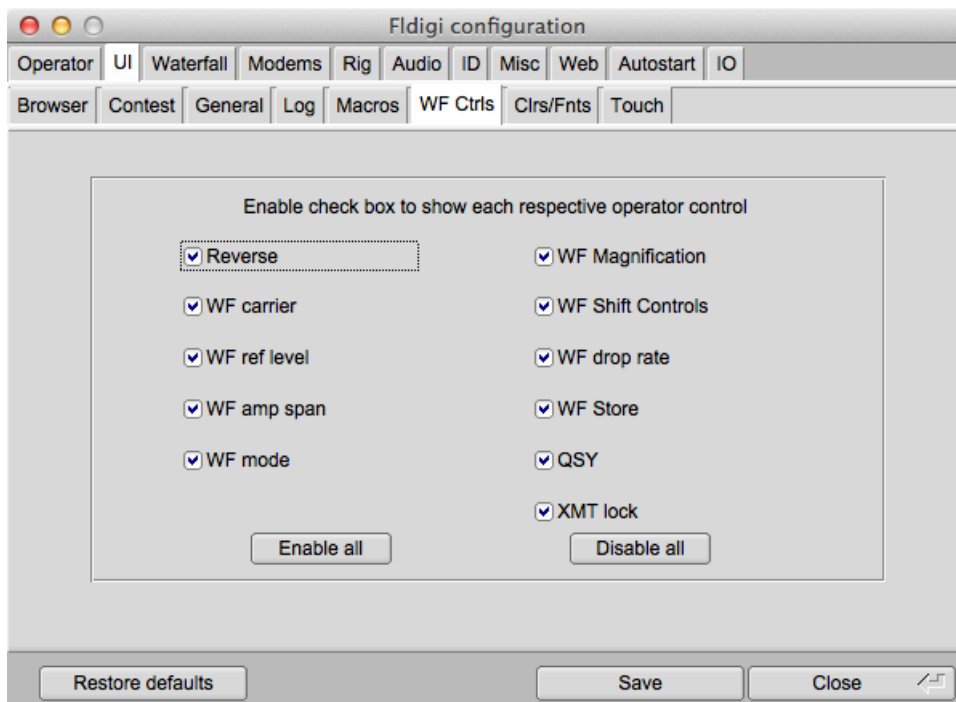


Figure 2.83: UI Configure - WF Controls

You can configure the appearance of fldigi in a variety of ways, including the suppression of unused waterfall controls.

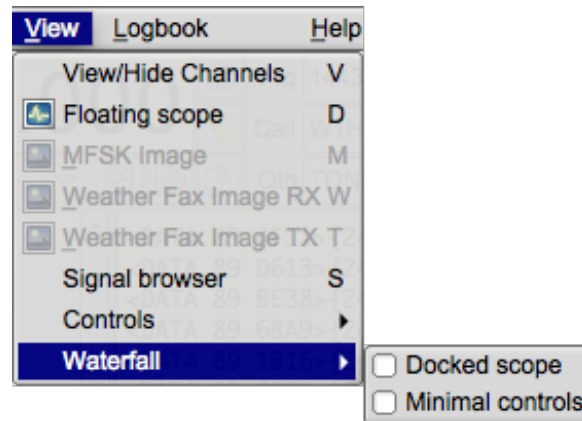


Figure 2.84: UI Configure - WF Controls 2

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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.26 Waterfall Configuration



Figure 2.85: UI Tabs

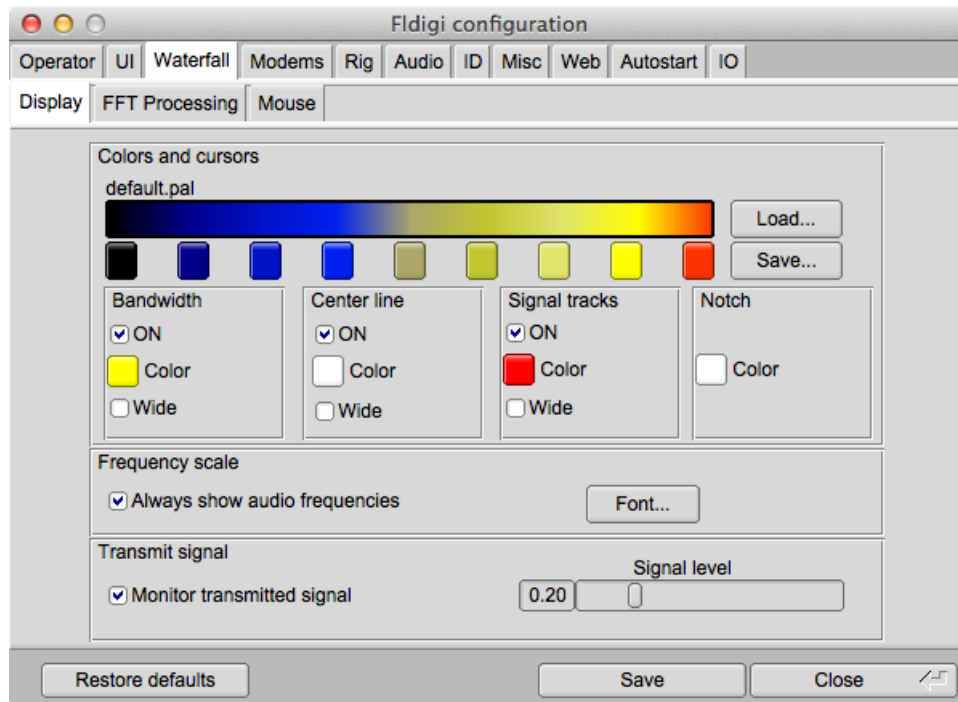


Figure 2.86: UI Waterfall Display

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!

Fldigi can set a notch from the waterfall when used with flrig and a transceiver that supports CAT control of a manual notch filter. When the notch is engaged a dotted vertical line is placed on the waterfall at the notch location. You can configure the color of the dotted notch indicator.

### 2.26.1 FFT Waterfall

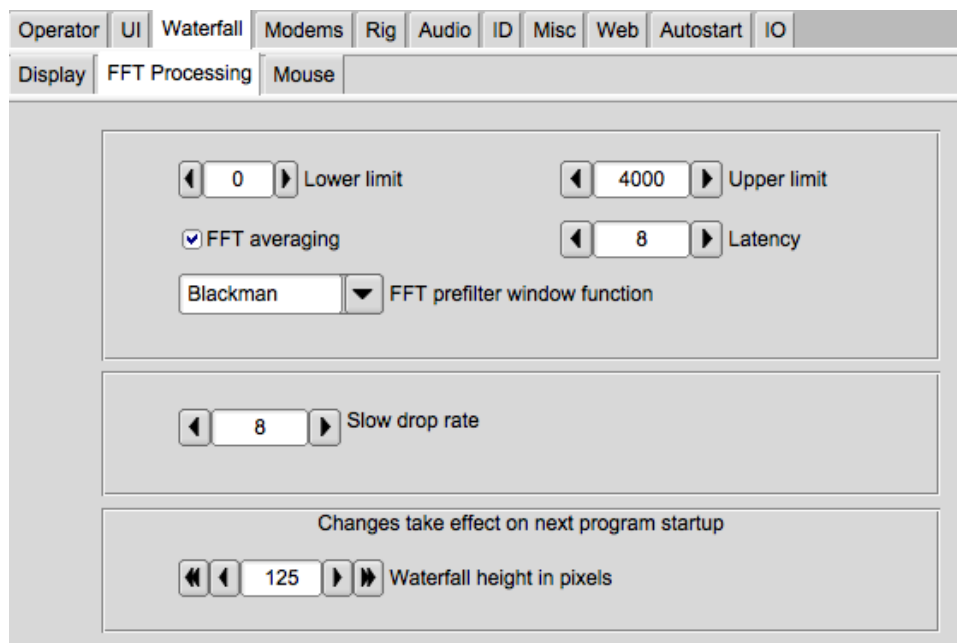


Figure 2.87: UI Waterfall Display

You can extinguish the display of received signals below a particular audio frequency.

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 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.

### 2.26.2 Waterfall Mouse Behavior

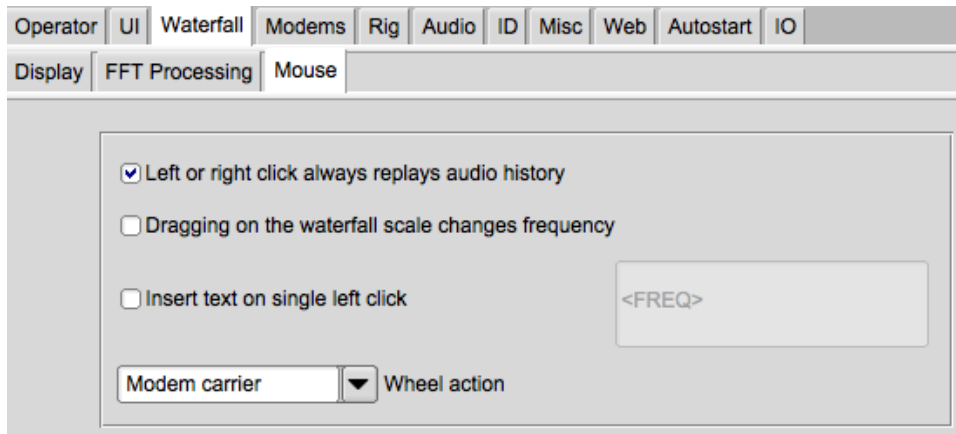


Figure 2.88: UI Waterfall Mouse Behavior

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.

#### 2.26.2.1 Waterfall Mouse Behavior Tailoring

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)

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.27 WEFAX

This modem is able to receive and transmit HF-Fax images, traditionally used for weather reports.

More technical information is available on the wikipedia article [Radiofax](#).

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

Many frequencies are available at <http://www.hffax.com/> for example.

When entering any Wefax mode, the reception window opens, and optionally the transmit window. it is always possible to bypass this with the menu bar:

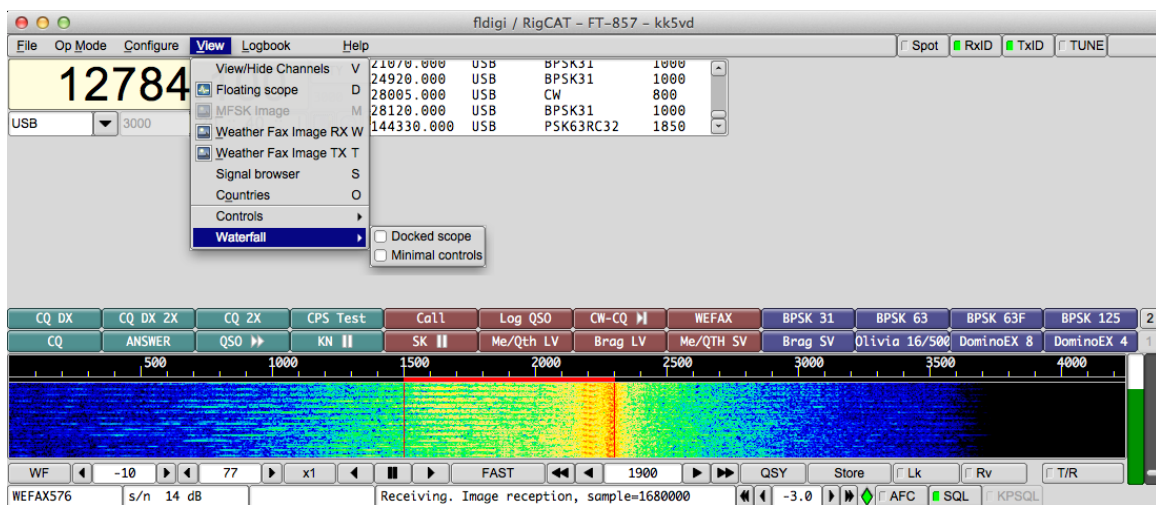


Figure 2.89: WEFAX

### 2.27.1 Configuration.

As for any modem, weather fax has its own tab in the configuration window.



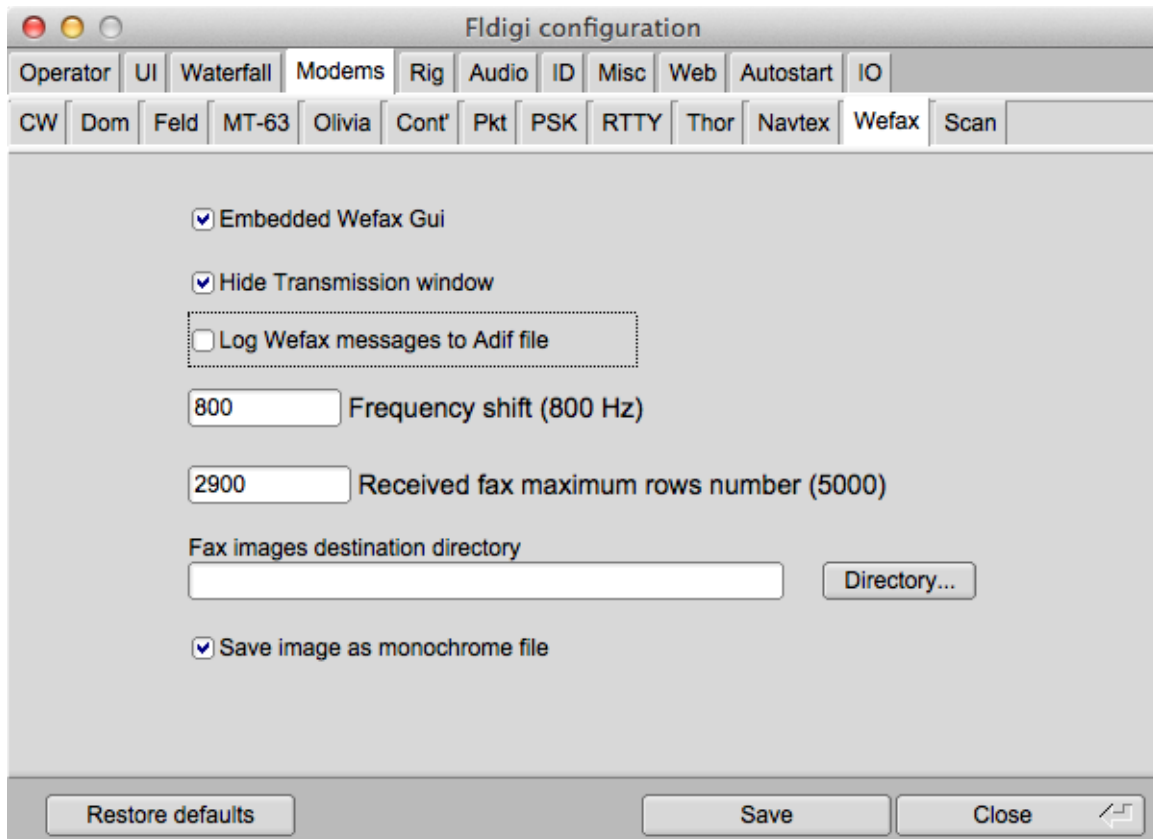


Figure 2.90: WEFAX Configuration

#### 2.27.1.1 Embedded vs floating display mode

There are two display modes for this modem:

- Embedded mode: This is the default mode, the normal receipt and transmit windows gets graphic (Like Hellschreiber mode).
- Floating mode: There are two separate windows for transmission and reception. This was the only available mode until fldigi 3.21.49.

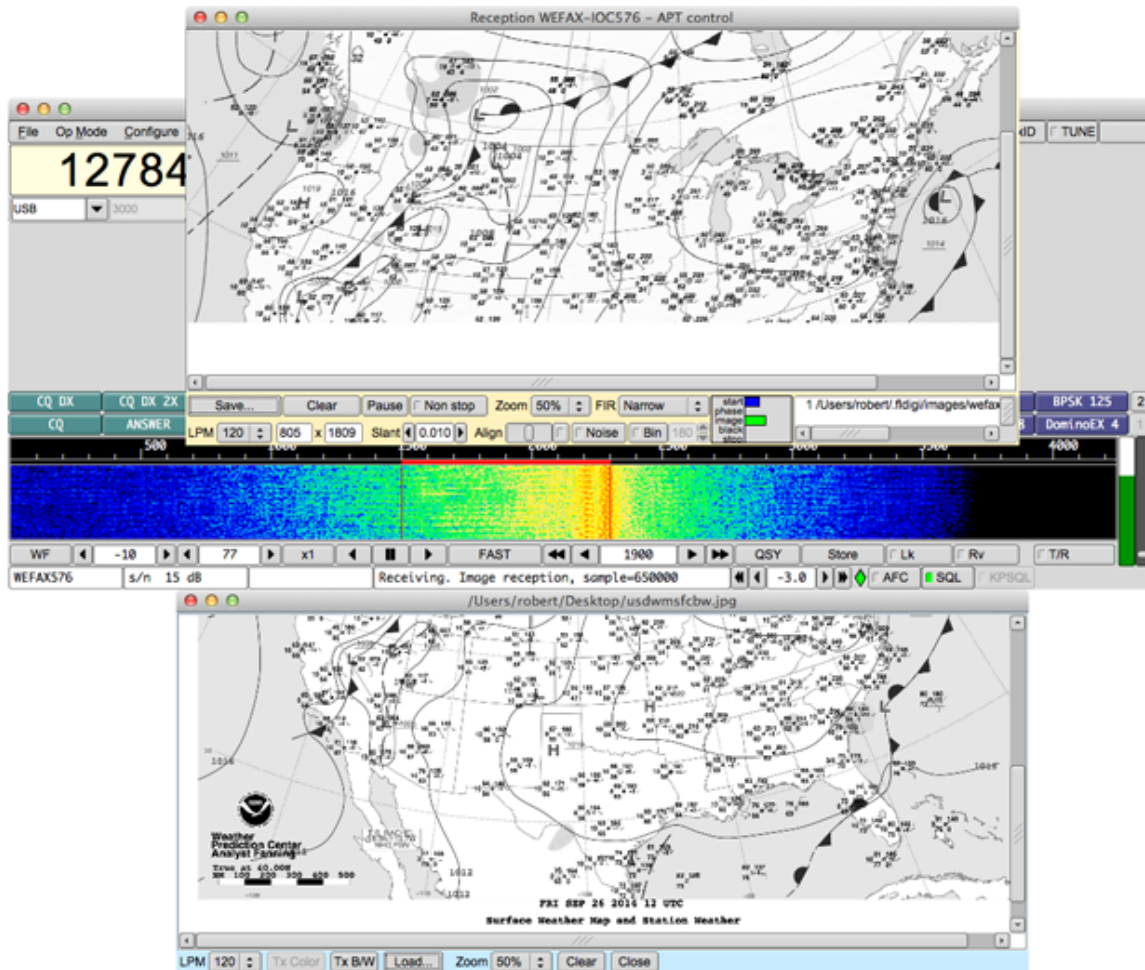


Figure 2.91: Detached WEFAX Window

#### 2.27.1.2 Hide transmission window.

This option, allows to closed by default, the transmission window, when entering Wefax mode. In embedded mode, this means that the entire window is used for reception (Most common mode). In floating mode, this implies that the transmission window is not opened by default. However, it is always possible to manually open or close the transmission window at any time.

#### 2.27.1.3 Logging messages to ADIF file

Each time an image is saved, it is possible to log this event, with the frequency and reception time, to the Adif file. This option is disabled by default.

#### 2.27.1.4 Frequency shift adjustment

The default frequency is 800 Hz. However, it is possible to adjust for example to 850 Hz for Deutsche Wetter Dienst.

### 2.27.1.5 Maximum rows number

In non-continuous (Non non-stop) reception mode, an image is automatically saved when it has more than this number of lines (Default 2500 lines). Once this number of rows is reached, the image is saved and a new image is read with the same parameters. This feature has two applications:

- In automatic mode (APT control), if an image end is not detected, we can guarantee that the result will take no more than, for example, the size of two faxes. Typical faxes have about 1300 lines, so the max lines parameters can be tuned to, for example, 200 lines.
- In manual mode, where images are read continuously, this cuts the received images into chunks of equal size.

### 2.27.1.6 Destination directory for saved images

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.

### 2.27.1.7 Monochrome images

Fax images are monochromes and are saved as such by default. However, it is possible to bypass this parameter and save them as color RGB images.

## 2.27.2 Transmitting an Image

To open the transmit window, you must of course select one of the two Wefax modems, and then right-click on the transmit (blue) window:

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:

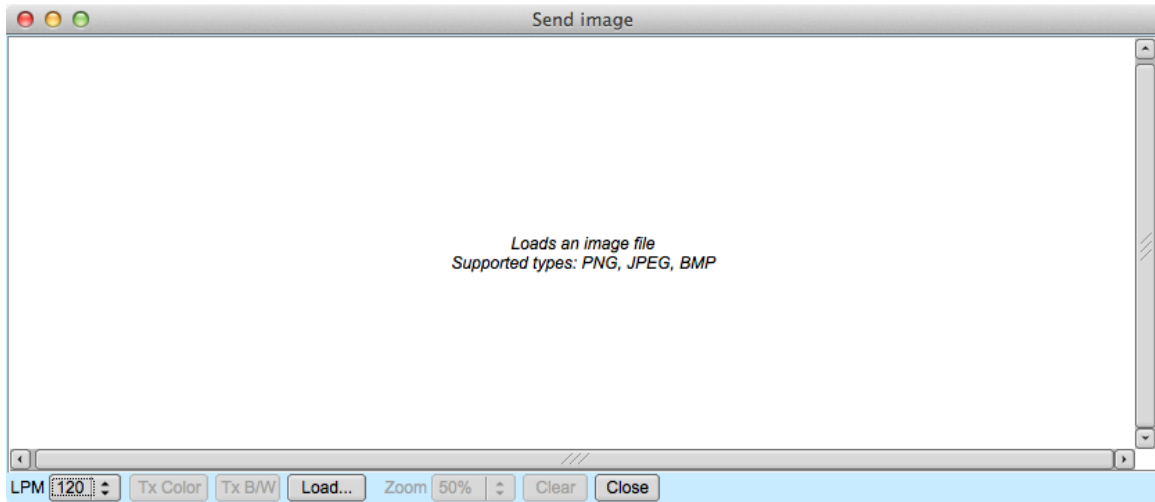


Figure 2.92: Transmitting

Now, to start the transmission, you just need to click "Tx B/W" 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 FIDigi status line displays the estimation transmit time, and the current stage (Start, phasing etc...); Color transmission ('Tx Color') is intentionally disabled at the moment.

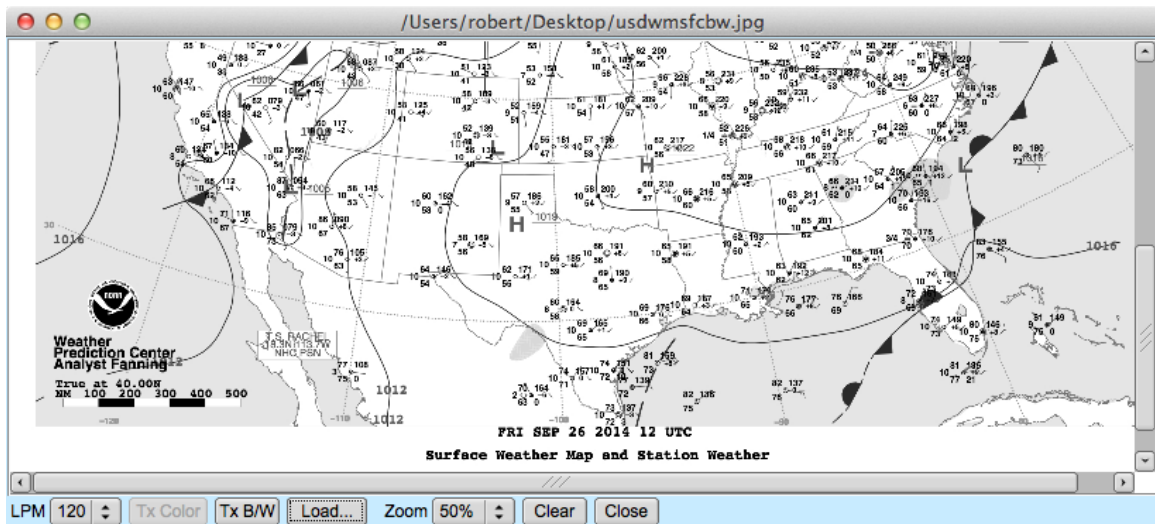


Figure 2.93: Sending

### 2.27.3 Receiving an image

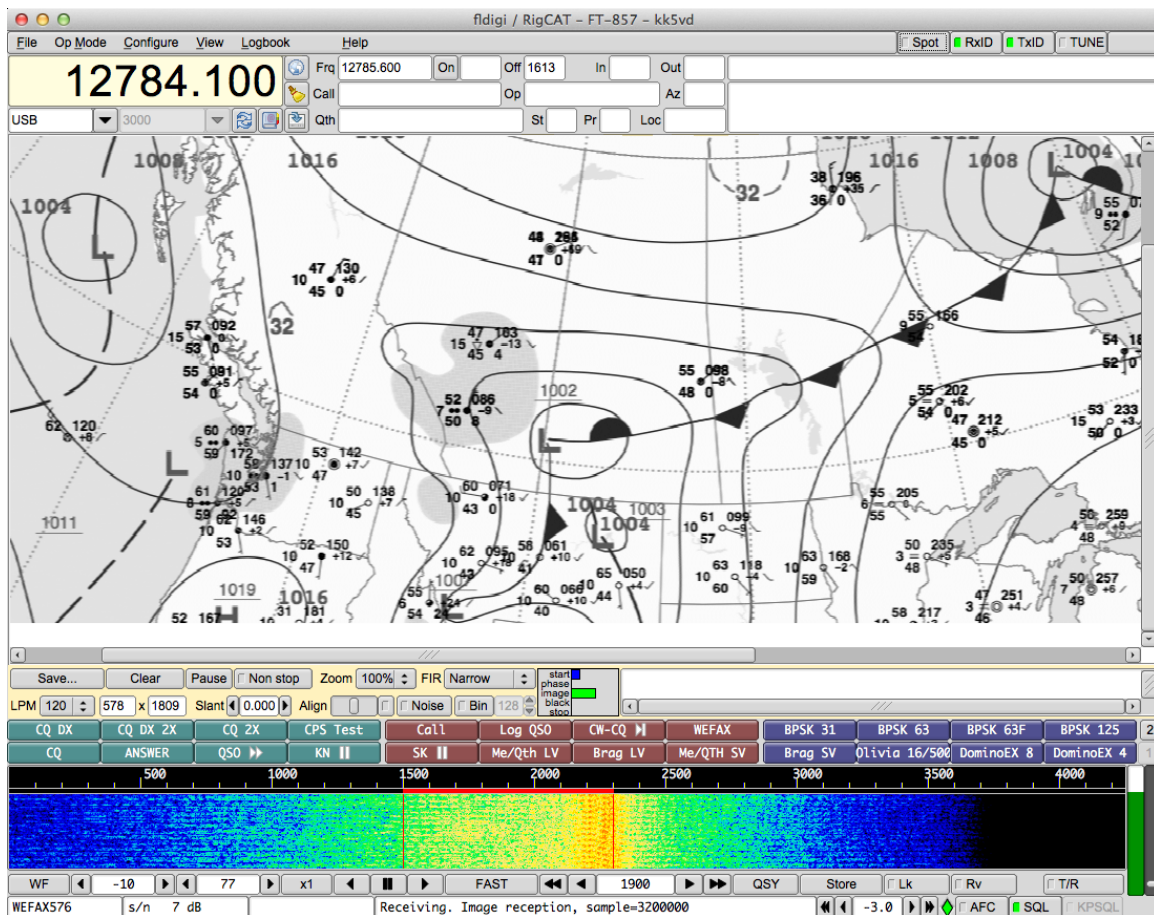


Figure 2.94: Reception

To enter reception mode, one can 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.
- Non-stop: 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:

- 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.

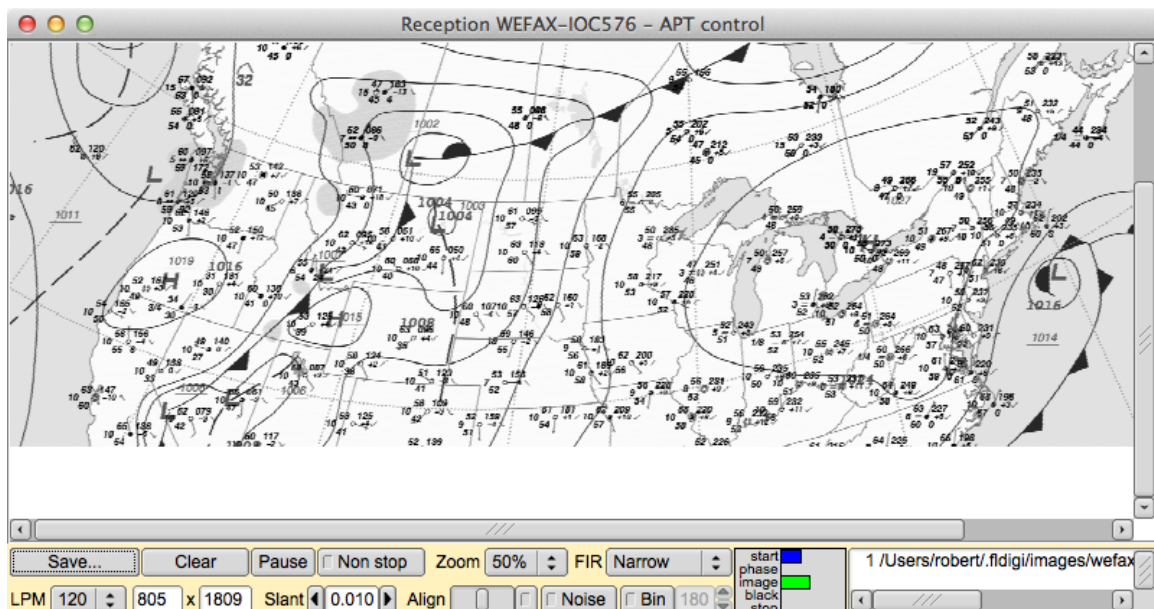


Figure 2.95: Received WEFAX File

### 2.27.3.1 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.

### 2.27.3.2 Manual reception mode (Non-Stop)

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.

### 2.27.3.3 Input FIR filters

There are three input Finite Impulse Response filters available. Here are their frequency characteristics:

Narrow filter, the default one, give the better results.

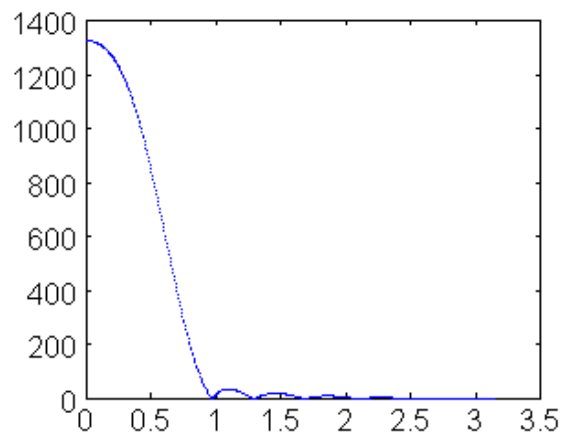


Figure 2.96: Narrow filter response

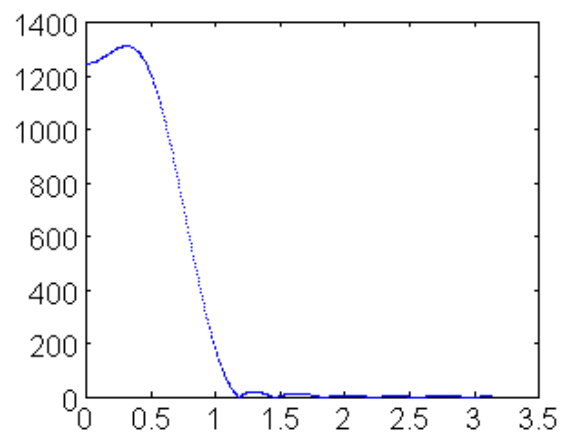


Figure 2.97: Middle filter response

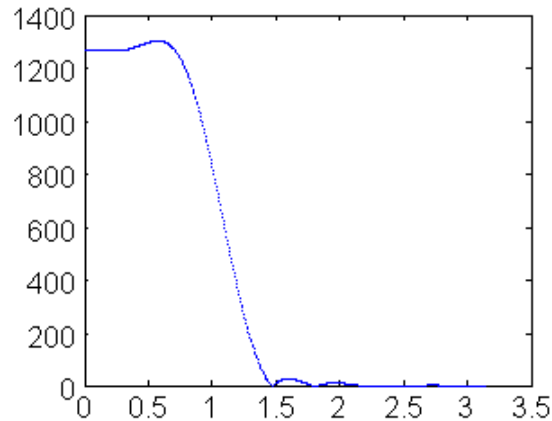


Figure 2.98: Wide filter response

#### 2.27.3.4 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.

#### 2.27.3.5 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.

#### 2.27.3.6 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 derivatives 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 contrasted details.

#### 2.27.3.7 Image detection based on signal power

The APT control - inherited from the Hamfax 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 traditionally method does not detect anything.

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

#### 2.27.3.8 AFC: Automatic Frequency Control

This option controls the frequency on the complete spectrum width. After several hunderads of lines have been correctly loaded (That is, with a high line-to-line correlation), the AFC locks until the frequency or mode is manually changed.

#### 2.27.3.9 Noise elimination

This option eliminates short-lived noise, individual pixels with a different value from their horizontal neighbors. They are modified using median values. This is based on the fact that no line should be narrower than two pixels, otherwise such an image would not be broadcast, because not correctly readable. It is therefore impossible to have one single pixel, simultaneously very different from its left and right neighbors.

#### 2.27.3.10 Binary images

Faxes can be stored as binary or grey level images. The cutoff level between black and white (Default 128) can be adjusted. No information is lost until the image is saved, therefore this level can be freely modified until image end. The purpose of this option is to save disk space.

#### 2.27.3.11 Displaying received files

Each time the end of an image is detected, an image file is created and its name appears on the file list of the reception window. By clicking on a file name, it is displayed in the transmit window.

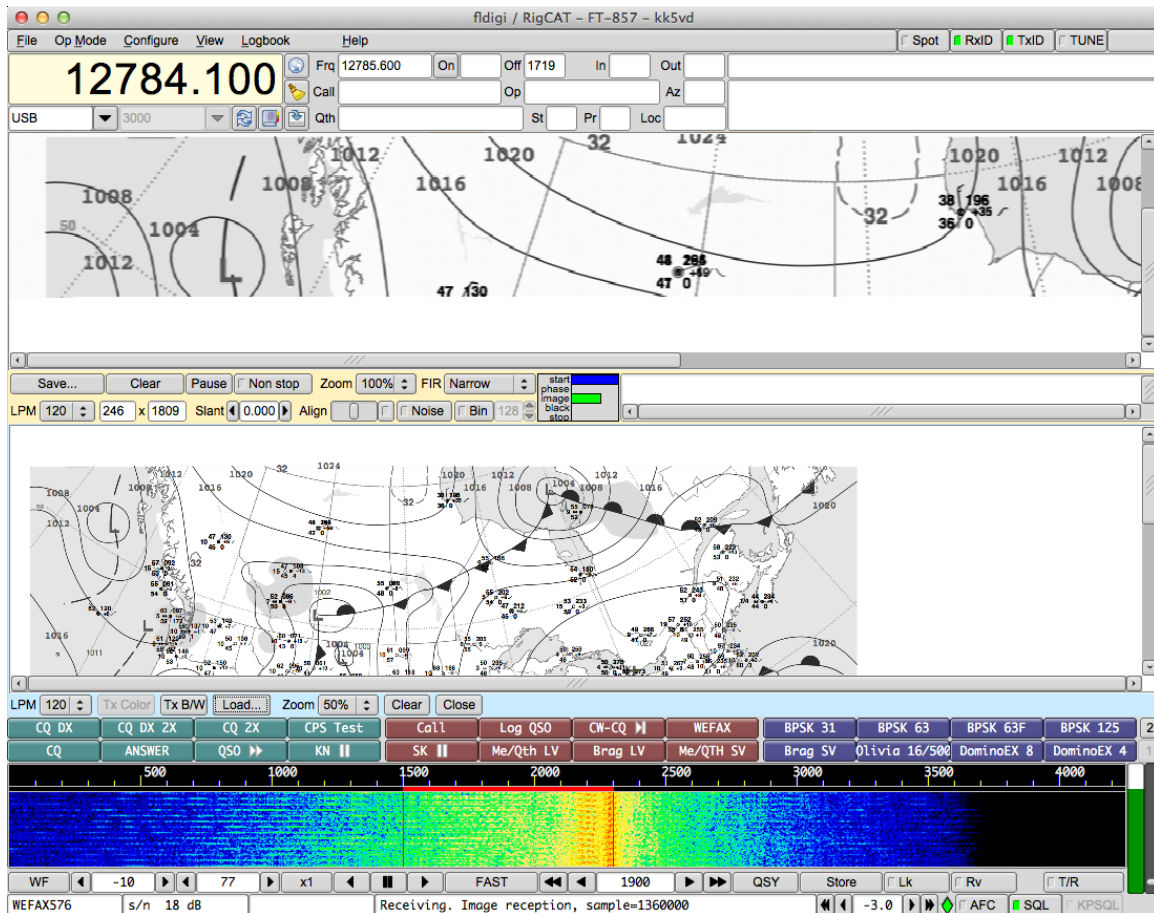


Figure 2.99: Received Image

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.28 Working Logs

### 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

Everytime you start or stop fldigi that event is recorded in a daily log file. The daily log is named 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 time stamped. 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): o ur 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"

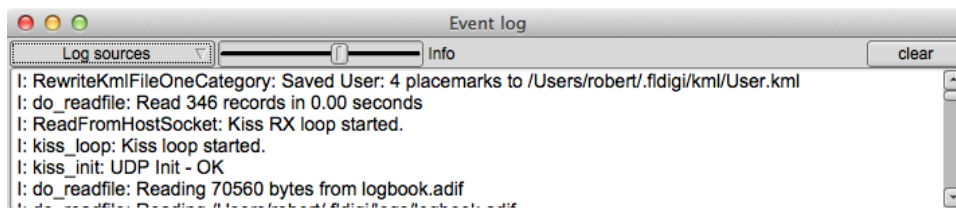


Figure 2.100: Event Log

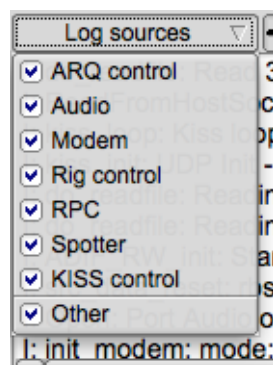


Figure 2.101: Event Log Menu

There are six levels of event logging with increasing depth of reports:

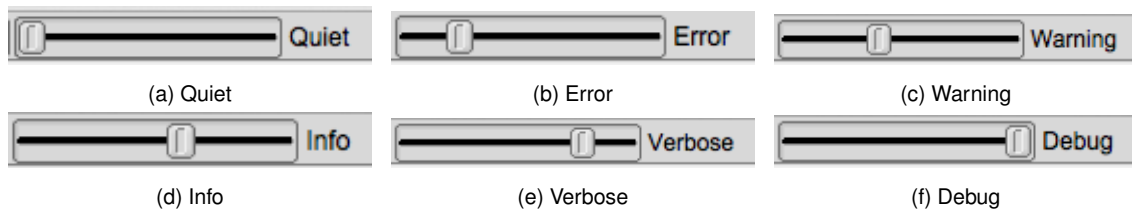


Figure 2.102: Event Logging Levels

The default level for logging events is "warning."

The event log show 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 psk reporter 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.

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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.29 Contestia Configuration

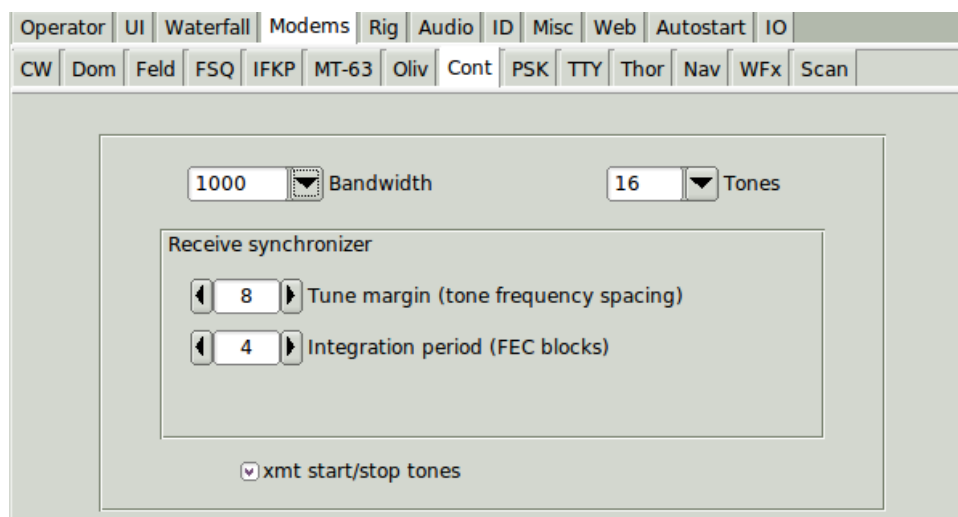


Figure 2.103: Contestia Configuration

Configuration of Contestia is similar to Olivia as Contestia is a derivative of Olivia.

See [Contestia](#) for additional information.

[Return to Top of Page](#)  
[Return to Main Page](#)

## 2.30 CW Configuration

The image shows the 'General' tab of the CW Configuration dialog. The 'Receive' section has the following settings:  SOM decoding, Lower: 0.40, Upper: 0.60,  Matched Filter, Filter bandwidth: 150,  FFT filter,  Tracking, Rx WPM: 0, Range, WPM: 10. The 'Transmit' section has: TX WPM: 18, Default: 24, Lower limit: 5, Upper limit: 50,  Use Farnsworth timing, F-WPM: 18.

Figure 2.104: General CW Configuration

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 announced on the status bar in the two boxes next to the mode indicator.

The CW signals are converted to a baseband signal. It is the digital equivalent of tuning an analog USB transceiver so that the carrier is exactly at the CW carrier frequency. The CW decoder can use one of two different DSP filters. The Fast Fourier Transform (FFT) filter is implemented with a  $\sin(x)/x$  impulse response. This is a very steep sided low pass filter. Unchecking the FFT filter causes the decoder to use a Finite Impulse Response (FIR) filter with a cutoff slope that is not as steep. The FFT filter is optimum when receiving CW in a white noise environment. The FIR filter may give better response with impulse noise. Selected a "Matched" filter for either the FFT or FIR implementation optimizes the filter bandwidth for white noise suppression. A lot of impulse noise (static) can cause either filter to ring and increasing the filter bandwidth might improve detection in that electrical environment.

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. When tracking is enabled the tracking filter is reset every time the transmit WPM is adjusted.

CW detection is basically an amplitude demodulator. You can set the threshold for detecting when the signal transitions from off-to-on and on-to-off. These signal levels are relative to the average signal level. The on-to-off is the "Lower" value and the off-to-on the "Upper." This implementation provides a hysteresis detector. Early fldigi CW decoders used this scheme but the detection levels were not adjustable.

The RxWPM control is an indicator and is not used for setting the operation of the CW decoder.

"SOM decoding" provides a fuzzy logic implementation to match the RX stream detected on-off sequence to a "best fit" character. It can increase the probability of correctly identifying the text character under very noisy conditions.

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 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.

### 2.30.1 Timing

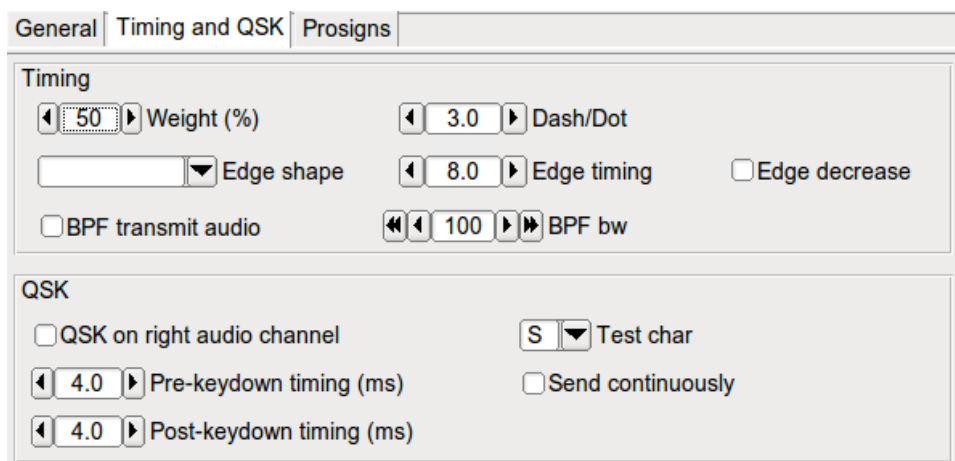


Figure 2.105: Timing

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 knowledgeable 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 adusted 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.
- The transmitted signal can be further wave shaped using a "Windowed-sinc" bandpass filter. It is enabled by checking the "BPF transmit audio" check box. The bandwidth of the filter can be adjusted from 10 to 1000 Hertz with the BPF bw control. The filter will always be centered on the AFCW injection frequency (waterfall TX cursor position). You should both visually observe and listen to the resulting CW signal. The easiest way to do this is to enable the "Send continuously", select a test character and then press the TX button on the main dialog. The weight, dash/dot, edge shape selection, edge timing, BPF and bandwidth can be changed with immediate effect.

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.

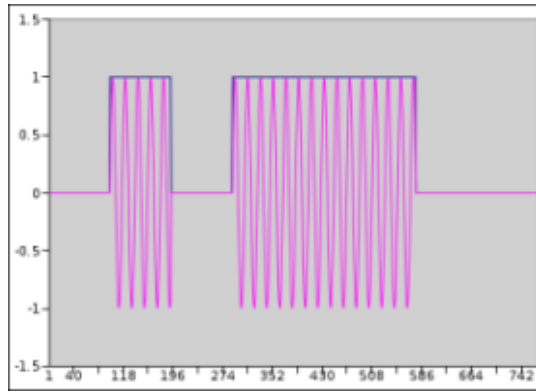


Figure 2.106: Dash/Dot = 3.0, Edge = 0.0

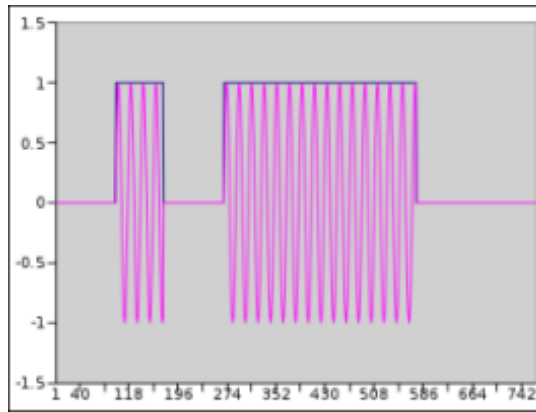


Figure 2.107: Dash/Dot = 4.0, Edge = 0.0

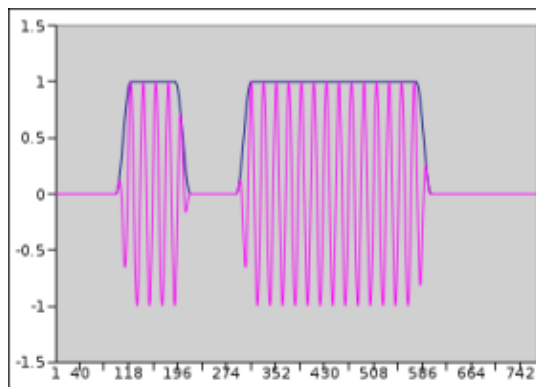


Figure 2.108: Dash/Dot = 3.0, Edge = 3 msec



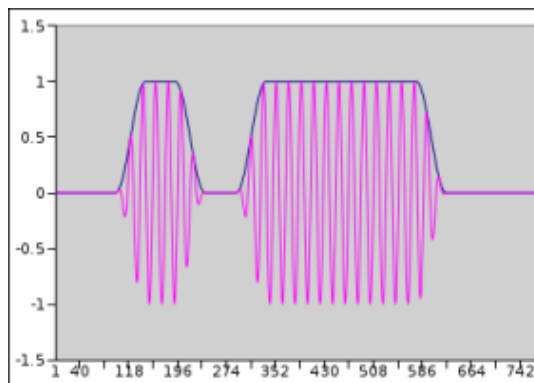


Figure 2.109: Dash/Dot = 3.0, Edge = 6 msec

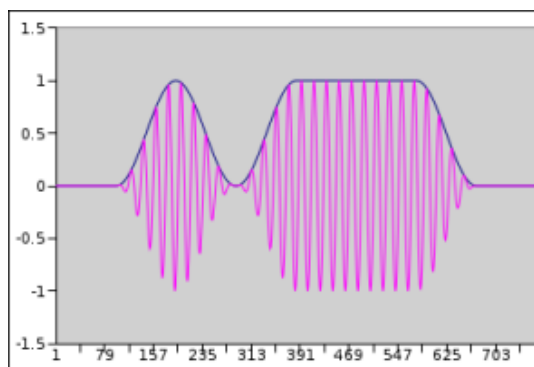


Figure 2.110: Dash/Dot = 3.0, Edge = 12 msec

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.

### 2.30.2 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.

## 2.30.3 PROSIGNS

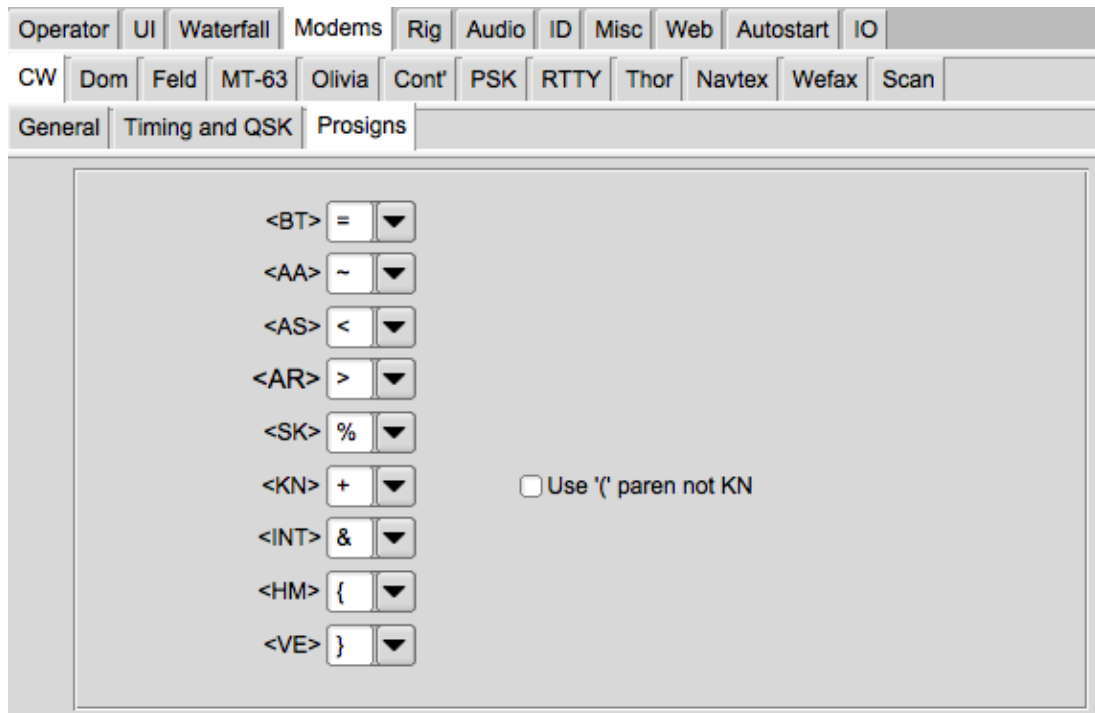


Figure 2.111: CW Prosigns

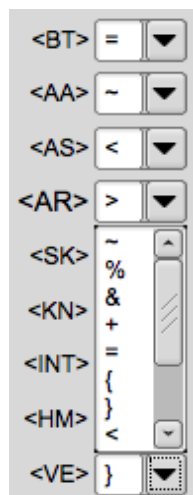


Figure 2.112: Prosign Popup

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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.31 DominoEX Configuration

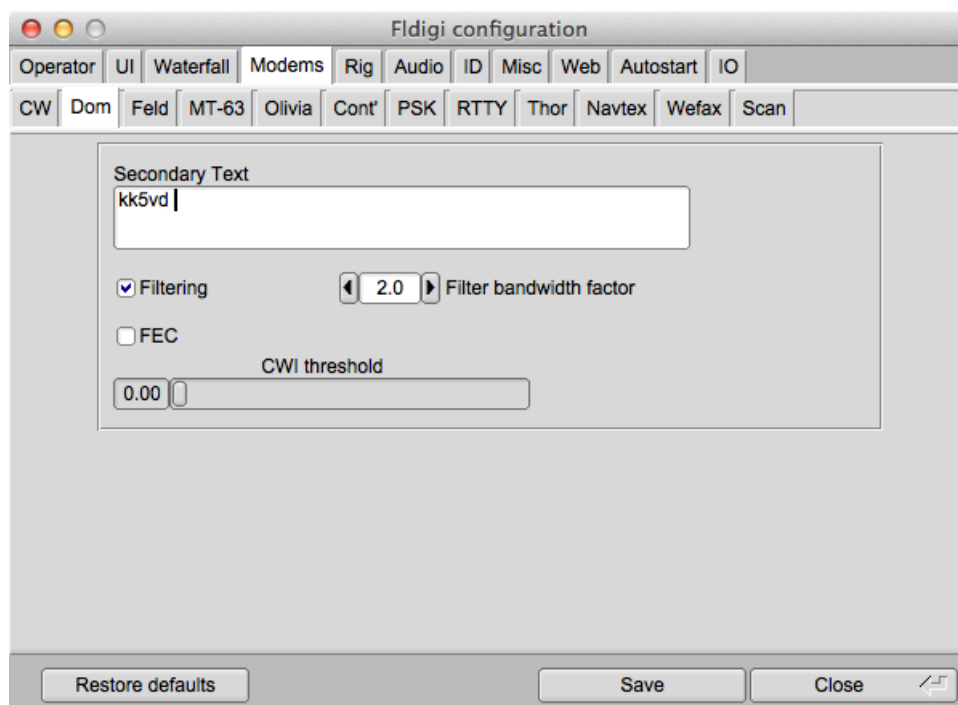


Figure 2.113: DominoEX Configuration

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.

DominoEX operations are described in [DominoEX](#) .

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.32 Feld Hell Configuration

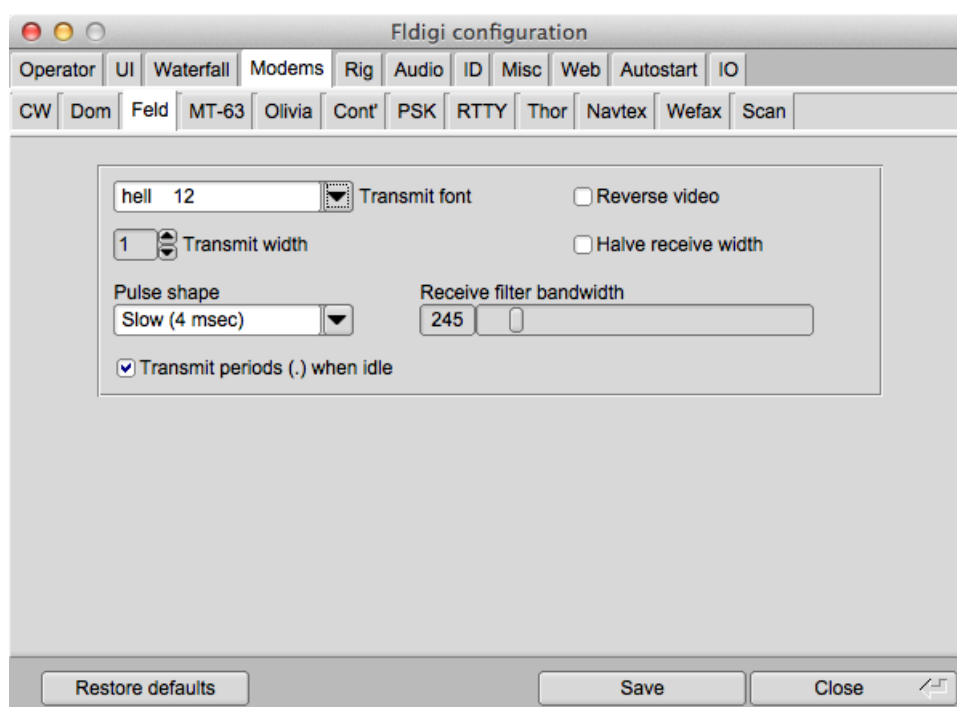


Figure 2.114: Feld Hell Config Panel

The Hellschreiber modes all use a video display that is basically a character-by-character facsimile. The shape of the characters is determined at the transmitting station. You can select from 15 different fonts, all of which have been designed for Feld Hell use. In normal use each dot of the character font is transmitted twice. You can increase the effective video s/n by transmitting the dots 2 or 3 times the normal. That is controlled by the Transmit width.

FeldHell is a pulse amplitude modulated signal. The shape of the pulse is a raised cosine. This helps to control the bandwidth of the transmitted signal. It is customary to use a 4 millisecond risetime for the raised cosine, especially on HF. You can change that to 2 milliseconds. The video edges will be sharper, but the bandwidth twice as large. You might find 2 msec a better choice for VHF and above.

Fldigi provides three controls for the receive function. The video is normally black on white. You can change that to white on black. You can also compress the horizontal scan rate of the video. You might find this effective in

displaying received fonts that are broad (or if the other end is using a multiple dot transmit width).

As you change Hellschreiber modes the optimum filter bandwidth will be set. You can change that and might find a narrower filter effective if you are experiencing CWI interference. FeldHell is susceptible to CWI. If you narrow the filter the received video will become blurred.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.33 FSQ Configuration

Operator	UI	Waterfall	Modems	Rig	Audio	ID	Misc	Web	Autostart	IO		
CW	Dom	Feld	FSQ	MT-63	Olivia	Cont'	PSK	RTTY	Thor	Nav	WFax	Scan
<b>Rx Parameters</b> Min Hits: <input type="text" value="3"/> <input type="range" value="3"/> <input type="text" value="Min Hits"/> MovAvg: <input type="text" value="1"/> <input type="range" value="1"/> <input type="text" value="MovAvg"/> Image LPF: <input type="text" value="500 Hz"/> <input type="button" value="v"/> Heard aging: <input type="text" value="Never"/> <input type="button" value="v"/>												
<b>Tx Parameters</b> <input type="radio"/> 2 baud <input type="radio"/> 3 baud <input checked="" type="radio"/> 4.5 baud <input type="radio"/> 6 baud Sounder: <input type="text" value="OFF"/> <input type="button" value="v"/> QTC: <input type="text" value="cq cq cq de w1hkj"/>												
<b>Logging</b> Audit log: <input type="text" value="fsq_audit_log.txt"/> <input type="button" value="Enable"/> <input type="button" value="Select"/> Heard log: <input type="text" value="fsq_heard_log.txt"/> <input type="button" value="Enable"/> <input type="button" value="Select"/>												

### 2.33.1 Rx Parameters

The FSQ decoder is essential a set of single frequency filters spaced by the interval  $\text{samplerate} / \text{symbol\_length}$ , 2.93 Hz. The filters are achieved using a Fast Fourier Transform. Each spectral line (or bin) of the FFT is then filtered using a time domain filter called a moving average (boxcar) filter. The moving average filter is the optimum filter for locating a signal transition in the time domain. The moving average filter can be adjusted to average 1 to 4 samples. A setting of 1 disables the filter. The default value for the moving average filter is 3.

The filtered spectral lines are then tested for maximum value. The Min Hits adjustment sets the number of consecutive times a spectral line is the maximum before being designated as a detected signal. Min Hits can be set over the range 3 to 6. The default value is 3.

Images are detected by first converting the signal to a base band. This is accomplished by mixing a 1500 Hz sine wave with the rx stream. This is equivalent to an analog receiver that using direct conversion. The down converted signal can be filtered to reduce the effect of noise. Your choice is NONE, 300, 400 or 500 Hz for the low pass filter 6 dB cutoff frequency. Use the narrower filters for noisy conditions, but remember that filtering will both reduce the image noise and also blur the image.

Callsigns that satisfy the parsing requirements will be added to the FSQ heard list. You can elect to have the heard list periodically managed so that older entries are removed or aged. The user can elect no aging, Never; or 1, 5 10,

20 minutes intervals for aging.

### 2.33.2 Tx Parameters

FSQ transmissions can occur at 2, 3, 4.5, or 6 baud. The default rate is 4.5. The baud rate can be selected on the fsq configuration tab, or by left clicking on the left most entry on the fldigi main dialog status bar.

FSQ "Sounding" is a periodic transmission of your call sign followed by a colon. The Sounding signal is a way of announcing your presence on the channel with the smallest possible data stream. Other FSQ stations will add you to their Heard list if they detect your Sounding signal. You set the Sounding interval to OFF, 1, 10 or 30 minutes.

Your QTC text can be sent using the "QTC" button on the main dialog. It can also be requested for auto reply by another FSQ station.

All FSQ transmissions are designed to be inhibited so long as the software detects a received signal (open squelch).

### 2.33.3 Logging

Any of the text in the Rx Panel or the Monitor Panel can be captured for copy, and saving to a file. Simply right click on the panel for a pop menu.

Fldigi's FSQ also saves a continuous log of all Heard stations, and an Audit log that includes all received data. These logs are named fsq\_heard\_log.txt and fsq\_audit\_log.txt respectively. The logs are never cleared unless the operator does so using the operating system file handling facilities. Data is appended to each log as it occurs. These logs are located in the fldigi temp folder:

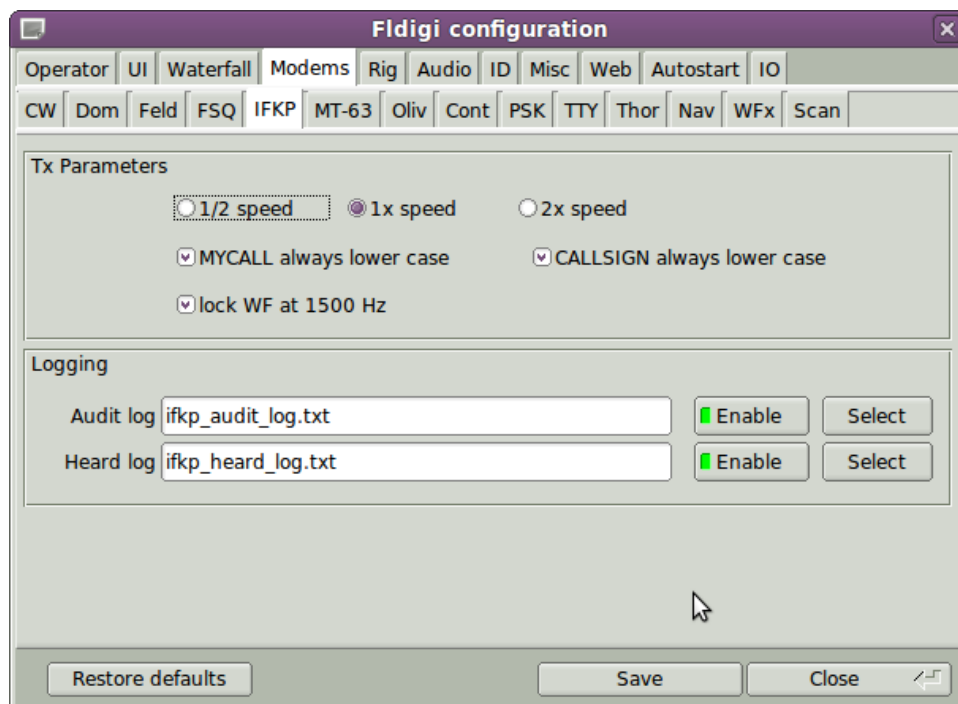
XP	C:\Documents and Settings\ <username>\fldigi.↵ files\temp</username>
Vista	C:\Users\ <username>\fldigi.↵ files\temp</username>
Win7	C:\Users\ <username>\fldigi.↵ files\temp</username>
Win8	C:\Users\ <username>\fldigi.↵ files\temp</username>
Linux	/home/<user>/.fldigi/temp
OS X	/home/<user>/.fldigi/temp
Unix	/home/<user>/.fldigi/temp

You can elect to disable the logging functions. You may also rename the log to suit your particular operations.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.34 IFKP Configuration



### 2.34.1 Tx Parameters

IFKP transmissions can occur at 0.5, 1, or 2 times the nominal baud rate. The baud rate can be selected on the IFKP configuration tab, or by left clicking on the left most entry on the fldigi main dialog status bar.

Lower case letters are transmitted within a single symbol length, while upper case letters require two symbols. You may configure IFKP to always transmit your own call sign, MYCALL, and the other call, CALLSIGN, using lower case letters.

The transmit center frequency can be locked to 1500 Hertz.

### 2.34.2 Logging

Any of the text in the Rx Panel for copy, and saving to a file. Simply right click on the panel for a pop menu.

Fldigi's IFKP can save a continuous record of all received data to an audit log. The heard list can also save a record of all stations added to the IFKP heard list. These logs are named IFKP\_audit\_log.txt and IFKP\_heard\_log.txt respectively. The logs are never cleared unless the operator does so using the operating system file handling facilities. Data is appended to each log as it occurs. These logs are located in the fldigi temp folder:

XP	C:\Documents and Settings\ <username>\fldigi.\files\temp</username>
----	---



Vista	C:\Users\ <username>\fldigi.↔ files\temp</username>
Win7	C:\Users\ <username>\fldigi.↔ files\temp</username>
Win8	C:\Users\ <username>\fldigi.↔ files\temp</username>
Linux	/home/<user>/.fldigi/temp
OS X	/home/<user>/.fldigi/temp
Unix	/home/<user>/.fldigi/temp

You can change the name of each log and elect to disable the logging functions.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.35 RSID / Video / ID Configuration

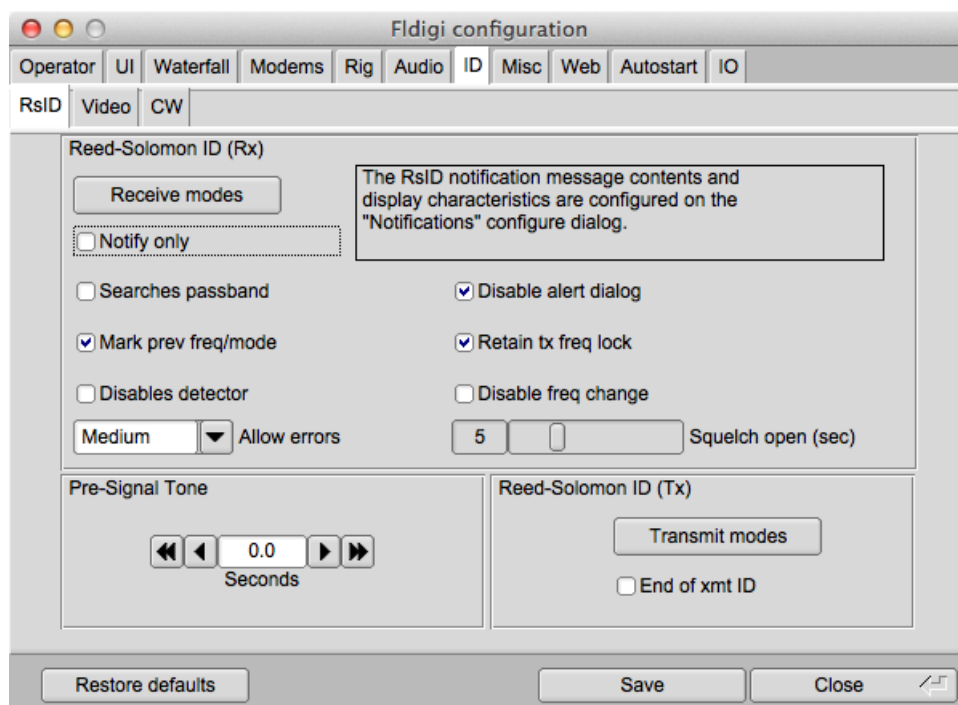


Figure 2.115: RSID Config Panel

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](#).

### 2.35.1 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 RSID 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.

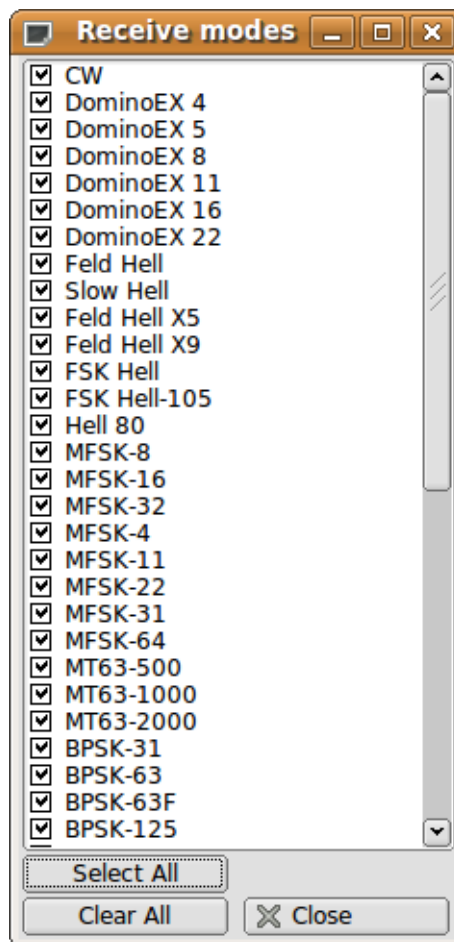


Figure 2.116: Receive Modes

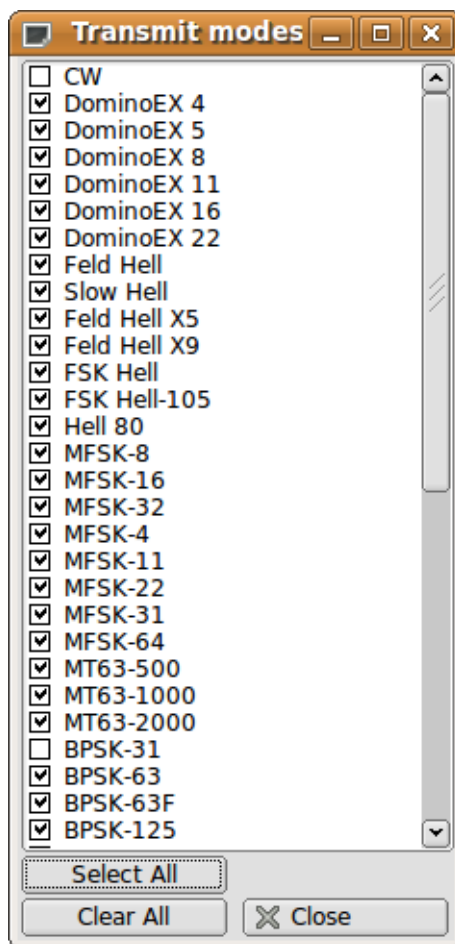


Figure 2.117: Transmit Modes

### 2.35.2 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.



Figure 2.118: Small Font



Figure 2.119: 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.

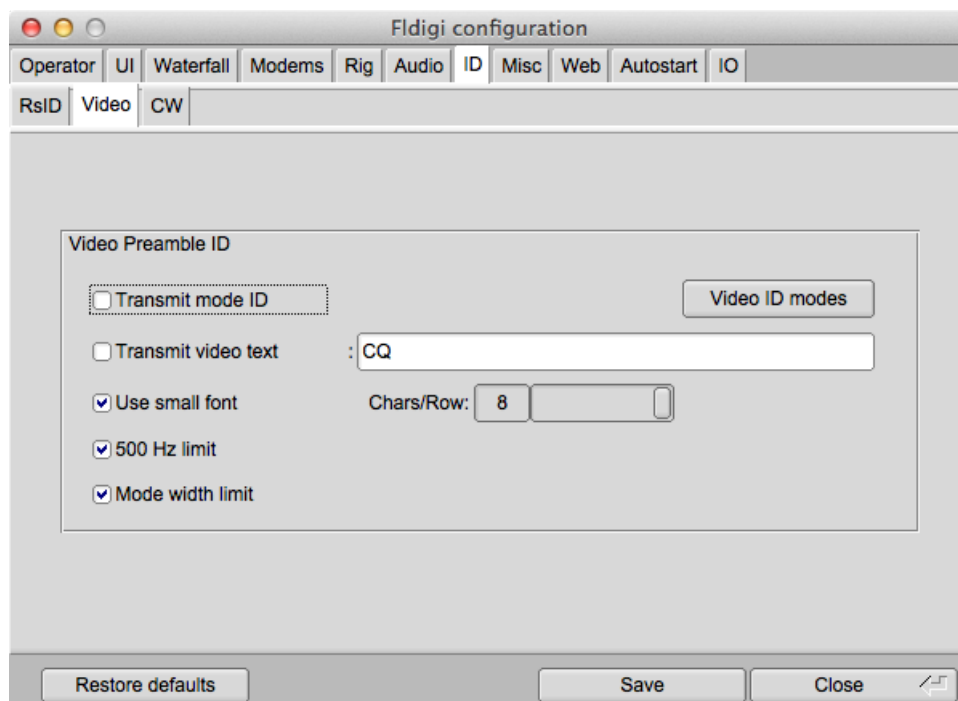


Figure 2.120: Video ID Config Panel

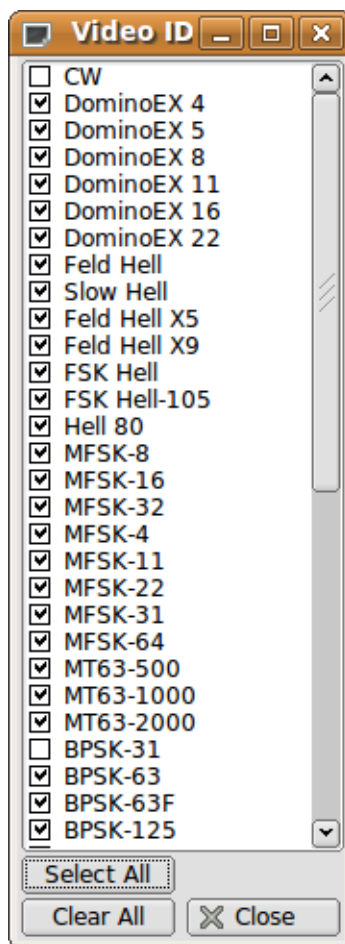


Figure 2.121: Video ID Modes

### 2.35.3 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.

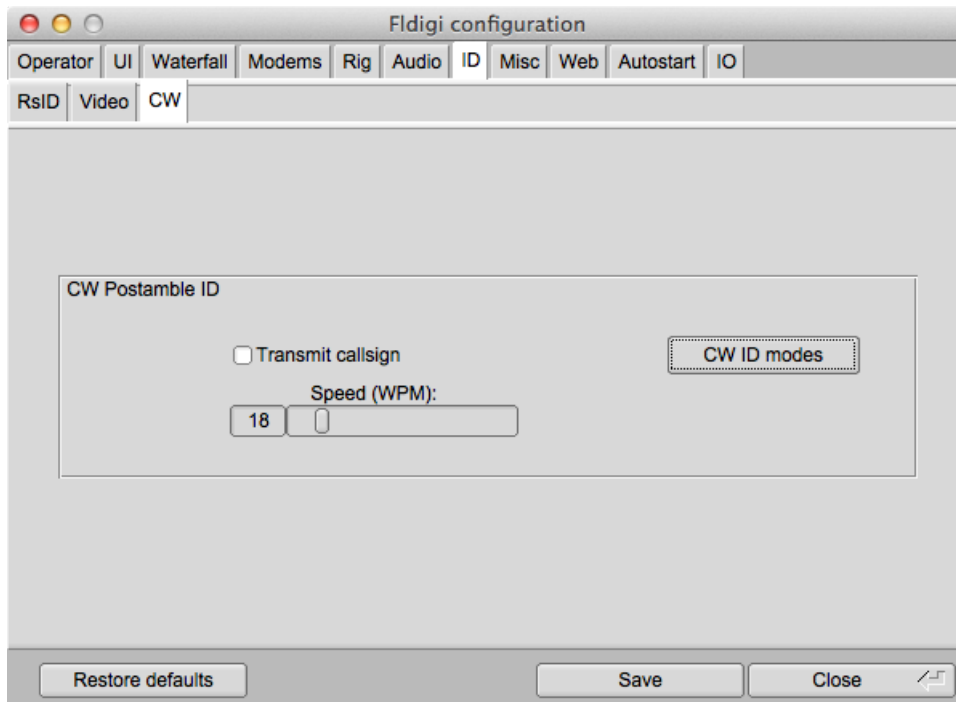


Figure 2.122: CW Postamble ID Config Panel

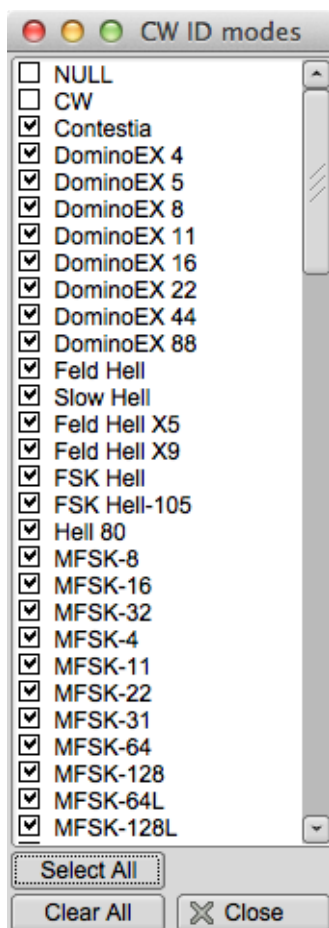


Figure 2.123: CW ID Modes

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.36 Miscellaneous Configuration

### 2.36.1 Sweet spot

Figure 2.124: Misc Sweet Spot Config

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](#).

The K3 A1A lower sideband can be selected for any transceiver that only provides LSB in the CW mode.

### 2.36.2 Callsign spotting

Figure 2.125: Callsign spotting

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.

If your CPU performance is marginal you can disable spotting when no signal browser is visible.

### 2.36.3 CPU performance

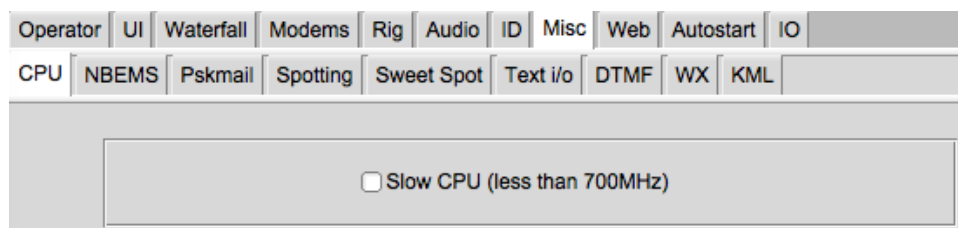


Figure 2.126: Misc CPU

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.

### 2.36.4 Text i/o

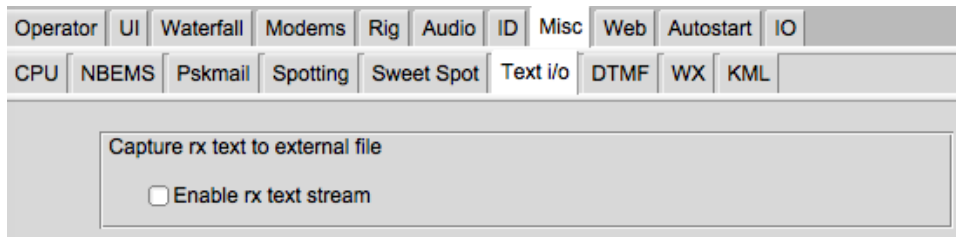


Figure 2.127: Misc Text I/O

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.

### 2.36.5 NBEMS (flmsg / flwrap) interface

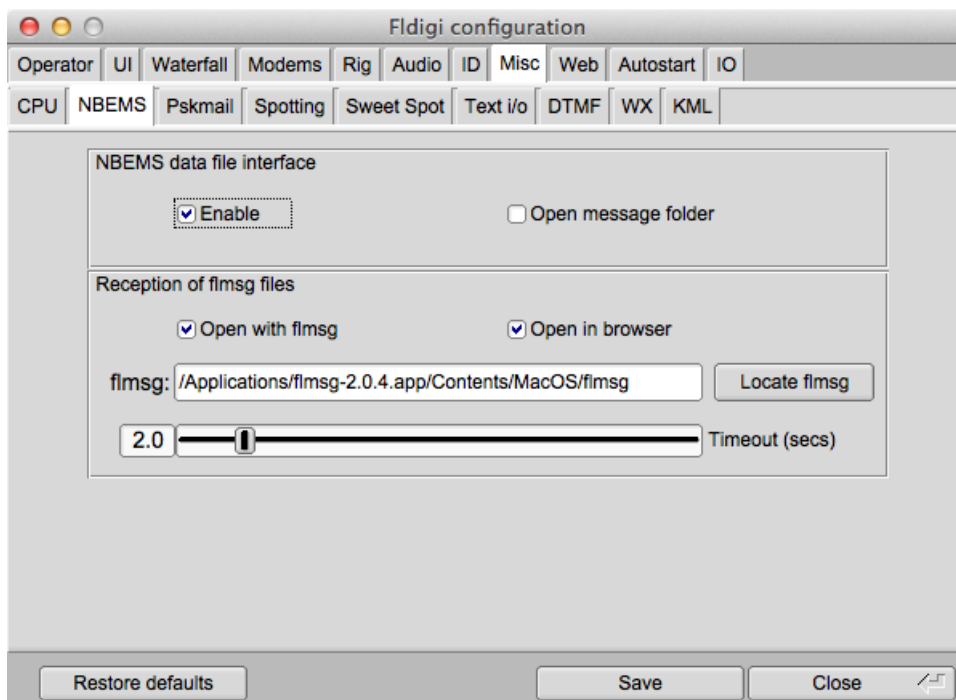


Figure 2.128: Config Misc NBEMS

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. Additional information on flarq is available here:

- `flarq help system`.
- `wrap help system`.
- `flmsg help system`

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.

### 2.36.6 DTMF decoding

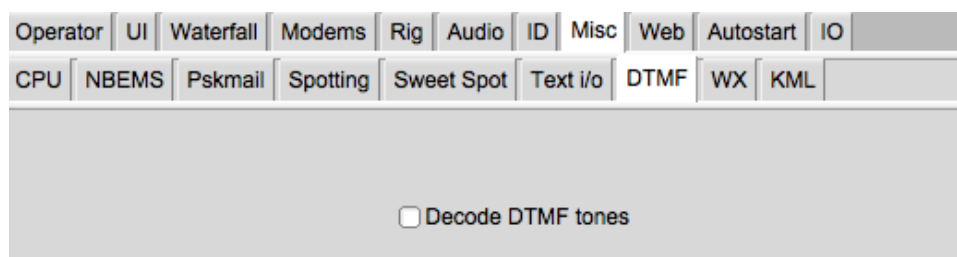


Figure 2.129: Misc DTMF Decoding

Fldigi can encode and decode DTMF tone sequences. Enable this check box to display the decoded tone sequence in the Rx panel. DTMF encoding is accomplished by a [MACRO](#) tag.

### 2.36.7 WX rss-xml queries

Figure 2.130: Misc WX

Fldigi provides an automated query of a specified RSS-XML feed to obtain and format weather data. The report is added to the transmit text stream using the appropriate [MACRO](#) tag. The report for the above configuration is:

```
Huntsville, Madison County Executive Airport, AL, United States (KMDQ) 34-51-41N 086-33-26W
Cond: overcast
Wind: 210 at 12 mph 19 kph
Temp: 62 F 17 C
Baro: 30.04 in Hg 1017 mbar
```

The full report option creates this report:

```
Huntsville, Madison County Executive Airport, AL, United States (KMDQ) 34-51-41N 086-33-26W
Feb 29, 2012 - 07:15 AM EST / 2012.02.29 1215 UTC
Wind: from the SSW (210 degrees) at 12 MPH (10 KT) gusting to 17 MPH (15 KT):0
Visibility: 10 mile(s):0
Sky conditions: overcast
Temperature: 62 F (17 C)
Dew Point: 60 F (16 C)
Relative Humidity: 93%
Pressure (altimeter): 30.04 in. Hg (1017 hPa)
```

The "Search on web" button will open your browser to Greg Thompson's [global listing](#) of METAR station.

RSS feeds for world wide airports can be searched here <http://www.airrouting.com/content/←AirportLocatorForm.aspx>.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.37 MT63 Configuration

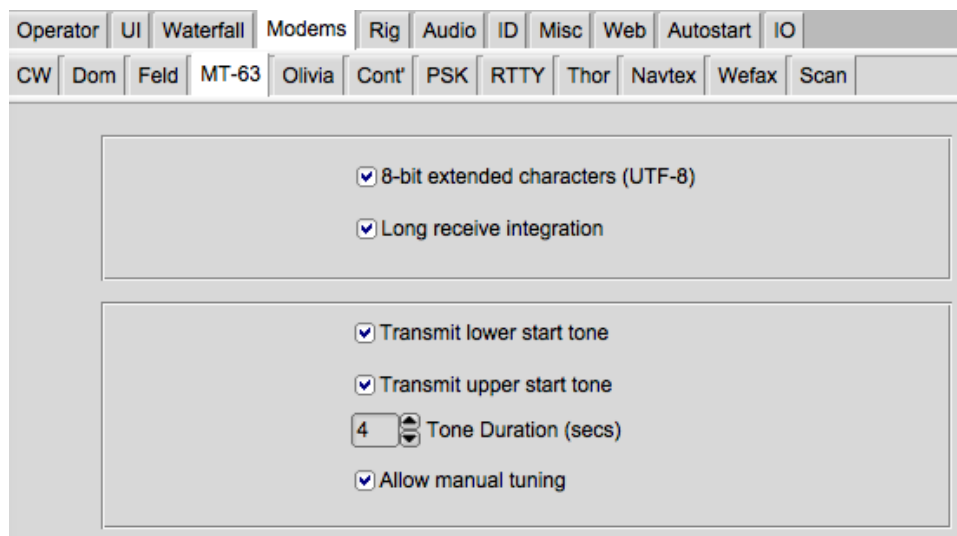


Figure 2.131: Modems MT63 Config Panel

MT63 is an orthogonal frequency division multiplexed mode consisting of 64 parallel carriers each carrying a part of the transmitted signal. There are 3 bandwidths and baudrates that fldigi implements in MT-63:

- 500 Hz - 5 baud
- 1000 Hz - 10 baud
- 2000 Hz - 20 baud

The lowest frequency transmitted is always 500 Hz. If you have a scheduled MT63 qso or are trying to copy what you think is MT63 you should tune the signal so that the lowest observable signal is at 500 Hz. Fldigi is capable of decoding signals that are mistuned by as much as +/- 100 Hz.

Selection of interleave (short/long) is made from the modem menu; MT63-500S, MT63-500L, MT63-1000S, MT63-1000L, MT63-2000S, MT63-2000L.

Interleave and 8-bit extended characters are usually agreed upon before a QSO exchange begins. The default is to use the long interleave. 8-bit extended characters allow the transmission of Latin-1 accented characters.

To assist the Rx operator you can transmit a short tone at the lowest (and highest) tone frequencies. You can set the tone duration.

You may also elect to use manual tuning (position on the waterfall) of the Tx / Rx signal. Be careful when doing this so that the signal does not extend beyond the limits of your transceiver SSB filters.

MT63 is used extensively in the Military Affiliate Radio System (MARS).

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.38 Olivia Configuration

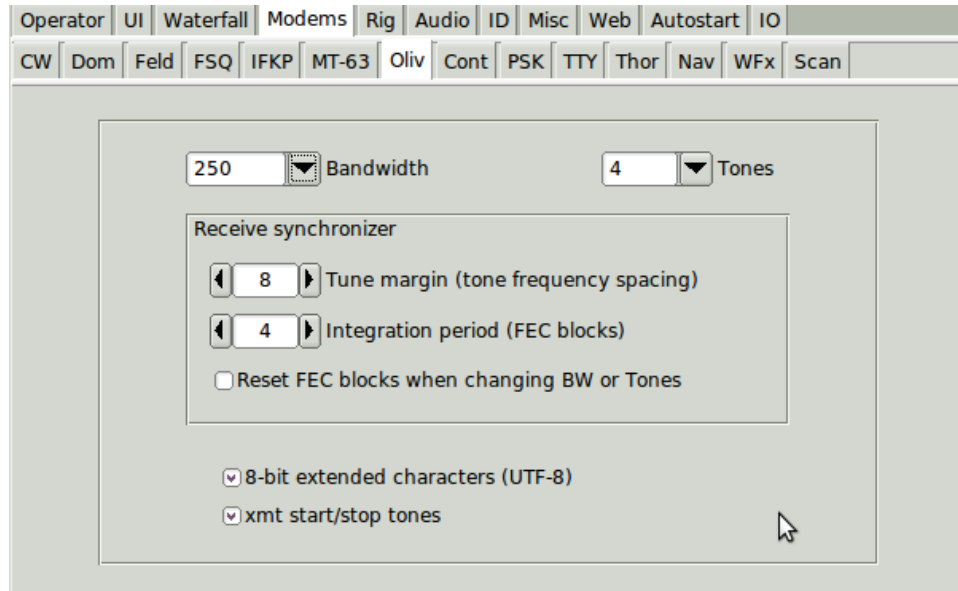


Figure 2.132: Description

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. 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.

Each Olivia transmission can start and end with two distinctive tones. These tones are exactly at the lowest and uppermost tones for the selection of tones and bandwidth. They can assist the receiving station in tuning the Olivia signal on the waterfall.

The start/stop tones can be disabled.

See [Operating Olivia](#) for additional information.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.39 PSK Configuration

The screenshot shows the 'Modems PSK General' configuration window. It is organized into four distinct sections:

- AFC behavior:** Contains two spinners. The first is labeled 'Acquisition search range (Hz)' and is set to 50. The second is labeled 'Acquisition S/N (dB)' and is set to 9.
- S/N and IMD behavior:** Contains a dropdown menu set to 'Dim' followed by the text 'after' and a spinner set to 15, with the unit 'seconds'.
- Multi-Channel Signal Processing:** Contains a text label 'Disable on very slow CPUs of if signal browser is not used' and a checked checkbox labeled 'Multi-channel detector'.
- 8 psk:** Contains an unchecked checkbox labeled 'Pilot tone' and a spinner set to -30, with the unit 'pilot power (dB)'.

Figure 2.133: Modems PSK General

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. The browser is described here:

8psk (8 phase states) does not use AFC for signal acquisition or tracking. RsID or operator adjustment of tracking point is necessary to align the decoder with the received signal. This mode includes the optional use of a pilot tone. The level of the pilot tone can be set relative to the total signal power.

This table shows the amount of transmit power used by the pilot tone

dB	-20	-26	-29	-32	-36	-40
Pwr Lvl	10%	5%	3.5%	2.5%	1.6%	1.0%

[Signal Browser](#)

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.40 RTTY / FSK Configuration

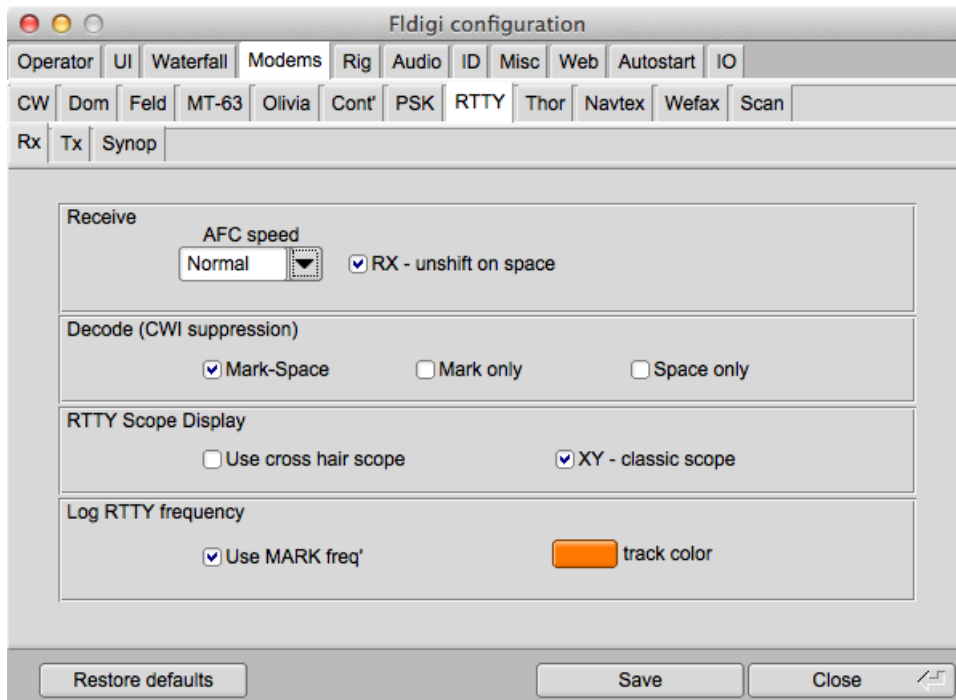


Figure 2.134: RTTY RX

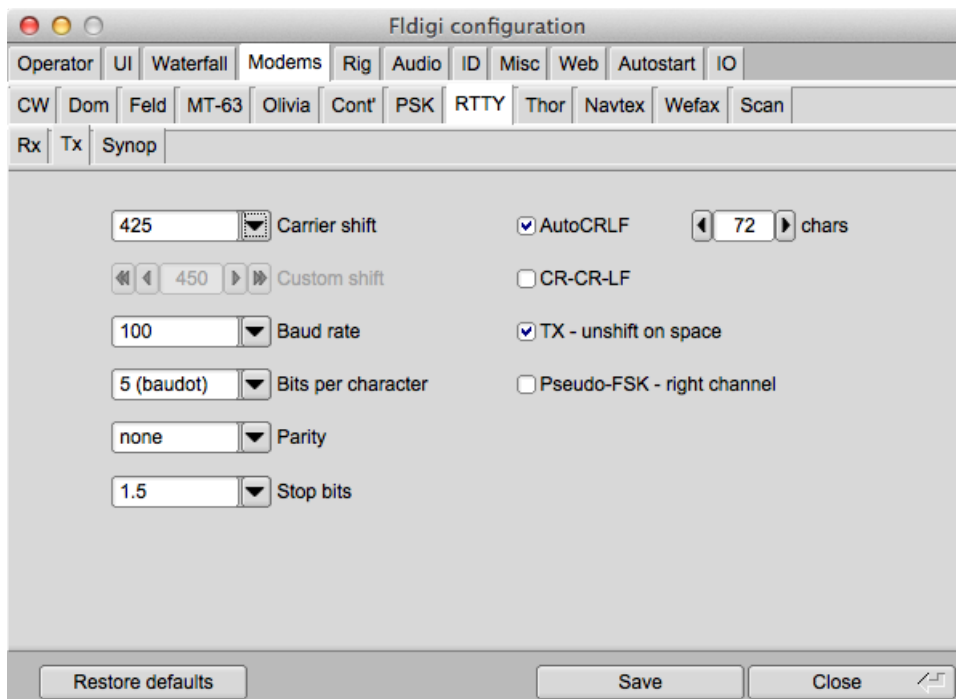


Figure 2.135: RTTY TX



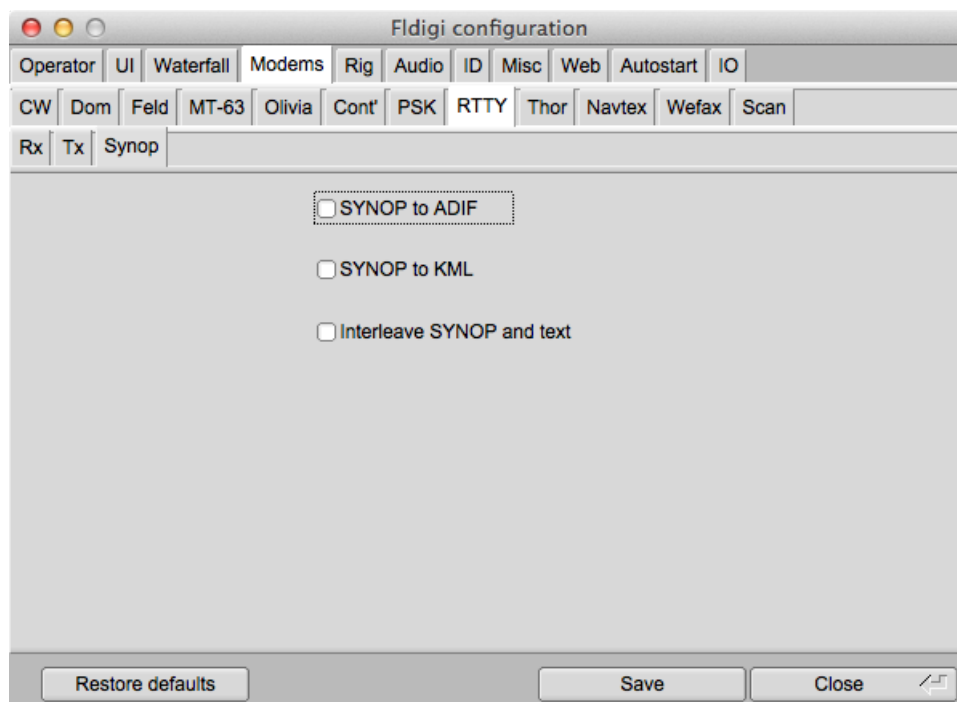


Figure 2.136: RTTY SYNOP

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 received signal processing consists of a tuned Raised Cosine Filter followed by an optimized Automatic Threshold Correcting (ATC) detector. The resulting bit stream is then processed to extract the byte data.



Figure 2.137: Classic RTTY XY Scope

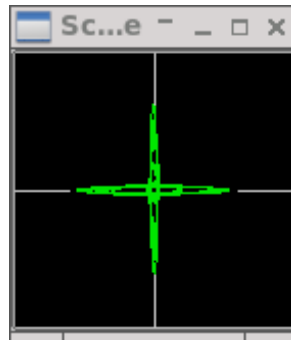


Figure 2.138: Pseudo Signal XY Scope

The Digiscope display can be defaulted to the X-scope or the Signal scope.

The classic XY-scope is similar to older hardware scopes that show the output of the separate Mark-Space filters, one applied to the horizontal and one to the vertical axis.

The Pseudo XY-scope is similar with the exception that the horizontal and vertical are phase related to the Mark-↔ Space signals, but not the actual signal.

The Classic XY-scope will appear noisier than the Pseudo XY-scope.

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. See [Pseudo FSK](#) for additional information and a suitable keying circuit.

Your log can record either the center frequency between the Mark and Space or the Mark frequency. You can also select the color that should distinguish the Mark frequency.

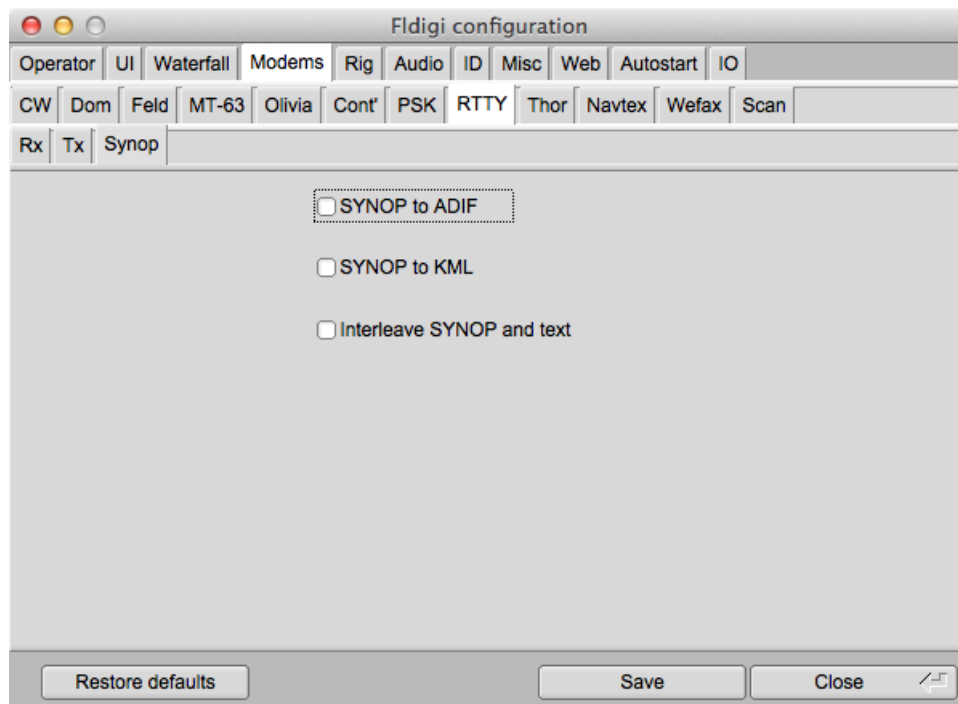


Figure 2.139: RTTY SYNOP

See [Synop](#) and [RTTY](#) for additional information.

[Return to Top of Page](#)

[Return to Main Page](#)

## 2.41 Thor Configuration

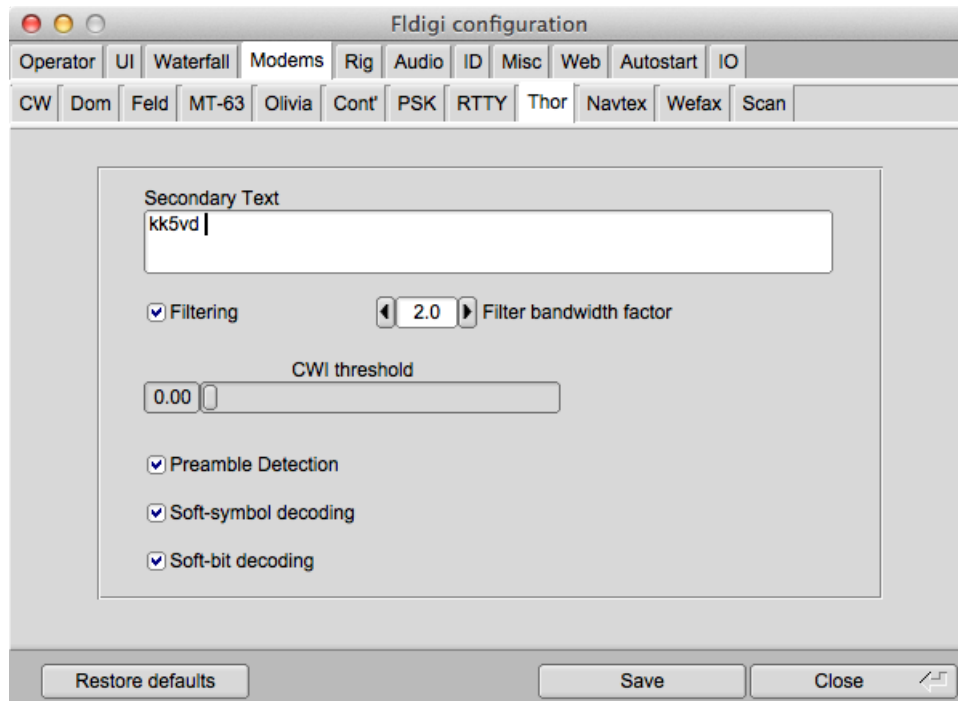


Figure 2.140: Modems Thor

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 increasing 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](#).

[Return to Top of Page](#)

[Return to Main Page](#)

## Chapter 3

# Logging

- [Cabrillo Report](#)
- [Exporting Logbook Data](#)
- [QSO Logbook](#)
- [User Interface Configuration - Logging](#)

### 3.1 Cabrillo Report

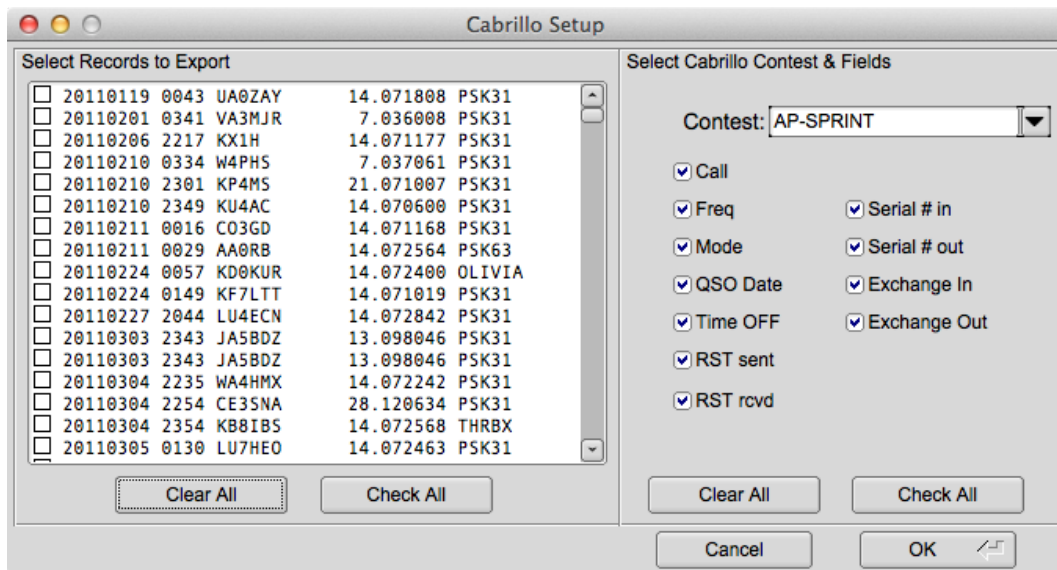


Figure 3.1: Cabrillo Report

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

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

If you want to export every record press the "Check All" in the left panel.

Select the Contest type from the pull down menu in the right panel. Fldigi knows how to format the various fields for each contest. When satisfied with the setup press OK. You will then have the opportunity to specify the location and name of the Cabrillo output file.

If you use the [Contest - How To](#) <XBEG>...<XEND> to capture your outgoing exchange, OR if you choose to capture all received exchange information in the "Exch" field on the [QSO Logbook](#) panel then you should be careful not to duplicate the data in the Cabrillo report. For example:

You captured outgoing exchange data as <XBEG><RST> <CNTR><XEND>. Do not enable the "RST sent" and "Serial # out" check boxes as this info is already contained in the "Exchange in" field.

You captured incoming exchange data into the Exch field as RST SER# TIME. Do not enable the "RST rcvd" or the "Serial # in" check boxes.

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,
# CQ-WPX-SSB, CQ-VHF, CQ-WW-CW, CQ-WW-RTTY, CQ-WW-SSB, DARC-WAEDC-CW,
# DARC-WAEDC-RTTY, DARC-WAEDC-SSB, FCG-FQP, IARU-HF, JIDX-CW, JIDX-SSB,
# NAQP-CW, NAQP-RTTY, NAQP-SSB, NA-SPRINT-CW, NA-SPRINT-SSB, NCCC-CQP,
# NEQP, OCEANIA-DX-CW, OCEANIA-DX-SSB, RDXC, RSGB-IOTA, SAC-CW, SAC-SSB,
# STEW-PERRY, TARA-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 WOSD 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:
```

[Return to Top of Page](#)

[Return to Main Page](#)

## 3.2 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 is available from the menu "File/Log/Export ADIF", "File/Log/Export Text", and "File/Log/Export CSV".

### 3.2.1 Export ADIF

Selecting the Export ADIF menu item opens the following dialog:

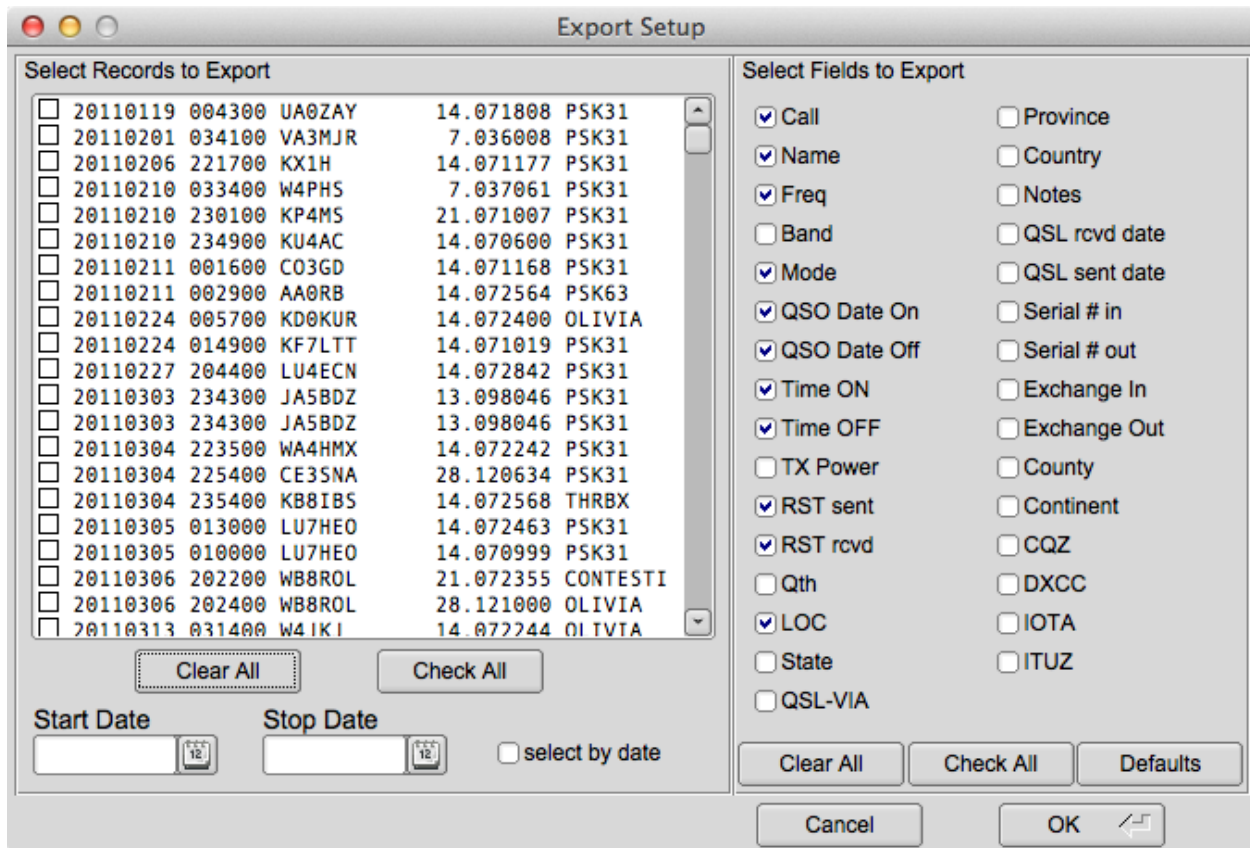


Figure 3.2: Export Setup

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.

### 3.2.2 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.

[Return to Top of Page](#)  
[Return to Main Page](#)



## 3.3 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.

### 3.3.1 List of Log Fields

The complete set of logbook fields are:

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

- - 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 word-processor and for printing. The CSV can be read into many spreadsheet programs such as Excel, Open Office or Gnumeric.

### 3.3.2 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. Please read further on using both [RST and RSQ signal reports](#).

### 3.3.3 Capturing QSO Data

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

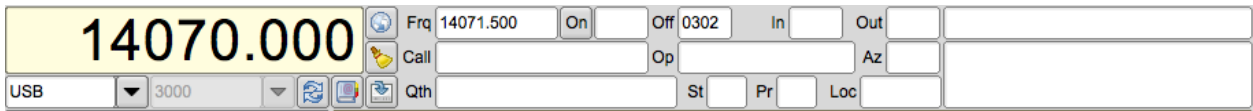


Figure 3.3: QSO capture panel

and the second for contest fields

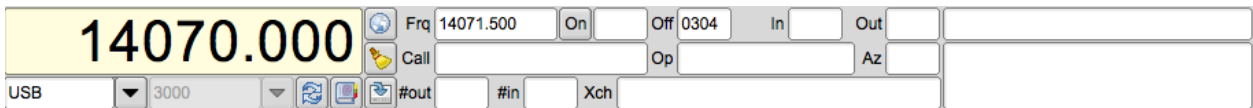


Figure 3.4: Contest capture panel

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:



Figure 3.5: Minimal QSO Capture Panel



Figure 3.6: Minimal Contest Capture Panel

These are selectable from the View menu:

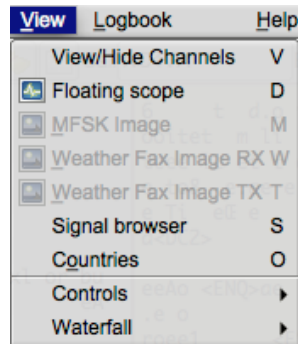


Figure 3.7: View menu

These three buttons are associated with logbook entries.

- The globe button provides access to [QRZ queries](#)
- The brush button clears all of the logging entries
- The save-into button saves the current logging entries into the logbook



Figure 3.8: Log Butons

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.

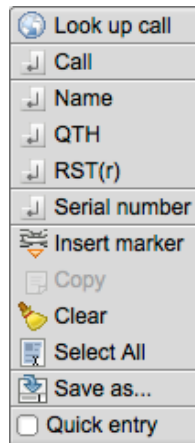


Figure 3.9: Normal Menu

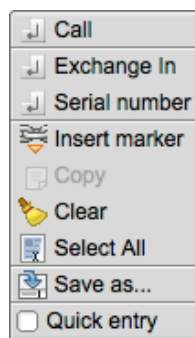


Figure 3.10: Contest Menu

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:

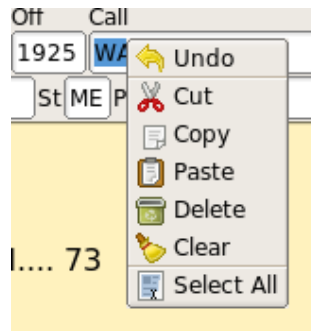


Figure 3.11: Menu Popup Edit Field

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 Logbook menu; Logbook/View. 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.

Date	Time	Callsign	Name	Frequency	Mode
20140402	14:34	RZ3DZ	Vyacheslav	28.120669	PSK31
20140404	17:41	UX2IO	ROMAN	28.121390	PSK31
20140406	01:58	WB2HVF		10.147015	MFSK32
20140412	16:47	KC1AAQ	Frank	7.037500	OLIVIA
20140412	16:47	KC1AAQ	Frank	7.037500	OLIVIA
20140510	04:40	KK5VD	Robert	10.147004	MFSK32
20140531	16:23	N7WWA	Randy	14.067000	OLIVIA
20140601	01:10	VA3VD	Dave	14.070743	PSK31
20140629	05:10	HB9HAE	Gion	10.143720	OLIVIA
20140629	05:18	HB9HAE	Gion	10.143720	OLIVIA
20140629	05:20	HB9HAE	Gion	10.143720	OLIVIA
20140823	16:37	YV4DHS	Homero	14.071813	PSK31
20140823	16:40	N0MHL	Marvin	14.072690	PSK250
20140825	01:35	W5FEC	Floyd	10.142510	PSK31
20140825	23:25	N6PDB	DENNIS	14.073184	MFSK16
20140830	18:55	WD9DUI	Gary	21.072940	OLIVIA
20140901	16:52	N1KHL	Ed	14.070846	PSK31
20140901	18:48	EG5SDC	Juan Carlos	14.073170	MFSK16
20141003	20:03	W1HKJ		14.072795	PSK63RC5

Figure 3.12: Log Book

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: [Log Exports](#).

Cabrillo reporting: [Contest Reports](#).

[Return to Top of Page](#)

[Return to Main Page](#)

### 3.4 User Interface Configuration - Logging



Figure 3.13: UI Tabs

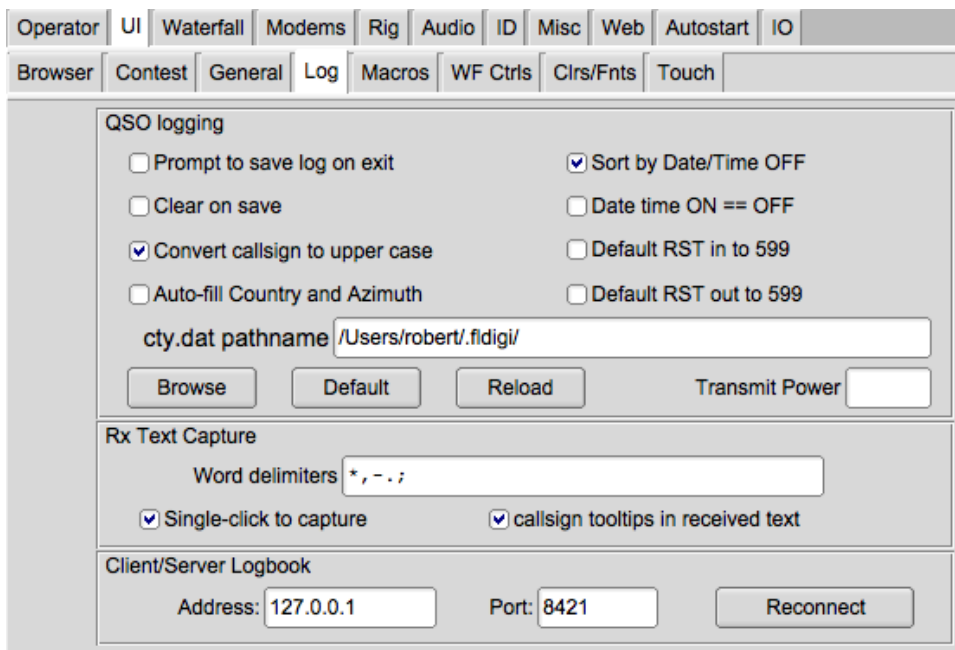


Figure 3.14: QSO Logging

#### 3.4.1 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 "cty.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 elect to use a single left click method if you prefer. The word pointed to by the cursor is parsed to determine it's intended field use; call, RSQ, operator name, and QTH. The call will be overwritten, but the other log fields need to be blank for the data transfer to be accomplished. 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. Clicking on the 599 NJ and Bozo would treat each as a separate word. You may also right click on

a word (or a phrase that has been highlighted by dragging the cursor over the text). That will open a pop-up menu with selections to transfer the data into a number of different log fields.

You can elect to have the RST in/out preset to 599 after you clear the QSO entry fields.

If you check the "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:

### 3.4.2 Pop ups

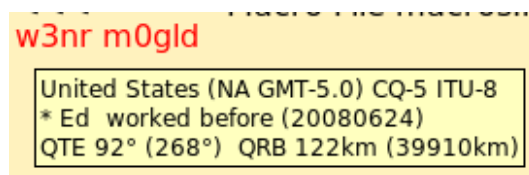


Figure 3.15: Pop ups

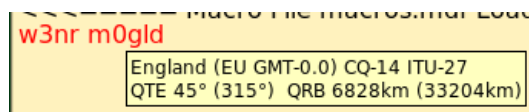


Figure 3.16: Pop ups 2

### 3.4.3 Country Files

The data is derived by parsing the callsign and referral to both the logbook and the "cty.dat" file. 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/>

The default location for this file is in the fldigi default files folder. You have the option of changing that file location by either entering the new folder pathname in the edit control, or by using the "Browse" or "Default" button. If you change the contents of cty.dat while fldigi is running you should force fldigi to reload the data from the file. This data is normally only read when the program starts.



### 3.4.4 Internal Logbook

Fldigi has an internal logbook. That log should be used for single operator operations. There may be times that you need to share a log, either between programs on a single computer, or with other operators running fldigi on other computers on a LAN (or even WAN).

In lieu of the internal logbook you can elect to use a common logbook server. This logbook is maintained by a separate logbook program, `fllog`. `fllog` provides access to read, query and update records via an xmlrpc socket interface. `fllog` provides the server function and connecting applications are clients.

You need to specify both the socket address and socket port. The defaults are as shown and are for the instance when both `fllog` and `fldigi` are on a single computer. Unless configured otherwise, `fllog` will always use the port address 8421.

### 3.4.5 Logbook Network Address






	<b>IP Address / Name</b>
	192.168.1.89 / HPBA49A1
	192.168.1.92 / linux-dev
	192.168.1.93 / HelensDell
	192.168.1.94 / fl-mac
	192.168.1.96 / dave-vista

Figure 3.17: Network Address

My home network has IP address assignments as shown. If `fllog` were running on the mini-mac (`fl-mac`) and `fldigi` running on the `linux-dev` machine. I would enter the server address 192.168.1.94 into the `fldigi` configuration for the Client/Server Logbook.

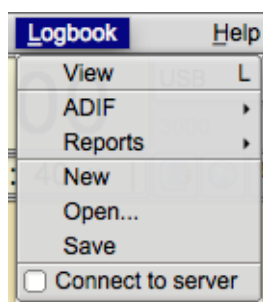


Figure 3.18: Connect to Server

The "Logbook/Connect to server" menu item allows you to connect to the remote logbook. If successful then the toggle remains checked and the menu items for accessing the internal logbook are disabled.

[Return to Top of Page](#)

[Return to Main Page](#)

# Chapter 4

## Macros

- [Macros](#)
- [Inline Macros](#)
- [Delayed Execution Macro Tags](#)
- [Advanced QSY operations](#)
- [Exec Macro](#)

### 4.1 Macros

Macros are short text statements that contain imbedded references to text data used by the program fldigi. A macro is accessed via the macro button bar or function key.

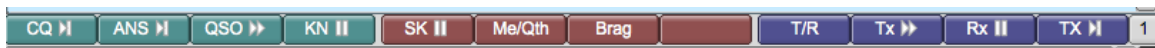


Figure 4.1: Macro Buttons

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. All 48 macro buttons can be displayed in a matrix.

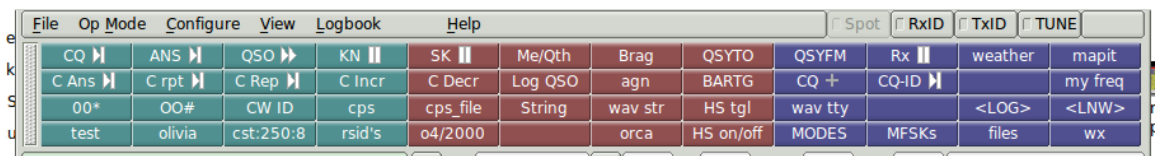
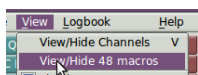


Figure 4.2: Macro Buttons Matrix

The matrix display is toggled on and off using the menu item



The matrix display can be dragged off and restored to its position below the main menu. The drag button is the knurled vertical button on the left edge of the matrix. Left click on the drag button and drag the matrix with the mouse.

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.

#### 4.1.1 Macro tags

Macro	Description
<FREQ>	my frequency
<MODE>	mode
<MYCALL>	configuration call
<MYLOC>	configuration locator
<MYNAME>	configuration name
<MYQTH>	configuration QTH
<MYRST>	my RST
<ANTENNA>	configuration antenna
<VER>	Fldigi version

Macro	Description
<CALL>	other stations callsign
<INFO1>	S/N or other data contained in first info field of status bar
<INFO2>	IMD or other data contained in second info field of status bar
<LOC>	other stations locator
<NAME>	other stations name
<QTH>	other stations QTH
<RST>	other stations RST

Macro	Description
<MAPIT>	map other stations locator on google
<MAPIT:adr   lat   loc>	map other stations address, latitude-longitude locator as specified
<CLRRX>	clear RX pane
<CLRRTX>	clear TX pane

Macro	Description
<FOCUS>	rig freq has keyboard focus

Macro	Description
<GET>	text to NAME/QTH
<TALK:on off t>	Digitalk On, Off, Toggle; this is a Windows only tag and Digitalk must be running
<LOG>	save QSO data to logbook immediately
<LNW>	log at xmt time
<CLRLOG>	clear log fields
<EQSL:[message]>	submit the current log entry to www.eQSL.cc This macro tag should be put before <LOG> or <LNW>

Macro	Description
<QSOTIME>	insert current logbook time HHMM, ie 0919
<ILD>	insert current local date-time in iso-8601 format, ie 2011-08-28 04:16-0500
<LDT>	insert Local date-time, ie 2011-08-28 04:16-0500
<IZDT>	insert Zulu date-time in iso-8601 format, ie 08/28/2011 04:16 CDT
<ZDT>	insert Zulu date-time, ie 2011-08-28 09:16Z
<LT>	insert local time, ie 0416
<ZT>	insert zulu time, ie 0916Z
<LD>	insert local date, ie 2011-08-28
<ZD>	insert Zulu date, ie 2011-08-28
<WX> <WX:xxxx>	insert current weather data from METAR as specified on WX configuration tab. see WX configure replace xxxx with the 4 letter METAR designator for a report on a station other than the one specified on the weather configuration tab.

Macro	Description
<CNTR>	insert current value of contest counter
<DECR>	decrement contest counter
<INCR>	increment contest counter
<XIN>	send exchange in string
<XOUT>	send exchange out string
<XBEG>	mark exchange in start
<XEND>	mark exchange in end
<SAVEXCHG>	save marked text to contest exchange in

Macro	Description
<RX>	receive
<TX>	transmit
<TX/RX>	toggle Transmit / Receive

Macro	Description
<SRCHUP>	search UP for next signal
<SRCHDN>	search DOWN for next signal

Macro	Description
<GOHOME>	return to waterfall cursor to sweet spot
<GOFREQ:NNNN>	move waterfall cursor to freq NNNN Hz
<QSYTO>	same as left-clk on QSY button
<QSYFM>	same as right-clk QSY button
<QSY:FFF.F[:NNNN]>	Qsy to transceiver frequency in kHz, optional waterfall (Audio) frequency in Hz (If not specified, it is not changed). Several QSY frequencies, or ranges of frequencies provided by one increment, can be proposed, in which case the first frequency after the current frequency is chosen.

Macro	Description
<RIGMODE:mode>	send CAT command to transceiver to change to a valid mode
<FILWID:width>	send CAT command to transceiver to change to a valid filter width example to QSY to sweetspot (center of bandpass filter) and select narrow filter

Macro	Description
<FILE:>	insert text file; a file selection box will open when this tag is selected during editing
<IMAGE:>	insert image file; a file selection box will open when this tag is selected during editing
<IMAGE:G,>	insert image file; transmit in grey scale
<VIDEO:text>	transmit video text

Macro	Description
<IDLE:NN.nn>	transmit idle signal for NN.nn sec
<TIMER:NN>	repeat this macro every NN sec
<TUNE:NN>	transmit single tone tune signal for NN sec
<WAIT:NN>	insert delay of NN seconds before transmitting
<REPEAT>	repeat macro continuously
<SKED:hhmm[:YYYYDDMM]>	schedule execution to begin at time and optionally date specified

Macro	Description
<CWID>	transmit a CW callsign identifier
<ID>	transmit mode ID using waterfall video text
<TEXT>	transmit video text defined on ID configuration tab
<TXRSID:on off t>	transmit RSID on, off, toggle
<RXRSID:on off t>	receive RSID on, off, toggle
<NRSID:NN>	transmit multiple RsID bursts NN < 0 will transmit NN bursts for current modem and then return to Rx NN > 0 will transmit NN bursts for current modem and continue in Tx NN = 0 will transmit 1 bursts and continue (same as NN = 1)
<DTMF:[Wn:][Ln:]>	tones transmit DTMF tone sequence at start of transmission; options W-wait n msec, default 0 L-tone symbol length n in msec; default 50 msec '-', ' ' and ',' insert silence symbol eg: <DTMF:W250:L100:1-256-827-3200>

Macro	Description
<POST:+/-nn.n>	CW QSK post-timing in milliseconds
<PRE:nn.n>	CW QSK pre-timing in milliseconds
<RISE:nn.n><RISE:nn.n>	CW rise time in milliseconds
<WPM:ww[:ff]>	CW WPM, ww = word rate, optional ff = Farnsworth rate

Macro	Description
<RIGCAT:hex hex:retnbr>	RigCAT user commands
<RIGCAT:"string":retnbr>	hex hex ... are sequential hexadecimal values string is Ascii char sequence retnbr is number of bytes in xcvr response 'retnbr' is optional; retnbr set to 0 if missing

Macro	Description
<AFC:on off t>	AFC on, off, toggle
<LOCK:on off t>	lock waterfall cursor; on, off, toggle
<REV:on off t>	Reverse on, off, toggle

Macro	Description
<QRG:text>	Insert current operating info with "text" into Rx stream, ie: info text <<2013-01-12T21:18Z RTTY @ 14005000+0760>> which can be used to return to a mode, rf, audio frequency.

<PAUSE>	Cause transmission to pause at place of occurrence in macro text. "Pause/Break" key on keyboard resumes transmission
<TXATTEN:nn.n>	set fldigi tx attenuator to value -30 dB <= val <= 0
<COMMENT:text>	allow macro to contain a comment field; everything between < and > is ignored by macro parser
<SAVE>	save the current macro definitions to the current file

Macro	Description
<MACROS:>	load a new macro defs file; file prompt when editing macro

Macro	Description
<CPS_TEST>	modem timing test internal string
<CPS_FILE:>	modem timing test, spec' file
<CPS_STRING:s>	modem timing test, string 's'

Macro	Description
<WAV_TEST>	WAV file; internal string
<WAV_FILE:>	WAV file; spec' file
<WAV_STRING:s>	WAV file; string 's'

#### 4.1.2 Modem macro tags

Macro tags are also assigned to each supported modem type and sub-modem type that is supported by fldigi:

Data Modems				
<MODEM:8PSK250>	<MODEM:8PSK500>	<MODEM:8PSK1000>	<MODEM:8PSK1333>	<MODEM:BPSK1000>
<MODEM:BPSK125>	<MODEM:BPSK250>	<MODEM:BPSK31>	<MODEM:BPSK500>	<MODEM:BPSK63>
<MODEM:BPSK63F>	<MODEM:CTSTIA↔ :1000:16>	<MODEM:CTSTIA↔ :1000:8>	<MODEM:CTSTIA↔ :250:8>	<MODEM:CTSTIA↔ :500:16>
<MODEM:CTSTIA↔ :500:8>	<MODEM:CTSTIA>	<MODEM:CW>	<MODEM:DOMEX4>	<MODEM:DOMEX5>
<MODEM:DOMEX8>	<MODEM:DOMX11>	<MODEM:DOMX16>	<MODEM:DOMX22>	<MODEM:DOMX44>
<MODEM:DOMX88>	<MODEM:FELDHXL>	<MODEM:FSKH105>	<MODEM:FSKHELL>	<MODEM:HELL80>
<MODEM:HELLX5>	<MODEM:HELLX9>	<MODEM:MFSK11>	<MODEM:MFSK128>	<MODEM:MFSK128↔ L>
<MODEM:MFSK16>	<MODEM:MFSK22>	<MODEM:MFSK31>	<MODEM:MFSK32>	<MODEM:MFSK4>
<MODEM:MFSK64>	<MODEM:MFSK64L>	<MODEM:MFSK8>	<MODEM:MT63-1KL>	<MODEM:MT63-1KS>
<MODEM:MT63-2KL>	<MODEM:MT63-2KS>	<MODEM:MT63-500↔ L>	<MODEM:MT63-500↔ S>	<MODEM:NAVTEX>
<MODEM:Olivia-16-1K>	<MODEM:Olivia-16-500>	<MODEM:Olivia-32-1K>	<MODEM:Olivia-4-250>	<MODEM:Olivia-4-500>
<MODEM:Olivia-64-2K>	<MODEM:Olivia-8-1K>	<MODEM:Olivia-8-250>	<MODEM:Olivia-8-500>	<MODEM:OLIVIA↔ :1000:32>
<MODEM:OLIVIA↔ :1000:8>	<MODEM:OLIVIA↔ :250:8>	<MODEM:OLIVIA↔ :500:16>	<MODEM:OLIVIA↔ :500:8>	<MODEM:OLIVIA>



<MODEM:PSK1000↔ C2>	<MODEM:PSK1000R>	<MODEM:PSK1000↔ RC2>	<MODEM:PSK125↔ C12>	<MODEM:PSK125R>
<MODEM:PSK125R↔ C10>	<MODEM:PSK125R↔ C12>	<MODEM:PSK125R↔ C16>	<MODEM:PSK125R↔ C4>	<MODEM:PSK125R↔ C5>
<MODEM:PSK250C6>	<MODEM:PSK250R>	<MODEM:PSK250R↔ C2>	<MODEM:PSK250R↔ C3>	<MODEM:PSK250R↔ C5>
<MODEM:PSK250R↔ C6>	<MODEM:PSK250R↔ C7>	<MODEM:PSK500C2>	<MODEM:PSK500C4>	<MODEM:PSK500R>
<MODEM:PSK500R↔ C2>	<MODEM:PSK500R↔ C3>	<MODEM:PSK500R↔ C4>	<MODEM:PSK63R↔ C10>	<MODEM:PSK63R↔ C20>
<MODEM:PSK63R↔ C32>	<MODEM:PSK63R↔ C4>	<MODEM:PSK63R↔ C5>	<MODEM:PSK800C2>	<MODEM:PSK800R↔ C2>
<MODEM:QPSK125>	<MODEM:QPSK250>	<MODEM:QPSK31>	<MODEM:QPSK500>	<MODEM:QPSK63>
<MODEM:RTTY:170↔ :45.45:5>	<MODEM:RTTY:170↔ :50:5>	<MODEM:RTTY:850↔ :75:5>	<MODEM:RTTY>	<MODEM:SITORB>
<MODEM:SLOWHE↔ LL>	<MODEM:THOR100>	<MODEM:THOR11>	<MODEM:THOR16>	<MODEM:THOR22>
<MODEM:THO↔ R25x4>	<MODEM:THOR4>	<MODEM:THOR5>	<MODEM:THO↔ R50x1>	<MODEM:THO↔ R50x2>
<MODEM:THOR8>	<MODEM:THRBX1>	<MODEM:THRBX2>	<MODEM:THRBX4>	<MODEM:THROB1>
<MODEM:THROB2>	<MODEM:THROB4>	<MODEM:WEFA↔ X288>	<MODEM:WEFA↔ X576>	

### 4.1.3 Other Modems

The following modems perform other functions not involving data transportation:

Non Data Modems	Description
<MODEM:NULL>	Null modem - Rx stream is discarded. Tx stream is silent but PTT is enabled
<MODEM:SSB>	SSB modem (place holder)
<MODEM:ANALYSIS>	<a href="#">Analyze Frequency</a>
<MODEM:FREQSCAN>	<a href="#">Waterfall Power Measurement Frequency Scan</a>
<MODEM:WWV>	<a href="#">Calibrate Sound Card</a>

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.

### 4.1.4 Macro Editor

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.

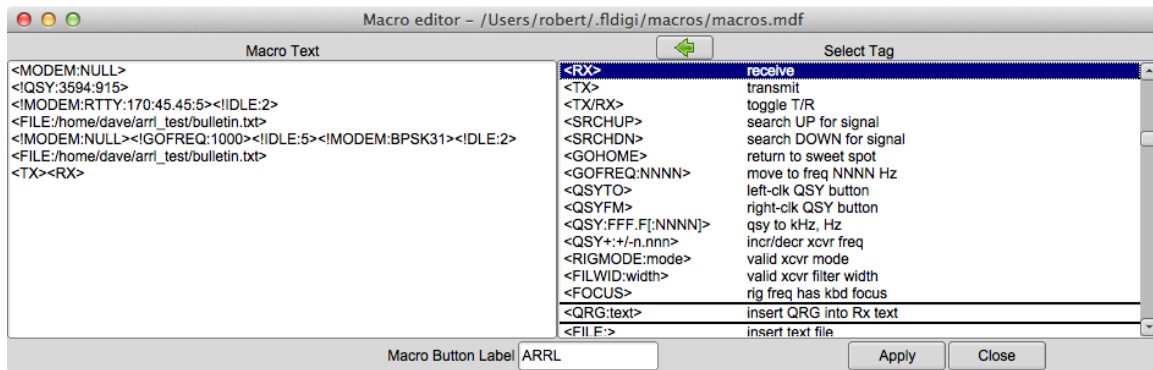


Figure 4.3: Macro Editor

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 transferred to the current editing cursor location. Highlight the desired macro reference and then press the double << arrow key for each occurrence 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:

## 4.1.5 Macro Display Symbols

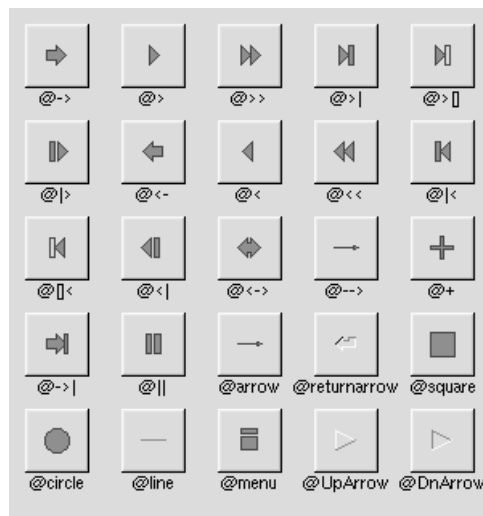


Figure 4.4: Symbols

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:

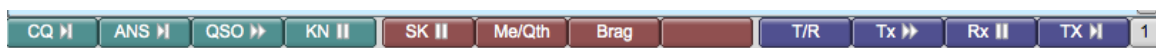


Figure 4.5: Macro Buttons

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](#) 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 [Exit Prompts](#) option selected.

#### 4.1.6 Contest macro tags

Refer to [Contest-How-To](#)

[Return to Top of Page](#)

[Return to Main Page](#)

## 4.2 Inline Macros

The following macro tags will be parsed and acted upon then they occur during the transmission of the macro.

Tag	Description
<!WPM:NN>	CW words per minute
<!POST:+/-nn.n>	CW post delay
<!PRE:nn.n>	CW pre delay
<RISE:nn.n><RISE:nn.n>	CW rise/decay time
<!MODEM: ,...>	Change to specified modem with parameters
<!GOHOME>	Move audio carrier to mode sweet spot frequency
<!GOFREQ:NNNN>	Move audio carrier to specific audio frequency
<!QSY:FFF.F[:NNNN]>	Move to specific RF and Audio frequency
<!IDLE:NN.nn>	Transmit idle signal for specified number of seconds
<!WAIT:NN>	Wait (no audio) for specified number of seconds
<!WPM:ww[:ff]>	CW - ww = WPM, optional ff = Farnsworth wpm

Note that each of these tags is identical to their immediate mode counterparts. The exception is the addition of the exclamation mark following the leading '<'. The use of these tags is best explained by example.

#### 4.2.1 CW Code Practice Transmission

```
<MODEM:CW><TX>
<!GOFREQ:600><!WPM:10:15>
NOW IS THE TIME - now 180 wpm
```

```
<!IDLE:2><!RISE:1.0><!PRE:0.4><!POST:+0.2><!WPM:180>
FOR ALL GOOD MEN TO COME TO THE AID of their country.
now 30 wpm<!IDLE:2>
<!WPM:30>de <MYCALL> k
<RX>
```

- Modem type is changed to CW before start of transmission (<MODEM:CW>) does not have the addition of the ! symbol
- Audio carrier is changed to 600 Hertz
- Transmit WPM set to 30 words per minute, followed by text
- Transmit is silent for 2 seconds
- Rise time, pre and post delays are adjusted and the WPM changed to 180 words per minute
- Text is sent at 180 WPM
- WPM changed to 30 words per minute, followed by identification string and signoff
- Return to receive

This is a more complex macro that might be used for a code practice transmission such as the W1AW broadcast on 3580 KHz.

```
<MODEM:NULL>
<!QSY:3579.200:800>
<TX>
<!MODEM:CW>
<!WPM:5:15><!IDLE:2><FILE:/home/dave/fldigi.ft950/scripts/practice-5wpm.txt>
<!WPM:10:15><!IDLE:5><FILE:/home/dave/fldigi.ft950/scripts/practice-10wpm.txt>
<!WPM:15:15><!IDLE:5><FILE:/home/dave/fldigi.ft950/scripts/practice-15wpm.txt>
<!WPM:18><!IDLE:10><FILE:/home/dave/fldigi.ft950/scripts/bulletin.txt>
<!IDLE:5>end of broadcast de <MYCALL> k
<RX>
```

#### 4.2.2 QSY:ffff.f:aaa test

```
<MODEM:NULL><TX>
<!QSY:3583.0:1750>
<!MODEM:RTTY:170:45.45:5>
RYRYRYRYRYRY de <MYCALL> k
<RX>
```

- Change modem type to NULL to suppress sending any signal when transmit enabled. This is necessary to allow the transmit data stream to be processed.
- Send frequency change command to transceiver, new frequency is 3583.0 kHz. Set audio frequency to 1750 Hz
- Change modem to RTTY, 170 Hz shift, 45.45 Baud, 5 Bit code (standard Baudot)
- Send RY... text
- Return to receive

### 4.2.3 Advanced QSY operations

Several QSY frequencies or ranges of frequencies provided by one increment, can be proposed, in which case the first frequency after the current frequency is chosen.

#### Advanced QSY operations

### 4.2.4 ARRL style broadcast in multiple modes

```
<MODEM:NULL>
<!QSY:3594:915>
<!MODEM:RTTY:170:45.45:5><!IDLE:2>
<FILE:/home/dave/arrl_test/bulletin.txt>
<!MODEM:NULL><!GOFREQ:1000><!IDLE:5><!MODEM:BPSK31><!IDLE:2>
<FILE:/home/dave/arrl_test/bulletin.txt>
<TX><RX>
```

- Change modem type to NULL to suppress transmission and start the Tx data stream processor
- Send frequency change command to transceiver, new frequency is 3594 kHz, audio frequency 915 Hz. Note that this puts the MARK tone at 1000 Hz.
- Change modem to RTTY 170 Hz shift, 45.45 baud (default is 5 bits). Idle for 2 seconds.
- Send the file contents of specified file
- Change to NULL modem. Turns off the RTTY diddle ... audio stream is silent.
- Change audio frequency to 1000 Hz. Idle for 5 seconds (complete silence of transmit)
- Change modem type to BPSK-31. Send BPSK idle for 2 seconds
- Send the file contents of specified file
- Return to receive (note that the <TX> can appear anywhere in the macro definition as it is executed when the macro text is parsed). <RX> is always moved to the end of the Tx buffer. It could also have appeared anywhere in the macro string.

### 4.2.5 Appearance of tx buffer

```
^!
^!^!
QST de W1HKJ
Test bulletin for 9/7/2011
QST de W1HKJ SK

^!^!^!^!^!
QST de W1HKJ
Test bulletin for 9/7/2011
QST de W1HKJ SK

^r
```

This is the contents of the ARRL broadcast macro text at the time the macro button is pressed. Each of the <!... macro tags has been specified by the string "^!" which the Tx processor interprets as the command to process the top most tag in the first-in, first-out sequence of tags. As each "^!" is executed the referenced tag is printed to the

Rx buffer using color coded text.

```
<!QSY:3594:915>
<!MODEM:RTTY:170:45.45:5>
<!IDLE:2>

QST DE W1HKJ
TEST BULLETIN FOR 9/7/2011
QST DE W1HKJ SK

<!MODEM:NULL>
<!GOFREQ:1000>
<!IDLE:5>
<!MODEM:BPSK31>
<!IDLE:2>

QST de W1HKJ
Test bulletin for 9/7/2011
QST de W1HKJ SK
```

[Return to Top of Page](#)

[Return to Main Page](#)

## 4.3 Delayed Execution Macro Tags

The following macro tags will be parsed and queued for action after fldigi returns to the receive mode, i.e. after the <RX> tag (^r) is executed.

Tag	Description
<@MODEM: , ... >	Change to specified modem with parameters
<@RIGCAT:hex hex:retnbr>	RigCAT user commands
<@RIGCAT:"string":retnbr>	hex hex ... are sequential hexadecimal values string is Ascii char sequence retnbr is number of bytes in xcvr response ':retnbr' is optional; retnbr set to 0 if missing
<@GOHOME>	return waterfall cursor to sweet spot
<@GOFREQ:NNNN>	move waterfall cursor to freq NNNN Hz
<@RIGMODE:mode>	send CAT command to transceiver to change to a valid mode
<@FILWID:width>	send CAT command to transceiver to change to a valid filter width example to QSY to sweetspot (center of bandpass filter) and select narrow filter
<@TXRSID:on off t>	transmit RSID on, off, toggle

Note that each of these tags is identical to their immediate mode counterparts. The exception is the addition of the @ symbol following the leading '<'. The use of these tags is best explained by example.

### 4.3.1 Check in to a net with video ID

```
<TXRSID:off>
<MODEM:NULL>
<TX>
```

```
<!GOFREQ:2200>
<VIDEO:K7KY>
<@GOFREQ:1200>
<@MODEM:MFSK32>
<@TXRSID:on>
<RX>\n
```

- TxRSID is disabled before any other action
- modem is changed to NULL to prevent anything but VIDEO transmission
- Audio carrier is changed to 2200 Hertz
- The callsign K7KY is transmitted as video text at 2200 Hertz
- Return to receive
- Waterfall carrier is set to 1200 Hertz
- Modem is changed to MFSK-32
- TxRSID is enabled

[Return to Top of Page](#)

[Return to Main Page](#)

## 4.4 Advanced QSY operations

The `<QSY:FFF.F[:NNNN]>` macro tag and its delayed flavour `<!QSY:FFF.F[:NNNN]>` allows the user to set the transceiver frequency to FFF.F (In kHz) and optionally set the audio frequency to NNNN (In Hz).

It is possible to give several frequencies, which specify a frequency set. When the macro is executed, it chooses the first frequency of this set greater than the current one. If the current frequency is greater than any frequency of the set, then the macro chooses the smallest, first frequency of this set.

The utility of this feature is to allow a scan of a range of frequencies by re-executing the same macro over and over. Each time the macro is executed, the next frequency of the set is selected. It can typically be used to iteratively try several frequency for a given test.

The frequency set can be specified in two distinct ways, which can be freely combined together.

### 4.4.1 Distinct frequencies

Several distinct separated with a semi-column, in increasing order. For example:

```
<QSY:2616.6;3287.6;3853.1;4608.1;4780.1;7878.1;8038.1:1900>
```

It means that the transceiver frequency will be set to 2616.6 kHz, then at next execution to 3853.1 kHz etc... and will loop back to 2616.6 kHz. Each time the same (optional) audio frequency will be set to 1900 Hz. In this specific case, it allows to test several Weather Fax broadcast stations, until the operator finds an active one.



#### 4.4.2 Frequencies with increments

A frequency can come with an increment: This means an implicit range of frequencies from this one to the next frequency. If the last frequency has an increment, it is never taken into account: The last frequency is always an upper limit. Let's consider this example:

```
<QSY:89000+100;102000>
```

It is equivalent to:

```
<QSY:89000;89100;89200;89300;...;101900;102000>
```

In this case, it allows fldigi to scan all FM frequencies by clicking the macro button.

#### 4.4.3 Combination with <TIMER> macro tag.

If the macro is automatically reexecuted using the <TIMER>, the same logic applies. At each run of the macro, the next frequency is chosen. The following macro transmits the same message on the frequencies 144800 MHz, 144900 ... until 146000 then loops back, waiting five seconds between each transmission.

```
<TX><QSY:144800.00+100;146000><MODEM:NULL><!MODEM:PACKET>  
<FREQ> CQ CQ de <MYCALL><RX><TIMER:5>
```

#### 4.4.4 Execution errors.

Several error messages can be displayed in the macro editor in case of a parameter. If this happens, the execution of the macro is stopped. Here is the list of possible messages:

#### 4.4.5 Invalid frequency range

There must be valid frequencies. This macro will display the message:  
<QSY:abcdef>

#### 4.4.6 Increment must be positive

The frequency increment must be positive. This macro will fail:  
<QSY:89000-1000;88000>

#### 4.4.7 Frequency not positive

All frequencies must be strictly positive.

#### 4.4.8 Frequencies must be increasing

The sequence of frequencies must be strictly increasing. Thus, this error message can appear with a macro such as:

```
<QSY:89000;88000>
```

#### 4.4.9 Inline Macro Tags

Unless otherwise noted all of the macro tags discussed thus far are meant to be executed at the time that the macro is invoked by the button closure. It is possible by execution of the tag to be delayed until it appears in the transmit data stream. This delayed execution capability is limited to a select number of tags. See [Inline Macro Tags](#) for a list of these tags and examples.

#### 4.4.10 Delayed Execution Macro Tags

It is possible for the execution of the tag to be delayed until after the macro execution returns to the receive state. This delayed execution capability is limited to a select number of tags. See [Delayed Execution Macro Tags](#) for a list of these tags and example.

[Return to Top of Page](#)

[Return to Main Page](#)

### 4.5 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.

#### 4.5.1 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.

#### 4.5.2 Detection of existing scripts

In anticipation of a collection of useful "fldigi scripts", the macro browser contains a `<exec>` `</exec>` 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:

```

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

```

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.

### 4.5.3 Querying an external database

The `<GET>` command captures returned text from the external process and parses it for the following content:

```
$NAMEtext_name$QTHtext_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](#). The matching macro key definition for the above is:

```
<EXEC>ualr-telnet.pl $FLDIGI_LOG_CALL</EXEC><GET>
```

which I named "ualr ?"

### 4.5.4 Google Earth Map

Here is a really cool perl script, [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>
```

### 4.5.5 Custom dates/times

You can use `<EXEC>` to create custom date/time entries. For example, BARTG contesters use HM, 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 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.

[Return to Top of Page](#)

[Return to Main Page](#)



# Chapter 5

## Modems

- [Contestia](#)
- [CW](#)
- [DominoEX](#)
- [Hellschreiber](#)
- [FSQ](#)
- [IFKP](#)
- [MFSK](#)
- [MT63](#)
- [NAVTEX and SITOR-B](#)
- [Olivia](#)
- [PSK Single and Multi-Channel Modems](#)
- [RTTY](#)
- [Thor](#)
- [Throb](#)
- [WEFAX](#)
- [WWV transmit mode](#)
- [Mode Table](#)

### 5.1 Contestia

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

Mode	Symbole Rate (Baud)	Typing speed (WPM)	Bandwidth (Hz)
Contestia 4-250	62.5	~ 40	250
Contestia 8-250	31.25	~ 30	250
Contestia 4-500	125	~ 78	500

Contestia 8-500	62.5	~ 60	500
Contestia 16-500	31.25	~ 30	500
Contestia 8-1000	125	~ 117	1000
Contestia 16-1000	62.5	~ 78	1000
Contestia 32-1000	31.25	~ 48	1000

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 2000hz) 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 - 3db 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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.2 CW

### 5.2.1 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.

### 5.2.2 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.



### 5.2.3 Pausing transmit

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

### 5.2.4 Aborting transmit

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

### 5.2.5 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.



Figure 5.1: The Rx and Tx WPM are shown in the status bar.

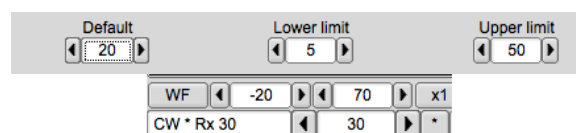


Figure 5.2: \* indicates that default WPM is selected

### 5.2.6 Farnsworth keying

You might want to use farnsworth keying to provide a character rate that is faster than the word per minute rate.

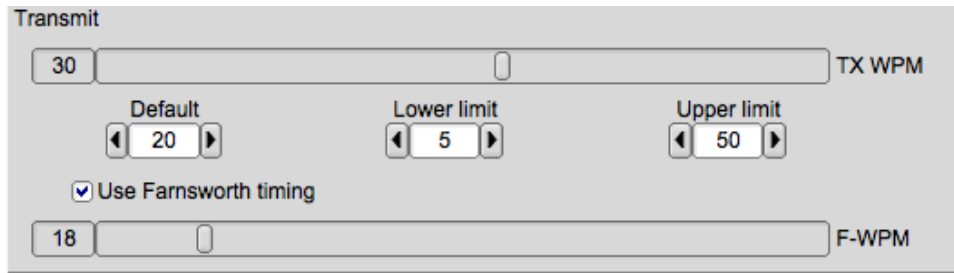


Figure 5.3: Farnsworth Timing

Set the F-WPM slider to the character rate and enable the Use Farnsworth timing check box. When the TX WPM is set to a speed below the Farnsworth value then the character rate will be at the Farnsworth setting and the word rate will be at the TX WPM rate. You can also set the TX WPM and F-WPM from within a [Macros](#).

### 5.2.7 CW configuration

The [CW Configuration](#) is easily reached from the Config menu or by right clicking on the left most entry in the status bar (CW).

### 5.2.8 Prosigns

The prosigns are configurable (see [CW Configuration](#) ). The defaults are:

PROSIGN	KEYBOARD	DISPLAYED AS
BT	=	<BT>
AA	~	<AA>
AR	>	<AR>
AS	<	<AS>
HM	{	<HM>
INT	&	<INT>
SK	%	<SK>
KN	+	<KN>
VE	}	<VE>

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.3 DominoEX

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 without the need for AFC. The AFC control does not alter the decoder in any way.

The waterfall and digiscope will appear as:

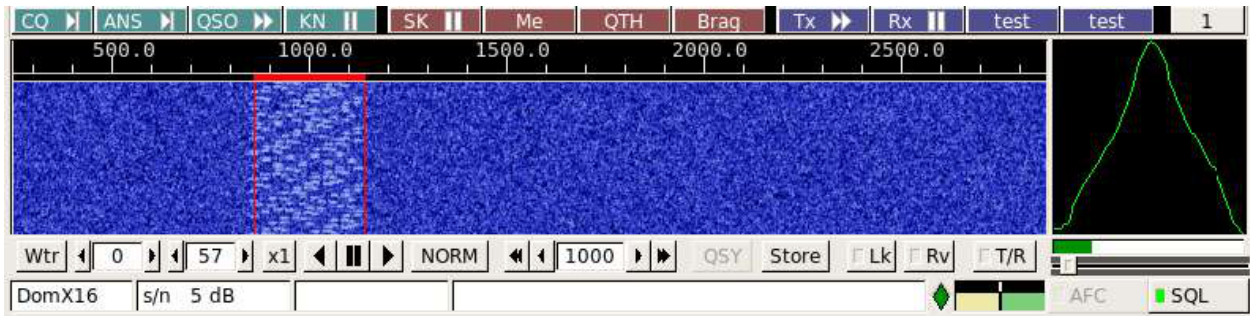


Figure 5.4: DominoEX

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).

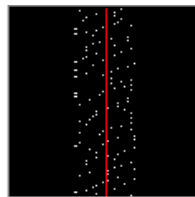


Figure 5.5: DominoEX

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:

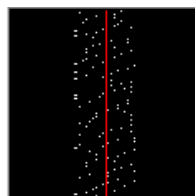


Figure 5.6: DominoEX Mistuned

and with the signal badly mistuned:

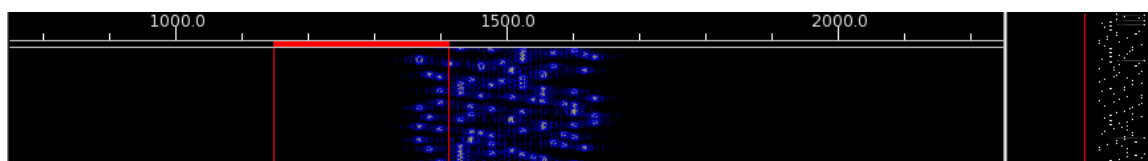


Figure 5.7: DominoEX Mistuned

See [DominoEX Configuration](#).

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.4 Hellschreiber

### 5.4.1 Hellschreiber modem

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 :

### 5.4.2 Hellschreiber modes

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

### 5.4.3 Hellschreiber Waterfall

Feld-Hell look like this when being received by fldigi:

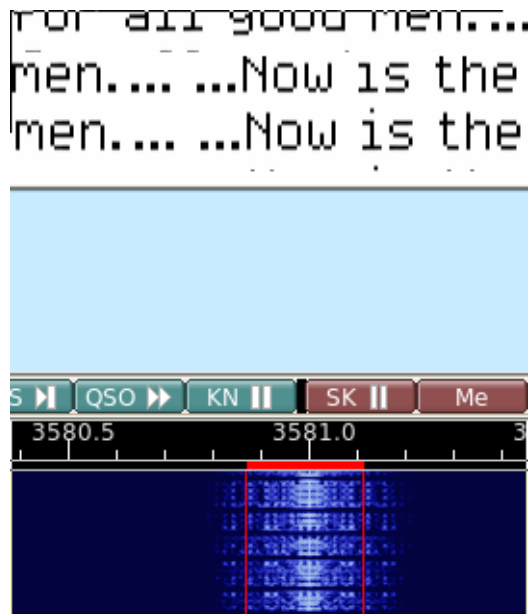


Figure 5.8: Feld-Hell, Slow Hell, Feld-Hell X5, and Feld-Hell X9 are all pulse

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. 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 on HF in the US. They can be used on VHF and UHF.

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.5 FSQ

FSQ, Fast Simple QSO, is an Incremental-Frequency-Keyed mode using an offset differential modulation scheme similar to DominoEX, and Thor. It is a unique mode for fldigi users in that it is a line by line transmission rather than character by character. FSQ uses 33 tones spaced by 3 times the 3 baud symbol rate or 8.8 Hz. The offset rotation of the IFKP sequence offers improved performance under NVIS conditions, because the rotation significantly reduces the risk of adjacent symbols causing inter-symbol interference.

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

Mode	Symbol Rate	WPM	Bandwidth
FSQ 2	1.95 baud	~20	290 Hz
FSQ 3*	2.93 baud	~30	290 Hz
FSQ 4.5	3.91 baud	~40	290 Hz
FSQ 6	5.86 baud	~60	290 Hz

- FSQ-3 is the base rate.

At the time of this release there is only one other amateur modem application that supports FSQ, FSQCALL, written by Con Wassilieff, ZL2AFP. FSQ is an invention of Murray Greenman, ZL1BPU. Both Murray and Con were very supportive during the development and testing of fldigi's FSQ implementation.

fldigi-fsq supports the undirected, directed and image modes of FSQCALL.

This FSQ help content borrows shamelessly from Murray Greenman's (ZL1BPU) "FSQ Explained" document.



Figure 5.9: fsq main dialog

The main dialog fsq user interface provides an Rx-text panel, a Tx-text panel, a "Heard" list, fsq specific control buttons, and a signal to noise indicator.

The Rx-text panel displays all decoded text in the UnDirected Mode. The Tx-text panel is for entering all transmitted text, including directives. These two panels can be resized vertically by dragging the dividing bar which separates them.

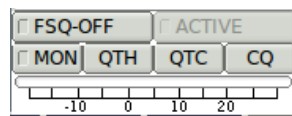


Figure 5.10: Undirected Mode

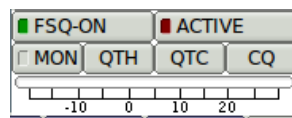


Figure 5.11: Directed Mode

The Rx-text panel displays only recognized text in the Directed Mode. The text parser must be satisfied that the sending stations callsign/crc8 agree and that the directed station callsign is your own or "allcall". FSQ is case sensitive, and your sending case must match the senders case, i.e. w1hkj and W1HKJ are not the same for the purposes of FSQ Directed Mode.

- FSQ-ON
  - when on the xmt and rcv are in FSQ directed mode
  - when off the xmt and rcv are in FSQ undirected mode
- ACTIVE
  - when on the decoder will accept all incoming directed traffic
  - when off the decoder will only respond to the wake-up \* trigger
- MON
  - when on the monitor panel is visible

- QTH  
appends the myQTH text to the tx panel
  
- QTC  
appends the myQTC text to the tx panel
  
- CQ  
sends the FSQ cq call 'cqccq'

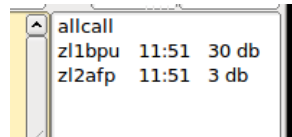


Figure 5.12: Heard List Panel

The "Heard" list displays the callsign, s/n, and time of reception of all correctly decoded transmissions.

NOTE: fldigi function keys are inoperative in the FSQ mode!

### 5.5.1 Modulation Scheme

FSQ is designed to use 32 tone differences between 33 equally spaced tones; so it is possible to allocate 29 different individual differences directly to the most frequently used characters. These characters are all lower case, a-z, plus the most common punctuation symbols: space, full stop (period), and new line (CR/LF). The three 'spare' differences are used to define further code tables.

The total alphabet includes upper case and a reasonable range of symbols, yielding a total alphabet of 104 ASCII characters. The extra characters are allocated to three additional code tables, which are sent as two sequential tone differences. These are characterized by an initial difference, describing the character, and a continuation difference, defining the code table.

When the receiver sees a tone difference in the range 0 – 29, followed by another in the range 0 – 29, it recognises a single-tone character (lower case etc). If the tone difference is followed by a difference in the range 30 – 31, it uses the second difference to decide which code table to select the character from, and the first difference to choose the character from that table. The encoding process is the reverse of this.

Symbol(s)	Char	Symbol(s)	Char	Symbol(s)	Char	Symbol(s)	Char
0	SPACE	0-29	@	0-30	~	0-31	=
1	a	1-29	A	1-30	1	1-31	[
2	b	2-29	B	2-30	2	2-31	\
3	c	3-29	C	3-30	3	3-31	]
4	d	4-29	D	4-30	4	4-31	^
5	e	5-29	E	5-30	5	5-31	
6	f	6-29	F	6-30	6	6-31	{
7	g	7-29	G	7-30	7	7-31	
8	h	8-29	H	8-30	8	8-31	}
9	i	9-29	I	9-30	9	9-31	`
10	j	10-29	J	10-30	0	10-31	±
11	k	11-29	K	11-30	!	11-31	÷
12	l	12-29	L	12-30	quote	12-31	°
13	m	13-29	M	13-30	#	13-31	×
14	n	14-29	N	14-30	\$	14-31	£
15	o	15-29	O	15-30	%	15-31	
16	p	16-29	P	16-30	&	16-31	
17	q	17-29	Q	17-30		17-31	
18	r	18-29	R	18-30	(	18-31	
19	s	19-29	S	19-30	)	19-31	
20	t	20-29	T	20-30	*	20-31	
21	u	21-29	U	21-30	+	21-31	
22	v	22-29	V	22-30	-	22-31	
23	w	23-29	W	23-30	/	23-31	
24	x	24-29	X	24-30	:	24-31	
25	y	25-29	Y	25-30	;	25-31	
26	z	26-29	Z	26-30	<	26-31	
27	.	27-29	,	27-30	>	27-31	BS
28	CRLF	28-29	?	28-30	IDLE	28-31	DEL

Figure 5.13: FSQ Varicode Table

IFK codes the data (the initial and continuation codes) as differences between two tones, rather than as an absolute tone. This is the MFSK equivalent of differential BPSK, as used in PSK31, etc. MFSK modes such as Piccolo, Coquelet and MFSK16 use direct MFSK modulation, assigning the code directly to a tone number. These modes may suffer from more inter-symbol interference than desirable, cannot tolerate frequency drift, and can be difficult to tune.

FSQ uses very narrow tone spacing. IFK+ reduces inter-symbol interference by ensuring that the chance of the same or an adjacent tone being used for sequential symbols would be very remote.

Since the tones always change due to the differential action and the tone rotation of IFK+, there is no opportunity for the tones to remain the same for two consecutive symbols, which allows the sync-less process to operate correctly. For example, without IFK+, repeated space characters could be seen as a single character.

Finally, since the coding is differential, any drift and frequency offset is cancelled out during decoding. IFK+ can handle frequency drift of about a third of the tone spacing per symbol, or in FSQ about 18 Hz drift per second at 6 baud. Frequency error tolerance is a matter of decoder design, but  $\pm 50$  Hz is tolerated easily by FSQ.

IFK+, an acronym for Offset Incremental Frequency Keying, was suggested by Murray Greenman ZL1BPU (2009), and first used in DominoEX. With each symbol transmitted, as well as adding the difference from the alphabet-coding table to the previous tone number, an addition rotation offset is added. In the case of FSQ, this value is ONE.

Only one tone is transmitted at a time.

In order for the first difference (first character) to be correctly decoded, a dummy tone needs to be transmitted first. This could be the lowest tone, but in fact any permitted tone will suffice since only the difference to the next tone is



important. The software simply sends a space character to achieve this.

## 5.5.2 Transmit Functions

The user types text into the Tx panel, which can be edited before transmission. Transmission starts when an E↵ NTER character is pressed. It will also start as a result of one of the automatic processes, such as 'Sounding' or automated replies.

The characters in the Tx panel are then encoded, and transmitted at the currently set symbol rate. As they are transmitted, the characters are transferred to the receive pane as a record of what has been transmitted.

### BOT Sequence

At the start of transmission in directed and non-Directed Mode, the BOT sequence, a SPACE followed by a LIN↵ EFEED, is sent. The space serves as a dummy symbol to allow the next tone difference to be measured. The LF character ensures that reception starts on a new line.

In Directed Mode, the LF also serves as one delimiter for the callsign search algorithm. The other delimiter is the colon that follows the callsign. This also locates the two callsign checksum characters.

### EOT Sequence

At the end of transmission in UnDirected Mode, the transmitter is simply turned off once the last useful character is sent. In Directed Mode, however, a special character sequence is sent just before the transmission stops, to signal EOT.

The BS (backspace) character is not required in FSQ, so the corresponding character in the table has been reallocated as an EOT marker.

The purpose of the EOT sequence is to cause the Squelch to close quickly in Directed Mode. Print will continue through fades without fldigi closing and stopping print partway through a sentence. If the sentence transmission were to finish without the EOT, the squelch would close slowly, and junk would print unnecessarily. The EOT 'shorts out' the Squelch time constant, closing the Squelch instantly.

The transmission consists of 33 tones spaced 8.7890625 Hz, resulting in a spread of tones of 290.0390625 Hz. Using the ITU-R SM.1138 assessment method, the signal easily meets 300 Hz useful bandwidth, at all signalling speeds, and its ITU Emission Designator is therefore 300HF1B.

In Directed mode, a conventional network model CSMA protocol decides when the station may have access to the radio channel. This protocol defines three classes of message (chat, responses and soundings), and applies priorities to them. Priority is provided by means of delays after Squelch has closed. In addition, to limit clashes between stations wanting access for the same priority message, random delays are also added.

Priority	Delay	Message Type
1	Short	Chat sentences
2	Med	Automated responses to commands
3	Long	Sounding

### Sounding

In Directed Mode, the user may elect to send regular 'soundings'. These are very short messages with no trigger command and no message body. The message consists only of the callsign, colon and checksum of the FSQCall preamble.

These sentences do not print at any station, but appear in the Heard List and are logged in the Heard Log. This allows all stations on the channel to know what other stations are there with minimum use of channel bandwidth.

Sounding is sent with the lowest message priority. Sounding is essentially random, although the time between Sounds is selectable at 1, 10 and 30 minute intervals. The start time is set by user action, and essentially random.

### 5.5.3 FSQ Protocol

FSQ is designed as a 'chat' mode, not a long-over QSO mode. It is capable of sending multiple sentences in each transmission, but it is easiest and quickest when just one sentence is sent.

Each transmission starts with the fldigi station callsign (converted to lowercase).

For example, if the user (say z11xyz) types:

Hello John – how are you this evening<ENTER>

The software actually transmits:

z11xyz:Hello John – how are you this evening

Nothing could be simpler. Every station within range will print this message.

#### Performance

There is no error correction involved, so users rely on the inherent robustness of the mode, and if there are errors, either due to 'typos' or reception errors, the users will quickly recognise these. If an error causes confusion, they can always ask for a repeat.

Operating 20W on the 40m band during the middle of the day, an automated station 300 km away was used to return transmitted sentences. From 250 words transmitted, at 6 baud, 223 were returned correct (89.2% copy), while at 3 baud, copy was 100%. For practical purposes, 80% copy is considered adequate for a comfortable QSO.

Tests using an ionospheric simulator bear these measurements out. Reception is essentially 100% for white-noise tests with the signal above -15dB S/N.

### 5.5.4 Directed Mode

FSQ Directed Mode operates as a simple extension of the FSQ protocol. The preamble is changed a little, a direction (address and command) is applied, and finally a short EOT signal is added to the end of each message. These differences are simple to apply, as the user only needs to learn a few simple commands and remember to add a direction and command to each sentence. The rest is automatic.

Further, directed sentences in FSQ mode can be read easily in undirected FSQ mode, and undirected text is visible in the monitor panel which can be viewed in Directed Mode. In Directed Mode, only text directed to your callsign will print in the Rx panel, and only commands directed to your station will be responded to.

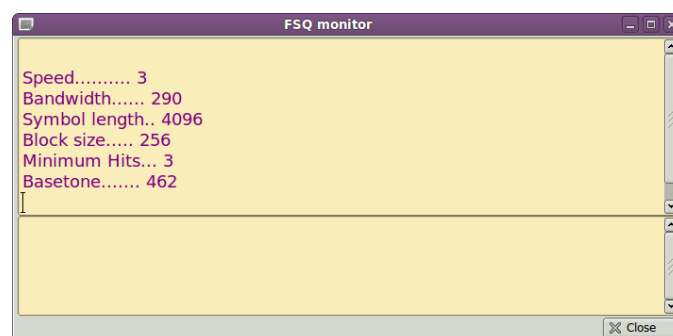


Figure 5.14: FSQ Monitor Panel

The Monitor Panel is only shown during Directed Mode operation. It is hidden when Undirected Mode is in use. All incoming traffic is printed on the Monitor Panel. This includes both Directed and Undirected traffic. The decoding is affected by the fldigi squelch setting.

You may see what appears to be a perfectly good Directed data stream on the Monitor Panel which never appears on the Rx Panel. When this occurs, the data stream has failed to parse correctly:

1. the BOT is missing or corrupt,

2. the CRC8 check has failed,
3. the EOT is missing or corrupt,
4. or it is not directed to your callsign or allcall.

The lower half of the monitor panel displays text that is queued for transmission, for example the response to a ? or a \$ trigger. Automated responses are not sent immediately, but after a small random delay AND when the squelch is not open.

### FSQCall Preamble

The FSQCall preamble consists of the sending station's callsign, a colon, and a two character hexadecimal checksum. For example, the simplest transmission, a Sounding message, consists of only this. For example:

```
z11bpu:b6
```

The checksum is a standard CCITT CRC8 calculated on the callsign and the colon. The maximum callsign length is 16 characters, and the CRC is capable of detecting up to four errors in this distance reliably. Pseudo-callsigns (such as place names or personal names) can be used, which is very convenient for Public Service and Emergency applications.

### Directions and Triggers

All FSQ Directed Mode messages, apart from the Sounding, carry at least one direction and a trigger. The direction is the callsign or callsigns of the recipient station or stations, in other words, where the message is directed. The trigger is one of a short list of one-letter commands sent to the directed station, which define what it should do with the message. There may or may not be a message body following the directions and triggers.

Here are some simple examples:

```
z11bpu:b1z11abc hello john
w1hkj:efw3fqk21bm hello guys
w1hkj:efk21bm?
```

Color coding:

```
[ ] sending station
[ ] crc8 characters
[ ] directed station(s)
[ ] trigger|
```

Figure 5.15: Directed Mode Examples

In the first example, ZL1BPU is simply chatting to ZL1ABC. He starts typing by giving the direction, the other station's callsign. The trigger used is a space, which is understood by the receiving station as an invitation to print the message that follows.

The second example illustrates how you can send the same sentence to two stations. There is also an 'all stations' direction (allcall) that can be used to send messages and some commands to all stations within range.

The third is an example of a simple automated query. It is the FSQ way of asking for a signal report. The queried station will respond with your s/n report as measured by the FSQ program that is being used.

### Error Checking

FSQ Directed Mode sentences are checked at the receiver in two ways. First, the sender's callsign is checked against the checksum received: if this is incorrect, the sentence is rejected. If it is correct, the rest of the sentence is checked to find the recipient's callsign. If this is not received correctly, the sentence is again rejected. If the callsign is received correctly, only the remainder of the message following the recipient's callsign is printed or acted upon.

When (and if) the incoming message is displayed, the preamble is shown without the checksum, so it looks like the preamble in the UnDirected Mode. However, the callsign displayed is in this case the callsign verified by the checksum.

Checking the validity of the originator's callsign ensures that the message is correctly attributed, is correctly logged, and most significantly, responses can be sent to the correct station.

### Logging

In Directed Mode, FSQ maintains a log of all stations heard, in chronological order. With each entry, verified callsign, the date/time, and measured SNR are recorded in CSV (comma separated value) format. This is suitable for examination as a spreadsheet. The heard log file is named "fsq\_heard\_log.txt" and is located in the fldigi.files/temp folder.

A complete audit log of transactions is also maintained in Directed Mode. This includes regular time stamps, all raw received text, and all transmitted sentences. The audit log is named "fsq\_audit\_log.txt" and is located in the fldigi.files/temp folder.

### Directed Callsigns

1. Each station responds to three 'callsigns'.
2. The operator callsign (fldigi converts operator call to lowercase).
3. The 'callsign' allcall, to which the software will act upon certain messages, including chat, file and image transfer, and alerts. This mechanism is widely used in nets to send chat to everyone in range.
4. The 'callsign' cqccq is recognised as a legitimate chat 'callsign'.

### Use of Lower Case

FSQ is case sensitive, so messages directed (for example) to 'ZL1EE' will not be recognised by the station with its callsign set to 'z11ee'. Lower case is not mandatory for operator callsigns, but is encouraged for these reasons:

1. Lower case is quicker and easier to type.
2. Lower case is faster to send (most characters are sent in one symbol).
3. The error rate is half that of upper case characters.

## 5.5.5 Triggers

All the command triggers (currently) consist of one reserved character. These are taken from a pool of characters that are not permitted in callsigns. As far as possible, the characters have a meaning easily associated with the character.

This list shows all the reserved characters, and their current or planned use. Blank entries are reserved, with no currently planned application.

CHAR	ASCII	USE
Space	32	Directed chat
!	33	Repeat command
"	34	--
#	35	Message or file transfer
\$	36	Request Heard List

%	37	Receive image
&	38	Report QTC message
'	39	--
(	40	--
)	41	--
*	42	Restore FSQ to ACTIVE from SLEEP
+	43	Message retrieve
,	44	--
-	45	Message delete (1)
.	46	--
:	58	Reserved, not used.
;	59	Relay command
<	60	Reduce sending speed
=	61	Reserved for ECC file transfer
>	62	Increase sending speed
?	63	Report reception
@	64	Report QTH (or position) message
[	91	Used in file transfers
\	92	--
]	93	Used in file transfers
^	94	Report FSQ program version
_	95	Report the squelch setting in dB (1)
'	96	--
{	123	--
	124	Send Alert
}	125	--
~	126	Delayed Repeat command

(1) not supported

### 5.5.6 Commands

Command sentences consist of a direction followed immediately by a command. Multiple directions can be used in the same sentence under limited circumstances. Two directions in the same sentence, which are understood by the same station, will cause confusion, and only the second will be responded to. For example, one should not send 'z1xyz@ allcall message', as the first part will be ignored by station z1xyz.

#### Command Syntax

The syntax for currently defined commands is listed below.

- **callsign text** Addressee station starts to print text. No trigger character used except space following callsign. You MUST use a space or the trigger will not be recognised.
- **callsign? text** (Is station hearing me?) Addressee station starts to print text. When squelch closes, station responds: origin\_callsign snr=xxdB
- **callsign\*** Switches station to ACTIVE if FSQSEL is in SLEEP. Addressee station starts to print. When squelch closes, station responds: callsign:CRC-VAL Active
- **callsign! message** (Repeat my message) Addressee station starts to print. When squelch closes, station responds: callsign:CRC-VAL message. This is a simple relay mechanism. message may contain further trigger commands.
- **callsign~ message** (Repeat my message later) Addressee station starts to print. When squelch closes, after a delay of about 15 seconds, station responds: callsign:CRC-VAL message This delayed relay is useful when stations mentioned in message can hear and respond to the original message.

- **callsign;** [*relay\_to*]{*trigger*}{*message*} Relay my message to the 'relay\_to' station. See Relay section for details.
- **callsign!sendto***message* Relay message to the 'sendto' station. Addressee station starts to print. When squelch closes, station relays the message.
- **callsign#[nnn]** (Send file to station) Addressee station starts to print. Text following [nnn] will be saved in or appended to text file nnn.txt in the same folder as the log/audit files. When squelch closes, station responds callsign:CRC-VAL Ack if the message stored OK. There is no response if the message was not received. [nnn] can be alpha, numeric or mixed.
- **callsign#[filename.txt]** (Send file to station) Addressee station starts to print. File is opened from menu, and filename appended to message before file body text, File will be saved in or appended to text file filename.txt in the same folder as the log/audit files. When squelch closes, station responds callsign:CRC-VAL if the message stored OK. There is no response if the message was not received. [nnn] can be alpha, numeric or mixed.
- **callsign@** (Request station position or location information) Addressee station starts to print. When squelch closes, station responds with pre-recorded sentence, typically containing QTH information. This could be GPS position, locator, or physical address. This is the message stored and retrieved by the QTH button.
- **callsign&** (Request station message) Addressee station starts to print. When squelch closes, station responds with pre-recorded sentence, typically containing station information or an 'Out to Lunch' message. This is the message stored by the QTC button.
- **callsign^** (Request Software Version) Addressee station starts to print. When squelch closes, station responds with current FSQCALL software version.
- **callsign%** (Send image) Addressed station receives the command to record an analog graphic image. The Rx Image Dialog will open and the image painted as it is being received. Images are in color or grey scale and always displayed in native size.. Images can be directed to allcall.
- **callsign|message** (Send alert) Addressee station starts to print. Alert pop-up box containing message is placed on the screen. When the operator closes this dialog, a response transmission is made: origin\_callsign  
Alert ack
- **cqqcq** text (Call for general chat) Following text prints.
- **allcall** text (All stations print) Following text prints at all stations.
- **callsign>** or **callsign<** (Speed change) > increases speed, < decreases speed at target station. Response is origin\_callsign: 4.5 baud (or whatever the new speed is). If the speed is already at the maximum or minimum speed, the response is the same, and there is no speed change.
- **allcall#[nnn]** (Send file to all stations) All stations within range start to print. Text following [nnn] will be saved in or appended to text file nnn.txt. When squelch closes, station responds callsign:CRC-VAL Ack if the message stored OK. There is no response if the message was not received. allcall#[filename.txt] works the same as callsign#[filename.txt].

### 5.5.7 Relaying messages

This is a relay mechanism which preserves the origin callsign as the message propagates so that any response can be relayed back to that station. message can contain further trigger commands.

Originating station: relay\_callsign; dest\_callsign message

```

Example 1. Simple relay with no response
k1a types: k2a;k3a hi
k1a sends: k1a:11k2a;k3a hi
k2a sees : k1a;;k3a hi
k2a sends: k2a:22k3a [k1a] hi
k3a sees : k2a:[k1a] hi
Example 2: Relay with response
k1a types: k2a;k3a@
k1a sends: k1a:11k2a;k3a@
k2a sees : k1a;;k3a@
k2a sends: k2a:22k3a [k1a]@
k3a sees : k2a:[k1a]@
k3a sends: k3a:33k2a;k1a vienna, va fm18iw
k2a sees : k3a;;k1a vienna, va fm18iw
k2a sends: k2a:22k1a [k3a] vienna, va fm18iw
k1a sees : k2a:[k3a] vienna, va fm18iw
Example 3: Relayed allcall
k1a types: k2a;allcall Calling NET. Any Relays?
k1a sends: k1a:11k2a;allcall Calling the NET. Any Relays?
k2a sees : k1a;allcall Calling the NET. Any Relays?
k2a sends: k2a:22allcall [k1a] Calling the NET. Any Relays?
A checking in station might type
k2a;k1a ka4cdn (mike)
k1a would see the check-in relayed through k2a as
k2a:[k1a] ka4cdn (mike)

[ ] sending station
[ ] crc8 characters
[ ] directed station(s)
[ ] trigger

```

Figure 5.16: Relay Examples

### 5.5.8 Special keys

Pressing the F1 key will transfer the "last call" selected from the heard list to the Tx panel. "last call" defaults to allcall on program startup.

Pressing the F2 key will transfer the last command to the Tx panel. The last command is an empty string on program startup.

### 5.5.9 Heard List

The FSQ heard list is automatically added to, modified, and aged in accordance with the FSQ user configuration parameters. Each time a valid callsign/crc8 pair is discovered it is either added to the heard list, or it's entry is updated with the last GMT and signal-to-noise ratio.

A single click on a heard list entry will make it the "last call" string.

A double click on a heard list entry will also add the call to the Tx panel. If multiple heard list entries are double clicked each one is added to the Tx panel separated by a space.

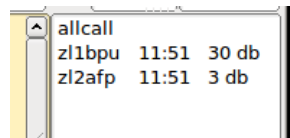


Figure 5.17: Heard List

Heard list entries have an associated pop-up menu, available by right-clicking on the desired entry:

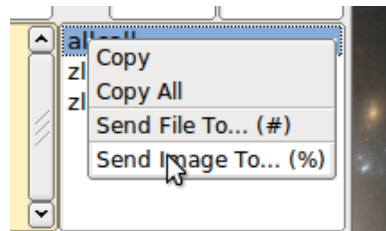


Figure 5.18: All Call Menu

### 5.5.10 Image Mode

fldigi-fsq has the ability to transmit and receive small digital images selected from those stored on the disk. The image transmission is similar to but not compatible with the MFSKpic modes. FSQ images are always transmitted in a fixed width to height ratio. The W x H format is not limited to a single ratio, nor is it limited to just color, or just gray scale.

The start of the picture is signalled digitally, and the receiver simply captures data continuously for W lines of H pixels, as requested, resulting in a WxH pixel image, then reception stops.

The images are approximately in 4:3 or 3:4 aspect ratio. Images must be sent and received in Directed Mode, and can be sent to just one recipient, to several recipients, or everyone on the channel, just as with other messages.

Available image aspect ratios / formats include:

- 160x120 color (1)
- 320x240 color (1)
- 640x480 grey (1)
- 640x480 color (1)
- 240x300 color
- 240x300 grey
- 120x150 color
- 120x150 grey



(1) compatible with the program FSQCALL

### Narrow-Band Images

FSQ Image Mode was designed for NVIS propagation on the lower HF bands. Unlike SSTV, the pictures are sent relatively slowly, in a narrow bandwidth (1/4 of SSTV), and as a result the picture signal to noise ratio is generally better than SSTV for the same transmission power. As a guideline, FSQ signals (before the picture is started) need to be about +10dB SNR for noise-free picture reception.

FSQ pictures also offer significantly better reception than SSTV under NVIS conditions. Because there is no picture sync, there is no risk of the picture tearing due to timing changes, or breaking up during selective fades. All that happens is that the image may noise up momentarily, but will stay correctly aligned. The mode is also significantly less affected by ghosting.

### Image Format

The image mode sampling rate is 12,000 samples/sec, the same as the text mode. Each pixel consists of 10 samples, and there are P pixels/line. Therefore the line period is  $10 \times P / 12000$  seconds, or the line rate  $12000 / (10 \times P)$  lines/sec. In colour, each line is sent three times, in BGR order.

As an example, if there were 320 pixels per line, and 240 lines, the line rate would be 3.75 lines/sec, and 240 x 3 (for RGB) lines would take 192 seconds, or 3.2 minutes for a colour picture containing 76,800 pixels.

There is no horizontal (line) sync sent with the picture, and no vertical sync at the start of the picture. As a result, it is possible for reception to start slightly earlier or later than the standard delay from the time of the command, depending on the relative speed of the computers used at either end. This has two effects: first, the picture may be recorded shifted left or right, and the pixel colour order may be wrong. This is easily fixed after reception using the Phase control.

If the sound cards in the transmitting and receiving computers differ markedly in sample rate, the resulting picture can also be skewed. A small timing difference is enough to cause a noticeable slant in the picture. This is easily fixed after reception using the Slant control. High quality sound cards should be better than 5ppm, and result in barely noticeable slant, but some cheaper computer built-in sound modules may be rather worse. If you plan to send FSQ images you should calibrate your sound card to WWV.

### Receive Image

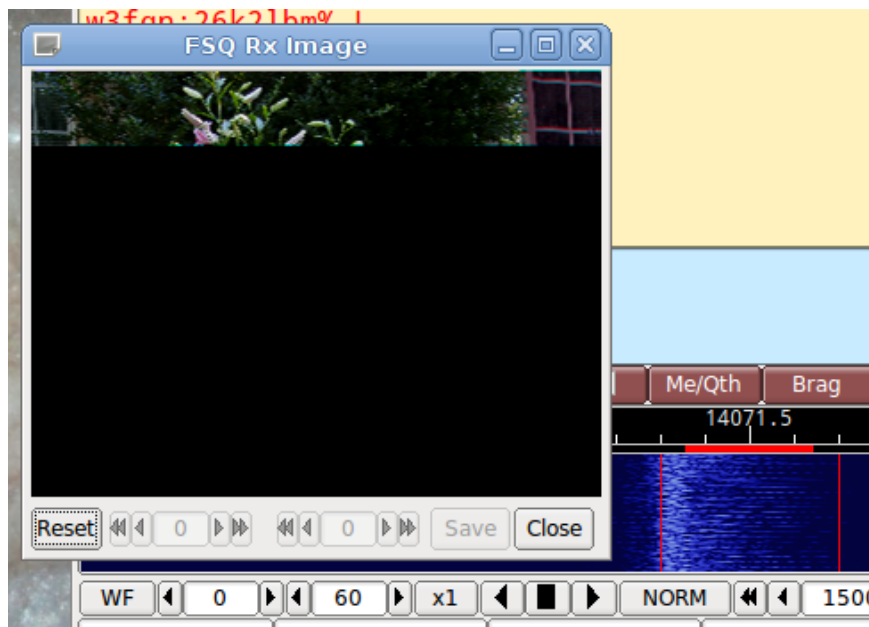


Figure 5.20: Image Reception

Reception is completely automatic. Provided the software is in Directed Mode, and the sending station uses the correct callsign and trigger command, the decoder will identify the picture start, and record the picture. In doing so, it automatically opens a separate "FSQ Rx Image" dialog.

Demodulation uses a phase detection algorithm based on quadrature demodulation using the current and previous samples at 12000 samples/sec. All the samples are converted to pixels and stored, but only the average over every 10 samples is displayed.

The raw image is stored in a temporary buffer of expanded size, which allows fine adjustment of the slant and phase to be made before the image is sampled and displayed.

If the transmitting station goes off the air or reception is lost during recording of a picture, the recording will continue until the necessary number of samples has been made, then reception stops.

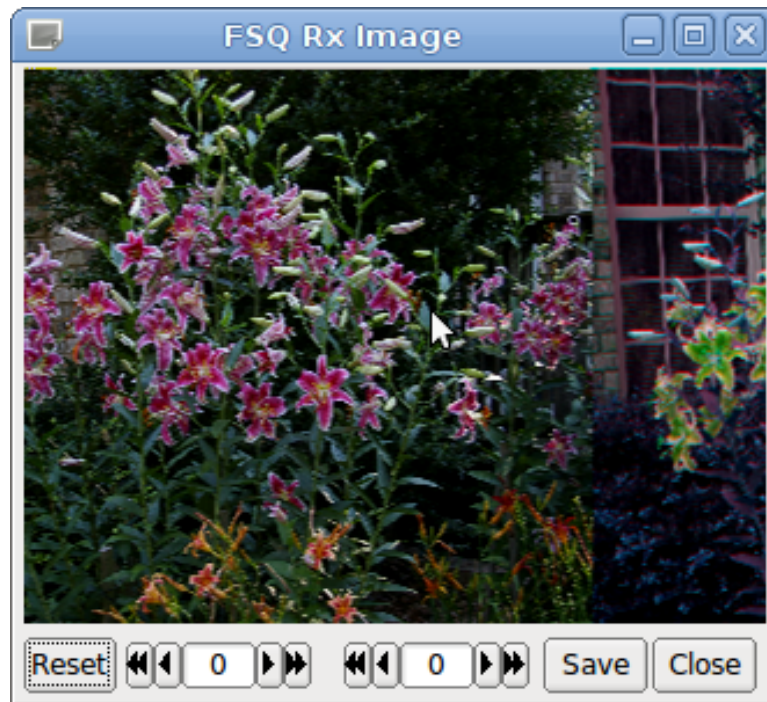


Figure 5.21: Received - no correction

The Rx Image dialog controls from left to right are:

- Reset: clears all rx data
- Phase adjustment: moves image left and right
- Slant adjustment: slews image diagonally
- Save: saves the image to the fldigi image folder with a date-time stamped file name
- Close: closes the image dialog

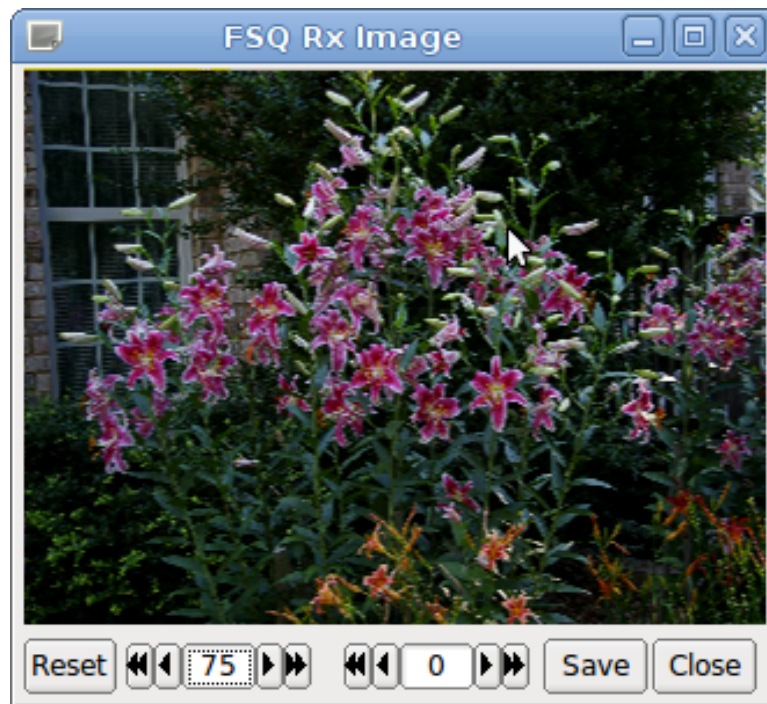


Figure 5.22: Received - phase corrected

Image phase and slant should not be adjusted before the picture reception has finished, as then only part of the image will be corrected.

#### **Transmit Image**

Access to the transmit image functions is only available via the heard list pop-up menus. Right click on the recipient call sign and select the "Send Image to ..." item from the pop-up menu.

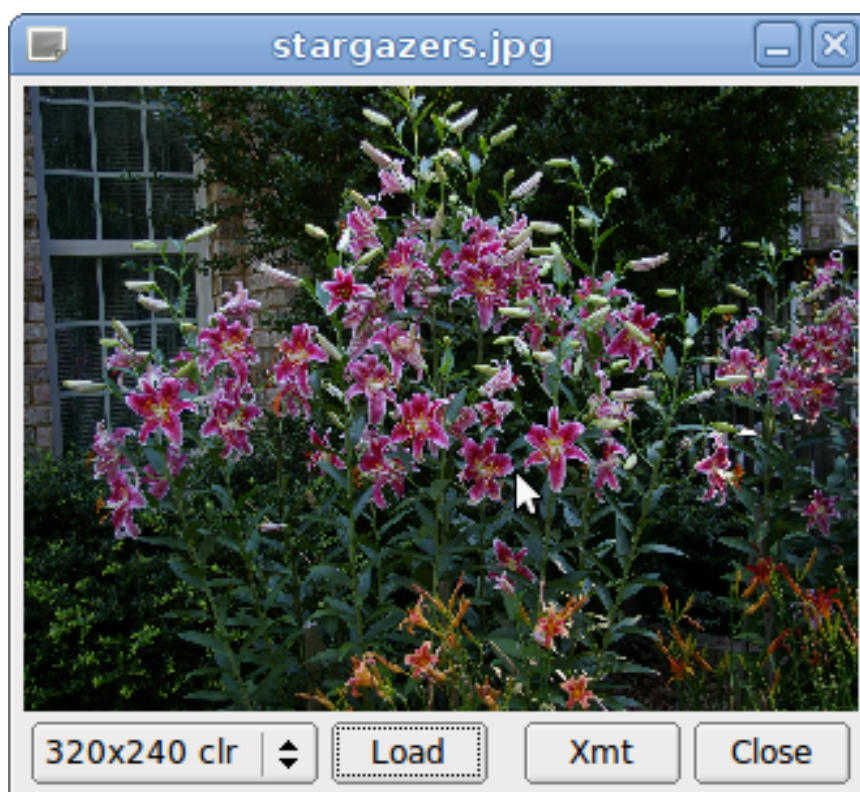


Figure 5.23: Image Transmission

A file open dialog is displayed when the "Load" button is pressed on the TX Image dialog. This allows the user to select any image in a number of different standard formats. The image is rendered to the selected image size using a resampling process. fldigi will try to force fit the image to the selected aspect ratio. You should try to work with images that are in a 4:3 or a 3:4 aspect ratio. The selected image does not need to be the same size. Fldigi will enlarge or reduce the image using a resampling technique to fit the selected image size. The example photo of Star Gazer lilies is a 3.2 MB file, 3008 x 2000 pixels. It was taken in my lily patch in June 2014.

Once the image has been loaded, the user starts the transmission by pressing the "Xmt" button on the transmit image dialog. fldigi formats the correct command sequence and starts the transmission.

The recipient callsign for image transmission can be a single callsign, or allcall. Sending to a single recipient or to allcall is accomplished as described above. To send to multiple addresses you must add their callsigns to the Tx panel, separated by spaces and with a trailing space. Then press the Xmt button on the Tx Image dialog.

fldigi returns to the receive mode at the conclusion of the image transfer.

### 5.5.11 Disable RxID

You will probably want to disable the RxID detector during FSQ operations. If RxID is enabled and an RsID signal is detected the program will switch to the mode specified by the RxID. That will completely alter the user interface.

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.6 IFKP

### 5.6.1 Incremental Frequency Keying Plus

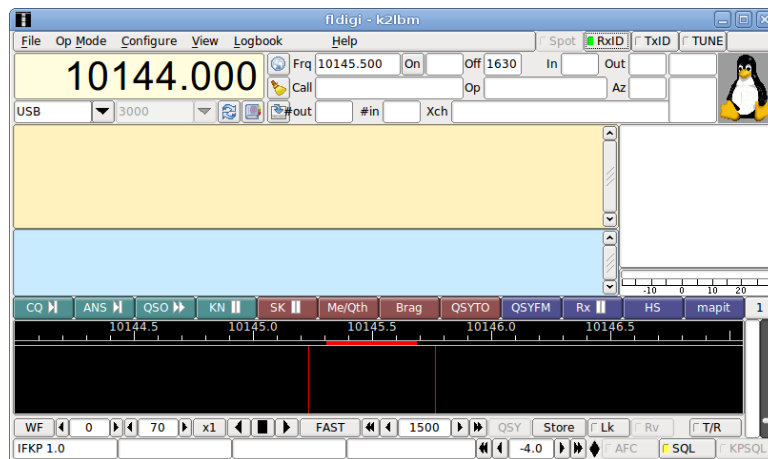


Figure 5.24: ifkp main dialog

The main dialog ifkp user interface provides an Rx-text panel, a Tx-text panel, a "Heard" list, and an ifkp signal to noise indicator, and an avatar image. The default avatar is the "Tux" logo. Sending, receiving and saving avatars is discussed in the avatar section.

Keyboard operation in ifkp is identical to other digital modes such as PSK, MFSK, DominoEX, etc. with a primary difference; the alphabet is a truncated version of the MicroSoft Code Page 1250, CP1250. This alphabet is described in the ifkp varicode section.

Unlike FSQ, which uses a similar incremental frequency keying technique, IFKP does not employ any special protocols. The exception to this rule is for the transmission and reception of IFKP images.

IFKP transmission and reception can occur anywhere within the nominal passband of 500 to 3500 Hertz. fldigi will adjust the waterfall tracking point if the center frequency selected would create a signal that is either above or below the nominal passband. The default audio frequency is 1500 and most operators will elect to use the default.

### 5.6.2 IFKP heard list

The IFKP decoder includes a parser which continuously examines the received text stream looking for a sequence of characters which begins with <SP>de<SP>, independent of case. <SP> is the space character. The ensuing word is then tested for validity as a callsign. The following examples would all be qualified as a valid callsign:

```
" DE W1HKJ "
" de k17cgf "
" de      W3FQN "
" de K2LBM<LF>"
"<LF>de N2IKY<LF>"
```

Note that the white space can be extended between the "de" and the callsign. The callsign can be terminated with either a space character or a new line character.

When a valid callsign is discovered it is added to the heard list along with the time and detected signal to noise ratio. New callsigns are always prepended to the top of the heard list. Any older heard list entries with the same callsign are purged.

Left clicking on a heard list entry inserts that callsign in the Tx panel at the cursor position. Right clicking on a heard list entry deletes that entry from the list.

### 5.6.3 IFKP specification

IFKP is a incremental frequency keyed mode with an offset of +1. It is designed for very high coding efficiency. This version uses 33 tones (32 differences), a Varicoded Alphabet, with a rotating difference frequency. Modem statistics:

Parameter	Value
Samplerate	16000
Bandwidth	386
Center freq	1500(1)
Symbol length	4096
Lower case cps	3.65

(1) nominal, adjustable from 500 to 3500 Hertz

Tone differences are expressed as decimal numbers 0 - 31. The alphabet design uses single-symbol and two-symbol character codes. The character set does not include much extended ASCII, and has only 116 different codes. It is a strongly 'unsquare' design (29 x 3), which restricts the number of codes, but we don't need the maximum anyway. This 'unsquare' code maximizes the single-symbol set to 29, so will fit all lower case in one symbol. Using 29 initials means using a comparison process to discover where the initial symbols are.

Amateur communications does not follow standard English text for letter frequency. It has a higher incidence of letters such as Q, X and Z (in Q codes and callsigns). Lower case letters are more common than upper case. Numbers are considered equal in frequency of occurrence to upper case characters.

The IFKP character set has been designed for one set of 29 single-symbol characters, and three two-symbol sets of upper case letters, number and punctuation. There is some spare space, and a few extended ASCII characters have been included ( $\pm \div \circ \pounds \times$ ).

There are 29 single-symbol characters, 87 two-symbol codes, and a total alphabet of 116 codes. This allows coding all upper case, lower case and punctuation. The most common symbol is the NULL, a non printing character and is assigned a single symbol entry in the table.

The nomenclature used for the character symbol numbers in the tables is

[1st\_symbol], ([2nd\_symbol])

expressed in decimal, where the [1st\_symbol] always has values less than 29, and the [2nd\_symbol] will have a value of 30 or 31.

### 5.6.4 IFKP Varicode

The IFKP varicode table is defined as shown in the following tables:

CHAR	ASCII	VAR	CHAR	ASCII	VAR	CHAR	ASCII	VAR
IDLE	0	0	@	64	0,29	'	96	9,31
!	33	11,30	A	65	1,29	a	97	1
"	34	12,30	B	66	2,29	b	98	2
#	35	13,30	C	67	3,29	c	99	3
\$	36	14,30	D	68	4,29	d	100	4



```
chars/sec: 3.65
```

```
text: ABCDEFGHIJKLMNOPQRSTUVWXYZ
chars/sec: 1.84
```

```
text: 0123456789
chars/sec: 1.84
```

```
text: !#$%&*()_+=[ ]{|$;:'", <>/?<bs><lf>
chars/sec: 1.84
```

### 5.6.5 IFKP xmt speed

fldigi-ifkp can transmit text in one of 3 speeds, 0.5X, 1.0X and 2.0X. The 1.0X speed is the natural speed of the modem. The 0.5X and 2.0X speeds are obtained by dividing or multiplying the symbol length by 2 respectively. You can expect to experience decreased decoding errors with the 0.5X speed. Use it when s/n conditions are well below minus 10 dB on the IFKP signal strength meter. The 2.0X speed should probably not be used unless the s/n is well above plus 10 dB on the IFKP signal strength meter.

The three speeds can be selected at the time that IFKP is chosen from the mode menu. The speed can also be changed using the IFKP modem configuration tab. The quickest way to change speed is to left click on the left-most status indicator on the fldigi status bar (lowest set of controls on the main dialog). A pop-up menu will appear with the speed selections. The quick mode change and the configuration tab changes will occur immediately even during transmission. The selection from the mode menu will cause a complete reset of the IFKP modem.

### 5.6.6 IFKP Image Mode

fldigi-ifkp has the ability to transmit and receive small digital images selected from those stored on the disk. The image transmission is similar to but not compatible with the MFSKpic modes. IFKP images are always transmitted in a fixed width to height ratio. The W x H format is not limited to a single ratio, nor is it limited to just color, or just gray scale. The image mode is nearly identical to the FSQ image mode with the exception of the RGB transmission sequence. IFKP transmits in Blue-Green-Red order, whereas IFKP transmits in Red-Green-Blue order. The IFKP image bandwidth is less than 400 Hz.

The start of the picture is signalled digitally, and the receiver simply captures data continuously for W lines of H pixels, as requested, resulting in a WxH pixel image, then reception stops.

IFKP Image Mode was designed for NVIS propagation on the lower HF bands. Unlike SSTV, the pictures are sent relatively slowly, in a narrow bandwidth (1/4 of SSTV), and as a result the picture signal to noise ratio is generally better than SSTV for the same transmission power. As a guideline, IFKP signals (before the picture is started) need to be about +10dB SNR for relatively noise-free picture reception.

IFKP pictures also offer significantly better reception than SSTV under NVIS conditions. Because there is no picture sync, there is no risk of the picture tearing due to timing changes, or breaking up during selective fades. All that happens is that the image may noise up momentarily, but will stay correctly aligned. The mode is also significantly less affected by ghosting.

#### Image Format

The images are approximately in 4:3 or 3:4 aspect ratio. IFKP images can be received by any IFKP capable decoder.

Available image aspect ratios / formats include:

- 59x74 color
- 160x120 color
- 320x240 color
- 640x480 grey scale



- 640x480 color
- 240x300 color
- 240x300 grey scale
- 120x150 color
- 120x150 grey scale

These formats correspond roughly to a portrait (3x4) and landscape (4x3) photo. fldigi will scale any computer stored image to the target image. You should edit the images off line to be at least close to the 3x4 or 4x3 ratio. That will prevent the fldigi loader from distorting the loaded image.

The image mode sampling rate is 16,000 samples/sec, the same as the text mode. Each pixel consists of 8 samples, and there are P pixels/line. Therefore the line period is  $8 \times P / 16000$  seconds, or the line rate  $16000 / (8 \times P)$  lines/sec. In colour, each line is sent three times, in Red-Green-Blue (RGB) order.

As an example, if there were 320 pixels per line, and 240 lines, the line rate would be 6.25 lines/sec, and 240 x 3 (for RGB) lines would take 115.2 seconds, or 1.92 minutes for a colour picture containing 76,800 pixels.

There is a single sync signal sent at the beginning of each image. This sync signal is a tone placed 186 Hz below the center frequency, or at 1384 Hertz. The sync signal is 1.5 symbol lengths in duration, or 5.8 seconds.

If the sound cards in the transmitting and receiving computers differ markedly in sample rate, the resulting picture can also be skewed. A small timing difference is enough to cause a noticeable slant in the picture. This is easily fixed after reception using the Slant control. High quality sound cards should be better than 5ppm, and result in barely noticeable slant, but some cheaper computer built-in sound modules may be rather worse. If you plan to send IFKP images you should calibrate your sound card to WWV.

### Transmit Image

Transmitting an image in IFKP is initiated by selecting the "Send image" menu item from the pop up Tx menu. Right click on the Tx panel

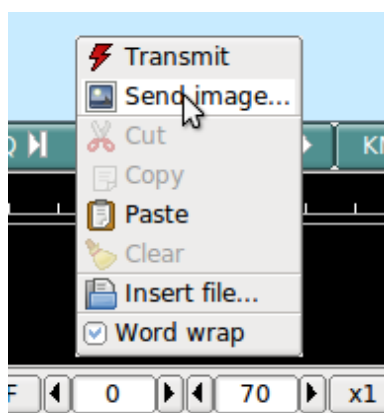


Figure 5.25: Send Image

This selection opens up the Send Image dialog



Figure 5.26: Send Image Dialog

shown with a 160x120 color image loaded and ready to transmit.

Transmission begins when you press the "Xmt" button. fldigi will insert the text preamble and immediately begin the image transmission. fldigi returns to the receive mode when the image transmission is completed.

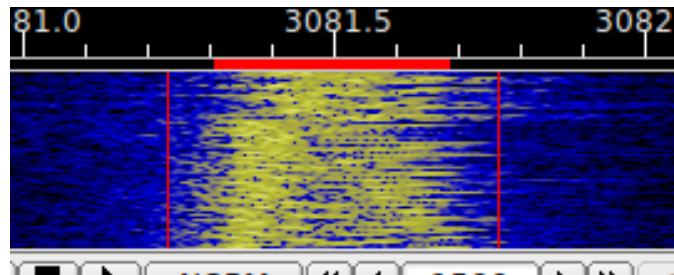


Figure 5.27: Image Mode Waterfall

### Receive Image

Reception is completely automatic. The decoder will identify the picture start, and record the picture. In doing so, it automatically opens a separate "IFKP Rx Image" dialog.

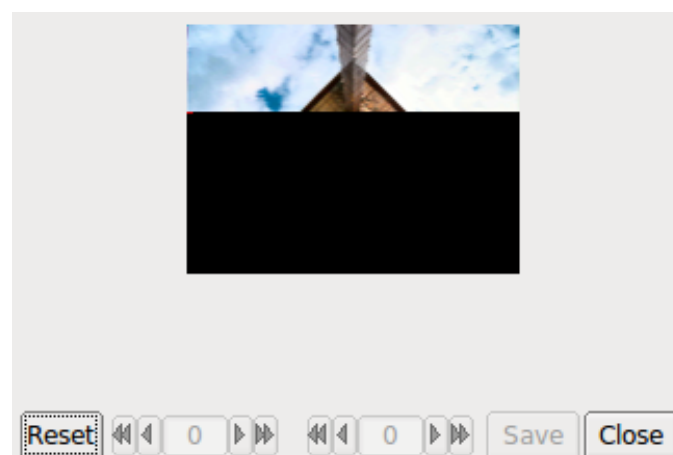


Figure 5.28: Image Reception

Demodulation uses a phase detection algorithm based on quadrature demodulation using the current and previous samples at 16000 samples/sec. All the samples are converted to pixels and stored, but only the average over every 16 samples is displayed.

The raw image is stored in a temporary buffer of expanded size, which allows fine adjustment of the slant and phase to be made before the image is sampled and displayed.

If the transmitting station goes off the air or reception is lost during recording of a picture, the recording will continue until the necessary number of samples has been made, then reception stops.

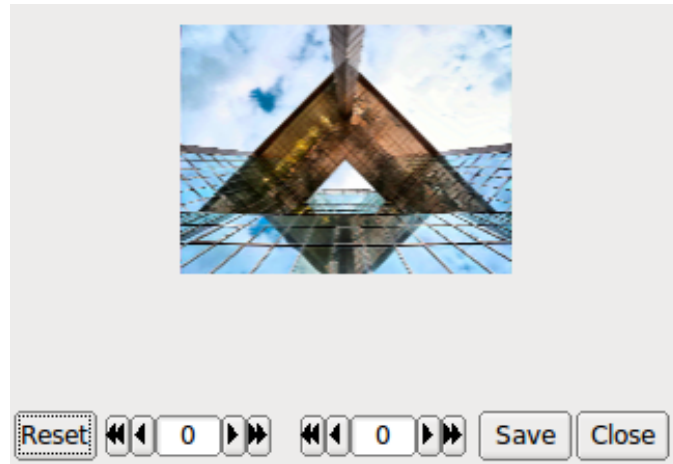


Figure 5.29: Image Reception Completed

The Rx Image dialog controls from left to right are:

- Reset: clears all rx data
- Phase adjustment: moves image left and right
- Slant adjustment: slews image diagonally
- Save: saves the image to the fldigi image folder with a date-time stamped file name
- Close: closes the image dialog

Image phase and slant should not be adjusted before the picture reception has finished, as then only part of the image will be corrected.

### 5.6.7 Avatars

Avatar images are 59 x 74 (width x height) color images that are associated with a call sign. The avatar images are stored in the folder

```
C:\Documents and Settings\\fldigi.files\avatars\  
C:\Users\\fldigi.files\avatars\  
/home/<user>/.fldigi/avatars/
```

as "png" images.

You should prepare an image file for your own call sign, in lower case. For example

```
/home/dave/.fldigi/avatars/w1hkj.png
```

in a 4x5 aspect ratio, preferably in the 59x74 format, but any larger 4x5 image can be used. fldigi will scale the image to 59x74 when it is read from the file.

You can send your avatar very simply by right clicking on the IFKP avatar.

If you receive an avatar from another station it will automatically be displayed in the avatar box. You should have previously entered the call sign of the other station in the Call logging control. Left clicking on the avatar will then save the image in the avatars folder with the appropriate callsign. The correct avatar will then be displayed the next time you enter that call sign in the Call logging control. This is an example of a received avatar with timing misalignment.

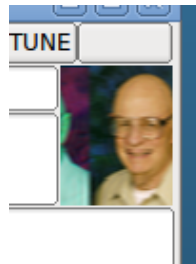


Figure 5.30: received avatar

The received image may be misaligned left or right in the image box. This is caused by an error in the start timing of the reception. You can realign the image using a combination of SHIFT, CONTROL and LEFT/RIGHT mouse clicks on the image. Be careful to use either the SHIFT or CONTROL in combination with the mouse button. Mouse left and right click without the SHIFT/CONTROL will cause either a save or an own avatar transmit to occur.

CONTROL LEFT click - move image left by 1 pixel  
 SHIFT LEFT click - move image left by 5 pixels

CONTROL RIGHT click - move image right by 1 pixel  
 SHIFT RIGHT click - move image right by 5 pixels

The same image after alignment.



Figure 5.31: received avatar

You will probably need to manage the avatar images using your OS file explorer. Easy access to the image folder is obtained by the menu item File/Folders.

The avatar image format is identical to the IFKP image 59x74 format. If you receive and save a 59x74 image you can always rename the saved image (date-time-stamped) to a "callsign.png" in the avatar folder.

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.7 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.

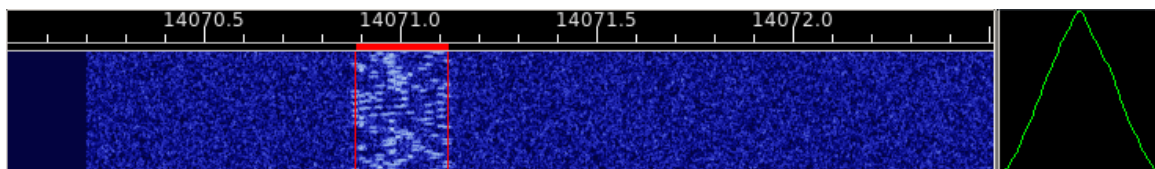


Figure 5.32: MFSK16 signal

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

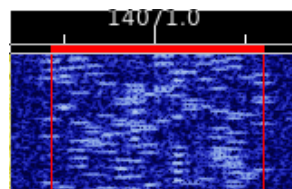


Figure 5.33: MFSK16 signal

### 5.7.1 MFSK Picture Mode

Fidigi 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.

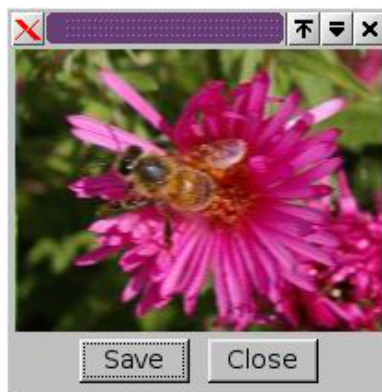


Figure 5.34: Received MFSK Image

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:

Time(sec) =  $W * H / 1000$  for black and white

Time(sec) =  $W * H * 3 / 1000$  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.



Figure 5.35: Picture received from K0OG

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).

## 5.7.2 Transmitting an Image



Figure 5.36: Xmit Picture Dialog box

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.



Figure 5.37: Xmit Picture Dialog Box with Image

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.



Figure 5.38: Received MFSK Image



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.

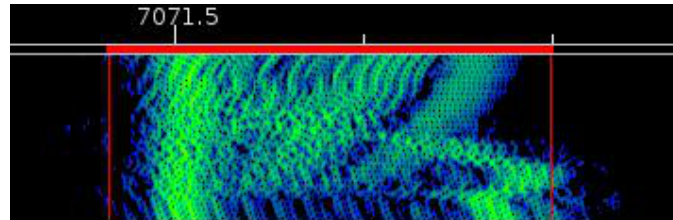


Figure 5.39: Waterfall of a MFSK Image

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.

### 5.7.3 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 MFSK pic'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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.8 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 fading. 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:



Mode	Symbol Rate	Typing Speed	Bandwidth
MT63-500	5.0 baud	5.0 cps (50 wpm)	500 Hz
MT63-1000	10.0 baud	10.0 cps (100 wpm)	1000 Hz
MT63-2000	20 baud	20.0 cps (200 wpm)	2000 Hz

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 stations 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 of 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.

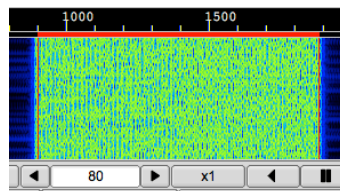


Figure 5.40: MT63-1000

You can also elect to operate the MT63 modem in a "manual tune" mode ([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:

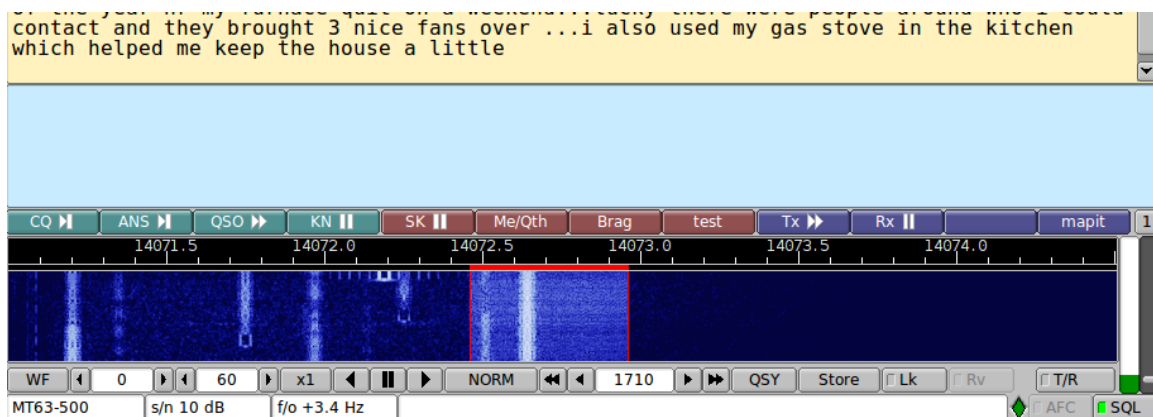


Figure 5.41: MT63-500 with QRM

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 Jalocho'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:

Bandwidth	Symbol Rate	Character Rate	Interleave / Char.
500 Hz	5 baud	5 char / sec	6.4 or 12.8 sec
1000 Hz	10 baud	10 char / sec	3.2 or 6.4 sec
2000 Hz	20 baud	20 char / sec	1.6 or 3.2 sec

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 side band 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. Signal synchronization is improved by filtering the timing measurement of the received signal. For very low signal to noise ratios you can improve the synchronization by selecting "Long Receive Integration" factor. Think of the process as a digital AFC. PSK31 does something very similar. One would expect there to be some correlation since mt63 is the equivalent of 64 simultaneous binary phase shift signals. The synchronizer filtering can be reduced to 1/2 it's normal value by selecting "Long Receive Integration". It takes twice as long for decoding to begin and the synchronizer is more immune to noise.

You probably never need to enable "Long Receive Integration" unless you are trying to receive an mt63 signal buried in the noise.

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 than  $10^{-4}$ .

If you have calibrated your sound card to [WWV](#), then you will meet this requirement.

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.9 NAVTEX and SITOR-B

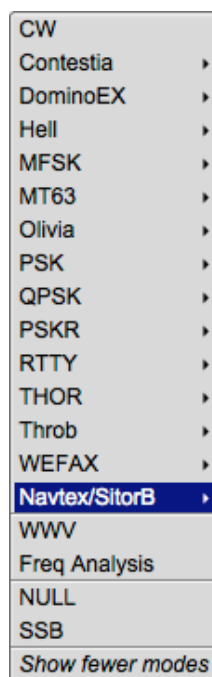


Figure 5.42: Op Mode Select

**NAVTEX** (Navigational Telex) is an international automated service for delivery of meteorological and marine safety information to ships. These broadcasts are sent with the **SITOR collective B-mode** (Also known as AMTOR-B or AMTOR-FEC), using the **CCIR 476** character set. SITOR-B is also used in **amateur radio**, where it is known as **AMTOR-B** or AMTOR-FEC.

It transmits at 100 baud FSK modulation with a frequency shift of 170 Hz. The frequencies are:

- 518 kHz : International frequency, always in English.
- 490 kHz : Regional transmission, in local languages (Not used in the United States).
- 4209.5 kHz : Marine Safety Information.

The implementation reflects this structure: The Navtex modem is a specialization of the SITOR-B modem. Fldigi supports both modes. We will specify when their behaviors are different.

### 5.9.1 Transmitting a text

Transmitting a text is done with the usual GUI. The only difference between the SITOR-B and Navtex modes, is that data (Whether sent with the GUI or with XML/RPC) are untouched in SITOR-B. In Navtex, on the contrary:

- They are preceded by a phasing signal of several seconds.
- A "ZCZC B<sub>1</sub>B<sub>2</sub>B<sub>3</sub>B<sub>4</sub>" preamble is sent.
- Then the original message is transmitted.
- ... followed by the "NNNN" terminator,
- ... and another phasing signal.

### 5.9.2 Receiving a text

The only difference between the SitorB and Navtex modes, is that messages are processed (Parsed and optionally stored) only in Navtex mode. Here is a typical transmission showing:

- The end of a previous message : "NNNN"
- The preamble of a new message: "EA85": 85th *navigational warning* ('A') of the Niton station in England ('E').
- The message itself, without the terminator (Which should come soon).

The modem has some flexibility when interpreting messages, and is able to deal with missing or incomplete preamble and terminator.

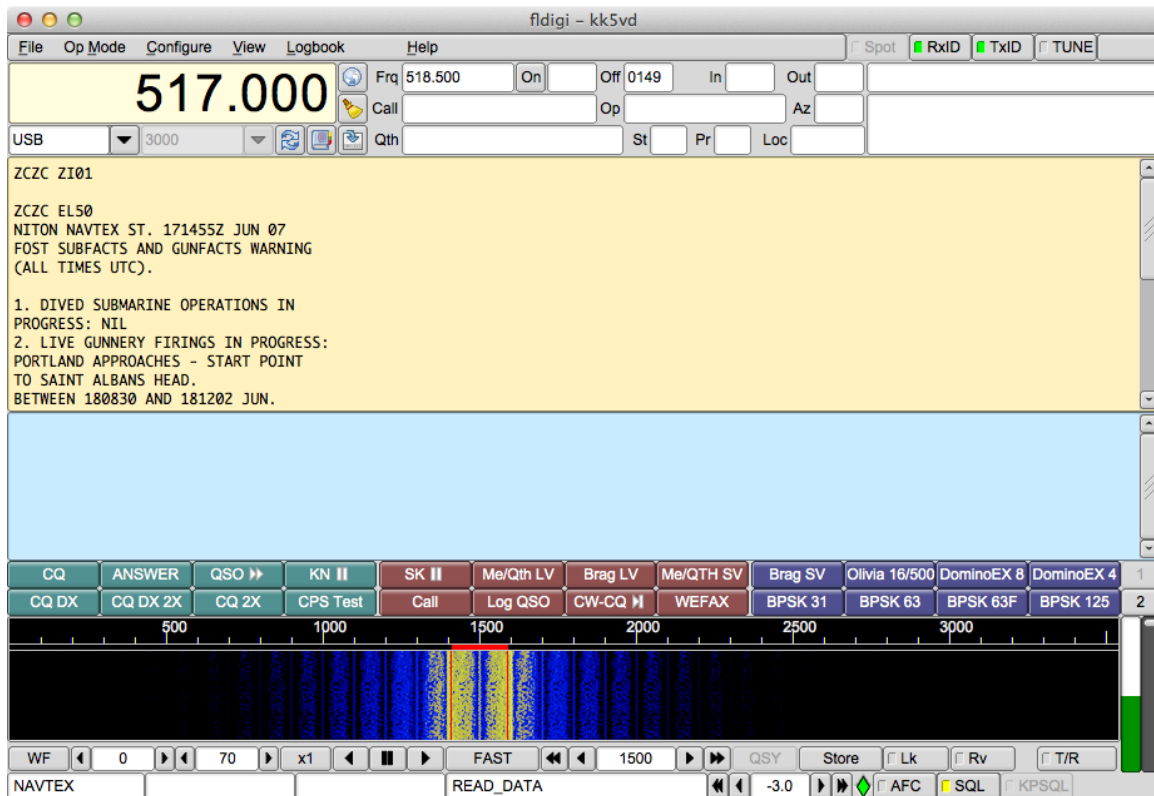


Figure 5.43: Navtex Reception

### 5.9.3 AFC: Automatic Frequency Control

As shown in the reception screen hard-copy, it is possible to tick "AFC". The consequence is that the frequency is continuously monitored.

### 5.9.4 Logging Navtex data to ADIF files

Navtex messages are delimited with the usual separators ZCZC and NNNN. Their format is:

```
ZCZC B1B2B3B4
(message text ...)
NNNN
```

These four characters are:

- B<sub>1</sub> : The station origin, used for lookups in the Navtex stations file. The same character is associated to several stations. Therefore, we use other criteria such as the frequency and the distance to the receiver, to eliminate the ambiguity. It is therefore important to specify correctly your Maidenhead locator.
- B<sub>2</sub> : The subject indicator, used by the receiver to identify the message class, such as ice reports or meteorological messages.
- B<sub>3</sub>B<sub>4</sub>: A serial number between 00 and 99.

### 5.9.5 Configuration

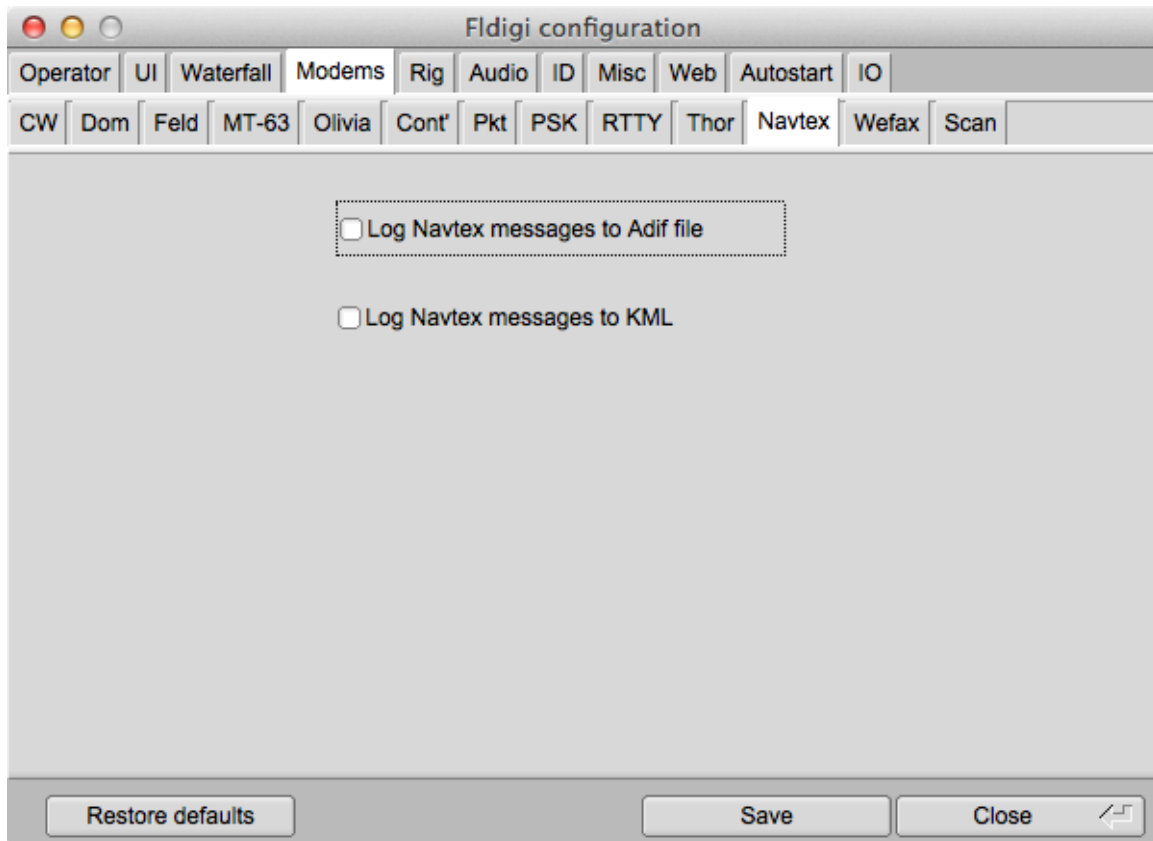


Figure 5.44: Navtex Configuration

As for any modem, there is a specific configuration tab for Navtex which allows:

- To specify whether messages should be stored in the current Adif log file.
- To choose the Text file containing stations definitions.



## 5.9.6 ADIF journal file

The screenshot shows the 'Logbook - Meteo.adif' window. The top section contains input fields for logging a contact: On Date (20120622), Heure (0801), Off Date (20120622), Heure (0801), Indicatif (EAV), and Nom (Valencia (Cabo de la Nao)). Below these are fields for Fréq. (1400.000000), Mode (TOR), In, Out, Tx Power, QSL-reçu, and QSL-env. Further down are fields for Qth (Spain), St, Pr, Pays (Spain), and Loc (JM09AC). There are also fields for County, IOTA, CQZ, TIUZ, CONT, and DXCC. A Notes field contains the text: "TC NOV ;~WZ 1196~SELF CANCELLING.~CANCEL WZ 1192 (GA92) (MA33).~WALKER LIGHTBUOY NORMAL CONDITIONS RESTORED." Below the notes are fields for Ser# out, Exchange Out, Ser# in (35), and Exchange In (Navigational warning). At the bottom left, there are buttons for 'Recs' (117), 'Nouveau', 'Mise à jour', 'Supprimer', and 'Dial'. On the right, there is a 'Call Search' field and two arrow buttons. The bottom section is a table of contacts:

Date	Heure	Indicatif	Nom	Fréquence	Mode
20120622	0759		Unknown station	1400.000000	TOR
20120622	0759	FRC	CROSS Corsen	1400.000000	TOR
20120622	0800	FRC	CROSS Corsen	1400.000000	TOR
20120622	0801	FRC	CROSS Corsen	30.000000	TOR
20120622	0801	EAV	Valencia (Cabo de la Nao)	1400.000000	TOR
20120622	0802		Unknown station	1400.000000	TOR

Figure 5.45: Logging Contacts

Navtex messages can be logged to the ADIF file. This is done with other extra data:

- Date and time of the contact.
- Frequency and mode.
- Station name, country, Maidenhead locator and callsign deduced from the message header and the Navtex stations file (See below). The locator is calculated using the station coordinates.
- The message content itself. Note that carriage-returns which cannot be displayed on a single line, are transformed into a tilde "~".

### 5.9.7 Stations file

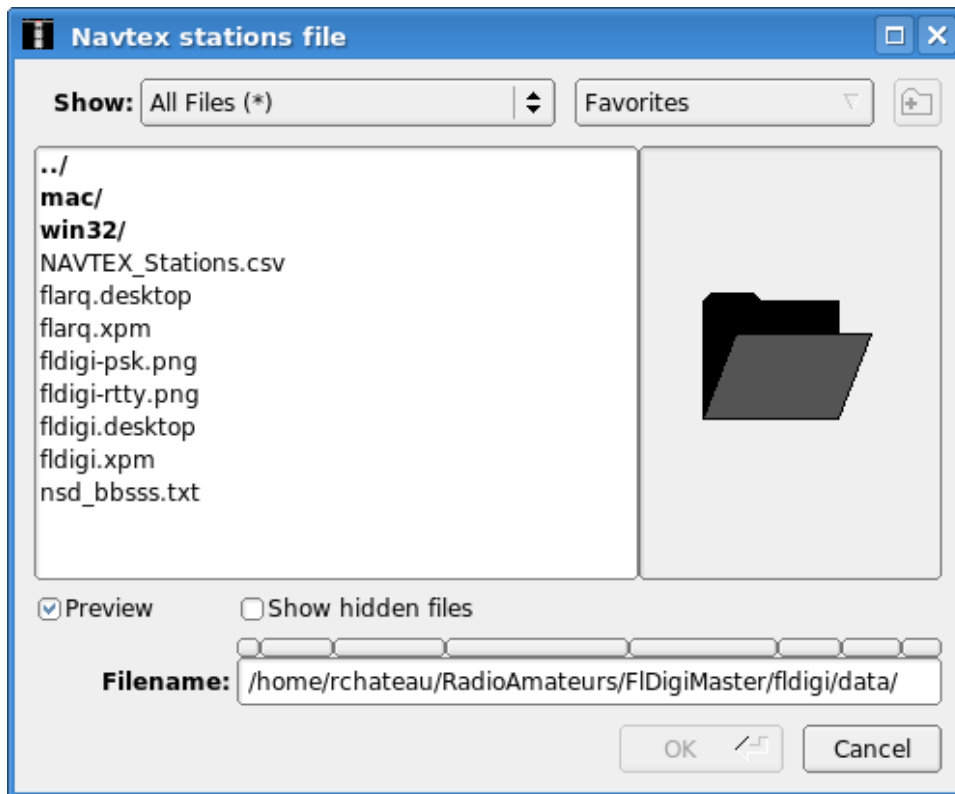


Figure 5.46: Choosing Stations File

The emitter station is extracted from the Navtex message, and used to extract latitude, longitude, station name and other characteristics from a file containing well-known station, data/NAVTEX\_Stations.csv. It is possible to edit this text file, or to choose another one.

### 5.9.8 XML/RPC functions

Two XML/RPC functions are create:

navtex.get_message	Returns next Navtex/SitorB message with a max delay in seconds passed as an integer parameter. Empty string if timeout.
navtex.send_message	Send a Navtex/SitorB message, passed as a string. Returns an empty string if OK otherwise an error message

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.10 Olivia

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

Mode	Symbol Rate	Typing Speed	Bandwidth
Olivia 8-250	31.25 baud	1.46 cps (14.6 wpm)	250 Hz
Olivia 8-500	62.5 baud	2.92 cps (29.2 wpm)	500 Hz
Olivia 16-500	31.25 baud	1.95 cps (19.5 wpm)	500 Hz
Olivia 32-1000	31.25 baud	2.44 cps (24.4 wpm)	1000 Hz

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:

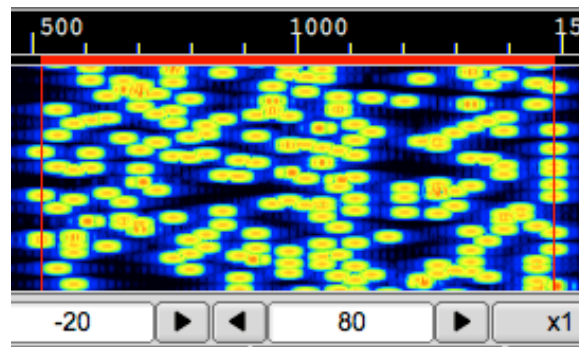


Figure 5.47: Olivia 32/1000

### Excerpts from the web pages of [Gary, WB8ROL](#)

Olivia 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. It doesn't do much good to use a digital mode like Olivia that can decode signals -14 db below the noise lever 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 settings 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 Periods 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 for his turn – 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 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 about 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 identify 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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.11 PSK Single and Multi-Channel Modems

### 5.11.1 BPSK, QPSK, 8PSK modems

PSK are narrow band low to moderate symbol rate modes using either single or multiple carrier differential **Phase Shift Keying**.

Current supported variants:

BPSK: Binary, 2 constellations (1)

QPSK: Quadrature, 4 constellations

8PSK: Octal, 8 constellations

(1)

A convenient way to represent PSK schemes is on a constellation diagram. This shows the points in the complex plane where, in this context, the real and imaginary axes are termed the in-phase and quadrature axes respectively due to their 90° separation.

**NOTE:**

Multi-Channel PSK modems use BPSK modulation.

PSK63FEC and the PSKxxxR modes are forward error correcting modes. PSK63FEC is compatible with the Multi↔ Psk 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 used 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 straight forward 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 and 8PSK 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.

Many of the multi-carrier and 8PSK modes exceed the baud and bandwidth limitations imposed by the FCC (US operators only). These modes are intended for use on VHF/UHF and have proven to be very robust on FM even when used with repeaters.

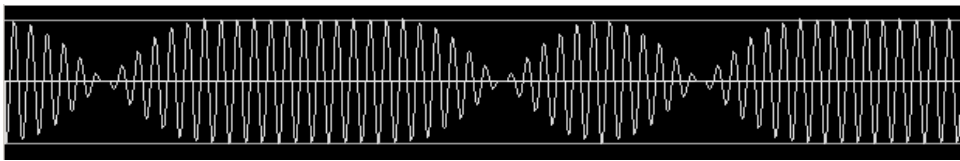


Figure 5.48: PSK63 signal transmitting text data - oscilloscope view

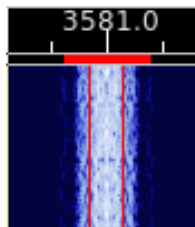


Figure 5.49: PSK63 signal transmitting text data - waterfall view

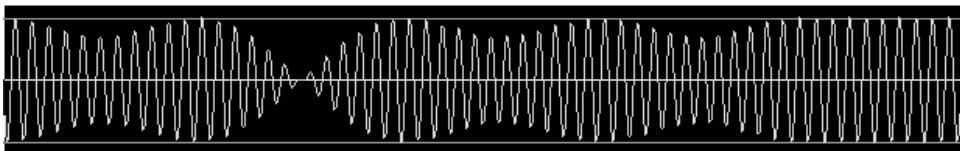


Figure 5.50: QPSK63 signal transmitting text data - oscilloscope - waterfall view

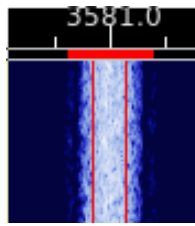


Figure 5.51: QPSK63 signal transmitting text data - waterfall view

### 5.11.2 Multi-Channel BPSK modems

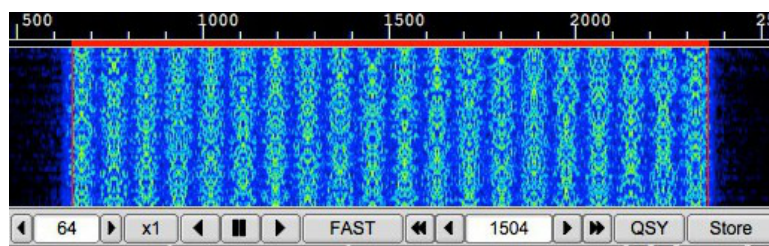


Figure 5.52: PSK63R20C signal transmitting text data - waterfall view

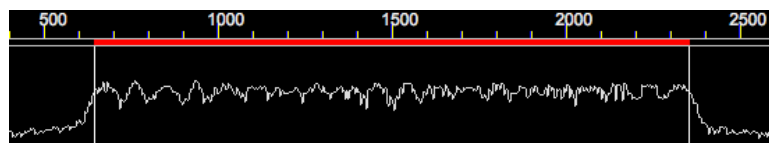


Figure 5.53: PSK63R20C signal transmitting text data - oscilloscope view

### 5.11.3 8PSK modems

The 8 PSK modes are intended for use on VHF/UHF FM systems. They provide a very high data rate suitable for use with both flmsg and flamp and the transfer of digital data.

Mode	Baud	WPM	Mode-FEC	Baud	WPM
8PSK125	125	635	8PSK125F	125	317
8PSK250	250	1270	8PSK250F	250	635
8PSK500	500	2540	8PSK500F	500	1690
8PSK1000	1000	5080	8PSK1000F	1000	3386

-----	-----	-----	8PSK1200F	1200	4170
-------	-------	-------	-----------	------	------

The FEC modes do not all use the same Viterbi polynomials to achieve the forward error correction. That is why the WPM rates for the FEC modes are not multiples of the base 125 baud. The WPM rates are only an indication of relative rates. The actual transfer rate is highly dependent on the data content. This is always true for modes which use a VARICODE.

The 8psk signal is similar to both bpsk and qpsk, but with 8 possible phase states instead of the 2 and 4 associated with bpsk, qpsk. The format of the signal does not lend itself easily to a conventional AFC. Instead, the modes should be used with RsID enabled. The RsID signal will both determine the mode and the mode center frequency (to the nearest 2.6 Hz). A finer resolution of the mode center frequency can be made using the optional pilot carrier. This pilot carrier is placed at the frequency

$f_0 - \text{samplerate} / \text{symbolen}$ ,  $f_0$  is modem center frequency

$\text{samplerate} / \text{symbolen}$  is the same as the mode bandwidth. The waterfall and spectrum signature for the 8psk500 mode is shown here with and without the pilot carrier:

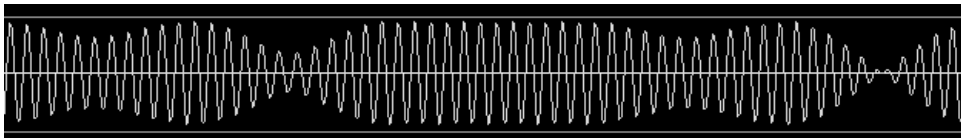


Figure 5.54: 8PSK125 signal transmitting text data - oscilloscope - waterfall view

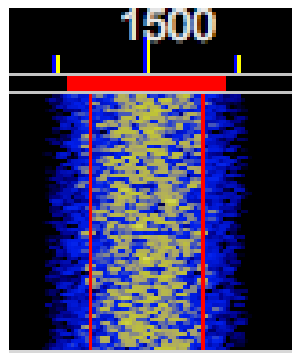


Figure 5.55: 8PSK125 signal transmitting text data - waterfall view

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



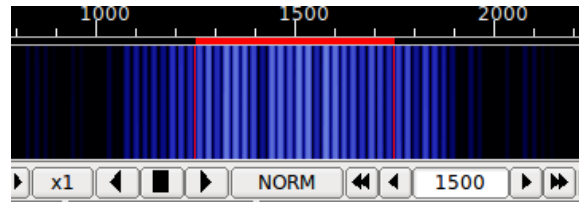


Figure 5.56: 8PSK500F idle signal without pilot

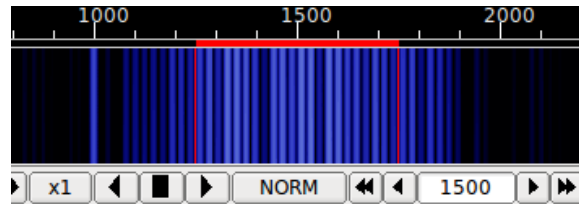


Figure 5.57: 8PSK500F idle signal with pilot

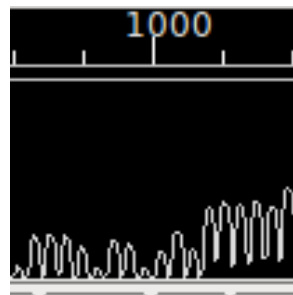


Figure 5.58: 8PSK500F idle signal without pilot

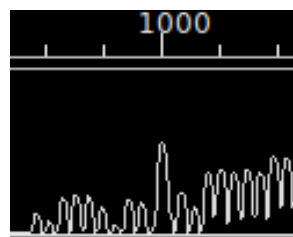


Figure 5.59: 8PSK500F idle signal with pilot

Decoding errors are reduced as the tracking point nears the actual transmit center frequency. The loss of signal power is more than offset by the decoder improvement. Field testing has shown that the pilot tone needs to be at

-40 dB or greater (less negative).

Pilot tone detection does require a bit more cpu power. The pilot tone is detected using a sliding fast Fourier transform, `sfft`, which computes the frequency of the pilot to approximately 1 Hz resolution. The `sfft` only evaluates the signal at 11 discrete frequencies, so it is necessary that either the `RslD`, or manual tuning is used for the initial signal acquisition. The detector is set to provide lock when the pilot s/n is 2:1 or better. The signal tracking point is then adjusted to place the pilot tone at the correct frequency location. For example, if the `RslD` put the tracking point at 1502 Hz, the pilot would then adjust for 1500 (if that is the correct tracking frequency). The adjustment is made once per second. Unlike AFC, which is continuous, the pilot adjustment is discrete and occurs in 1 Hz steps. If the pilot s/n is less than 2:1 then no adjustment is made to the tracking point. The pilot tone is transmitted during the 8psk preamble as well as during the data transmission. You should see the tracking point adjust once at the beginning of the transmission and then stay fixed.

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. 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.

For further information about [Phase Shift Keying](#)

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.12 RTTY

The RTTY modulator and demodulator have been extensively changed with version 3.21.67. The new design was a cooperative effort of Stefan, DO2SMF, and Dave, W1HKJ with extensive testing performed by Ed, W3NR, and Dick, AA5VU. Chen, W7AY, was a silent contributor to the design by virtue of his excellent technical papers on RTTY modulation and demodulation, which he so generously placed in the public domain.

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

Mode	Symbol Rate	Typing Speed	Bandwidth
RTTY 45	45.45 baud	6.0 cps (60 wpm)	270 Hz

RTTY 50	50.0 baud	6.6 cps (66 wpm)	280 Hz
RTTY 75	75.0 baud	10.0 cps (100 wpm)	370 Hz

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.

### 5.12.1 RTTY modulator

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. Each of the generated Mark / Space signals are on-off-keyed (OOK), bandwidth limited signals. The resultant waveform is not an FM type signal of constant amplitude. Therefore the **transmit audio and RF amplifiers must be linear**, just like the requirement for PSK signals. There are performance gains using this approach. The principal being a reduction in inter symbol interference which gives much improved performance by the receiver. The waterfall, time domain, and spectrum signatures of the transmitted signal look like this:

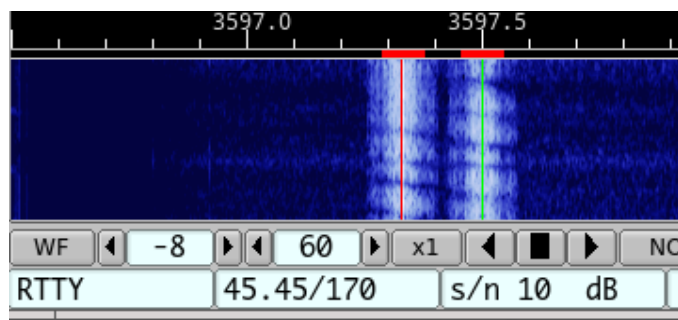


Figure 5.60: W1AW on air signal

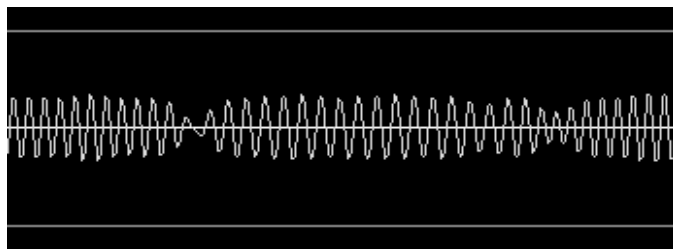


Figure 5.61: AFSK signal

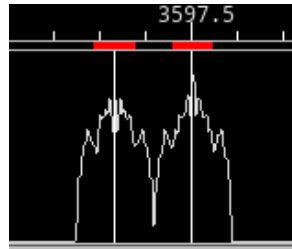


Figure 5.62: Spectrum

You must operate your transceiver in the USB mode for the fldigi RTTY signal to be the correct polarity. If your transceiver is set to LSB then use the fldigi "Rev" button to reverse the sense of the mark and space signals.

You must maintain transmitter LINEARITY in the AUDIO AMPLIFIERS. Do not think that you can improve performance by over driving the audio input. A good operating procedure for most transceivers is the set the audio level to the level for which there is just barely a hint of ALC. Then reduce the input to 80% of that power level. Over driving an AFSK signal is as disastrous as over driving a PSK signal. This is an actual on air signal that was being over driven (but not on purpose):

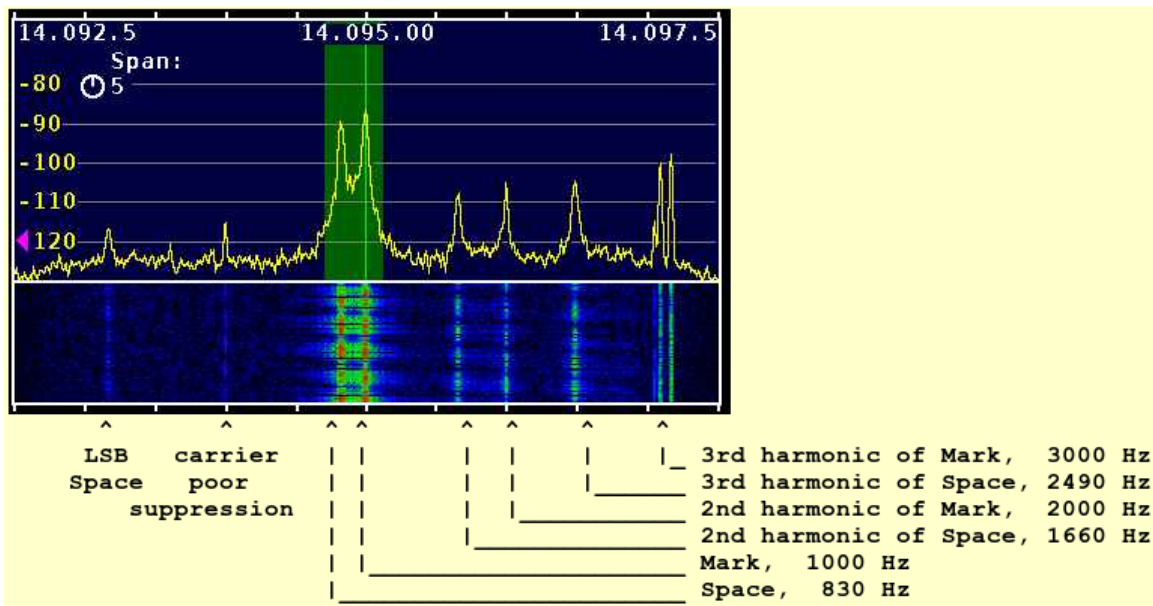


Figure 5.63: Overdriven RTTY Signal

Joe also performed a series of tests on an Icom 706 mkIIg transceiver. The results of those tests are very enlightening.

"Two views - the 2 KHz span and a 10 KHz span. The 10 KHz span shows the one failing of the IC-706mkIIg and other rigs with analog modulation - opposite sideband and carrier leakage. This one isn't too bad but one can see carrier at -50 dBc and opposite sideband at -55 dBc +/- . I do use a high audio frequency to minimize harmonic issues.

For fun I've attached versions at 70 W in 10K, 5K, and 2K spans. The narrow spans clearly show the benefits of reducing the audio until output power drops 1.5 dB.

Audio was connected to the IC-706mkIIg via the "DATA" jack rather than the mic connector or "Mod In" pin of the ACC jack. Using this input avoids several potential problems - including the constant swapping between mic and digital connections and remembering to turn off the compressor when switching to digital operation. The "Data" input is also 6 dB less sensitive than "Mod in" making it that much less likely that one will significantly over drive the the transceiver and create distortion in the audio stages ahead of the modulator".

The green area is 400 Hz wide.

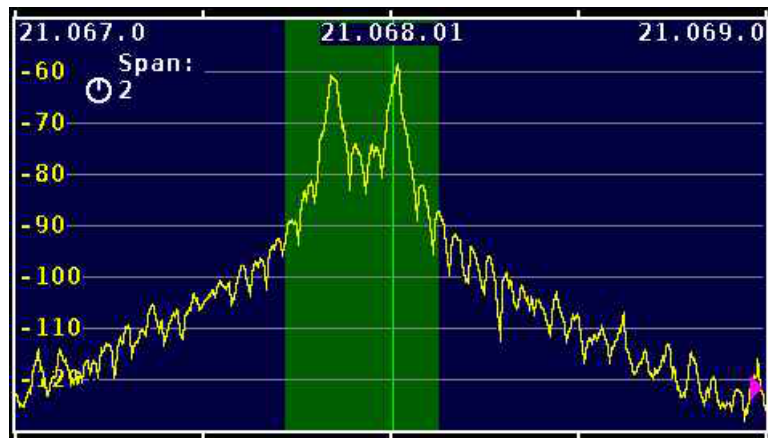


Figure 5.64: Image A

Transceiver operated in FSK mode  
Power: 100 W

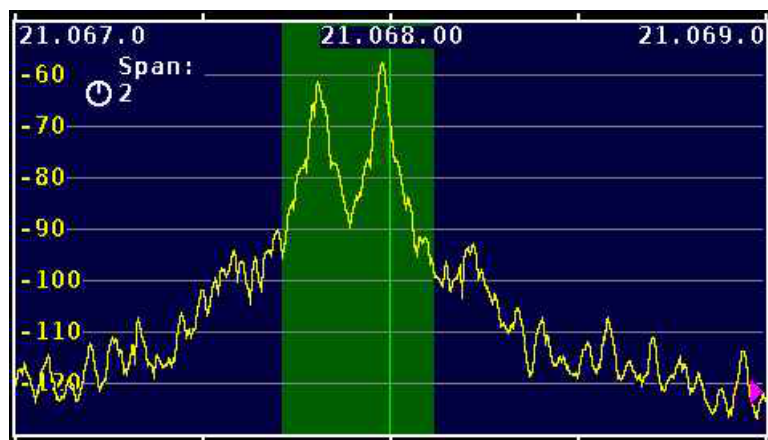


Figure 5.65: Image B

Transceiver in USB, fldigi AFSK audio drive  
Space at 2125, Mark at 2295 Hz  
Power: 100 Watts, ALC just extinguished

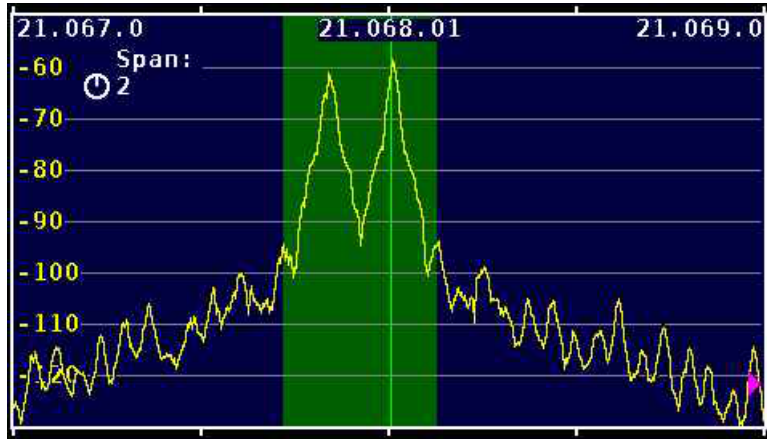


Figure 5.66: Image C

Transceiver in USB, fldigi AFSK audio drive  
Space at 2125, Mark at 2295 Hz  
Power: reduced to 80 Watts (-1 dB)

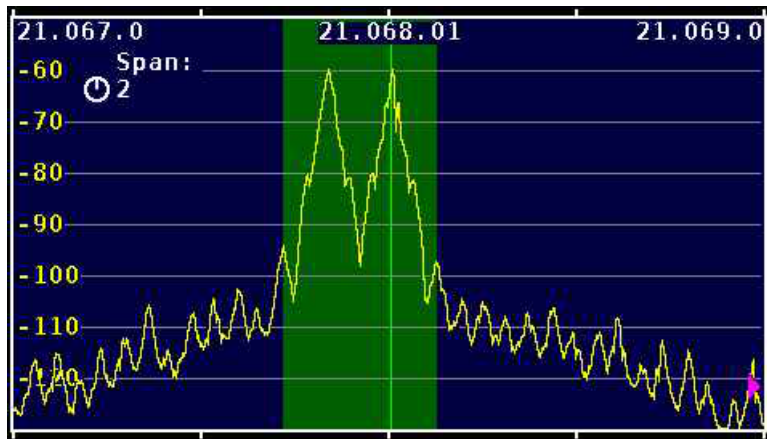


Figure 5.67: Image D - 2 K span

Transceiver in USB, fldigi AFSK audio drive  
Space at 2125, Mark at 2295 Hz  
Power: reduced to 70 Watts (-1.5 dB)

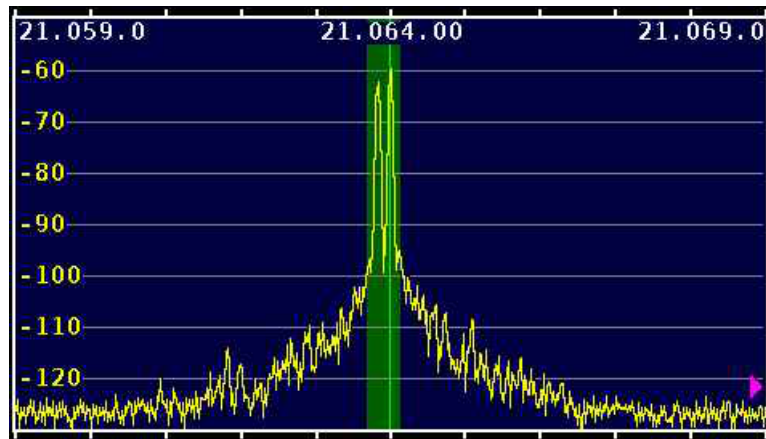


Figure 5.68: Image F - 10 K span

Transceiver in USB, fldigi AFSK audio drive  
Space at 830 Hz, Mark at 1000 Hz  
Power: 70 Watts. The LSB leakage is clearly  
seen at approximately -55 dB

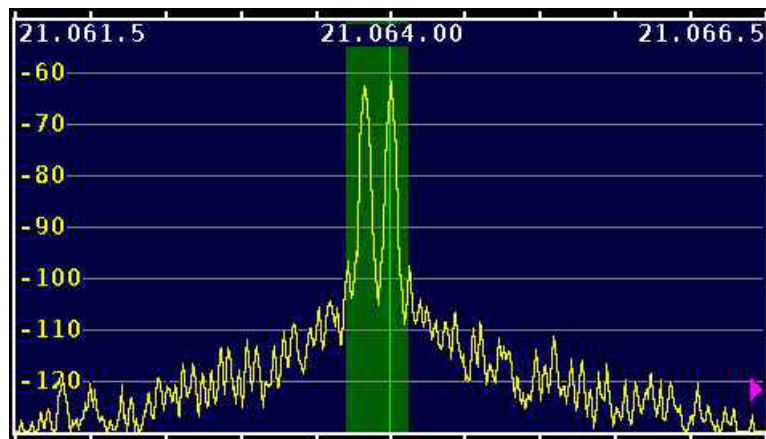


Figure 5.69: Image G - 5 K span

Transceiver in USB, fldigi AFSK audio drive  
Space at 830 Hz, Mark at 1000 Hz  
Power: 70 Watts. The LSB leakage is clearly  
seen at approximately -55 dB.

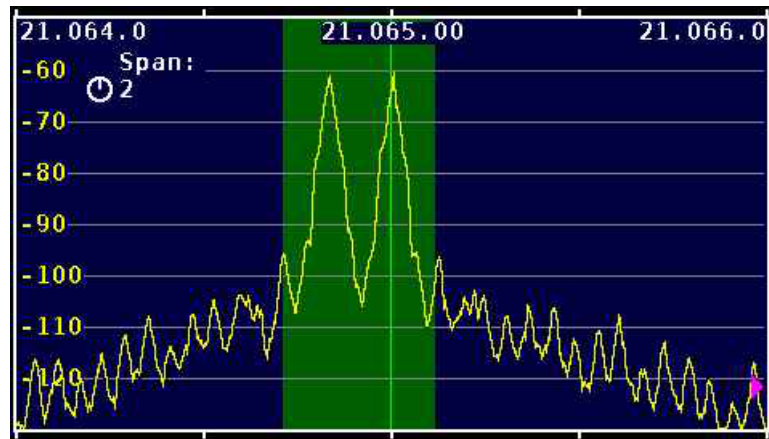


Figure 5.70: Image H - 2 K span

Transceiver in USB, fldigi AFSK audio drive  
 Space at 830 Hz, Mark at 1000 Hz  
 Stop Bits set to 2 vice 1.5  
 Power: 70 Watts. Compare this to image D

### 5.12.2 RTTY demodulator

Fldigi's demodulator uses a design based on theoretical work published by Kok Chen, W7AY, <http://www.w7ay.net/site/Technical/ATC/>.

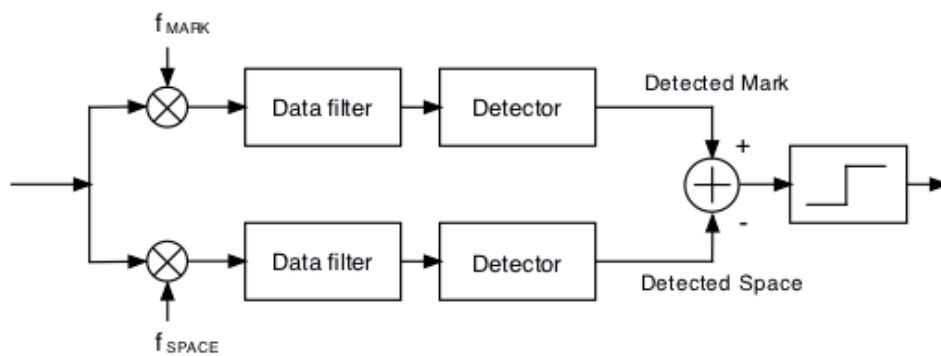


Figure 5.71: Demodulator

The mark and space signals are converted to base band and then filtered in a low pass filter. Each filter is a variant of Chen's enhanced Nyquist filter. It is implemented using a Fast Overlap-and-Add Fourier Transform.

Each time the baud rate is selected the program must "rebuild" the digital RTTY filter. The filter parameters are optimized for the baud rate.



The detector is an optimized Automatic Threshold Correcting (ATC) type described in Chen's paper.

To start decoding a signal simply left click on the signal and the AFC should lock on to the signal.

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

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.

See [RTTY Configuration Page](#)

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.13 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](#).

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 audible 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 without the need for AFC. The AFC control does not alter the decoder in any way.

The fldigi implementation of the Thor modem includes the ability to send and receive images and avatars. The default avatar is the "Tux" logo. Sending, receiving and saving avatars is discussed in the avatar section.

The waterfall and digiscope will appear as:

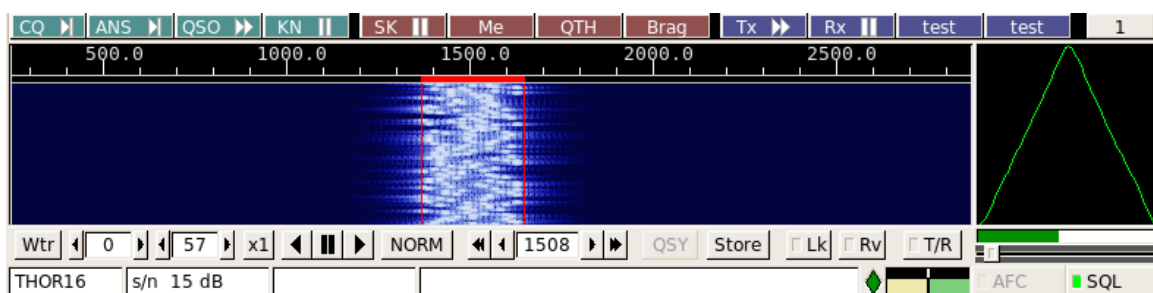


Figure 5.72: Thor

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).

### 5.13.1 Thor Image Mode

fldigi-thor can send and receive images and avatars when the modem is set to 11, 16 or 22 baud.

Small digital images are selected from those stored on the disk. The image transmission is similar to but not compatible with the IFKP image mode. Thor images are always transmitted in a fixed width to height ratio. The W x H format is not limited to a single ratio, nor is it limited to just color, or just gray scale. Thor transmits the image in Blue-Green-Red order. The thor image bandwidth is the same as the signal bandwidth for for Thor mode in use.

The start of the picture is signalled digitally, and the receiver simply captures data continuously for W lines of H pixels, as requested, resulting in a WxH pixel image, then reception stops.

A brief period of silence, followed by a sync signal will be transmitted at the beginning of each image or avatar. The sync signal is used to align the start of the received image. This is the equivalent of a vertical sync signal in an analog television signal. There is no equivalent of horizontal sync, therefore there is no risk of the picture tearing due to timing changes, or breaking up during selective fades. All that happens is that the image may noise up momentarily, but will stay correctly aligned.

#### Image Format

The images are approximately in 4:3 or 3:4 aspect ratio. thor images can be received by any thor capable decoder.

Available image aspect ratios / formats include:

- 59x74 color
- 160x120 color
- 320x240 color
- 640x480 grey scale
- 640x480 color
- 240x300 color
- 240x300 grey scale
- 120x150 color
- 120x150 grey scale

These formats correspond roughly to a portrait (3x4) and landscape (4x3) photo. fldigi will scale any computer stored image to the target image. You should edit the images off line to be at least close to the 3x4 or 4x3 ratio. That will prevent the fldigi loader from distorting the loaded image.

The image mode sampling rate is the same as the base baud samplerate:

- Thor-11 11025 samples/sec
- Thor-16 8000 samples/sec
- Thor-22 11025 samples/sec

Each pixel consists of 10 samples, and there are P pixels/line. Therefore the line period is  $10 \times P/\text{sample-rate}$  seconds, or the line rate  $\text{sample-rate}/(10 \times P)$  lines/sec. In colour, each line is sent three times, in Red-Green-Blue (RGB) order.

The is a single sync signal sent at the beginning of each image is a tone placed 186 Hz below the center frequency. The sync signal is 20 symbol lengths of silence followed by 20 symbol lengths of the tone.

If the sound cards in the transmitting and receiving computers differ markedly in sample rate, the resulting picture can be skewed. A small timing difference is enough to cause a noticeable slant in the picture. This is easily fixed after reception using the Slant control. High quality sound cards should be better than 5ppm, and result in barely noticeable slant, but some cheaper computer built-in sound modules may be rather worse. If you plan to send thor images you should calibrate your sound card to WWV. There should be no slant if both the transmit and receive sound cards are calibrated to WWV

### Transmit Image

Transmitting an image in thor is initiated by selecting the "Send image" menu item from the pop up Tx menu. Right click on the Tx panel

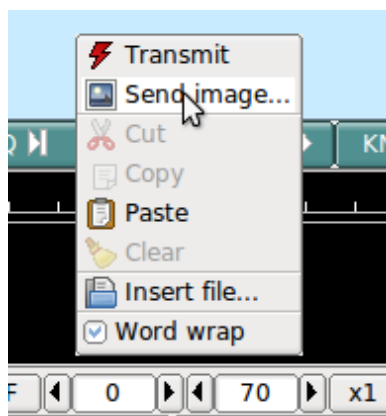


Figure 5.73: Send Image

This selection opens up the Send Image dialog



Figure 5.74: Send Image Dialog

shown with a 160x120 color image loaded and ready to transmit.

Transmission begins when you press the "Xmt" button. fldigi will insert the text preamble and immediately begin the image transmission. fldigi returns to the receive mode when the image transmission is completed.

### Receive Image

Reception is completely automatic. The decoder will identify the picture start, and record the picture. In doing so, it automatically opens a separate "thor Rx Image" dialog.

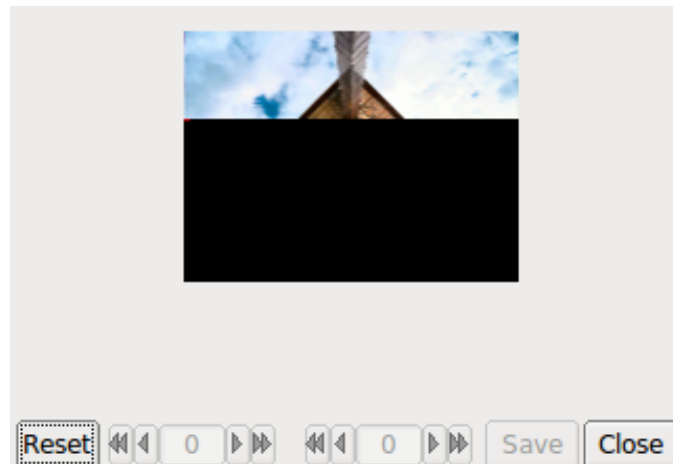


Figure 5.75: Image Reception

Demodulation uses a phase detection algorithm based on quadrature demodulation using the current and previous samples. All the samples are converted to pixels and stored, but only the average over every 10 samples is displayed.

The raw image is stored in a temporary buffer of expanded size, which allows fine adjustment of the slant and phase to be made before the image is sampled and displayed.

If the transmitting station goes off the air or reception is lost during recording of a picture, the recording will continue until the necessary number of samples has been made, then reception stops.

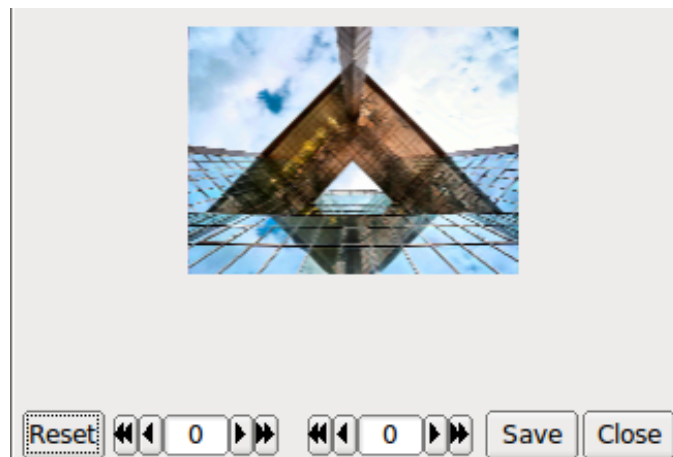


Figure 5.76: Image Reception Completed

The Rx Image dialog controls from left to right are:

- Reset: clears all rx data
- Phase adjustment: moves image left and right
- Slant adjustment: slews image diagonally
- Save: saves the image to the fldigi image folder with a date-time stamped file name
- Close: closes the image dialog

Image phase and slant should not be adjusted before the picture reception has finished, as then only part of the image will be corrected.

### 5.13.2 Avatars

Avatar images are 59 x 74 (width x height) color images that are associated with a call sign. The avatar images are stored in the folder

```
C:\Documents and Settings\\fldigi.files\avatars\  
C:\Users\\fldigi.files\avatars\  
/home/<user>/.fldigi/avatars/
```

as "png" images.

You should prepare an image file for your own call sign, in lower case. For example

```
/home/dave/.fldigi/avatars/wlhkj.png
```

in a 4x5 aspect ratio, preferably in the 59x74 format, but any larger 4x5 image can be used. fldigi will scale the image to 59x74 when it is read from the file.

You can send your avatar very simply by right clicking on the thor avatar.

If you receive an avatar from another station it will automatically be displayed in the avatar box. You should have previously entered the call sign of the other station in the Call logging control. Left clicking on the avatar will then save the image in the avatars folder with the appropriate callsign. The correct avatar will then be displayed the next time you enter that call sign in the Call logging control. This is an example of a received avatar with timing misalignment.

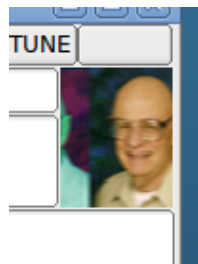


Figure 5.77: received avatar

The received image may be misaligned left or right in the image box. This is caused by an error in the start timing of the reception. You can realign the image using a combination of SHIFT, CONTROL and LEFT/RIGHT mouse clicks on the image. Be careful to use either the SHIFT or CONTROL in combination with the mouse button. Mouse left and right click without the SHIFT/CONTROL will cause either a save or an own avatar transmit to occur.

```
CONTROL LEFT click - move image left by 1 pixel  
SHIFT LEFT click - move image left by 5 pixels
```

```
CONTROL RIGHT click - move image right by 1 pixel  
SHIFT RIGHT click - move image right by 5 pixels
```

The same image after alignment.



Figure 5.78: received avatar

You will probably need to manage the avatar images using your OS file explorer. Easy access to the image folder is obtained by the menu item File/Folders.

The avatar image format is identical to the thor image 59x74 format. If you receive and save a 59x74 image you can always rename the saved image (date-time-stamped) to a "callsign.png" in the avatar folder.

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.14 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:

### 5.14.1 Throb baud rates and tone spacings

Mode	Symbol Rate	Typing Speed	Bandwidth
THROB1	1.0 baud	1.0 cps (10 wpm)	72 Hz
THROB2	2.0 baud	2.0 cps (20 wpm)	72 Hz
THROB4	4.0 baud	4.0 cps (40 wpm)	144 Hz

THROBX1	1.0 baud	1.0 cps (10 wpm)	94 Hz
THROBX2	2.0 baud	2.0 cps (20 wpm)	94 Hz
THROBX4	4.0 baud	4.0 cps (40 wpm)	188 Hz

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.15 WEFAX

This modem is able to receive and transmit HF-Fax images, traditionally used for weather reports.

More technical information is available on the wikipedia article [Radiofax](#).

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

Many frequencies are available at <http://www.hffax.com/> for example.

When entering any Wefax mode, the reception window opens, and optionally the transmit window. It is always possible to bypass this with the menu bar:

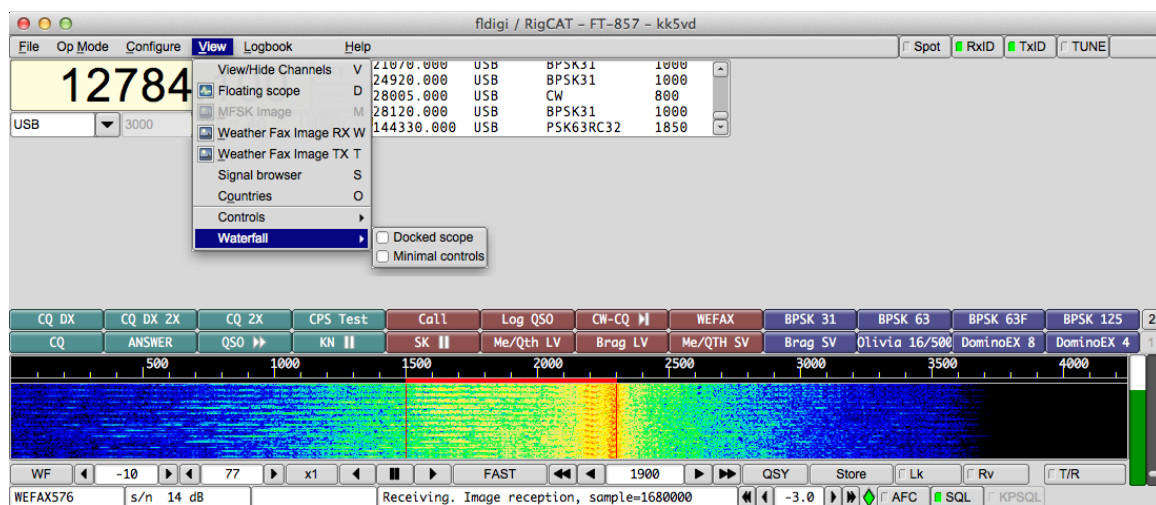


Figure 5.79: WEFAX

### 5.15.1 Configuration.

As for any modem, weather fax has its own tab in the configuration window.

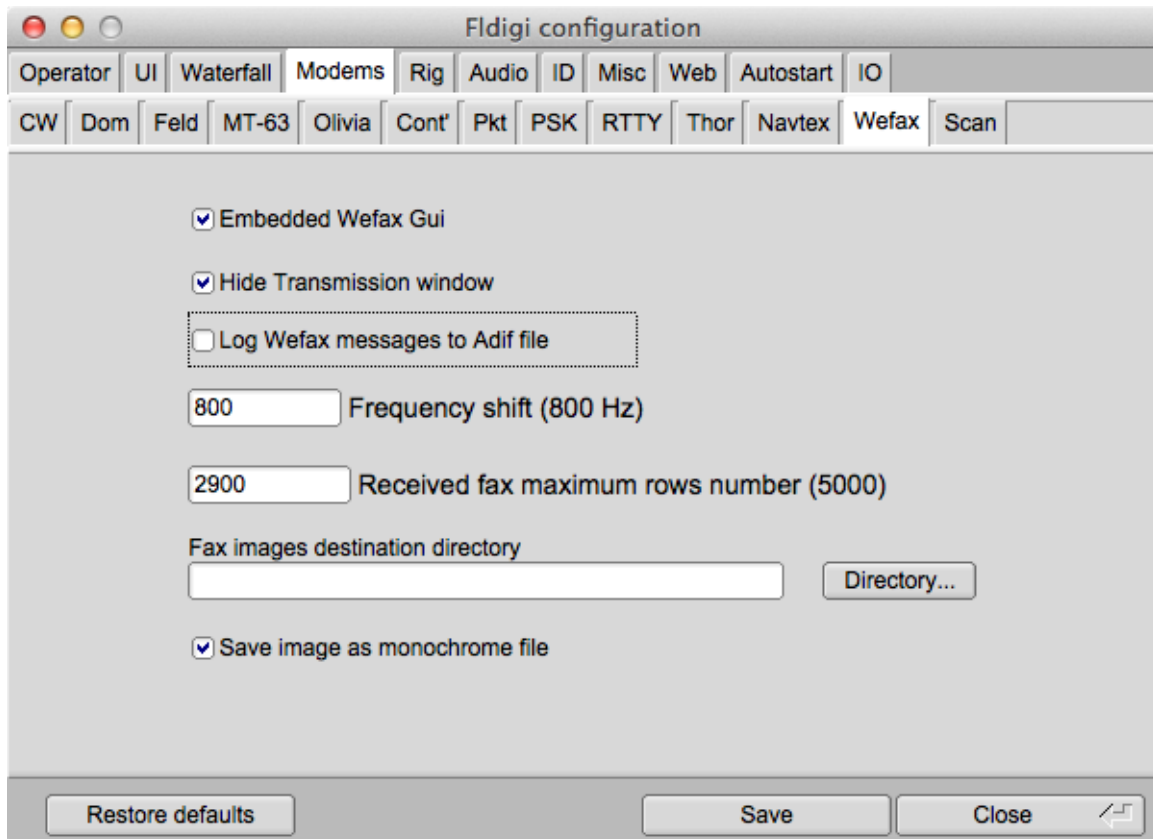


Figure 5.80: WEFAX Configuration

#### 5.15.1.1 Embedded vs floating display mode

There are two display modes for this modem:

- Embedded mode: This is the default mode, the normal receipt and transmit windows gets graphic (Like Hellschreiber mode).
- Floating mode: There are two separate windows for transmission and reception. This was the only available mode until fldigi 3.21.49.



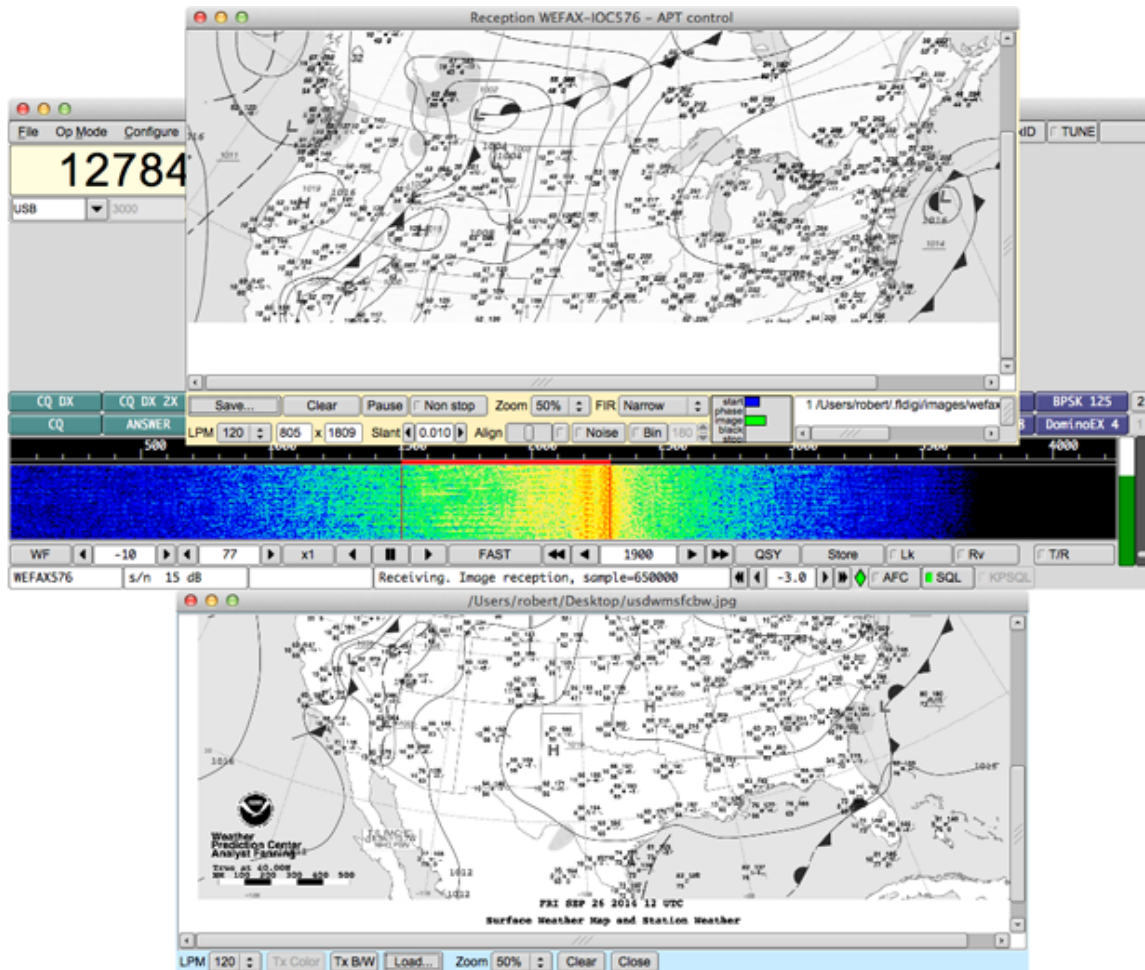


Figure 5.81: Detached WEFAX Window

#### 5.15.1.2 Hide transmission window.

This option, allows to closed by default, the transmission window, when entering Wefax mode. In embedded mode, this means that the entire window is used for reception (Most common mode). In floating mode, this implies that the transmission window is not opened by default. However, it is always possible to manually open or close the transmission window at any time.

#### 5.15.1.3 Logging messages to ADIF file

Each time an image is saved, it is possible to log this event, with the frequency and reception time, to the Adif file. This option is disabled by default.

#### 5.15.1.4 Frequency shift adjustment

The default frequency is 800 Hz. However, it is possible to adjust for example to 850 Hz for Deutsche Wetter Dienst.

#### 5.15.1.5 Maximum rows number

In non-continuous (Non non-stop) reception mode, an image is automatically saved when it has more than this number of lines (Default 2500 lines). Once this number of rows is reached, the image is saved and a new image is read with the same parameters. This feature has two applications:

- In automatic mode (APT control), if an image end is not detected, we can guarantee that the result will take no more than, for example, the size of two faxes. Typical faxes have about 1300 lines, so the max lines parameters can be tuned to, for example, 200 lines.
- In manual mode, where images are read continuously, this cuts the received images into chunks of equal size.

#### 5.15.1.6 Destination directory for saved images

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.

#### 5.15.1.7 Monochrome images

Fax images are monochromes and are saved as such by default. However, it is possible to bypass this parameter and save them as color RGB images.

### 5.15.2 Transmitting an Image

To open the transmit window, you must of course select one of the two Wefax modems, and then right-click on the transmit (blue) window:

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:

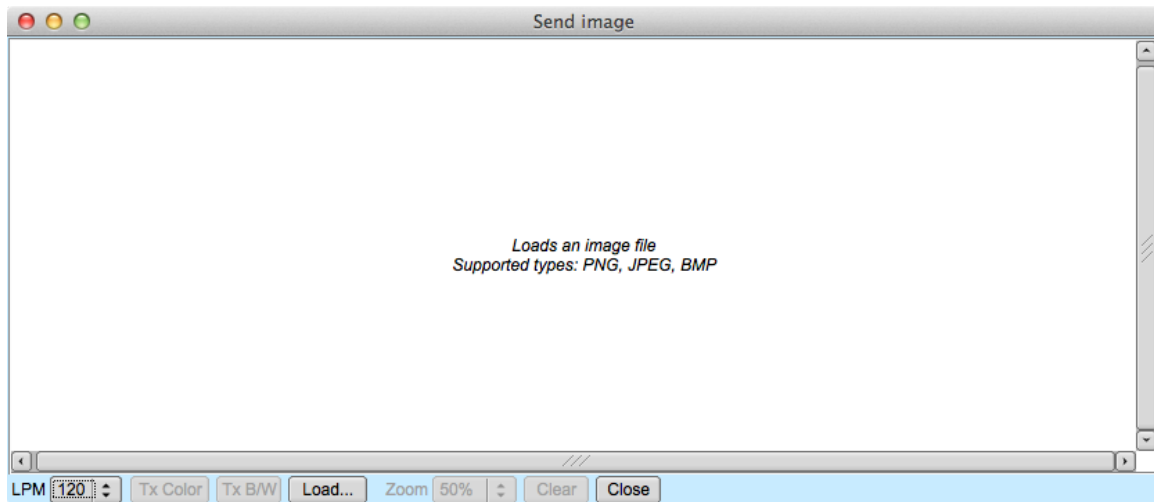


Figure 5.82: Transmitting

Now, to start the transmission, you just need to click "Tx B/W" 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 FIDigi status line displays the estimation transmit time, and the current stage (Start, phasing etc...); Color transmission ('Tx Color') is intentionally disabled at the moment.

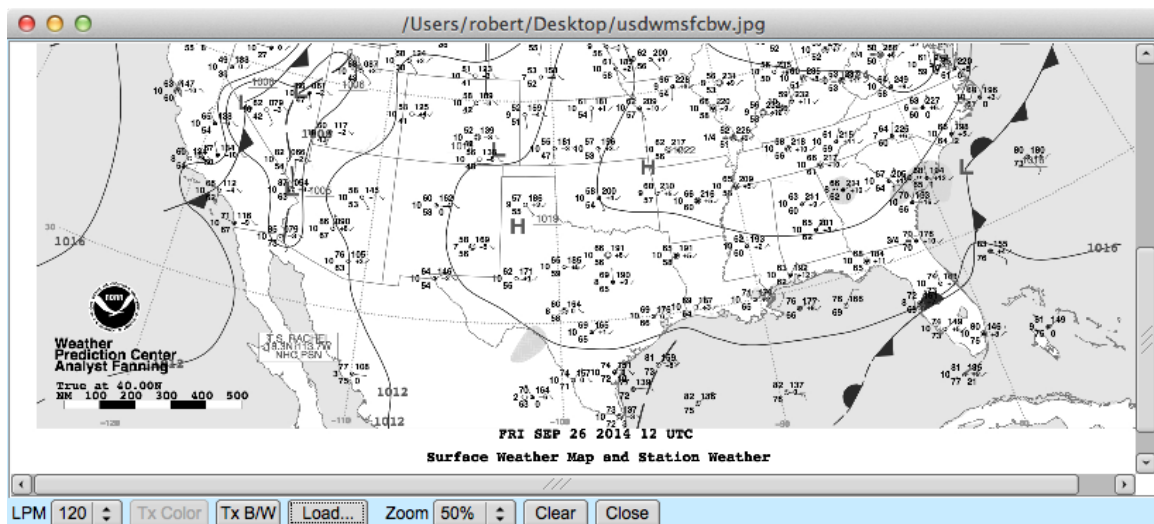


Figure 5.83: Sending

### 5.15.3 Receiving an image

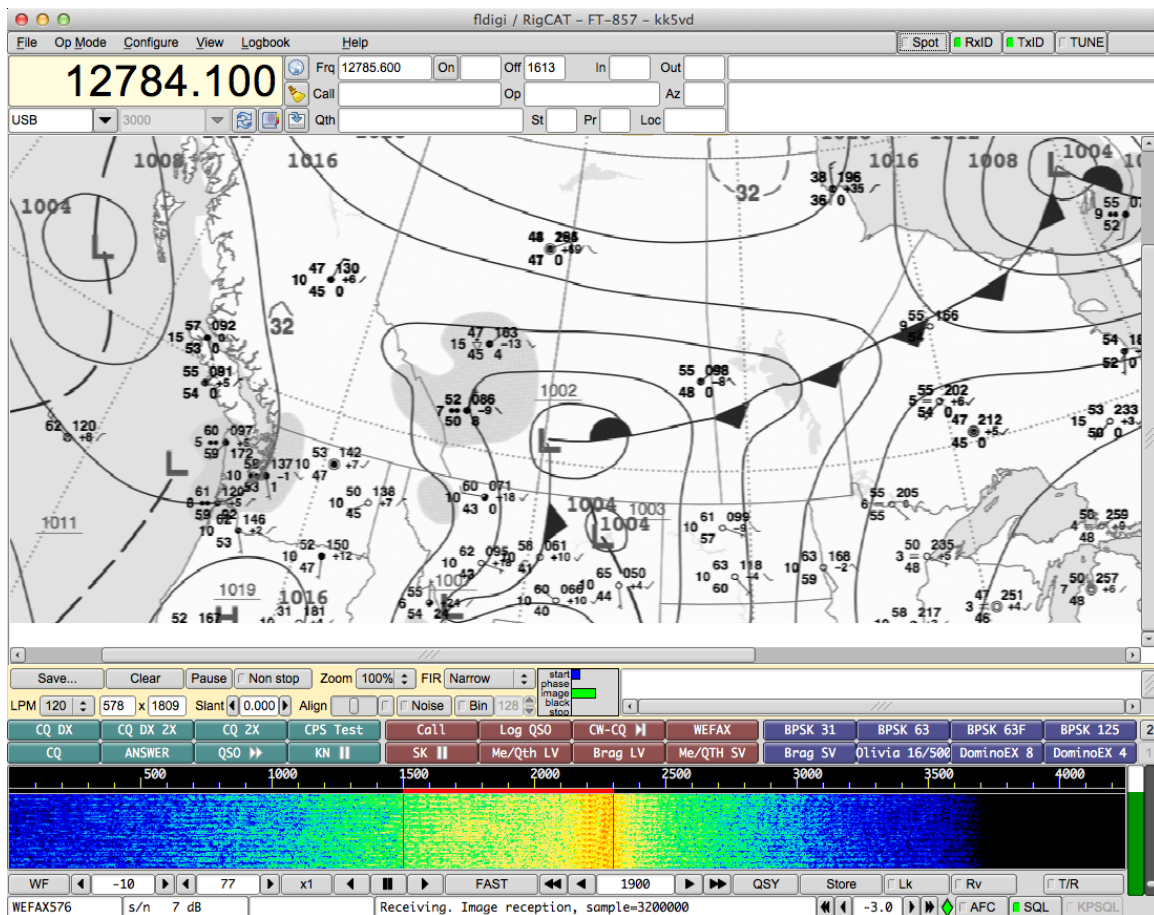


Figure 5.84: Reception

To enter reception mode, one can 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.
- Non-stop: 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:

- 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.

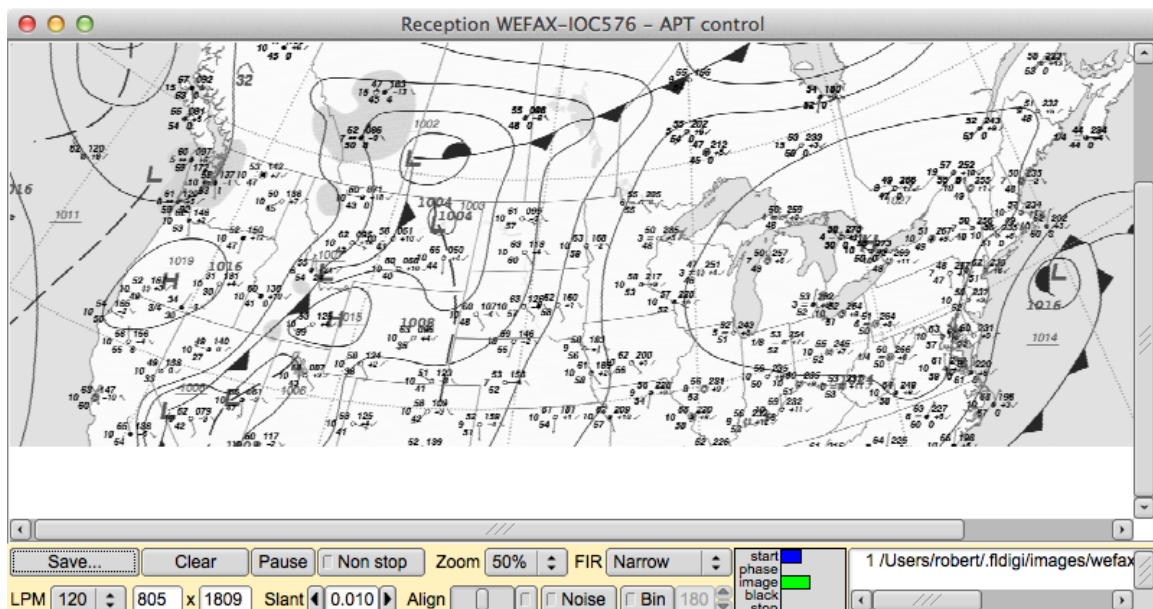


Figure 5.85: Received WEFAX File

### 5.15.3.1 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.

### 5.15.3.2 Manual reception mode (Non-Stop)

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.

### 5.15.3.3 Input FIR filters

There are three input Finite Impulse Response filters available. Here are their frequency characteristics:

Narrow filter, the default one, give the better results.

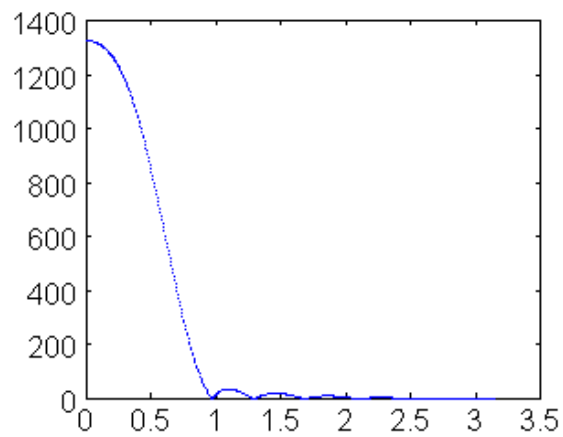


Figure 5.86: Narrow filter response

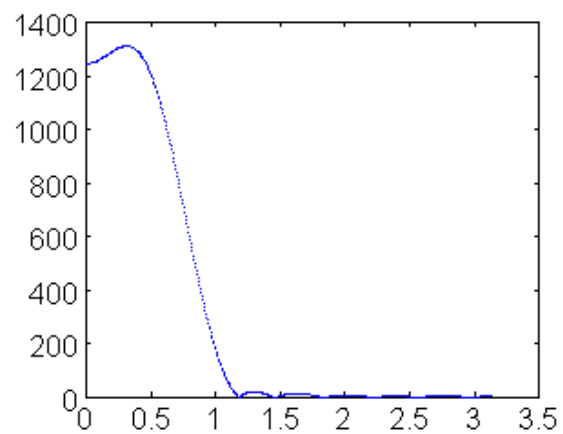


Figure 5.87: Middle filter response

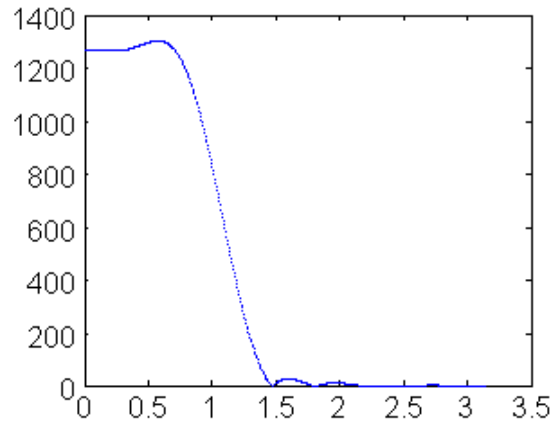


Figure 5.88: Wide filter response

#### 5.15.3.4 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.

#### 5.15.3.5 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.

#### 5.15.3.6 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 derivatives 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 contrasted details.

#### 5.15.3.7 Image detection based on signal power

The APT control - inherited from the Hamfax 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 traditionally method does not detect anything.

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

#### 5.15.3.8 AFC: Automatic Frequency Control

This option controls the frequency on the complete spectrum width. After several hunderads of lines have been correctly loaded (That is, with a high line-to-line correlation), the AFC locks until the frequency or mode is manually changed.

#### 5.15.3.9 Noise elimination

This option eliminates short-lived noise, individual pixels with a different value from their horizontal neighbors. They are modified using median values. This is based on the fact that no line should be narrower than two pixels, otherwise such an image would not be broadcast, because not correctly readable. It is therefore impossible to have one single pixel, simultaneously very different from its left and right neighbors.

#### 5.15.3.10 Binary images

Faxes can be stored as binary or grey level images. The cutoff level between black and white (Default 128) can be adjusted. No information is lost until the image is saved, therefore this level can be freely modified until image end. The purpose of this option is to save disk space.

#### 5.15.3.11 Displaying received files

Each time the end of an image is detected, an image file is created and its name appears on the file list of the reception window. By clicking on a file name, it is displayed in the transmit window.



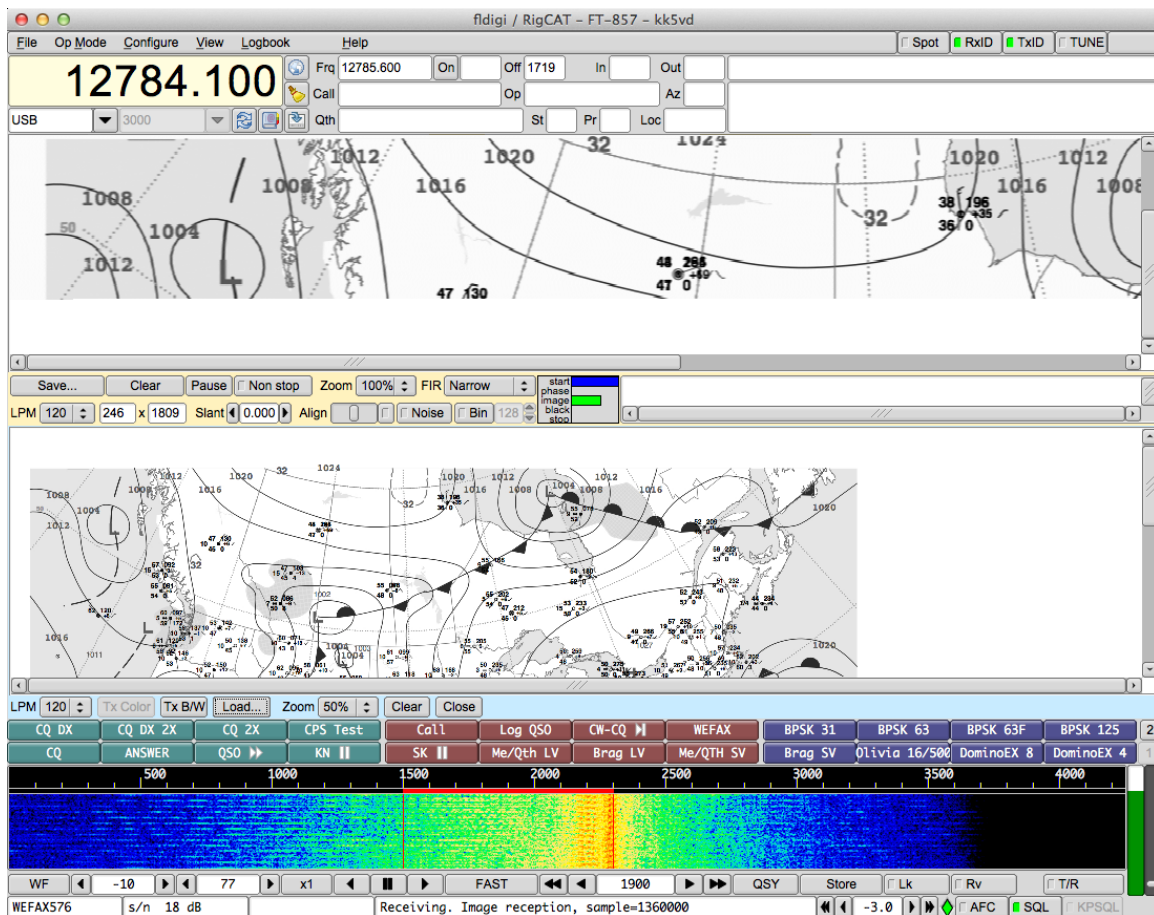


Figure 5.89: Received Image

[Return to Top of Page](#)  
[Return to Main Page](#)

## 5.16 WWV transmit mode

The WWV mode is normally used to measure the offset of the sound card oscillator. (see [WWV ppm measurement](#)).

It can also be used to transmit time tick signals similar to the format that is used by WWV. The WWV modem sends a time tick which is a 200 msec wide pulse at a once per second rate when the T/R button is enabled. This pulse is shaped with a 4 msec raised cosine shape on the leading and trailing edges to reduce key clicks. The accuracy of the transmitted time tick is solely dependent on the accuracy with which the WWV ppm measurement was performed.

The purpose of the WWV time tick transmission is to allow other stations to calibrate their sound cards against your calibrated system. This can be used to align all systems in a VHF/UHF net for example. It is only necessary for one of the net members to be able to calibrate his or her sound card against WWV. The other's would then be calibrated by proxy use of the WWV time tick transmit mode. This can even be used in the case where no member has access to a HF transceiver. The "master" station would set the Rx and Tx ppm settings to zero. It would then transmit the time tick signal for the other stations to calibrate their sound cards against the master sound card. Having all of

the stations calibrated in this way will insure that the modem decoders will give maximum performance. Here is an example of an [advanced macro](#) that will send a CW announcement, 2 minutes of time ticks and end with another CW announcement.

```
<MODEM: CW>
<!GOFREQ:1000>
<!WPM:24>
QRZ QRZ de <MYCALL> <MYCALL>
2 minute time tick cal run follows
<IDLE:2>
<!MODEM: WWV><!IDLE:120>
<!MODEM: CW><!IDLE:2>
end of time tick run
de <MYCALL> k
<TX><RX>
```

[Return to Top of Page](#)

[Return to Main Page](#)

## 5.17 Mode Table

### 5.17.1 PSK

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
PSK31	31.25	50	80.00%	31	1-PSK		31HG1B	1	
PSK63	62.5	100	80.00%	63	1-PSK		63HG1B	2	
PSK125	125	200	80.00%	125	1-PSK		125H↔ G1B	4	
PSK250	250	400	80.00%	250	1-PSK		250H↔ G1B	126	
PS↔ K250C6	250	2400	80.00%	2000	6-PSK		2000H↔ G1B	263	63
PSK500	500	800	80.00%	500	1-PSK		500H↔ G1B	173	
PS↔ K500C2	500	1600	80.00%	1200	2-PSK		1200H↔ G1B	263	27
PS↔ K500C4	500	3200	80.00%	2600	4-PSK		2600H↔ G1B	263	28
PS↔ K800C2	800	2300	80.00%	1400	2-PSK		2300H↔ GB1	263	57
PSK1000	1000	1600	80.00%	1800	1-PSK		1800H↔ G1B	263	50
PS↔ K1000C2	1000	3200	80.00%	3600	2-PSK		3600H↔ G1BC	263	52

### 5.17.2 PSKR / PSK-FEC

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
PSK63↔ FEC	62.5	55	80.00%	63	1-PSK		63HG1B	22	
PSK63↔ RC4	63	220	80.00%	330	4-PSKR		330H↔ G1BC	263	1

PSK63↔ RC5	63	275	80.00%	416	5-PSKR		416H↔ G1BC	263	2
PSK63↔ RC10	63	550	80.00%	850	10-PS↔ KR		850H↔ G1BC	263	3
PSK63↔ RC20	63	1100	80.00%	1725	20-PS↔ KR		1725H↔ G1BC	263	4
PSK63↔ RC32	63	1760	80.00%	2775	32-PS↔ KR		2775H↔ G1BC	263	5

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
PS↔ K125R	125	110	80.00%	125	1-PSKR		125H↔ G1B	183	
PS↔ K125RC4	125	352	80.00%	650	3-PSKR		650H↔ G1B	10	
PS↔ K125RC5	125	440	80.00%	825	4-PSKR		700H↔ G1BC	11	
PS↔ K125R↔ C10	125	1100	80.00%	1700	10-PS↔ KR		1700H↔ G1BC	12	
PS↔ K125R↔ C16	125	1760	80.00%	2750	16-PS↔ KR		2750H↔ G1BC	13	

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
PS↔ K250R	250	220	80.00%	250	1-PSKR		250H↔ G1B	186	
PS↔ K250RC2	250	440	80.00%	600	2-PSKR		600H↔ G1BC	263	20
PS↔ K250RC3	250	660	80.00%	950	3-PSKR		950H↔ G1BC	263	21
PS↔ K250RC5	250	1100	80.00%	1650	5-PSKR		1760H↔ G1BC	263	22
PS↔ K250RC6	250	1320	80.00%	2000	6-PSKR		2000H↔ G1BC	263	65
PS↔ K250RC7	250	1540	80.00%	2350	7-PSKR		2350H↔ G1BC	263	23

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
PS↔ K500R	500	440	80.00%	500	1-PSKR		500-H↔ G1BC	187	
PS↔ K500RC2	500	880	80.00%	1400	2-PSKR		1400H↔ G1BC	263	24
PS↔ K500RC3	500	1320	80.00%	1900	3-PSKR		1900H↔ G1BC	263	25
PS↔ K500RC4	500	1760	80.00%	2600	4-PSKR		2600H↔ G1BC	263	26

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
PS↔ K800RC2	800	1280	80.00%	1400	2-PSKR		800HG↔ B1C	263	54

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
PS↔ K1000R	1000	880	80.00%	1800	1-PSKR		1800H↔ G1B	263	51
PS↔ K1000↔ RC2	1000	1760	80.00%	3600	2-PSKR		3600H↔ G1BC	263	53

### 5.17.3 QPSK

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
QPSK31	31.25	50	80.00%	31	1-QPSK		31H↔ G1B	110	
QPSK63	62.5	100	80.00%	63	1-QPSK		63HG1B	3	
QPSK125	125	200	80.00%	125	1-QPSK		125H↔ G1B	5	
QPSK250	250	400	80.00%	250	1-QPSK		250H↔ G1B	127	
QPSK500	500	800	80.00%	500	1-QPSK		500H↔ G1B		

## 5.17.4 8PSK

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
8PSK125	125	620	80.00%	125	1-8PSK		250H↔ G1B	56	1066
8PSK250	250	620	80.00%	250	1-8PSK		250H↔ G1B	56	1071
8PSK500	500	1650	80.00%	500	1-8PSK		500H↔ G1B	56	1076
8PS↔ K1000	1000	3300	80.00%	1000	1-8PSK		1000H↔ G1B	56	1047
8PS↔ K125F	125	620	80.00%	125	1-8PSK		250H↔ G1B	56	1037
8PS↔ K250F	250	620	80.00%	250	1-8PSK		250H↔ G1B	56	1038
8PS↔ K500F	500	1650	80.00%	500	1-8PSK		500H↔ G1B	56	1043
8PS↔ K1000F	1000	3300	80.00%	1000	1-8PSK		1000H↔ G1B	56	1078
8PS↔ K1200F	1200	4200	80.00%	1200	1-8PSK		1200H↔ G1B	56	1058

## 5.17.5 Contestia

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
CONT↔ ESTIA-4- 250	62.5	40	100.00%	250	4-FSK	-10 dB	250H↔ F1B	55	
CONT↔ STIA-8- 250	31.25	30	100.00%	250	8-FSK	-13 dB	250H↔ F1B	49	
CONT↔ ESTIA-4- 500	125	78	100.00%	500	4-FSK	-8 dB	500H↔ F1B	54	
CONT↔ ESTIA-8- 500	62.5	60	100.00%	500	8-FSK	-10 dB	500H↔ F1B	52	
CONT↔ ESTIA- 16-500	31.25	30	100.00%	500	16-FSK	-12 dB	500H↔ F1B	50	
CONT↔ ESTIA-8- 1000	125	117	100.00%	1000	8-FSK	-5 dB	1K00↔ F1B	117	
CONT↔ ESTIA- 16-1000	62.5	78	100.00%	1000	16-FSK	-9 dB	1K00↔ F1B	53	

CONT↔ ESTIA- 32-1000	31.25	48	100.00%	1000	32-FSK	-12 dB	1K00↔ F1B	51	
----------------------------	-------	----	---------	------	--------	--------	--------------	----	--

### 5.17.6 DominoEX

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
Domino↔ EX4	3.9	29	100.00%	173			173H↔ F1B	84	
Domino↔ EX5	5.4	44	100.00%	244			244H↔ F1B	85	
Domino↔ EX8	7.8	58	100.00%	346			346H↔ F1B	86	
Domino↔ EX11	10.8	80	100.00%	262			262H↔ F1B	87	
Domino↔ EX16	15.6	115	100.00%	355			355H↔ F1B	88	
Domino↔ EX22	21.5	160	100.00%	524			524H↔ F1B	90	
Domino↔ EX44	43	312	100.00%	1600			1600H↔ F1B	263	45
Domino↔ EX88	86	614	100.00%	1600			1600H↔ F1B	263	46

### 5.17.7 MFSK

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
MFSK4	3.9	18	100.00%	154	32-FSK		154H↔ F1B		
MFSK8	7.8	36	100.00%	316	32-FSK		316H↔ F1B	60	
MFSK11	10.8	40	100.00%	218	16-FSK		218H↔ F1B	148	
MFSK16	15.6	58	100.00%	316	16-FSK		316H↔ F1B	57	
MFSK22	21.5	80	100.00%	435	16-FSK		435H↔ F1B	152	
MFSK31	31.3	55	100.00%	330	8-FSK		330H↔ F1B		
MFSK32	31.3	120	100.00%	630	16-FSK		630H↔ F1B	147	
MFSK64	63	240	100.00%	1260	16-FSK		1260H↔ F1B	263	30
MFSK128	125	480	100.00%	1920			1920H↔ F1B	263	31
MFS↔ K64L	63	240	100.00%	1260	16-FSK		1260H↔ F1B	263	30
MFS↔ K128L	125	480	100.00%	1920			1920H↔ F1B	263	31

### 5.17.8 MT-63

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
MT63-500	5	50	80.00%	500	64 x 2-PSK		500H↔ J2DEN	9	
MT63-1000	10	100	80.00%	1000	64 x 2-PSK		1K00↔ J2DEN	12	
MT63-2000	20	200	80.00%	2000	64 x 2-PSK		2K00↔ J2DEN	15	

## 5.17.9 Olivia

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
OLIVIA-4-250	63	20	100.00%	250	4-FSK	-12 dB	250H↔ F1B	75	
OLIVIA-8-250	31	15	100.00%	250	8-FSK	-14 dB	250H↔ F1B	69	
OLIVIA-4-500	125	40	100.00%	500	4-FSK	-10 dB	500H↔ F1B	74	
OLIVIA-8-500	63	30	100.00%	500	8-FSK	-11 dB	500H↔ F1B	72	
OLIVIA-16-500	31	20	100.00%	500	16-FSK	-13 dB	500H↔ F1B	70	
OLIVIA-8-1000	125	58	100.00%	1000	8-FSK	-7 dB	1K00↔ F1B	116	
OLIVIA-16-1000	63	40	100.00%	1000	16-FSK	-10 dB	1K00↔ F1B	73	
OLIVIA-32-1000	31	24	100.00%	1000	32-FSK	-12 dB	1K00↔ F1B	71	

## 5.17.10 RTTY

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
RTTY 45	45	60	100.00%	270			270H↔ F1B	39	
RTTY 50	50	66	100.00%	270			270H↔ F1B	40	
RTTY 75N	75	100	100.00%	370			370H↔ F1B	41	
RTTY 75W	75	100	100.00%	870			870H↔ F1B		

## 5.17.11 THOR

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modulation	S/N	ITU	RSID-1	RSID-2
THOR4	3.9	14	100.00%	173			173H↔ F1B	136	
THOR5	5.4	22	100.00%	244			244H↔ F1B	139	
THOR8	7.8	28	100.00%	346			346H↔ F1B	137	
THOR11	10.8	40	100.00%	262			262H↔ F1B	143	
THOR16	15.6	58	100.00%	355			355H↔ F1B	138	
THOR22	21.5	78	100.00%	524			524H↔ F1B	145	

THO↔ R25X4	24.3	88	100.00%	1800	4x tone spacing, 2 sec interleave		1800H↔ F1B	263	40
THO↔ R50X1	48.6	176	100.00%	900	1 sec interleave		900H↔ F1B	263	41
THO↔ R50X2	48.5	176	100.00%	1800	2x tone spacing, 1 sec interleave		1800H↔ F1B	263	42
THOR100	97	352	100.00%	1800	0.5 sec interleave		1800H↔ F1B	263	43

### 5.17.12 THROB

Mode	Baud	WPM	Duty Cycle	BW (Hz)	Modula- tion	S/N	ITU	RSID-1	RSID-2
THROB1	1	10	80.00%	72	1/2 of 9-FSK		72H0↔ F1B	43	
THROB2	2	20	80.00%	72	1/2 of 9-FSK		72H0↔ F1B	44	
THROB4	4	30	80.00%	144	1/2 of 9-FSK		144H↔ F1B	45	
THRO↔ BX1	1	10	80.00%	94	2 of 11-FSK		94H0↔ F1B	46	
THRO↔ BX2	2	20	80.00%	94	2 of 11-FSK		94H0↔ F1B	47	
THRO↔ BX4	4	40	80.00%	188	2 of 11-FSK		188H↔ F1B	146	

[Return to Top of Page](#)

[Return to Main Page](#)



## Chapter 6

# Operating

- [Contest - How To](#)
- [CW Keying](#)
- [Digiscope Display](#)
- [Digiscope Display - WWV mode](#)
- [DXCC List etal](#)
- [Configure FFT Scan](#)
- [Frequency Analyzer](#)
- [Keyboard Operation](#)
- [Extended Characters](#)
- [KML](#)
- [Menus](#)
- [Mouse and Keyboard Shortcuts](#)
- [Notifier](#)
- [Operating Controls and Displays](#)
- [Operating Multiple Copies](#)
- [PSK Reporter](#)
- [Rig Control](#)
- [RST and RSQ Reporting](#)
- [Signal Browser](#)
- [Supporting Data Files Acquisition](#)
- [SYNOP decoding](#)
- [UTF-8 support](#)
- [Using the FLdigi Rt. Channel for transmitter PTT](#)
- [Working Logs](#)
- [WWV transmit mode](#)

## 6.1 Contest - How To

Fldigi supports a basic contesting format. Select the menu item View/Contest fields to see how the QSO 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.

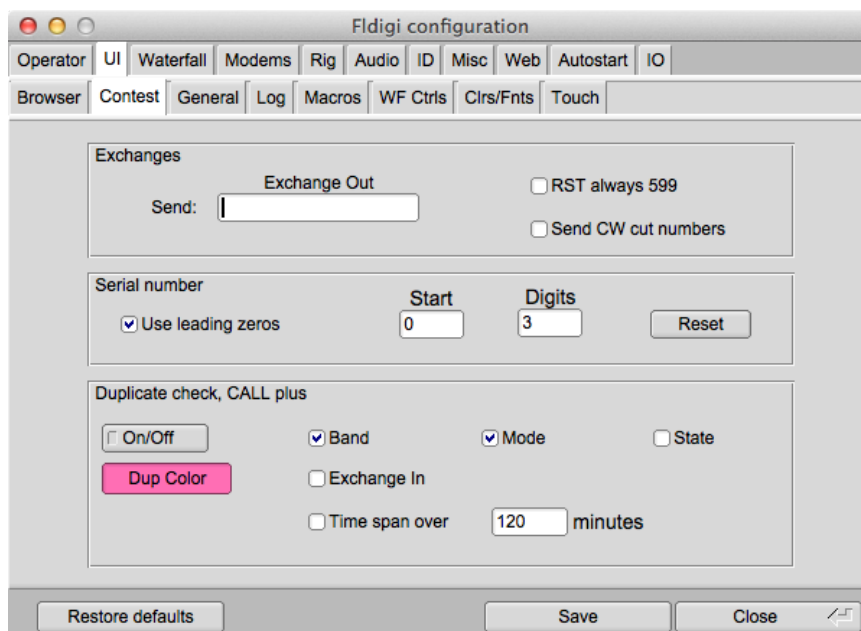
### 6.1.1 Contest Fields



The image shows a horizontal input area with three fields: "#Out" followed by a small text box, "#In" followed by a small text box, and "Xchg" followed by a larger text box.

Figure 6.1: Contest Fields

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.



The image shows the "Fldigi configuration" dialog box with the "Contest" tab selected. The "Exchanges" section has "Exchange Out" and "Send:" with a text box, and checkboxes for "RST always 599" and "Send CW cut numbers". The "Serial number" section has a checked "Use leading zeros" checkbox, "Start" (0) and "Digits" (3) text boxes, and a "Reset" button. The "Duplicate check, CALL plus" section has a checked "On/Off" checkbox, "Band" and "Mode" checked checkboxes, "State" unchecked checkbox, a pink "Dup Color" button, "Exchange In" unchecked checkbox, and "Time span over" (120) minutes.

Figure 6.2: UI Contest

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.

### 6.1.2 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

### 6.1.3 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 marker"

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

```
<<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.

#### 6.1.4 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 it 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.

#### 6.1.5 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 599-599-001-001-001 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
<ILDY>	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

See [Macros](#) for additional information on editing and using the fldigi macro system.

### 6.1.6 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
<RX>
```

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
<RX>
```

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.

### 6.1.7 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>
agn agn k
<RX>
```

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.

## &lt;XBEG&gt;

- 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\>\<S←AVEXCHG\>" 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 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

```
<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:

```
<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

```
<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.

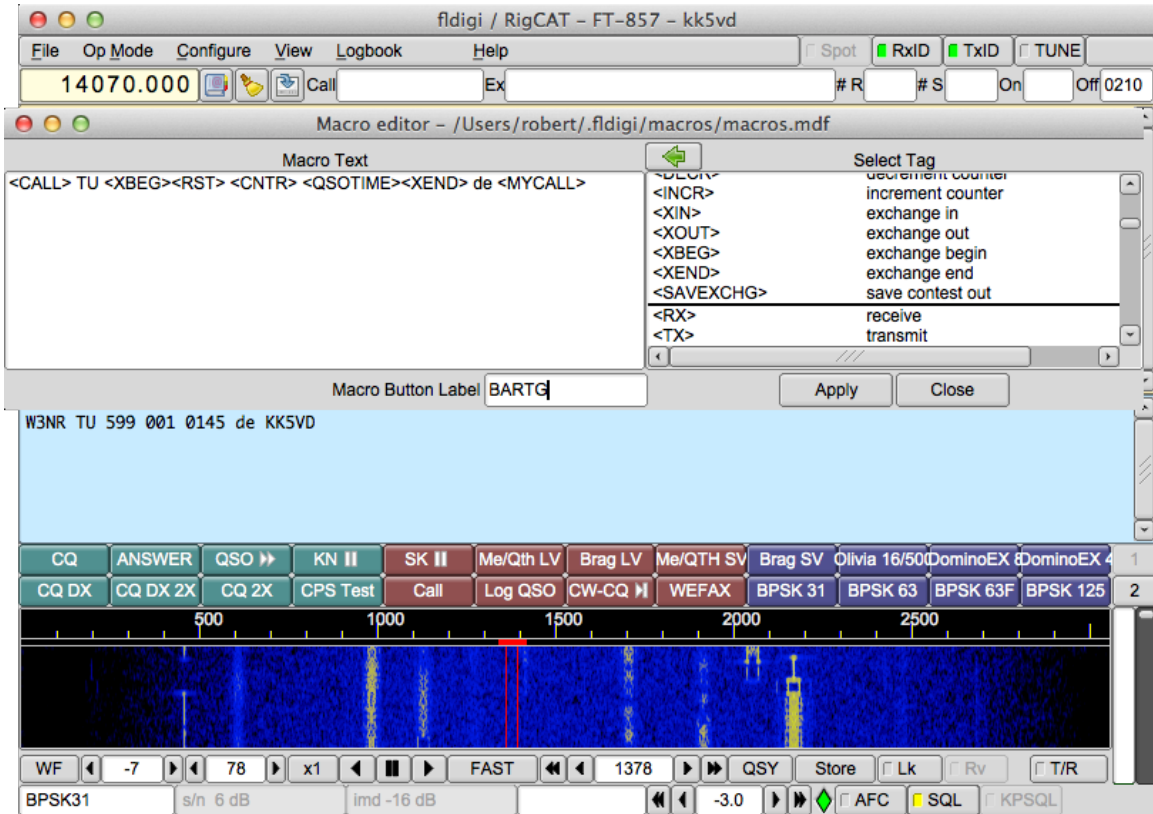


Figure 6.3: Exchange Begin-End

[Return to Top of Page](#)  
[Return to Main Page](#)



## 6.2 CW Keying

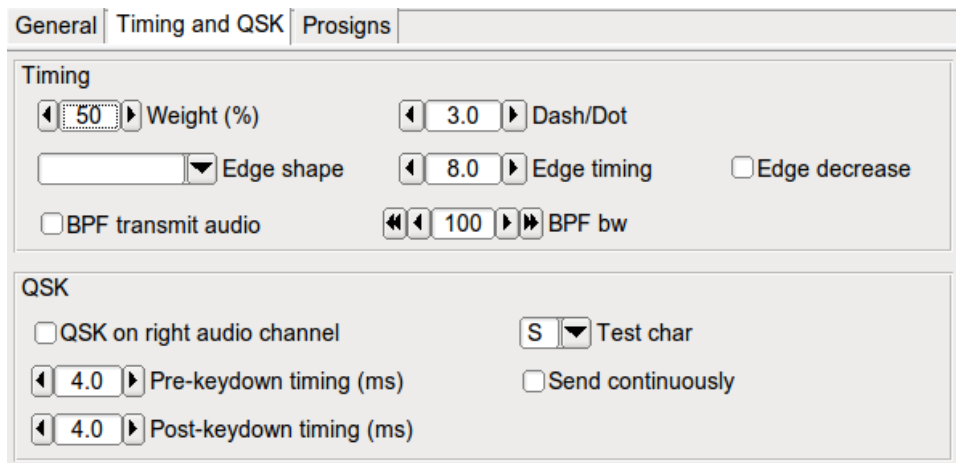


Figure 6.4: Timing and QSK

Click on the Configure menu item to open the *fldigi* configuration dialog. Click on the Modems tab and then on the QSK tab.

1. Set the Pre Timing and Post Timing to zero.
2. Activate the QSK feature by clicking "QSK on right channel".

Click the CW tab and adjust the CW settings to your preference. Use the Test char and the Send continuous controls to ease the adjustment process.

FLdigi is now ready to generate a 1600 hertz CW tone on the right channel of the stereo audio out of your sound card.

The left channel will be the normal raised cosine shaped CW wave form that you may use for your side tone.

The following circuit may be used to take the FLdigi QSK OUT signal from the right channel of your SOUND CARD to key your transmitter or a QSK circuit.

## FULL WAVE DIODE VOLTAGE DOUBLER

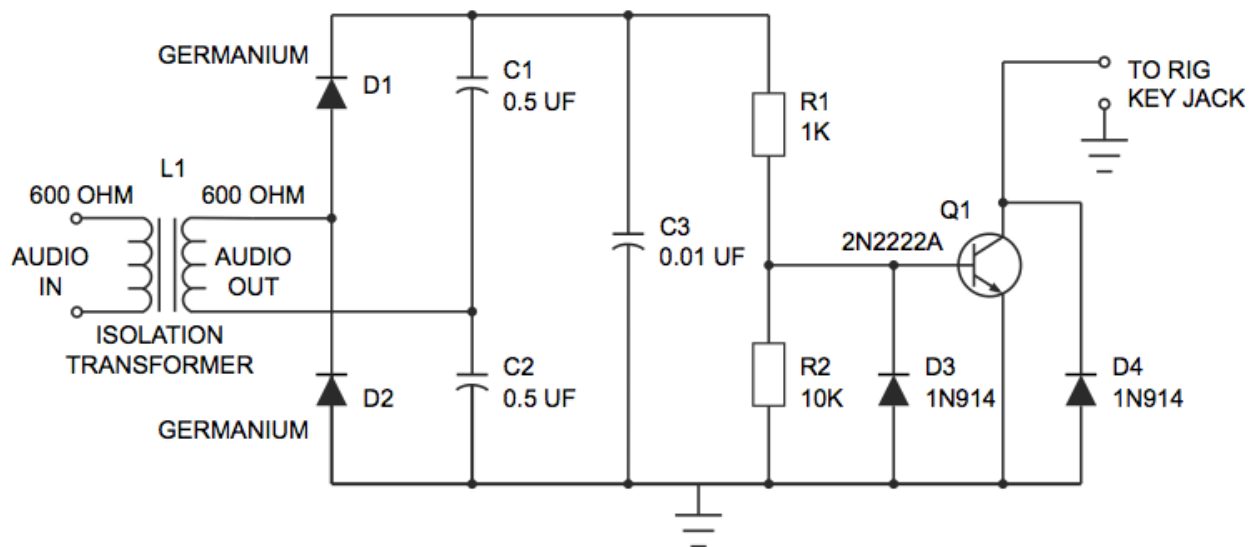


Figure 6.5: CW Keying Circuit

**NOTE:** L1 - Radio Shack has two items that may be used for this isolation transformer.

- Catalog # 270-054
- Catalog # 273-1374

Attach an audio cable from the Rt. Channel out of the your computer's SOUND CARD to the input of this QSK INTERFACE CIRCUIT (input of L1).

Attach another cable from the output of this circuit to your Rig's Keying Jack.mall>

Every CW 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, which takes the RIG'S CW KEYING JACK to ground and "keys" the transmitter.

You can adjust the start and stop timing of the QSK circuit relative to the CW waveform with the "pre" and "post" settings.

[Return to Top of Page](#)  
[Return to Main Page](#)

### 6.3 Digiscope Display

FLdigi provides several different views of the decoded signal with its waterfall, text and a scope displays. The scope display is either a separate moveable, resizable dialog that is opened from the "View/Digiscope" menu item or a docked scope.

## 6.3.1 CW

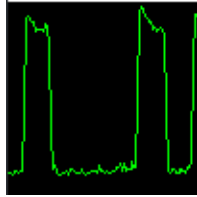


Figure 6.6: Digiscope CW

The CW signal will consist of the time domain amplitude detected signal. The horizontal timing is dependent on CW speed, so that the display will appear similar independent of CW speed.

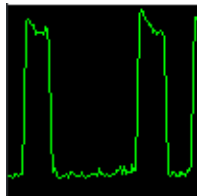


Figure 6.7: Digiscope CW

## 6.3.2 DominoEX / Thor

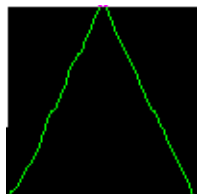


Figure 6.8: Digiscope DominoEX &amp; Thor

DominoEX and Thor have two alternate views available on the digiscope display. You can toggle between the views by left clicking on the digiscope display area. The triangular view shows data propagation through the interleave filter. As signal s/n degrades this display will become more wavy.



Figure 6.9: Digiscope DominoEX & Thor

The second view is the decoded data stream viewed in the frequency domain. The dots will be very distinct when the signal is fully acquired and decoding properly. It will be fuzzy when the decoder is not locked or there is interference present.

### 6.3.3 MFSK

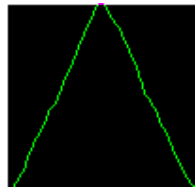
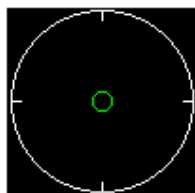


Figure 6.10: Digiscope MSFK

This is what you expect to see for all of the MFSK type modes. The number of steps in the slant lines will change with the various modes, but they will all have the same general appearance. If the signal is mistuned the sloped lines will become bowed and distorted.

### 6.3.4 PSK

The digiscope display just to the right of the waterfall displays signal quality in various formats. The display for PSK modes is the vector scope:



(a) The display with no signal or below squelch level. If the SQL is off this display will be random vectors driven by noise.



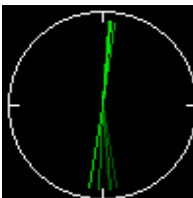
(b) The display with a normal psk31 signal. The vector flips between 0 and 6 o'clock



(c) AFC off and receive carrier set below the center of the received signal



(d) AFC off and receive carrier set above the center of the received signal.



(e) AFC enabled, Fading History Display Mode Selected (left click on scope)



(f) AFC enabled, Fading History / Amplitude Display Mode Selected (2nd left click on scope)

Figure 6.11: PSK Digiscopes

You can see the effect of mistuning by slewing the carrier control moving from low to high over the signal . You must do this with AFC off. Engage the AFC and the vectors will immediately snap to vertical positions.

You can alter the appearance of the phase vectors by left clicking on the digiscope display. One click will give you a history of phase vectors that fade with time. A second click will give you a history of phase vectors that both fade with time and are amplitude significant. The third click returns you to the original phase vector display.

The effect is the same with QPSK signals except you will see 4 vectors that are 90 degrees from each other.

### 6.3.5 RTTY

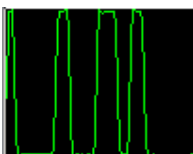


Figure 6.12: Digiscope RTTY

The signal can be viewed in two different ways on the digiscope. This is the time domain representation of the detected FSK signal. The two yellow lines represent the MARK and SPACE frequencies. This display is for Baudot, 45.45 baud, 182 Hz shift. If the transmitting station were transmitting at 200 Hz shift the signal extremes would lie above and below the yellow lines. Try tuning across the RTTY signal with the AFC disabled. You will see the signal move above and below the yellow lines as you tune. Then enable the AFC and the signal should rapidly move into

the center region of the display. This signal was about 3 - 6 dB above the noise floor. It looked marginal on the waterfall but still gave good copy.

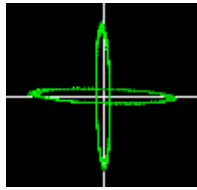


Figure 6.13: Digiscope RTTY

This is the other digiscope display for RTTY. You obtain this view by left clicking anywhere in the digiscope display window. You can toggle back and forth between these views. The MARK / SPACE frequencies are represented by the quadrature ellipses. When the RTTY signal is properly tuned in the lines will be in quadrature and aligned as shown. Tune across the RTTY signal and the MARK/SPACE lines will rotate around the center. If the sending station is using a shift that is smaller than you have the decoder setting then the two lines will close toward the NW/SE quadrants. If the sending station is using a shift that is greater than the decoder setting then the two lines will close toward the NE/SW quadrants.

[Return to Top of Page](#)  
[Return to Main Page](#)

## 6.4 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 is then 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.

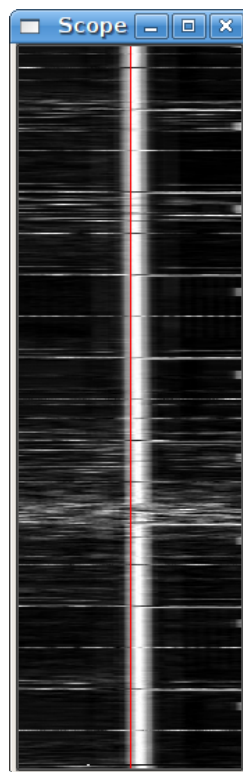


Figure 6.14: WWV corrected 20 minute trace 5x scale

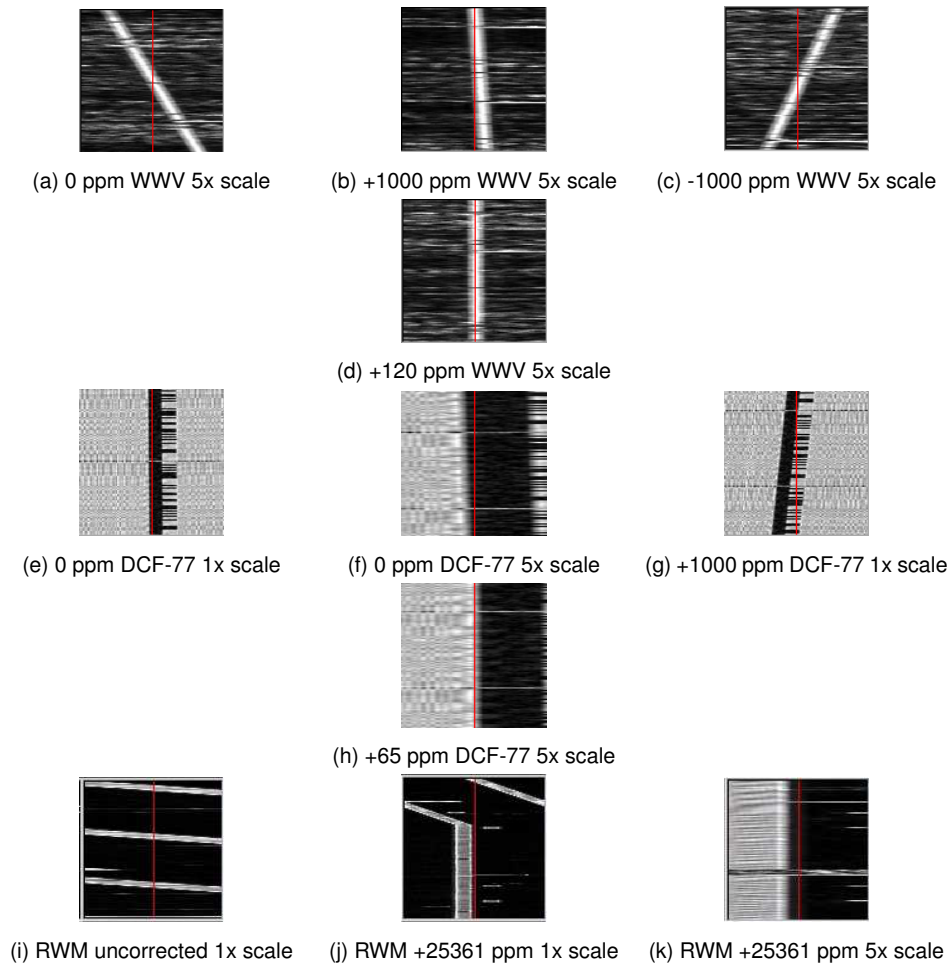


Figure 6.15: PSK Digiscopes

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 quater;

on 9996 kHz : 2nd wednesday of the 1st month of the quater;

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 – NON 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

See [Transmitting Simulated WWV Timing Tone](#)

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.5 DXCC List etal

Fidigi 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:

List Data	List Name	Web source
DXCC	cty.dat	<a href="http://www.country-files.com/cty/">http://www.country-files.com/cty/</a>
LOTW	lotw1.txt	<a href="http://www.hb9bza.net/lotw/lotw1.txt">http://www.hb9bza.net/lotw/lotw1.txt</a>
EQSL	AGMemberList.txt	<a href="http://www.eqsl.cc/QSLcard/DownloadedFiles/AGMemberList.txt">http://www.eqsl.cc/QSLcard/DownloadedFiles/AGMemberList.txt</a>

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".

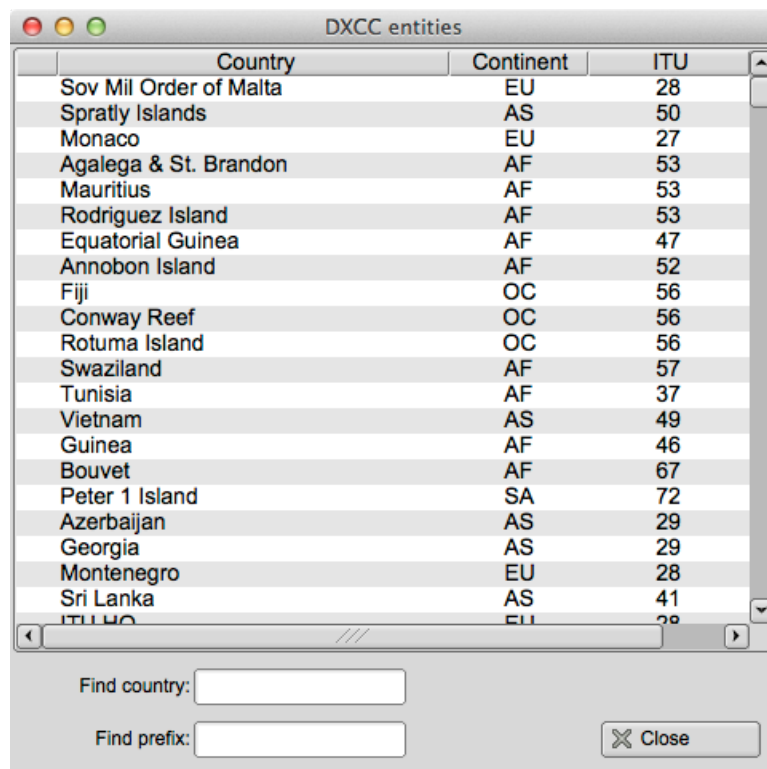


Figure 6.16: DXCC List

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

[Return to Top of Page](#)  
[Return to Main Page](#)

## 6.6 Configure FFT Scan

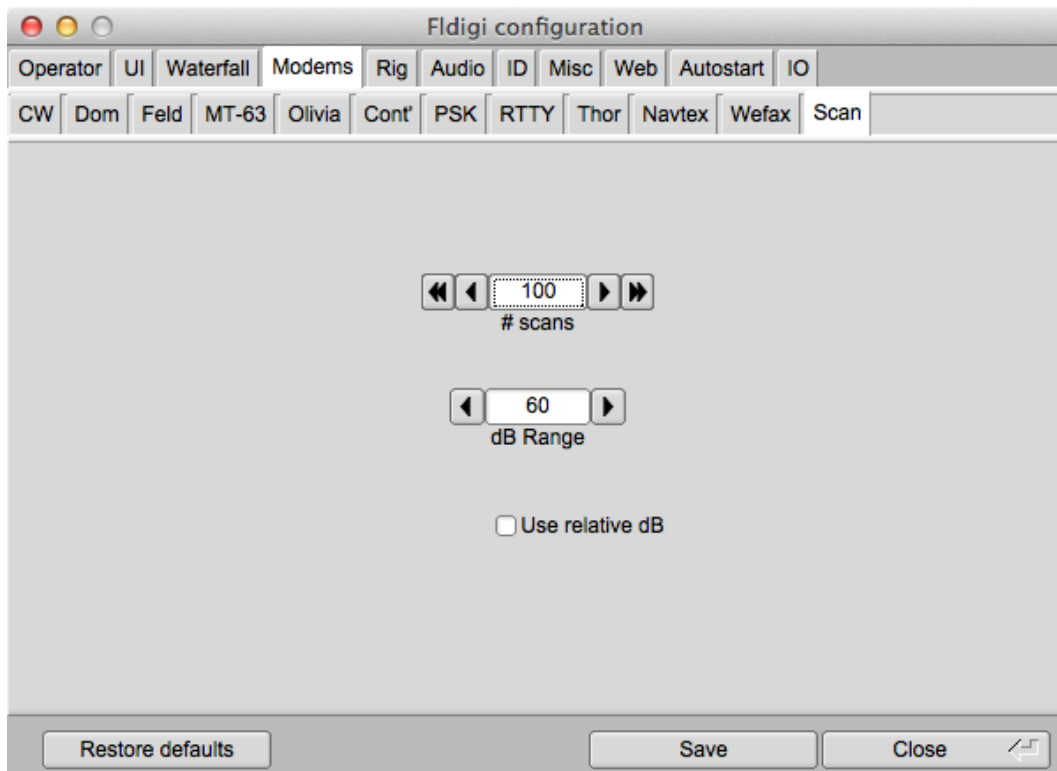


Figure 6.17: Scan Configuration Panel

# scans: The number of scans to perform (averaging).

db Range: The dynamic range in dB (vertical scale).

### 6.6.1 FFT Scan

Analysis tool used to evaluate Rx/Codec signal path. Can be used to measure noise floor of the codec; view the transceiver passband characteristics, Rx signal spectrum, etc.

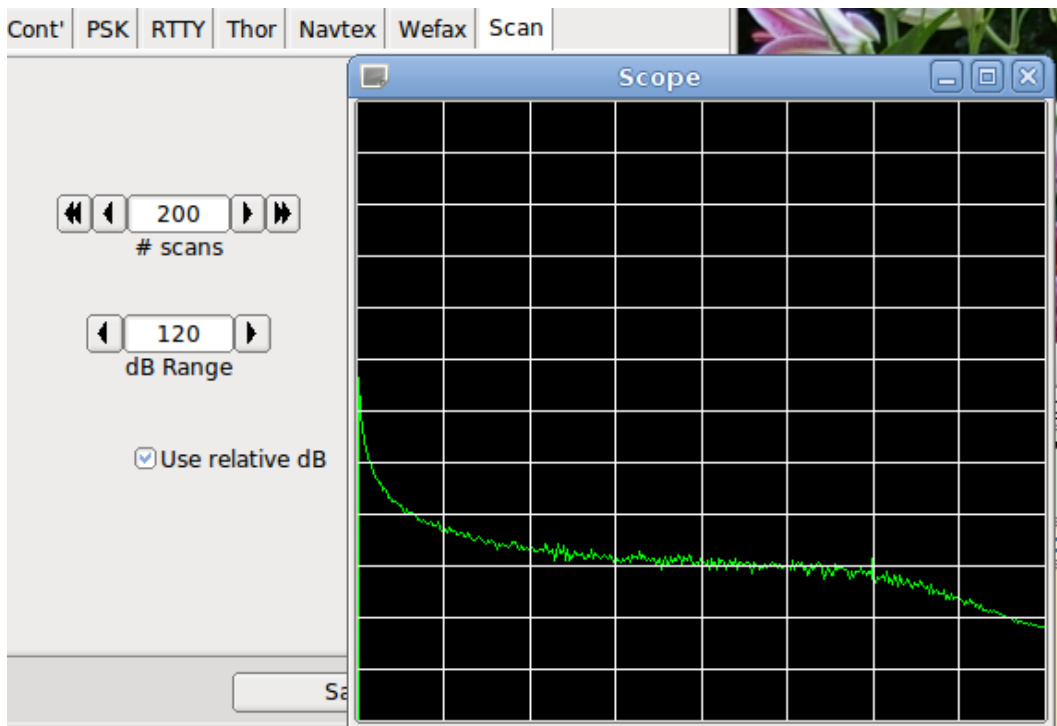


Figure 6.18: Scan results

The scope shows the results, both final and during the scan averaging. The # of scans specifies how many scans are averaged for the display. The dB range specifies the vertical scale as seen above. "Use relative dB" removes the vertical offset so that a wider range can be displayed with more detail.

Left clicking anywhere on the waterfall restarts the scan. You can observe the effect on the scope display. A comma separated file is also generated; `~/fldigi/fftscan.csv` which can be opened with any spreadsheet program.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.7 Frequency Analyzer

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

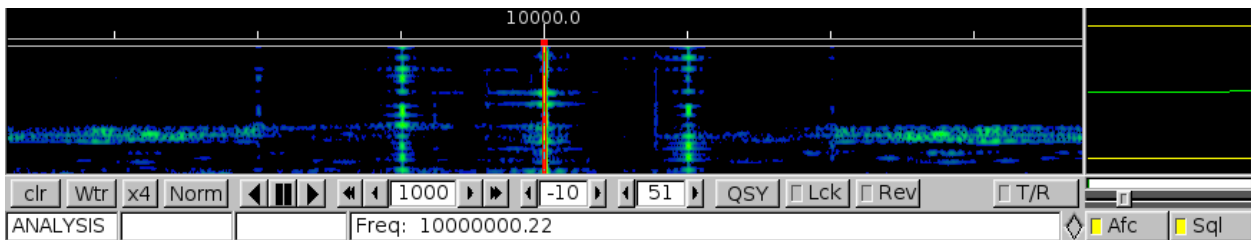


Figure 6.19: Frequency Analyzer

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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.8 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 oooops.

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, which fldigi uses as a meta character. The ^ 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.

Other useful meta combinations:

- ^I ^L - save the log entry when meta combination parsed
- ^p ^P - pause transmission at this point in transmit text
- ^r ^R - return to receive clear all transmit text

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 text insert cursor to the end of the transmitted text (which also pauses tx). Characters entered from the keyboard will be transmitted as entered until the tx text entry point is returned to the end of the tx buffer. A tab press at that position returns the cursor to the end of the text buffer. Transmitted text and text pending transmit are color coded red and black respectively (default colors):

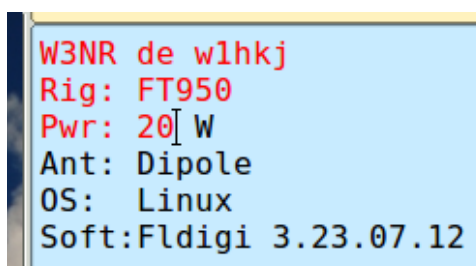


Figure 6.20: Tx Cursor Position

Be aware that fldigi will NEVER transmit past the text insert cursor! You may be caught by that behavior if you are modifying the buffer and transmitting at the same time. Or you may have accidentally mouse clicked in the middle of the Tx text with the result that fldigi suddenly stops transmitting. In either case the Tab key will return the text insert cursor to the end of the tx buffer.

CW operation is slightly different, see the help for [CW](#) .

- Extended UTF-8 characters can be entered into all text entry controls using Ctrl + three digits. These characters can also be entered using the right-click pop-up menu item "Spec Char ". The extended character transmitted by fldigi will actually be a 2 byte UTF-8 string.

see [Extended Characters](#)

### 6.8.1 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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.9 Extended Characters

Extended UTF-8 characters can be entered into all text entry controls using Ctrl + three digits. These characters can also be entered using the right-click pop-up menu item "Spec Char ". The extended character transmitted by fldigi will actually be a 2 byte UTF-8 string.

**Extended character set**

-	-	-	-	-	-	-	-
161	162	163	164	165	166	167	168
iexcl	cent	pound	curren	yen	brvbar	sect	uml
¡	¢	£	¤	¥	¦	§	¨
-	-	-	-	-	-	-	-
169	170	171	172	173	174	176	177
copy	ordf	laquo	not	shy	reg	macr	deg
©	ª	«	¬		®	-	°
-	-	-	-	-	-	-	-
177	178	179	180	181	182	183	184
pluSmn	sup2	sup3	acute	micro	para	midot	cedil
±	²	³	´	µ	¶	·	¸
-	-	-	-	-	-	-	-
185	186	187	188	189	190	191	192
sup1	ordm	raquo	frac14	frac12	frac34	iquest	Agrave
¹	º	»	¼	½	¾	¿	À
-	-	-	-	-	-	-	-
193	194	195	196	197	198	199	200

Acute	Acirc	Atilde	Auml	Aring	AElig	Ccedil	Egrave
Á	Â	Ã	Ä	Å	Æ	Ç	È
-	-	-	-	-	-	-	-
201	202	203	204	205	206	207	208
Eacute	Ecirc	Euml	Igrave	Iacute	Icirc	Iuml	ETH
É	Ê	Ë	Ì	Í	Î	Ï	Ð
-	-	-	-	-	-	-	-
209	210	211	212	213	214	215	216
Ntilde	Ograve	Oacute	Ocirc	Otilde	Ouml	times	Oslash
Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø
-	-	-	-	-	-	-	-
217	218	219	220	221	222	223	224
Ugrave	Uacute	Ucirc	Uuml	Yacute	THORN	szlig	agrave
Û	Ú	Ü	Û	Ý	Þ	ß	à
-	-	-	-	-	-	-	-
225	226	227	228	229	230	231	232
aacute	acirc	atilde	auml	aring	aelig	ccedil	egrave
á	â	ã	ä	å	æ	ç	è
-	-	-	-	-	-	-	-
233	234	235	236	237	238	239	240
eacute	ecirc	euml	igrave	iacute	icirc	iuml	eth
é	ê	ë	ì	í	î	ï	ð
-	-	-	-	-	-	-	-
241	242	243	244	245	246	247	248
ntilde	ograve	oacute	ocirc	otilde	ouml	divide	oslash
ñ	ò	ó	ô	õ	ö	÷	ø
-	-	-	-	-	-	-	-
249	250	251	252	253	254	255	-
ugrave	uacute	ucirc	uuml	yacute	thorn	yuml	-
ù	ú	û	ü	ý	þ	ÿ	-

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.10 KML



Figure 6.21: KML Logo

Section data\_source Data sources

### Keyhole Markup Language

(KML) is an XML file format for geographic visualization in two-dimensional maps such as Google Maps and three-dimensional earth browsers such as Google Earth or [Marble](#).



Fldigi can generate data with geographical locations, which can be used to generate KML data. This list might expand in the future

- The emitting station of a **Navtex** message.
- The origin of a **SYNOF** weather report.
- The Maidenhead locator of the user, as entered in fldigi user's profile.

### 6.10.1 KML generation from Navtex messages

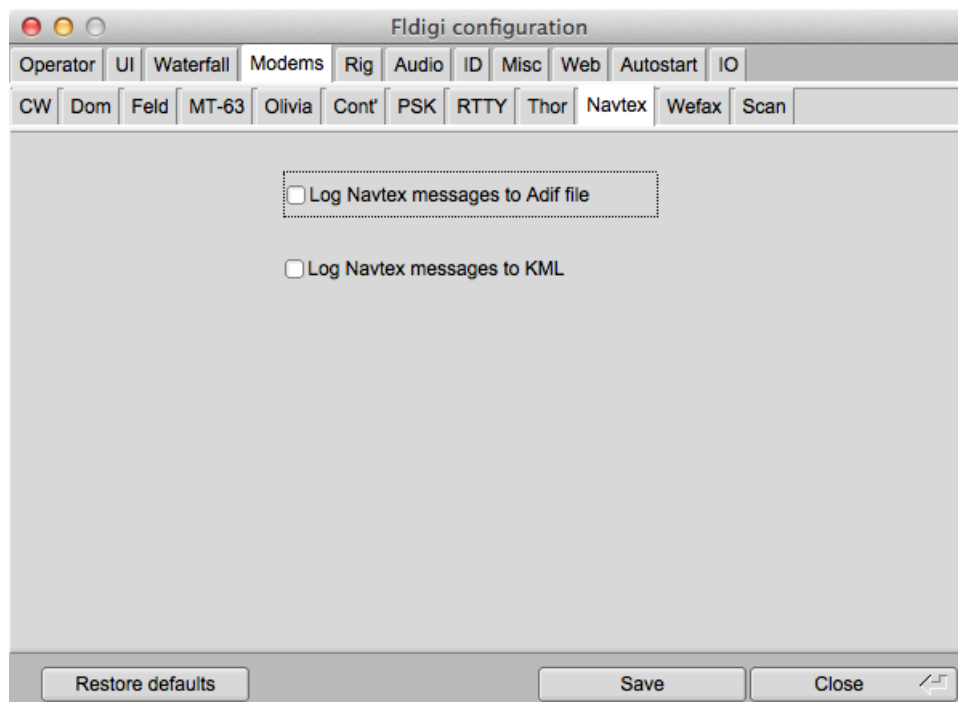


Figure 6.22: Navtex configuration tab with KML option

Each Navtex message comes with the code of the sending station, also called *origin*.

These messages are displayed, in KML files, at the coordinates of the sender. That is: KML placemarks are created or updated with these coordinates. Fldigi parses the Navtex reports, uses the station identifier to make a lookup in the Navtex stations file which contains geographical coordinates. These coordinates are used to create KML placemarks.

More explanation about how station coordinates are used, are given at the [Navtex page](#).

Navtex messages are quite often sent with embedded coordinates of the event they describe (Ship wreck, oil exploration etc...). For example: "LIGHT BUOY MARKING DANGEROUS WRECK 58-01.2 NORTH 005-27.1 WEST" or "THREE MEN OVERBOARD IN PSN 39-07, 7N 026-39, 2E". A future version will parse the content of the message, extracting raw coordinates, and will display a graphic entity at the location of the described event.

### 6.10.2 KML generation from SYNOP reports

**SYNOP** is a code used for reporting weather information and as such, is used to broadcast meteorological data by radio. One of the most important emitter is **Deutsche Wetterdienst** which transmits them in **RTTY**, and fldigi is able to decode them and generate KML placemarks at the location of the weather information.

### 6.10.3 KML files structure

The KML data are made of different files

<i>fldigi.kml</i>	Entry point. Only this one has to be loaded. It never changes.
<i>styles.kml</i>	KML style sheet. Freely changeable by the user, for example to customize the icons.
<i>User.kml</i>	Location of the user based on his/her <b>Maidenhead locator</b> .
<i>Synop.kml</i>	Synop weather reports displayed at the location of the WMO station, or ship, or buoy.
<i>Navtex.kml</i>	Navtex reports, displayed at the place of the emitting station. A future version will plot the position of the coordinates indicated in the Navtex messages themselves.

### 6.10.4 Extended data

When creating a new placemark, written in of the KML data files (*Synop.kml*, *Navtex.kml* etc... ) data are sent to the KML module in the form of key-value pairs and are written into two forms:

- HTML content, in the `<description>` tag, surrounded by `CDATA` directives. The HTML format is chosen exclusively for display purpose and might change at any new version.
- Regular `<ExtendedData>` XML tags: These data are internally used by Fldigi to reload the previous session. The format is stable and can be used by external applications. All useful data are saved.

## 6.10.5 Parameters

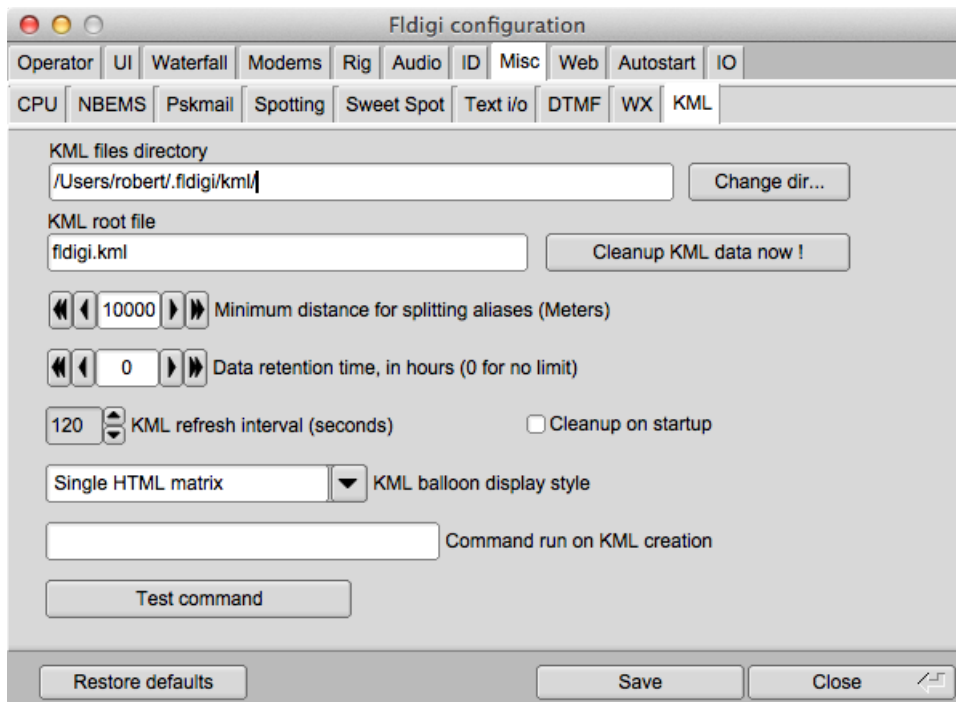


Figure 6.23: KML configuration tab

Fldigi maintains in a internal container, a set of placemarks which are data associated to geographical coordinates, an unique name, a set of key-value pairs and a timestamp. At regular intervals, a thread is woken up to save these geographical data to a KML file, in a specific directory. At this moment, a process can be started, running an external command. Depending on the type of data, a given file name will be used.

All KML files are accessible from an unique KML filename. Placemarks are identified with an unique name, for example a vessel name, or their WMO identifier. Placemark with a moving position such as ships, can have their path visualized because they still can be identified in two different reports. These reports can be kept as separate, or they can be merged into a single placemark: This depends on the distance between two placemarks with the same name, compared to the merging distance parameter.

Data can be kept for a given retention time, after which delay they are purged. At startup, former KML data can be reloaded, or cleaned up. Data as key-value pairs associated to a given placemarks can be displayed several ways.

All these parameters are controlled by the KML configuration tab.

## 6.10.6 Destination directory

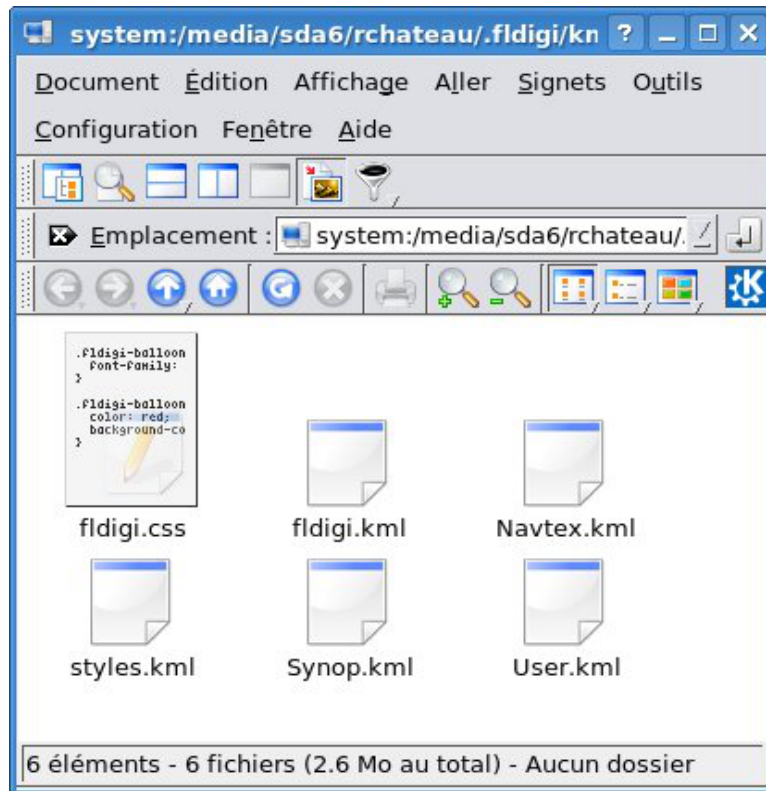


Figure 6.24: Directory of generated KML files

The default destination directory where KML files are saved is a subdirectory called /kml in the fldigi users directory. For example on Linux: `$HOME/.fldigi/kml/` and `<defaultpath>/fldigi.files/kml` on Windows™. This destination can be freely changed.

The file `fldigi.css` is created at installation, and is not changed later. Therefore it is possible to customize it by adding specific icons.

The file `fldigi.kml` is created by fldigi when it is not there, or when the refresh interval is changed.

If this destination directory is accessible from the internet, then it can be published to Google Maps.

**Note:**

Files updates are atomic. This means that a file is not accessible by a reader until it is completely written and closed. This is achieved by writing into temporary files, which are atomically renamed (POSIX function `rename()`) at the end of operation.

Therefore, the KML destination directory can safely be accessed by one writer and multiple readers. Several sessions of fldigi might also updates different KML files, as long as the main `fldigi.kml` file is not changed.

### 6.10.7 KML root file

This is the default name of the entry file of the generated KML document, which by default is fldigi.kml. If it does not exist, it is generated with the list of possible source of KML data (Synop, Navtex etc...). If [Google Earth](#) or [Marble](#) are installed on your machine, then they are associated to the file extension .kml and you just need to click on fldigi.kml to visualize it. It is automatically refreshed when fldigi adds new Synop weather reports or Navtex messages to it.

### 6.10.8 KML refresh interval

This delay, in seconds, is used at two places:

- This is the frequency at which new KML files are created, if new data is available
- This is the refresh interval specified in the KML file with tag `<refreshInterval>`.

This should not be too small, especially if the data files are big, otherwise fldigi will spend most of its time refreshing KML data, and accordingly Google Earth or Marble, reloading them.

### 6.10.9 Cleanup on startup

By default, at startup, fldigi reloads the existing KML files, extracting the key-value pairs contained in the "Extended↔Data" tags. However, it is possible to force fldigi to restart from scratch.

### 6.10.10 Merging distance

Different reports with the same placemark name can be merged into a single report if their distance is below a given threshold which is the merging distance. Otherwise, separate placemarks are created and joined by a red line, visible in the KML document.

### 6.10.11 KML balloon display type

Reports are inserted in the KML document one after the other. These description data are visible as *KML balloons*, or when getting placemark properties. If they have the same name and are within the merging distance, they will form a single placemark. The descriptions of each report will be displayed and merged by three possible ways.

#### 6.10.11.1 Plain text

Description are inserted without any HTML formatting. Only special HTML entities such as ampersands are re-formatted. This is especially useful if the KML document is later converted to GPX, because many GPS devices are not able to display HTML data.

## 6.10.11.2 HTML tables

Each description of placemark is transformed into a HTML table labelled with the time stamp of the insertion. Here is an example of two Navtex messages from the same station at different times:

2013-02-14 23:18	
Callsign	OST
Country	Belgium
Locator	JO11JE
Message number	35
Frequency	0
Mode	TOR
Message	191533 UTC NOV ; WZ 1196 SELF CANCELING. CANCEL WZ 1192 (GA92) (MA33). WALKER LIGHTBUOY NORMAL CONDITIONS RESTORED."
2013-02-14 23:13	
Callsign	OST
Country	Belgium
Locator	JO11JE
Message number	35
Frequency	0
Mode	TOR
Message	... etc ...

## 6.10.11.3 Distinct HTML matrix

For the same KML placemark, the key will typically the same for all reports. More, some data are numeric. This is therefore convenient to group them in matrices:

Here is an example for SYNOP weather data, made of three reports:

	2012-12-16 00:00	2012-12-17 06:00	2012-12-18 00:00
Dewpoint temperature		Undefined	Undefined
Figure		11	
Humidity	Unspecified		
Precipitations		Omitted, no observation	Omitted, no observation
Pressure change	Not specified	Not specified	Not specified
Sea level pressure	994 hPa	1000 hPa	1013 hPa
Ship average speed	0 knots	0 knots	
Ship direction	Calm	Calm	
Station type		Automated station. No observation (No 7WW)	Automated station. No observation (No 7WW)
Temperature	9.5 deg C	9.3 deg C	10.3 deg C

Value		37	
Visibility		4 km	4 km
Wave height	3.6 meters	4.7 meters	
Waves height	3.5 meters	4.5 meters	
Waves period	8 seconds	8 seconds	
Wind direction		265 degrees	275 degrees
Wind speed		33 knots (Estimated)	15 knots (Estimated)

### 6.10.12 Data Retention Time

Data may be automatically purged based on their time-stamp and a maximum retention time in hours. If the retention time is zero, then data are kept for ever.

### 6.10.13 Command run on KML creation

This command is executed at regular times, by default 180 seconds, and only if new data was written to any KML files. The first time this command is run, its process id is stored. Next time this command must be run, we check if this process is still running. If yes, no new process is created.

The intention is to handle the same way, programs which should always be running, for example KML visualizers, and on the other hand, one-shot scripts or converters. Typical situations are:

- Starting a program such as [Google Earth](#) or [Marble](#), only once per session.
- They will be automatically restarted if they crash, because their `process identifier` is not present anymore.
- Run as needed conversion programs such as GpsBabel, to another format ([GPX](#)). Or a [FTP](#) transfer to a remote platform, for inclusion of KML files in Google Maps.
- Accordingly, do not restart this conversion process as long it is not finished (FTP transfers might take long)

#### 6.10.13.1 Example of commands

##### 6.10.13.1.1 FTP Transfer

A new transfer - and a new process - must be initiated at each KML file save. A script is created for this purpose, and the command can be:

```
fldigi/scripts/ftp_kml_files.sh ftperso.free.fr MyFtpUserName MyPassword
kml
```

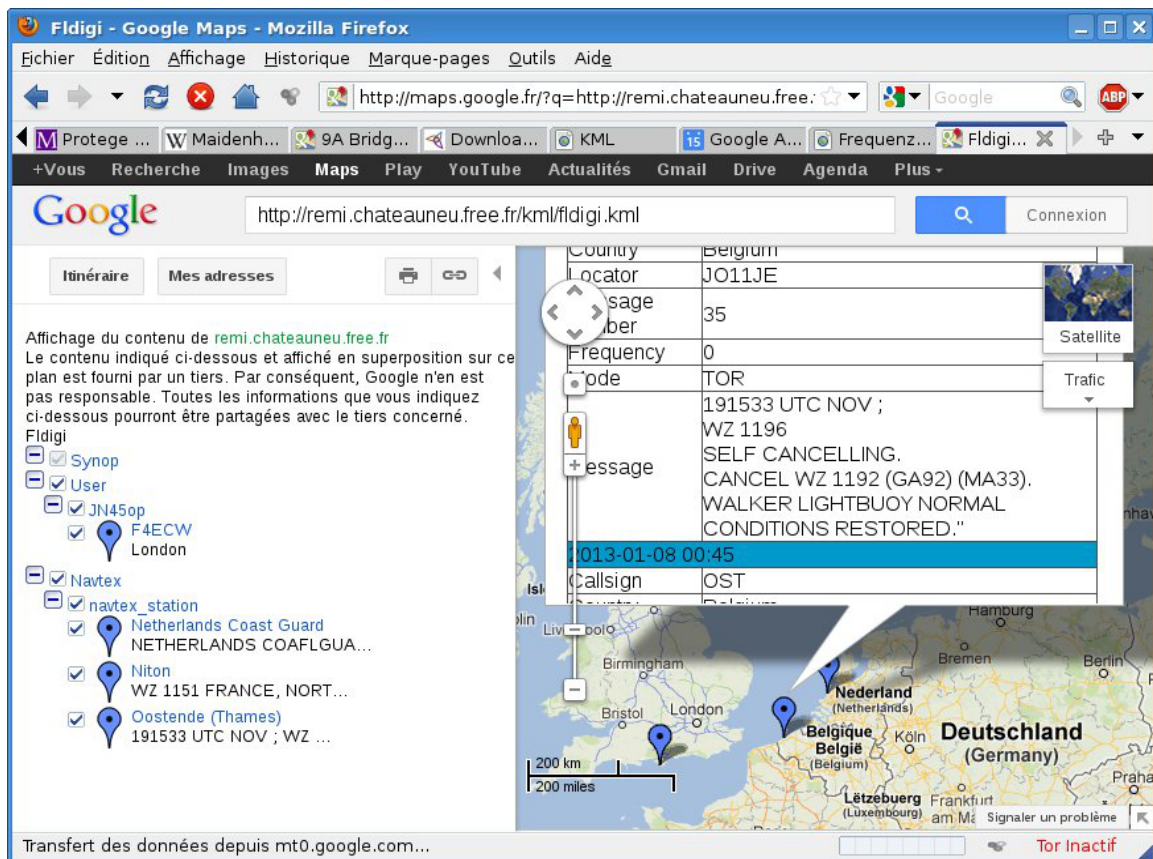


Figure 6.25: KML files displayed in Google Maps

An obvious use is to save these file to a remote machine where they can be accessed with a public URL. This URL can then be given as CGI parameter to [Google Maps](#) which will display the placemarks on a map. There are [limitations](#) on the maximum size of KML files which have to be smaller than 10 megabytes.

Note that KML files are for the moment not compressed into KMZ files.

An FTP copy is not necessary if the destination directory for KML files storage is public (That is, accessible from the Internet).

#### 6.10.13.1.2 Launch google-earth

The program will only be launched once, because its process id is still present. The command can be:

```
googleearth $HOME/.fldigi/kml/fldigi.kml
```

It is possible to change the icons by customizing the file `styles.kml`.



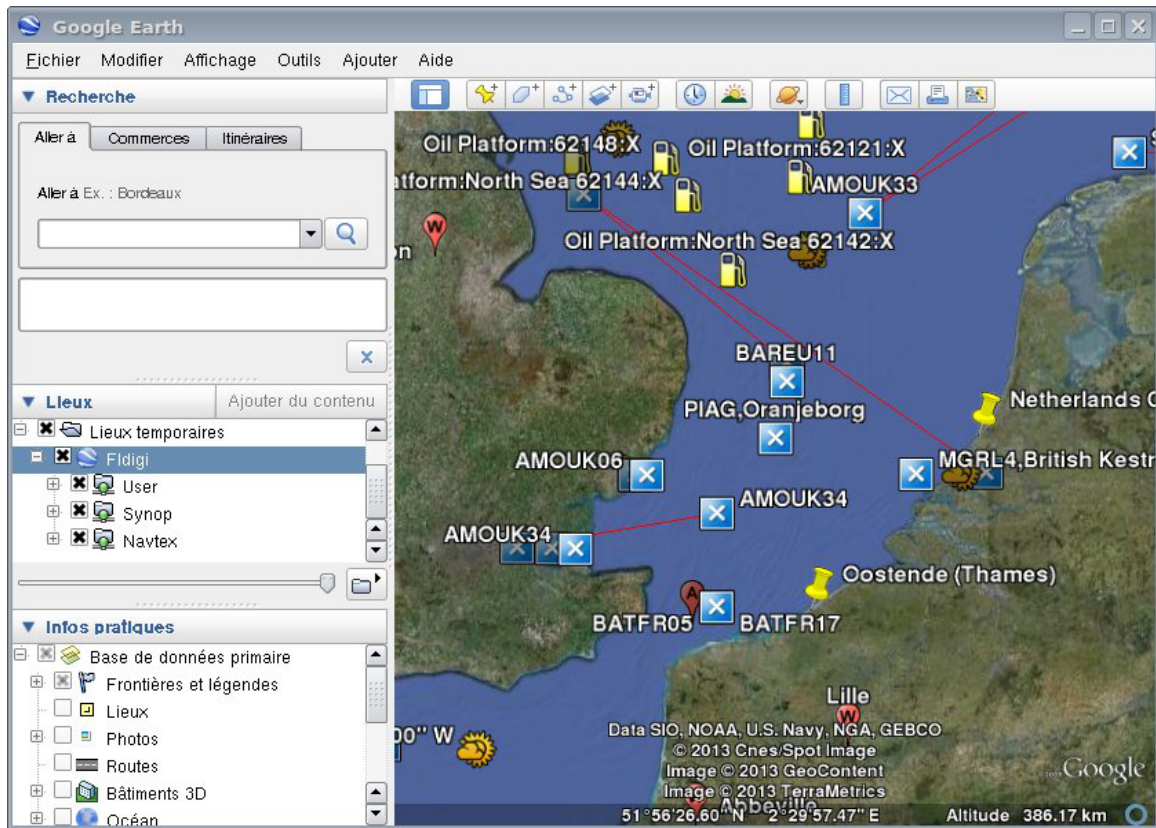


Figure 6.26: Google Earth

### 6.10.13.1.3 GPS Babel conversion

The command `GpsBabel`, for example, will selectively convert the KML file of Synop reports. It is generally advised to generate plain text description tags in the KML files, because GPS devices might not be able to correctly display HTML data. The command can be:

```
gpsbabel -i kml -f $HOME/.fldigi/kml/Synop.kml -o gpx -F out.gpx
```

The generated files can for example be fed into `Xastir`.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.11 Menus

### 6.11.1 Menu Heirarchy



Figure 6.27: Menu

## 6.11.1.1 Files

## 6.11.1.1.1 Folders

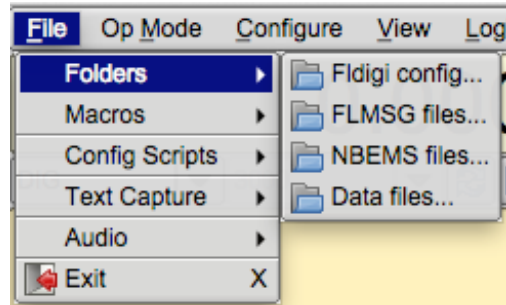


Figure 6.28: Folders

- Fldigi config... - open the OS native file explorer to the folder containing the fldigi operating & data files.
- NBEMS files... - open the OS native file explorer to the folder containing the NBEMS data files
- FLMSG files... - open the OS native file explorer to the folder containing the NBEMS/FLAMP data files
- Data files. See [Supporting Data Files Acquisition](#)

## 6.11.1.1.2 Macros

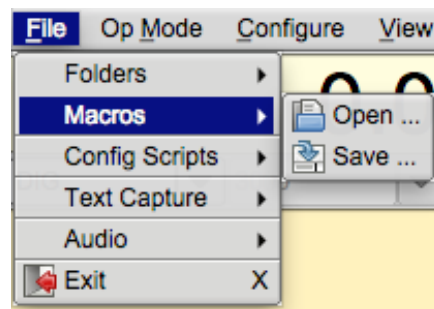


Figure 6.29: Macros

- Open Macros - open a macro definition file ... changes the MACRO keys immediately
- Save Macros - save the current macro definitions to a designated file

Additional information:

[Macros](#)

[User Interface Configuration - Macros](#)

## 6.11.1.1.3 Text Capture

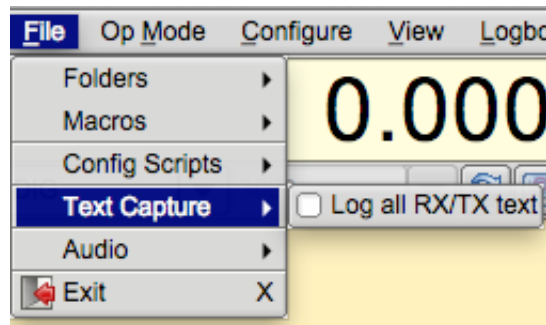


Figure 6.30: Text Capture

Capture all received and transmitted text to a date-time stamped file. Each day's data will be stored in a single file in which the text is appended: fldigiYYYYMMDD.log. This file will be in the "Fldigi config..." folder which can be accessed per the File/Folders menu item. A brief example log of Rx data:

```

--- Logging started at Wed Jan  5 18:42:51 2011 UTC ---
RX 14071955 : PSK31 (2011-01-05 18:42Z): d dx sk S
RX 14071756 : PSK31 (2011-01-05 18:42Z): PSE -lr dACQ CQ de WX1GRS WX1GRS
RX 14071756 : PSK31 (2011-01-05 18:42Z): CQ CQ de WX1GRS WX1GRS
RX 14071756 : PSK31 (2011-01-05 18:42Z): CQ CQ de WX1GRS WX1GRS
RX 14071756 : PSK31 (2011-01-05 18:42Z): PSE K aie =
--- Logging stopped at Wed Jan  5 18:43:04 2011 UTC ---
  
```

Each line contains the program state, RX or TX, the frequency, the mode, the date-time, and the data stream.

## 6.11.1.1.4 Audio

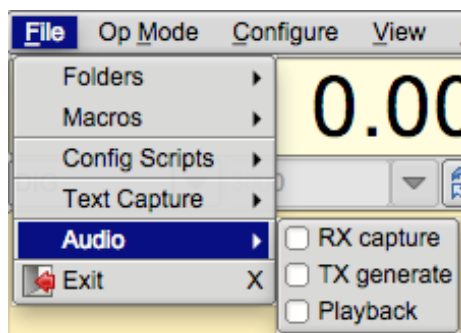


Figure 6.31: Audio

- Rx Capture - allows capturing the incoming audio to a wav file
- Tx Generate - allows capturing the generated tx audio to a wav file

- Playback - playback a previously captured or generated wav file

#### 6.11.1.1.5 Exit

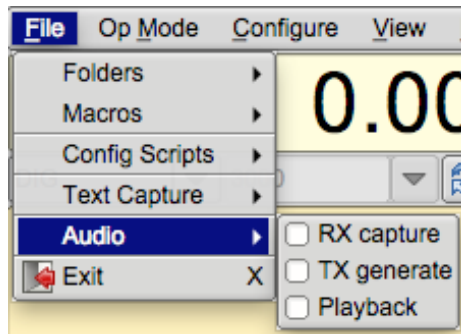


Figure 6.32: Exit

Exit - exit the program closing down the various interfaces in a nice controlled manner.

#### 6.11.1.2 Configure Scripts

Executing and Generating Scripts

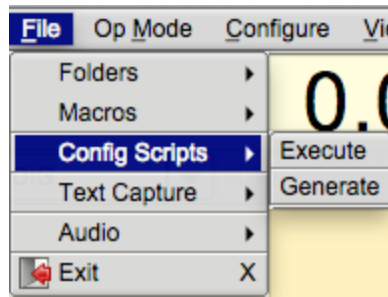


Figure 6.33: Config Scripts

Additional information:

[Configure Scripting](#)

[Configure Script Commands](#)

#### 6.11.1.3 Op Mode

Op Mode - the current operating mode will show as a highlighted menu item.

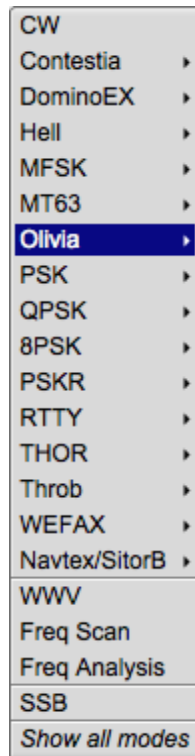


Figure 6.34: Op Mode

#### 6.11.1.3.1 CW

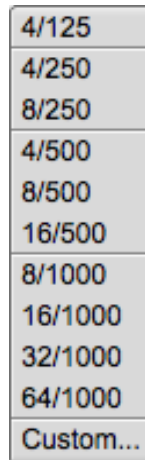
CW - receive CW 5 to 200 WPM and transmit on any audio frequency using AF CW.

Additional information:

[CW](#)

[CW Configuration](#)

## 6.11.1.3.2 Contestia



4/125
4/250
8/250
4/500
8/500
16/500
8/1000
16/1000
32/1000
64/1000
Custom...

Figure 6.35: Contestia

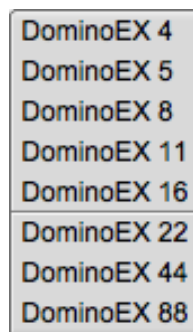
Additional information:

[Contestia](#)

[Contestia Configuration](#)

[Contestia Mode Table](#)

## 6.11.1.3.3 DominoEX



DominoEX 4
DominoEX 5
DominoEX 8
DominoEX 11
DominoEX 16
DominoEX 22
DominoEX 44
DominoEX 88

Figure 6.36: DominoEX

Additional information:

[DominoEX](#)

[DominoEX Configuration](#)

[DominoEX Mode Table](#)

## 6.11.1.3.4 Hell



Figure 6.37: Hell

Additional information:

[Hellschreiber](#)  
[Feld Hell Configuration](#)  
[Hell Mode Table](#)

## 6.11.1.3.5 MSFK

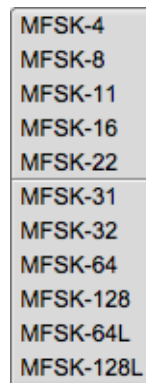


Figure 6.38: MSFK

Additional information:

[MFSK](#)  
[MSFK Mode Table](#)

## 6.11.1.3.6 MT-63

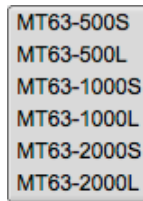


Figure 6.39: MT63

Additional information:

[MT63](#)

[MT63 Configuration](#)

[MT-63 Mode Table](#)

## 6.11.1.3.7 PSK

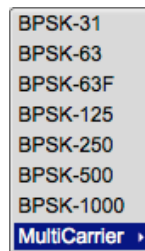


Figure 6.40: PSK

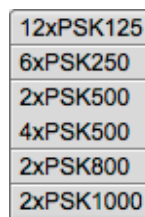


Figure 6.41: Multi Carrier PSK

Additional information:

[PSK Single and Multi-Channel Modems](#)

[PSK Mode Table](#)



## 6.11.1.3.8 PSKR

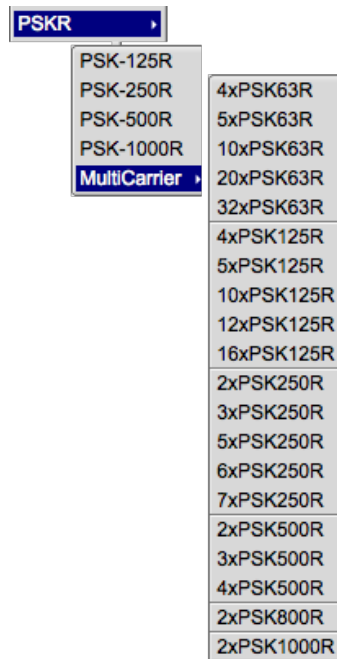


Figure 6.42: PSKR

All PSKR modes are phase shift keying, with FEC and interleaving.

Additional information:

[PSK Single and Multi-Channel Modems](#)

[PSK/PSKR Mode Table](#)

## 6.11.1.3.9 QPSK

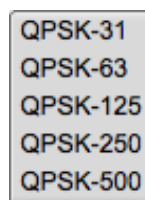


Figure 6.43: QPSK

Additional information:

[PSK Single and Multi-Channel Modems](#)

[QPSK Mode Table](#)

## 6.11.1.3.10 8PSK

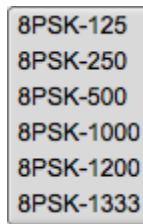


Figure 6.44: 8PSK

Additional information:

[PSK Single and Multi-Channel Modems](#)  
[8PSK Mode Table](#)

## 6.11.1.3.11 Olivia

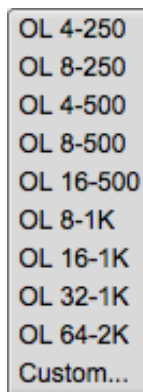


Figure 6.45: Olivia

Additional information:

[Olivia](#)  
[Olivia Configuration](#)  
[Olivia Mode Table](#)

## 6.11.1.3.12 RTTY

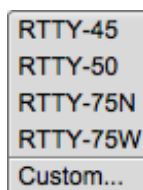


Figure 6.46: RTTY

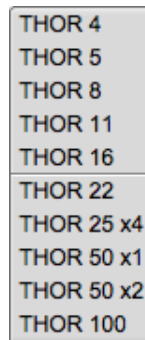
Additional information:

[RTTY](#)

[RTTY / FSK Configuration](#)

[RTTY Mode Table](#)

#### 6.11.1.3.13 THOR



THOR 4
THOR 5
THOR 8
THOR 11
THOR 16
THOR 22
THOR 25 x4
THOR 50 x1
THOR 50 x2
THOR 100

Figure 6.47: THOR

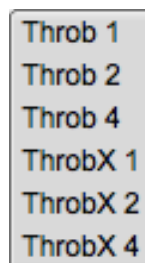
Additional information:

[Thor](#)

[Thor Configuration](#)

[Thor Mode Table](#)

#### 6.11.1.3.14 THROB



Throb 1
Throb 2
Throb 4
ThrobX 1
ThrobX 2
ThrobX 4

Figure 6.48: throb

Additional information:

[Throb](#)

[Throb Mode Table](#)

## 6.11.1.3.15 WEFAX

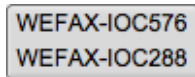


Figure 6.49: WEFAX

Additional information:

[WEFAX](#)

## 6.11.1.3.16 NAVTEX / SITOR

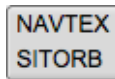


Figure 6.50: NAVTEX / SITOR

Additional information:

[NAVTEX and SITOR-B](#)

## 6.11.1.3.17 SSB, WWV, Freq Anal, and Freq Scan

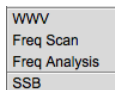


Figure 6.51: Analysis / Scan

- SSB - fldigi does not transmit, but can be used for rig control, signal frequency analysis and logging
- WWV - special receive only modem used for calibrating sound card
- Freq Scan - Evaluate Rx/Codec Signal Path
- Freq Anal - used for carrier detection and frequency measurement

Additional information:

[Frequency Analyzer](#)

[Digiscope Display - WWV mode](#)

[Configure FFT Scan](#)

## 6.11.1.4 Configure

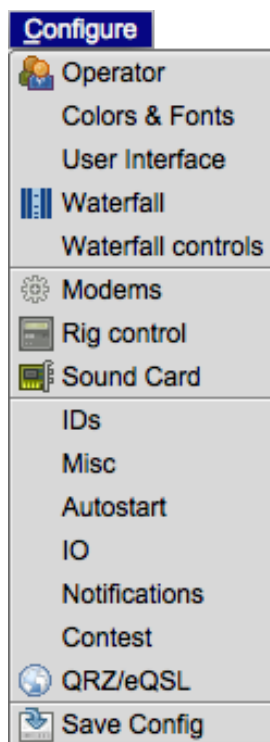


Figure 6.52: Configure

See [Configuration Page](#)

## 6.11.1.5 View

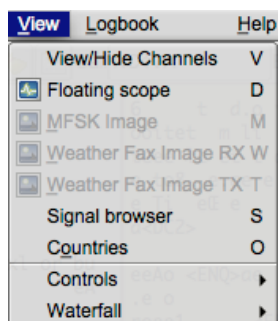


Figure 6.53: View

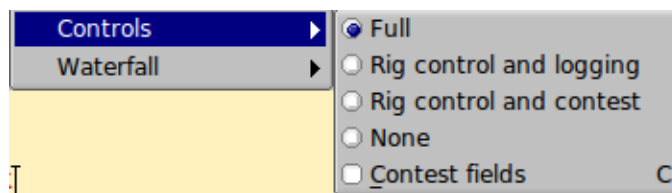


Figure 6.54: View Controls



Figure 6.55: View Waterfall

- View/Hide Channels - Opens / Closes multi-signal viewer, shares space with Rx/Tx panels
- Floating Scope - Opens up a resizable, moveable scope display
- MFSK Image - Opens the MFSK picture image (if being received)
- Weather Fax Image - Opens WEFAX send/receive dialog
- Signal browser - open the psk/rtty viewer dialog to display up to 30 simultaneously decoded psk signals Dialog viewer contains shared data / control with the View/Hide Channel viewer
- Countries - DXCC entities
- Controls
  - Full - show all logbook and rig controls
  - Rig control and logging - minimizes the logging panel for normal QSO entries
  - Rig control and contest - minimizes the logging panel for Contest entries
  - None - removes the Rig Control / Log panel from the main User Interface
  - Contest fields - Display alternate 2nd line in qso logging area; provides access to contest logging fields
- Waterfall
  - Docked scope - toggles the visibility of the docked scope display to the right of the waterfall
  - Min WF Controls - toggles the visibility of various waterfall controls as configured by the user

## 6.11.1.6 Logbook

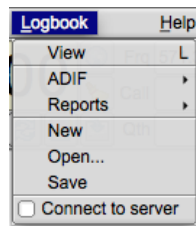


Figure 6.56: Logbook

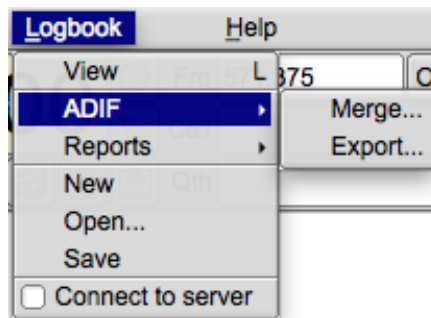


Figure 6.57: Logbook ADIF

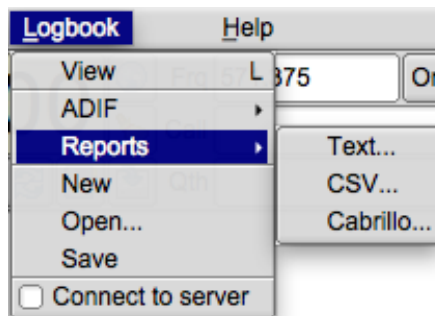


Figure 6.58: Logbook Reports

- View - open the logbook dialog
- New - create a new logbook
- Open - open an existing logbook
- Save - save the current logbook
- ADIF / Merge - merge adif data from an ADIF file

- ADIF / Export - export selected or all logbook records to an ADIF formatted file - see [Log Exports](#).
- Reports / Text ... - export selected or all logbook records to a text file suitable for printing
- Reports / CSV ... - export selected or all logbook records to a tab delimited file
- Reports / Cabrillo ... - create a Cabrillo contest report.
  - see [Cabrillo Reports](#).

#### 6.11.1.7 Help

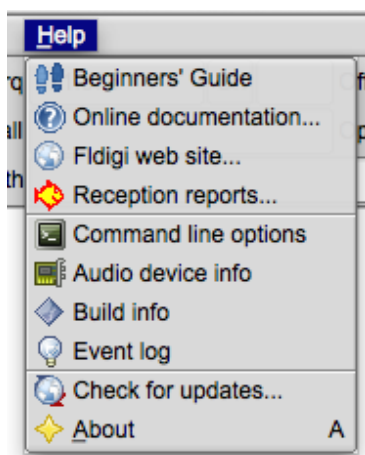


Figure 6.59: Help

- Beginners' Guide
- On line documentation... - open up default browser to the on-line Help site
- Fldigi web site... - open up default browser to the [www.w1hkj.com](http://www.w1hkj.com) primary web page
- Reception reports... - open up browser to the <http://pskreporter.info> web page preset to your callsign
- Command line options - display a list of all [command line switches](#) available to the fldigi user
- Audio device info - displays information about all audio devices detected on the computer system
- Build info - displays all relevant information regarding the compilation and link for the application
  - [build info](#)
- Event log - opens a text display window that records various events depending on the level of reporting depth selected. This is a useful window for reporting problems with the program to the developers.
- Check for updates... fldigi silently opens a download web site, checks and reports on whether a new version is available.
- About - Version number and a little about the programmers



## 6.11.1.8 Spot, Tx RSID, Rx RSID, and Tune Buttons



Figure 6.60: Menu

- Spot button - The "Spot" light button is visible if callsign spotting is enabled. Use this button to toggle the callsign spotting reporter 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. See [PskReporter](#) and [Notifier](#).
- RxID button - toggles the detection of [Reed Solomon Identification](#) codes.
- TxID button - toggles the transmission of the RSID signal.
- Tune button - toggles the "Tune" mode which causes fldigi to insert a tone at the current waterfall frequency. The peak-to-peak amplitude of this tone is the standard by which you should [set your transmitter drive](#) or adjust your antenna matching network.
- The right most button is normally not visible. This is the count-down timer button that is enabled when a macro button has been configured to repeat after a specified number of seconds. This button shows the count-down to the next transmission. Pressing the button disables the count and restores fldigi to normal keyboard operation.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.12 Mouse and 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!

### 6.12.1 Main window

#### 6.12.1.1 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. A full list can be found on the [FLTK web site](#)

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

##### 6.12.1.1.1 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.

### 6.12.1.1.2 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:

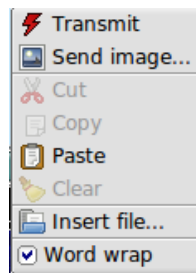


Figure 6.61: Fldigi Receiving Editing Tx text

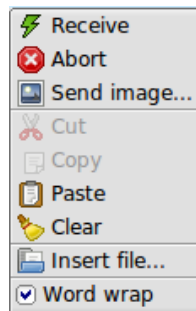


Figure 6.62: Fldigi Transmitting

- 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.

#### 6.12.1.2 Waterfall display

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

##### 6.12.1.2.1 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

##### 6.12.1.2.2 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
- Ctrl-Alt-Left click on the mouse when the mouse pointer is in the waterfall - if operating with flrig and a supported transceiver such as the FT-950, will set the transceiver manual notch frequency to the frequency under the mouse cursor. Repeat the Ctrl-Alt-Left click to disengage the notch. Disengagement is not dependent on where the cursor is on the waterfall, just that it be in the waterfall.

#### 6.12.1.2.3 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

#### 6.12.1.2.4 Digiscope display

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

#### 6.12.1.2.5 Rig control window

There are some shortcuts in addition to those described in the [Rig Control](#)

#### 6.12.1.2.6 Frequency display

- Left/Right arrow key - change the frequency by one 1 Hz
- Up/Down arrow key - change the frequency by 10 Hz

#### 6.12.1.2.7 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

#### 6.12.1.2.8 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

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.13 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.

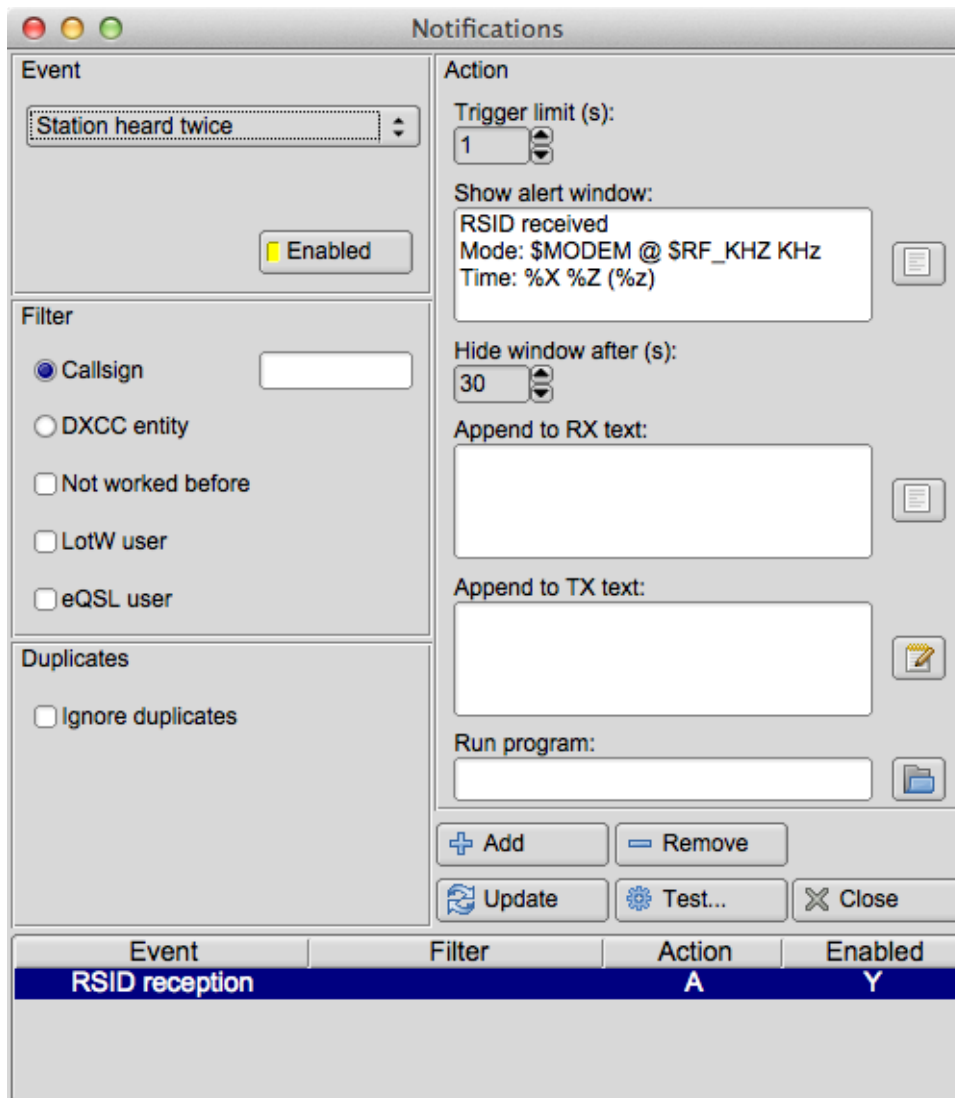


Figure 6.63: Notifications

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.

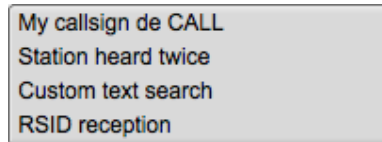


Figure 6.64: Notification Event Types

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.
4. RSID reception.

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.

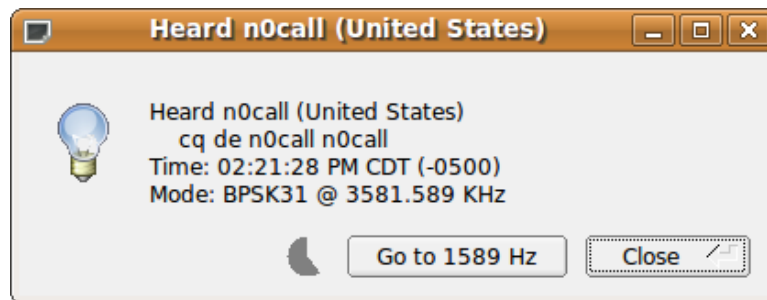


Figure 6.65: Notifier Popup

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.

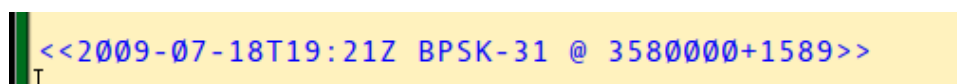


Figure 6.66: Notifier RX Text

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.

#### Additional 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.

```
-----snip
#!/usr/bin/perl
exec("notify-send", "-t", "5000", "-i", "/usr/share/pixmaps/flddigi.xpm",
    "Heard " . $ENV{"FLDIGI_NOTIFY_CALLSIGN"} . " (" . $ENV{"FLDIGI_NOTIFY_COUNTRY"} . ")",
    $ENV{"FLDIGI_NOTIFY_STR_0"});
snip-----
```

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.14 Operating Controls and Displays

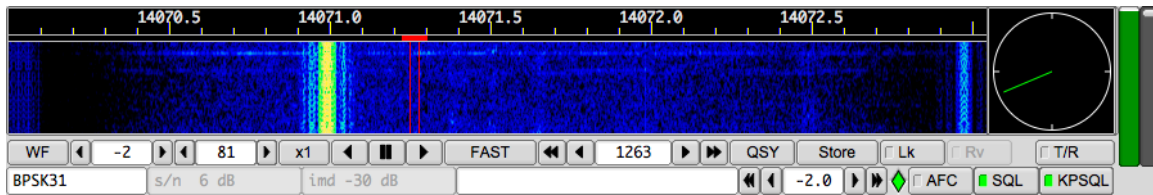


Figure 6.67: FLDIGI Controls

The main display for fldigi is the waterfall display shown above in color and in scale x1.

The button **WF** toggles the display between a waterfall, a spectrum display and an oscilloscope type view of the Rx and Tx signals. 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 **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.

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.

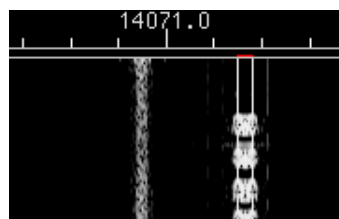


Figure 6.68: PSK Waterfall X1 scale

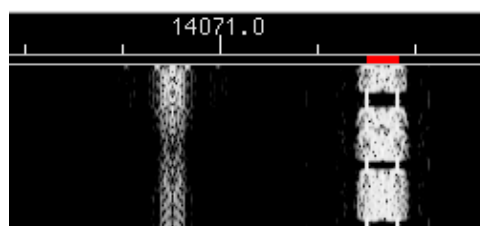


Figure 6.69: PSK Waterfall X2 scale



Figure 6.70: PSK Waterfall X4 scale

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 next control is your transceive audio frequency. In the display above you can see that the audio signal is 1500 Hz. The red cursor is centered beneath 14071.500 Mhz. The transceiver was set to 14070 Mhz. The arrow key pairs move up/down in cycles and tens of cycles. You can fine tune the receive point using this control.

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 51 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 affect.

The **QSY** button is very specific to transceivers interfaced with either hamlib, rigcat or when fldigi is used with flrig. Each transceiver has a sweet spot associated with its bandwidth controller. For the FT950 is 1500 when the

transceiver is set to PKT-U. As the transceivers bandwidth is changed the changes occur centered at this frequency. Set the "Sweet spot" frequency (Config / Misc / Sweet Spot) to 1500. 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 transceiver 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! The QSY button will be dimmed and not activated if you do not have a method of transceiver control enabled for your transceiver. Right click on the QSY button to undo the action.

This can also be achieved with the <QSYTO> and <QSYFM> macro tags. If you combine these two with the <FILWID:nnn> command you can center the signal in the passband and also narrow the transceiver passband, i.e. when used with the FT950

- <QSYTO><FILWID:200>
- <QSYFM><FILWID:2400>

The **M**> 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 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.

"Lck"	Before "Qsy"		After "Qsy"	
	RX	TX	RX	TX
OFF	1002 / 7071.002	1002 / 7071.002	1500 / 7071.002	1500 / 7071.002
ON	1002 / 7071.002	1000 / 7071.000	1500 / 7071.002	1500 / 7071.002
ON	1000 / 7071.000	1800 / 7071.800	1500 / 7071.000	1500 / 7071.000

With "Lk" off the TX audio frequency is always synchronized with the RX frequency.

With "Lk" 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 "Lk" is ON moving the cursor around will ONLY AFFECT the RX frequency and NOT the TX frequency.

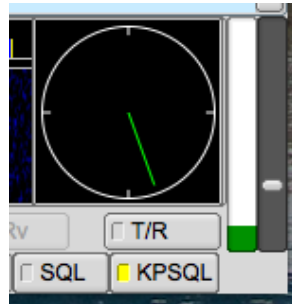


Figure 6.71: SQL/KPSQL Level Adjustment

The **AFC** and **SQL** buttons enable or disable the respective function in the software. 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.



Figure 6.72: KISS IO enabled



Figure 6.73: ARQ IO enabled

**KPSQL** squelch operates on the power aspect of a signal unlike SQL which measures the reception quality of the signal. For KPSQL to operate correctly the waterfall speed setting must be set to either NORM or FAST, FAST being the better option. The level setting slider is active for KPSQL when the KPSQL button is enabled (indicated by either a yellow or green square on the button surface). When disabled the slider adjustment affects SQL. KPSQL operates using two different threshold detection methods. With KPSQL button selected the user sets the threshold using the slider to the right of the vertical level indicator. With KPSQL deselected the detection level reverts to the histogram method. This method automatically adjust the threshold to just above the noise level and is used when the utmost signal sensitivity is desired.

**SQL** and **KPSQL** functional differences:

**SQL** - Inhibits character reception.

**KPSQL** - Inhibits the data transfer to the modem modulator.

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.

### 6.14.1 Mode Status Indicators

The lower left corner of the main display (MFSK-16) in the view above 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 heirarchy. 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.

### 6.14.2 Transmit level attenuator

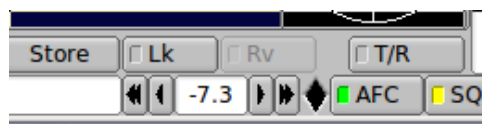


Figure 6.74: TX level

It is often difficult to adjust the audio drive for the point where ALC is just barely active. Mixer controls are OK, but not usually designed for very small changes. They are after all designed for adjusting listening levels. fldigi provides the ability to control the audio drive in increments of 0.1 dB over a 30 dB range. This control is located in the bottom right corner of the main dialog:

Set this control for -6 dB and then adjust the sound card mixer control for the best ALC level you can achieve. Then adjust the Tx-level control for best "just visible" ALC on the transceiver. That should give you a very clean PSK signal.

### 6.14.3 Operating split with fldigi / flrig

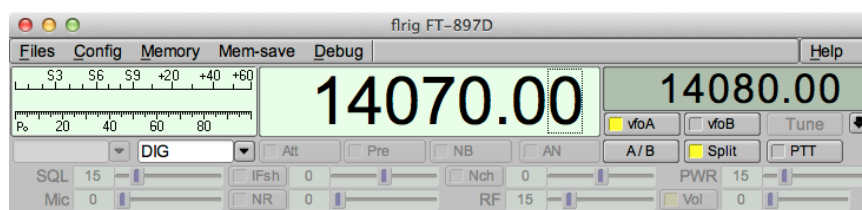


Figure 6.75: TTY Split Ops

Note: In the above example most features are disabled as they are not available for the indicated radio.

On flrig:

Unless specifically supported by the transceiver and implemented in flrig.

- VFO A is always the RX frequency.
- VFO B is always the TX frequency.
- Left Click on the A/B swaps A & B VFO frequencies.
- Right Click on the A/B assigns A VFO frequency to B VFO.
- Using the mouse wheel adjust VFO B to where you want to transmit.
- Click the SPLIT button, it will change color when it is active.
- Press the VFO B button to listen on the Tx frequency, be sure to press Split again before capturing that rare DX station.

On fldigi:

- Tune the waterfall cursor to the Rx signal
- Transmit as usual, be sure you are not doubling.

See [FLRIG Help](#) for specific operating procedures.

Some additional information can also be found here [Rig Control Page](#)

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.15 Operating Multiple Copies

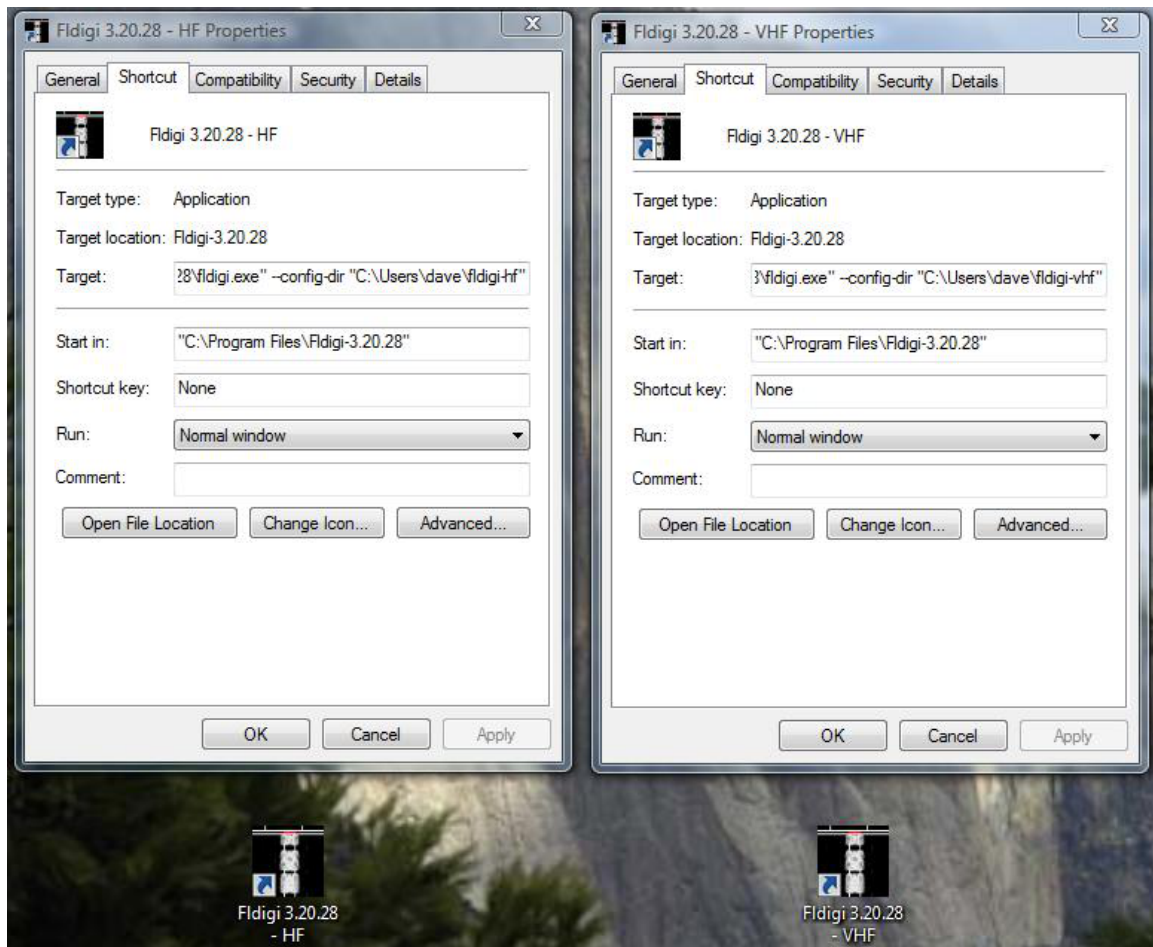


Figure 6.76: Multiple Copies of FLDIGI

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 it's xml-rpc socket port. You change the address/port pairs on both the fldigi launcher and the paired



application such as flarq or flrig.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.16 PSK Reporter

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

- The decoded text, obtained from the Rx Panel text or the multi-channel browser. Note: **the psk browser must be visible** or decoding and subsequent psk reporting will not occur. If you do not want the psk browser panel visible in the main dialog then use the separate browser dialog. It can be minimized and signal decoding will continue.
- The log data
- Data entered manually

The configuration for the PSK reporter in 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 color 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:

### 6.16.1 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.

### 6.16.2 Send reception report when logging a QSO

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

### 6.16.3 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 sub-band, and in that case they won't have to synchronise fldigi's frequency display with the rig all that often.

### 6.16.4 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.

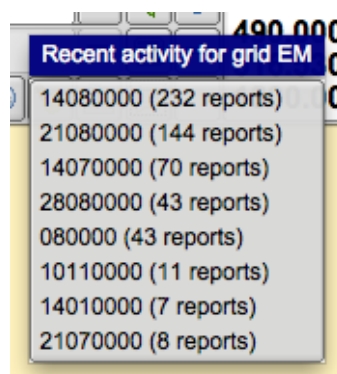


Figure 6.77: Activity Reports

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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.17 Rig Control

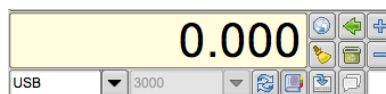


Figure 6.78: Initial Rig Setup

CAT not enabled



Figure 6.79: Docked Rig Control

CAT not enabled, Manual Entry of transceiver frequency

**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 [pskre-porter/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 increment by clicking/pressing on the upper half of the digit and decrement by clicking/pressing the lower half of the digit. 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 hovering the mouse pointer over the frequency display and entering the desired frequency (in Khz) via the keyboard. Pressing <return> or <enter> on completion sets the frequency to an active state, effecting rig control and future log entries. Don't forget the decimal point if you are entering a fractional KHz value.

The Smeter / Power-meter display is only active if fldigi is configured for xmlrpc control with flrig. The transceiver must support the smeter and power out CAT query. fldigi does not try to estimate the smeter reading based on the audio stream. You can toggle viewing the Smeter / Power-meter display using the button just to the right of the shared space used by the "Smeter/Power-meter" and the "Mode/Bandwidth" controls. Be sure to save the configuration for this to persist the next time fldigi is executed.

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.

### 6.17.1 Rig Configuration

#### Hardware PTT control

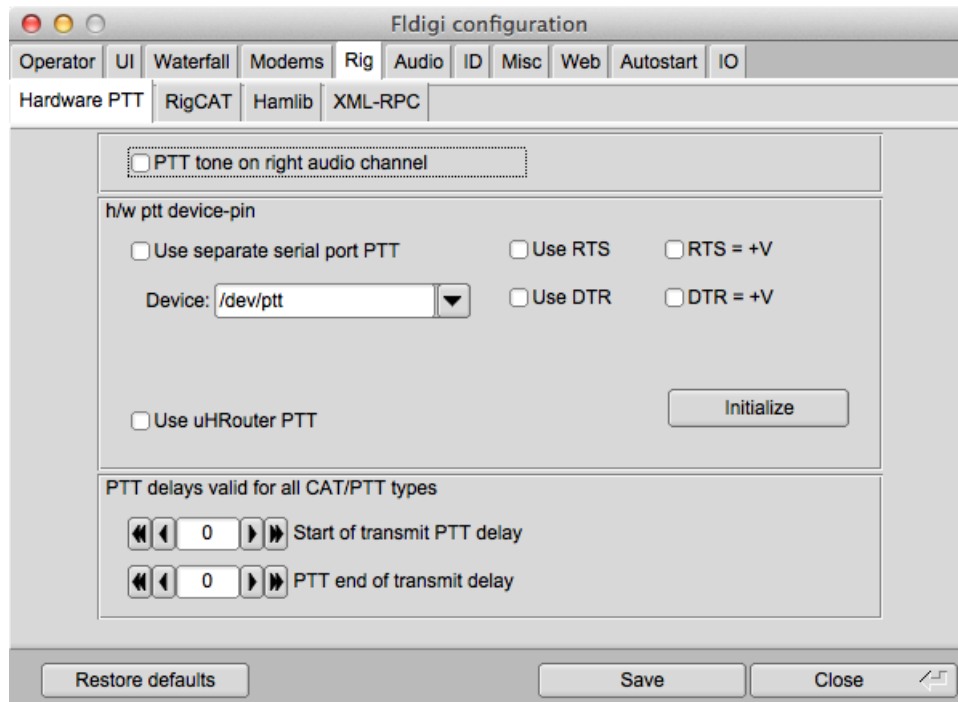


Figure 6.80: Hardware PTT control

### 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. Jim, W5ZIT, has provided details on building an [interface](#) for this type of PTT control.

### 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 your 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)

Fldigi 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.

### $\mu$ H Router (MacOS X)

Similar functionality can be achieved on the Macintosh operating system using ' $\mu$ H Router' by Kok Chen, W7AY. See  [\$\mu\$ H Router Website](#) for specific details and requirements. A selectable (checkbox) option will be avail-

able on the Rig->Hardware PTT Configuration panel.

### PTT delays

You can accommodate delays in transceiver switching between receive and transmit by adjusting the PTT delays. The control values are in milliseconds. These controls have no effect on external PTT circuits such as those implemented in the Signalink interfaces. They rely on detecting the audio data stream. You can use a combination of macro tags in a macro key definition to achieve a resolution. For example try a macro definition similar to this to insure that the RSID is sent via a slow FM xcvr (or via a VHF repeater)

```
<TX><MODEM:NULL><IDLE:2.5>
<!MODEM:MT63-500>
<TXRSID:on>
```

Change the idle time value (in fractional seconds) to suit your needs.

### 6.17.2 RigCAT control

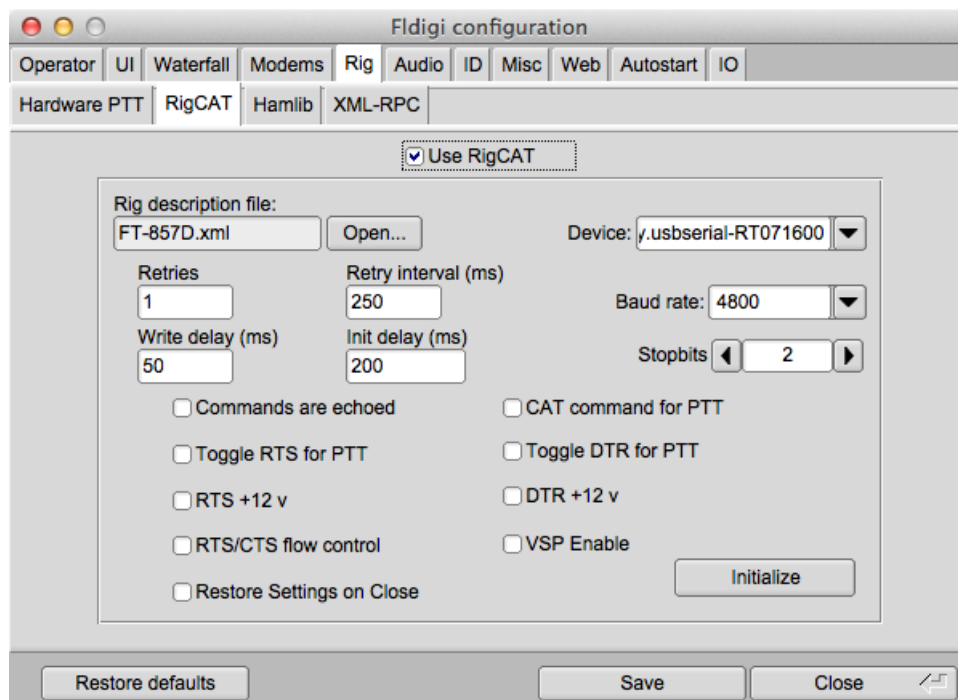


Figure 6.81: RigCAT control

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 specific xml 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/xm1s> or from the archives [web page](#). 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.

### 6.17.3 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.

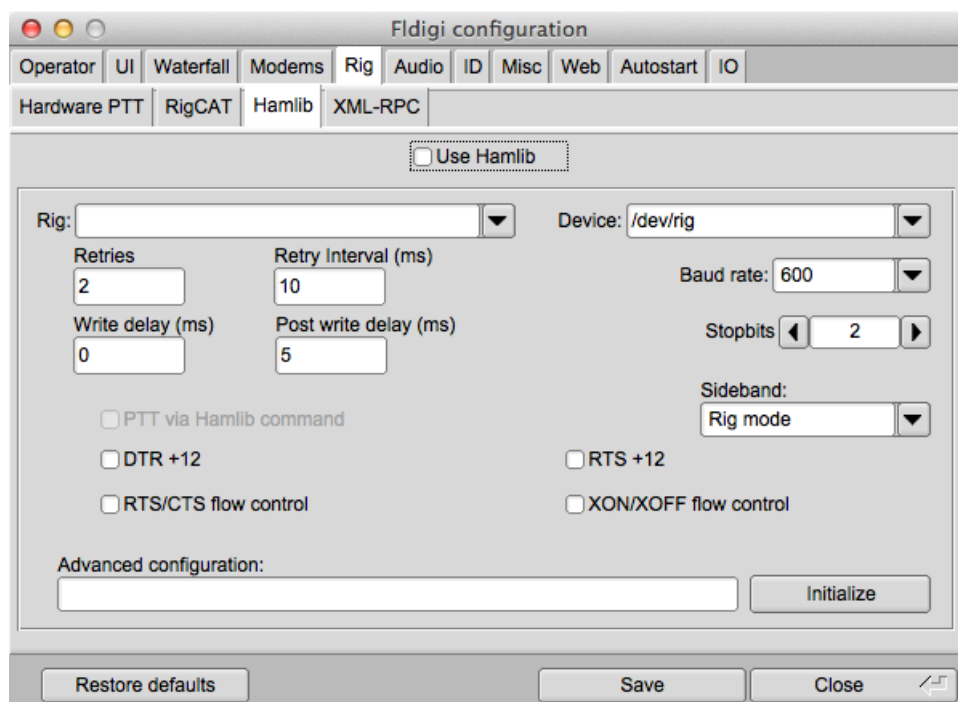


Figure 6.82: Hamlib CAT Control

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 annunciating them on the rig control frequency display.

#### 6.17.4 Xml-Rpc CAT

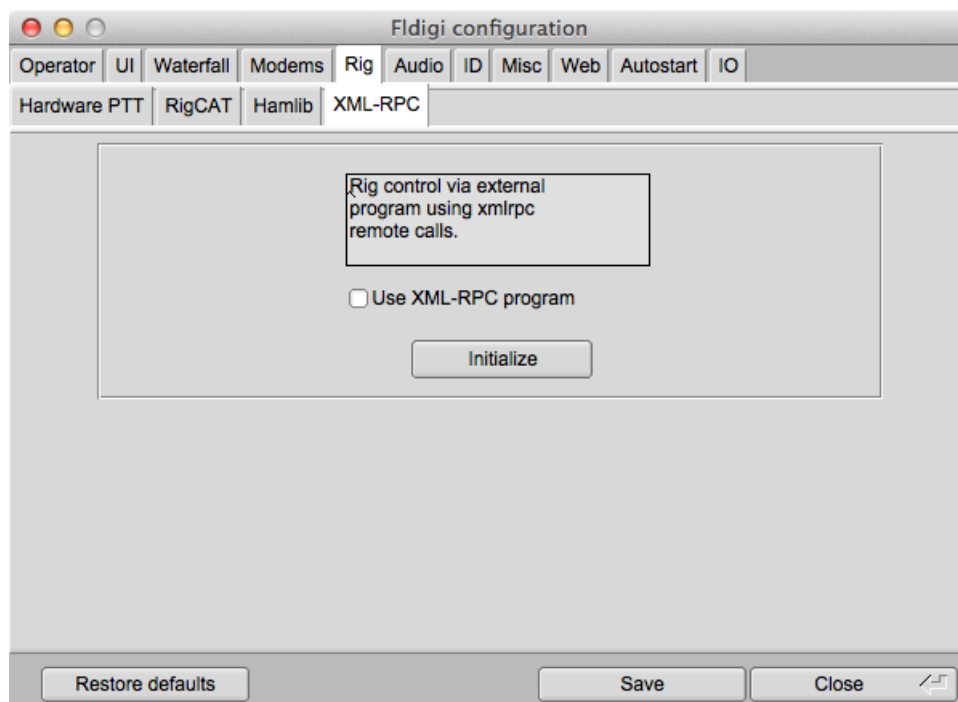


Figure 6.83: Xml-Rpc CAT Control

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.

[Return to Top of Page](#)



[Return to Main Page](#)

## 6.18 RST and RSQ Reporting

### 6.18.1 RST

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

#### 6.18.1.1 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 todays jargon)

#### 6.18.1.2 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

#### 6.18.1.3 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 trac of modulation
9. Perfect tone, no trace of ripple, or modulation of any kind

## 6.18.2 RSQ

Give the report as RSQ for digital modes, but especially BPSK and QPSK; see: <http://www.psb-info.net/RSQ-Reporting-Table.html>

### 6.18.2.1 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

### 6.18.2.2 STRENGTH

1. Barely perceptible trace
2. n/a
3. Weak trace
4. n/a
5. Moderate trace
6. n/a
7. Strong trace
8. n/a
9. Very strong trace

### 6.18.2.3 QUALITY

1. Splatter over much of the visible waterfall
2. n/a
3. Multiple visible pairs
4. n/a
5. One easily visible pair
6. n/a
7. One barely visible pair
8. n/a
9. Clean signal - no visible unwanted sidebar pairs

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.19 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.

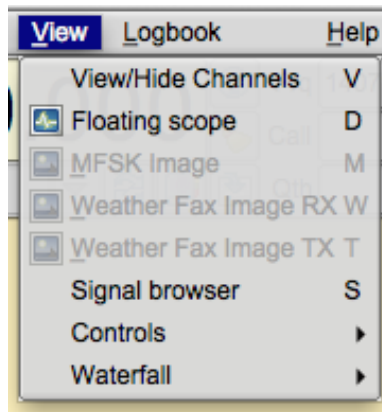


Figure 6.84: View Menu

It looks like this:

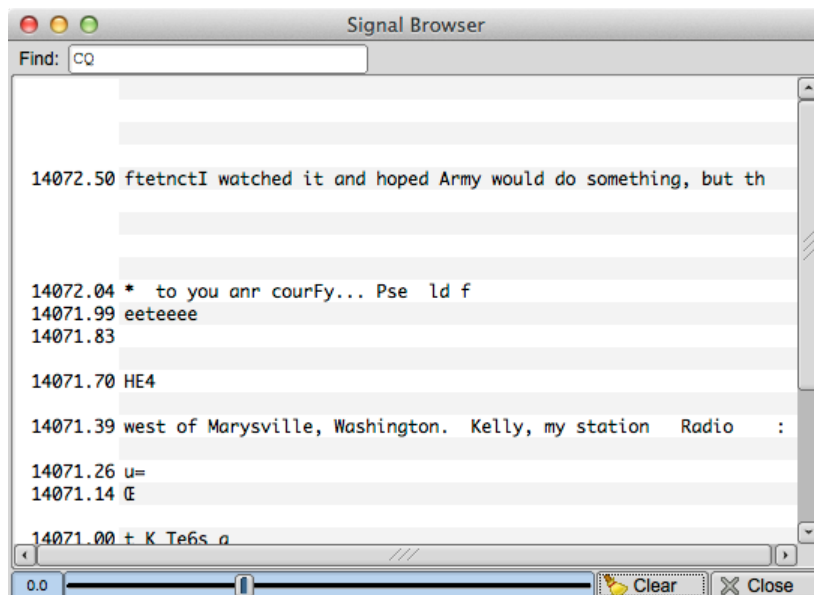


Figure 6.85: Browser Panel

Toggle the visibility of the embedded viewer by selecting the **View/Hide Channels** on the same menu list. Your main fldigi screen then opens a browser panel on the left side of the Tx/Rx panel(s).

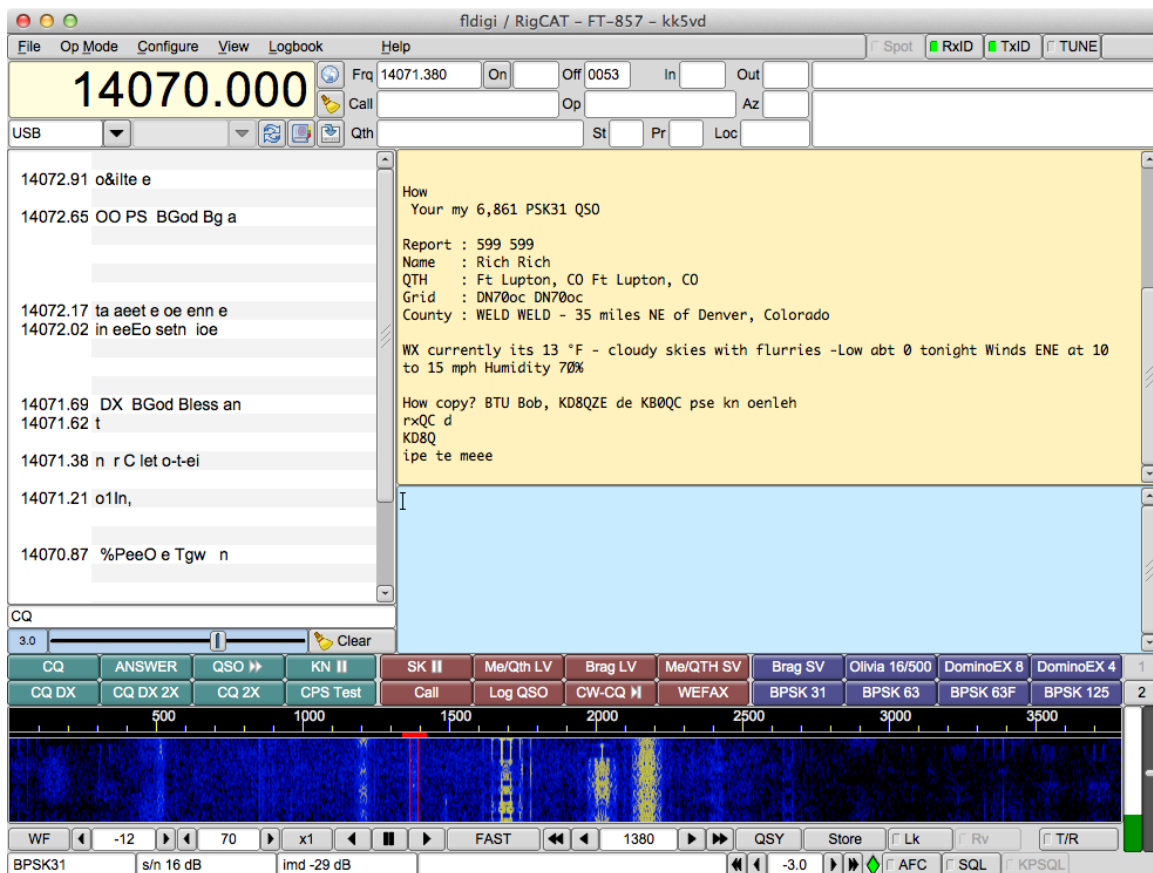


Figure 6.86: Main Dialog Browser

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.

Configuring the browser:

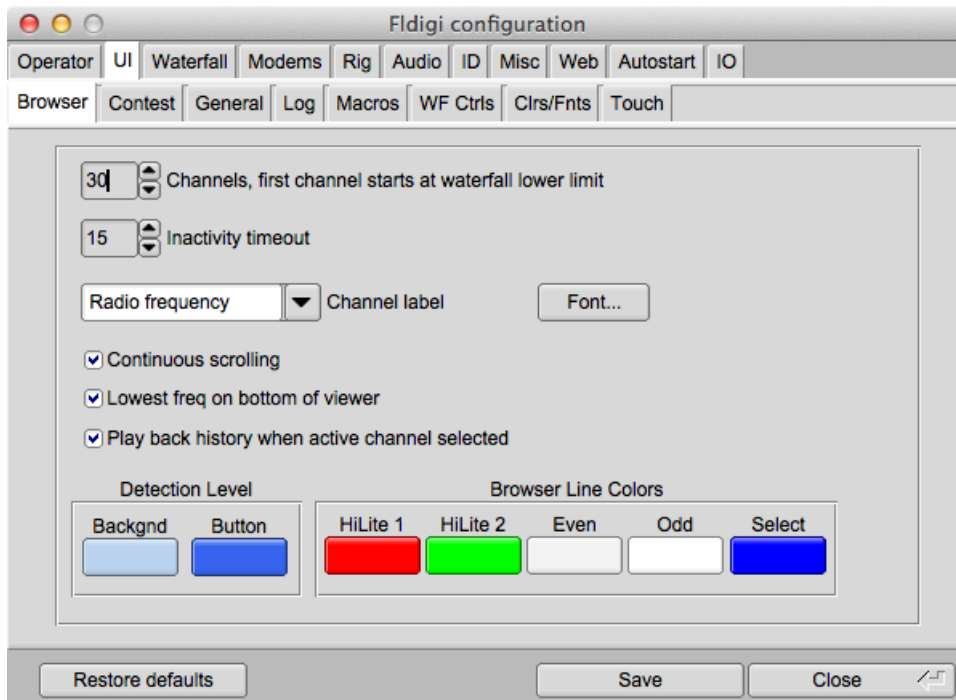


Figure 6.87: Browser Configure

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 whether you want a Marque type of continuous scrolling, or simply clear the line when it is filled.
5. select what kind of label annotation you want on each line
6. select the font and font size to be used in the browser
7. You can enter any text you want to search for in the **Seek Regular Expression** widget. This text can be a simple text snippet such as "CQ" or any regular expression. 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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.20 Supporting Data Files Acquisition

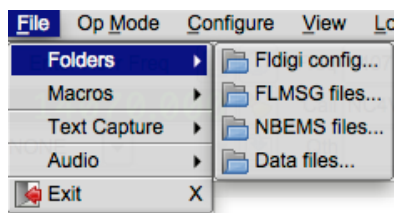


Figure 6.88: Data File Menu

To update support data file(s). Mouse click on the 'Data files...' menu item located in the above drop down menu.

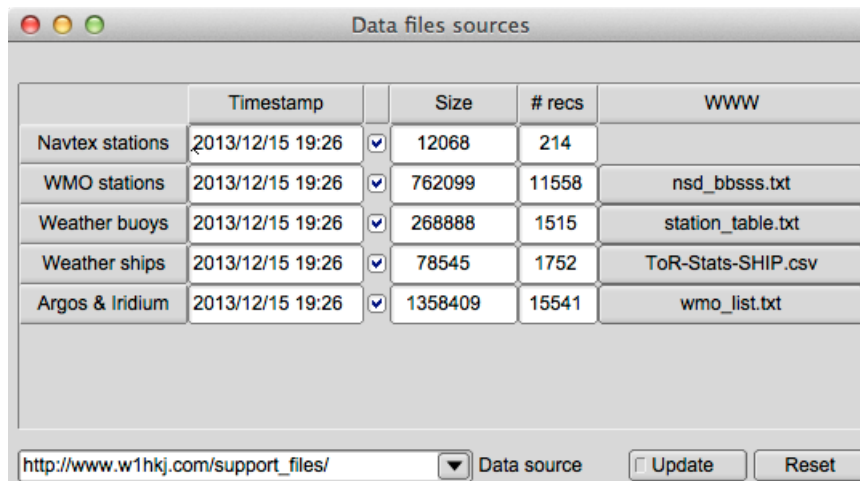


Figure 6.89: Data File Source

The dialog box contains a list of the available file(s) for downloading. Click on the 'Update' button to initiate the data transfer.

The buttons under WWW column are direct links to the data source. Selecting them either displays or downloads the data from your web browser.

**Note:** Data source selection contain a single entry. User selection not required.

[Return to Top of Page](#)  
[Return to Main Page](#)

## 6.21 SYNOP decoding

### 6.21.1 What is Synop

#### 6.21.1.1 Weather data

SYNOP, which stands for **surface synoptic observations** is a numerical code used for reporting weather observations made by manned and automated fixed land weather stations. It is also called FM-12 by the

World Meteorological Organization. It is closely associated to FM 13-XIV SHIP, for report of surface observation from a sea station (Created in 1982) and FM 14-XIV SYNOP MOBIL, for reports of surface observation from a mobile land station.

Here is an example of undecoded SYNOP weather report:

```
13586 31530 80000 10036 20029 39821 40218 53002 71022 886//
333 20029 88715=
```

Many SYNOP reports are available on web sites such as [Ogimet](#) .

#### 6.21.1.2 Synop reports broadcasts

The most important broadcaster on SYNOP data is DWD, in RTTY mode.

##### 6.21.1.2.1 Deutsche Wetterdienst schedule

SYNOP reports are typically sent every six hours by **Deutscher Wetterdienst** on **shortwave** and **low frequency** using **RTTY**.

The baud rate must be 50 bauds.

Freq	Station	Time	
147,300 kHz	DDH47	05.00 - 22.00	+ / - 42,5
4583,000 kHz	DDK2	00.00 - 24.00	+ / - 225,0
7646,000 kHz	DDH7	00.00 - 24.00	+ / - 225,0
10100,800 kHz	DDK9	00.00 - 24.00	+ / - 225,0
11039,000 kHz	DDH9	05.00 - 22.00	+ / - 225,0
14467,300 kHz	DDH8	05.00 - 22.00	+ / - 225,0

##### 6.21.1.2.2 Other broadcasters

Until recently, many shore stations used to transmit Synop reports by Morse Code ("*Ocean Yacht Navigator*", Kenneth Wilkes, 1976 ). Still, according to **WMO information for shipping**, a couple of maritime stations might broadcast these informations in different mode that RTTY, but this has not been tested yet. There are just anecdotically cited here.

#### Maritime Radio Station of Mumbai

Met codes: FM 12 - SYNOP, FM 13 - SHIP for Sea areas I.

Band	Freq
WT (MF)	500 kHz
WT (MF)	521 kHz
WT (HF)	8630 kHz
WT (HF)	12710 kHz

SYNOP decoding is not accessible from Morse CW modem yet.

#### Navtex station of Mumbai. COMCEN

FM 12 SYNOPSIS, FM 13 SHIP for sea area I (Time schedule 0900, 2100)

	Freq
Navtex (Also called MSI as Maritime Safety Information)	518 kHz

SYNOP decoding is not accessible from Navtex modem yet.

#### Marine Rescue Co-ordination Centre: Guayaquil, Radio IOA (Instituto Oceanografico Armada) o Radio Naval

Meteorological information in process: Text, SHIP, SYNOP.

Mode	Freq
AM	1515 kHz



### 6.21.2 Parameters

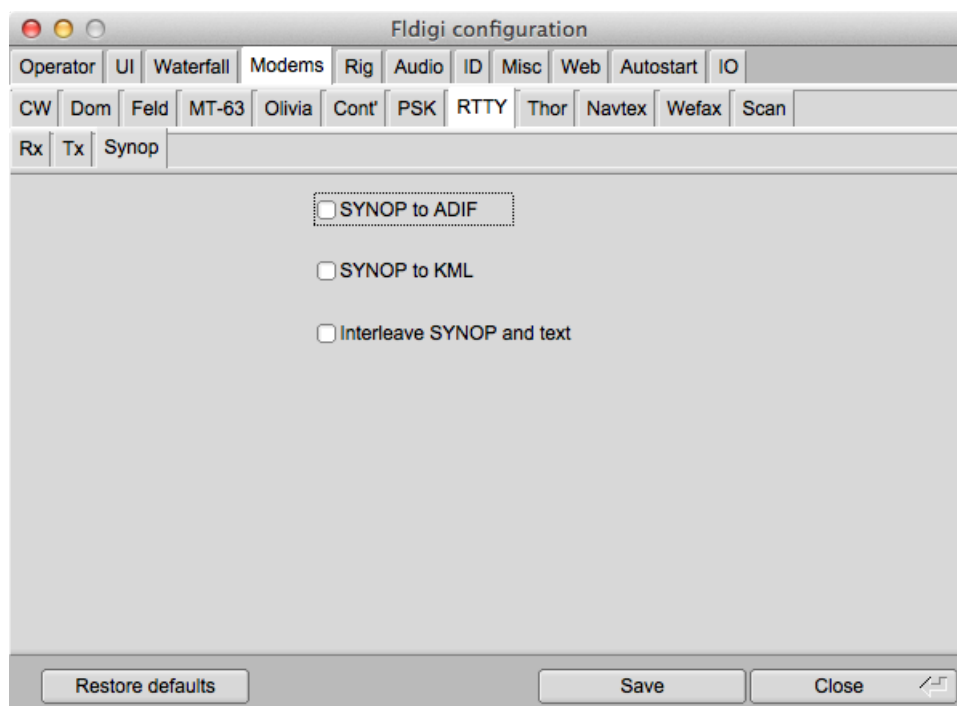


Figure 6.90: Synop configuration in RTTY tab

By default, there is not SYNOP decoding of RTTY reception, so its behavior does not change. However, it receives three new parameters related to Synop decoding.

SYNOP reports can be saved to the current ADIF log file: They are visible in the journal of QSOs with the content of the report, the timestamp, and the Maidenhead locator equivalent to the geographical coordinates of the emitter of the report.

SYNOP weather reports can also be plotted on KML files: They can then be visualized in KML viewers such as Google Earth.

If SYNOP reports are saved either to ADIF or KML files, fldigi attempts to detect and decode RTTY messages. When these messages are detected, they are displayed in the reception window. An extra parameter allows to replace the coded Synop message, giving clarity at the risk of losing information, or mix the decoded reports with the original group of digits (Interleaving of SYNOP and text).

### 6.21.3 Reception window

When SYNOP messages are detected, they are displayed in the reception window in "red" characters (depicted here in *italic*). Here are some examples (Interleaved text and SYNOP reports):

```
13661 16123 99276 70216 46///
WMO Station=13661
WMO station=WMO_13661
UTC observation time=2013-02-16 12:00
QLLLL token=Present
Longitude=-21.6
Latitude=27.6
```

```
/// 40206 52011
222// 00203;
Sea surface temperature=20.3 -C
Temperature type=Intake measurement

64522 16123 99591 70347 46///
WMO Station=64522
WMO station=WMO_64522
UTC observation time=2013-02-16 12:00
QLLLL token=Present
Longitude=-34.7
Latitude=59.1

./ 49920 57004.
::(1056;
, 26555 16123 998#6
WMO Station=26555
WMO station=WMO_26555
UTC observation time=2013-02-16 12:00

70269 46/// // 40168 52008;
WMO Station=70269
WMO station=WMO_70269
Precipitations=Omitted, no observation
Station type=Automated station. No observation (No 7WW)
Visibility=Missing
Wind direction=No motion or no waves
Wind speed=0 knots (Estimated)
Sea level pressure=1016 hPa
Pressure tendency=Increasing steadily. Raises
Pressure change=0.8 hPa

25617 16123 99867 1173!((6;
WMO Station=25617
WMO station=WMO_25617
UTC observation time=2013-02-16 12:00
QLLLL token=Present

149 52001;

13660 16123 992!;
WMO Station=13660
WMO station=WMO_13660
UTC observation time=2013-02-16 12:00

).'/ // 40222 52&17

52"// 08)02;

44551 16123 99376 70378 46//.
WMO Station=44551
WMO station=WMO_44551
UTC observation time=2013-02-16 12:00
QLLLL token=Present
Longitude=-37.8
Latitude=37.6
```

```
/// 40185 52022;
Sea level pressure=1018 hPa
Pressure tendency=Increasing steadily. Raises
Pressure change=2.2 hPa
```

```
62680 16123
9958"
WMO Station=62680
WMO station=Atbara
UTC observation time=2013-02-16 12:00
```

```
70265 66/? BMPROIO TUPEQ
```

```
222!/ 00078;
```

```
64527 16123 99593 70483 46///
WMO Station=64527
WMO station=WMO_64527
UTC observation time=2013-02-16 12:00
QLLLL token=Present
Longitude=-48.3
Latitude=59.3
/// 40095 52004
```

#### 6.21.4 Data files

Several data files containing public information are used to decrypt SYNOP data. They come from various places on the Internet, with redundancy.

NOAA	WMO identifier, name and coordinates of thousands of weather stations in the world.
<a href="http://weather.noaa.gov/data/nsd_bbsss.txt">http://weather.noaa.gov/data/nsd_bbsss.txt</a>	
NOAA	Weather, drifting, moored and wave riders buoys. Oil platforms. Fixed and mobile offshore drilling unit.
<a href="ftp://tgftp.nws.noaa.gov/data/observations/marine/stations/station_">ftp://tgftp.nws.noaa.gov/data/observations/marine/stations/station_</a>	
UK's National Weather Service	Name, country and description of weather ships
<a href="http://www.metoffice.gov.uk/media/csv/e/7/ToR-Stats-SHIP.csv">http://www.metoffice.gov.uk/media/csv/e/7/ToR-Stats-SHIP.csv</a>	
Joint WMO IOC Technical Commission for Oceanography and Marine Meteorology	Argos and Iridium buoys, with WMO identifiers.
<a href="ftp://ftp.jcommops.org/JCOMMOPS/GTS/wmo/wmo_list.txt">ftp://ftp.jcommops.org/JCOMMOPS/GTS/wmo/wmo_list.txt</a>	

There is a specific window for downloading the latest version of these data files. It can be reached with the command "File/Folders":

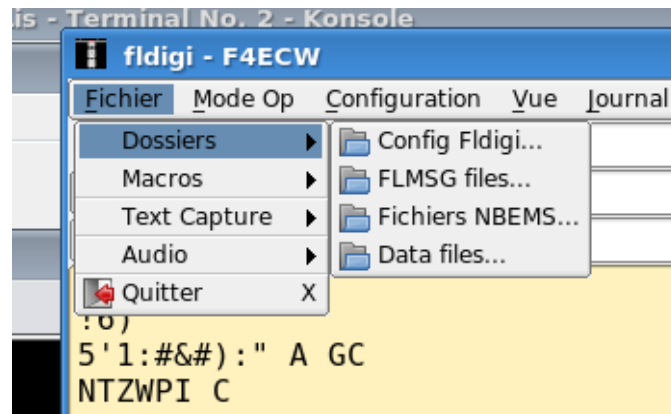


Figure 6.91: How to reach the data files sources menu

These data files can also be found:

- At installation time your fldigi binary comes with a package. All data files are silently installed on your machine.
- When building your own binary from the sources, with the command **make install**. Similarly, all files are installed.
- At first use and automatic installation, this opens the Data files sources menus, where you can download all or some data files used by fldigi.

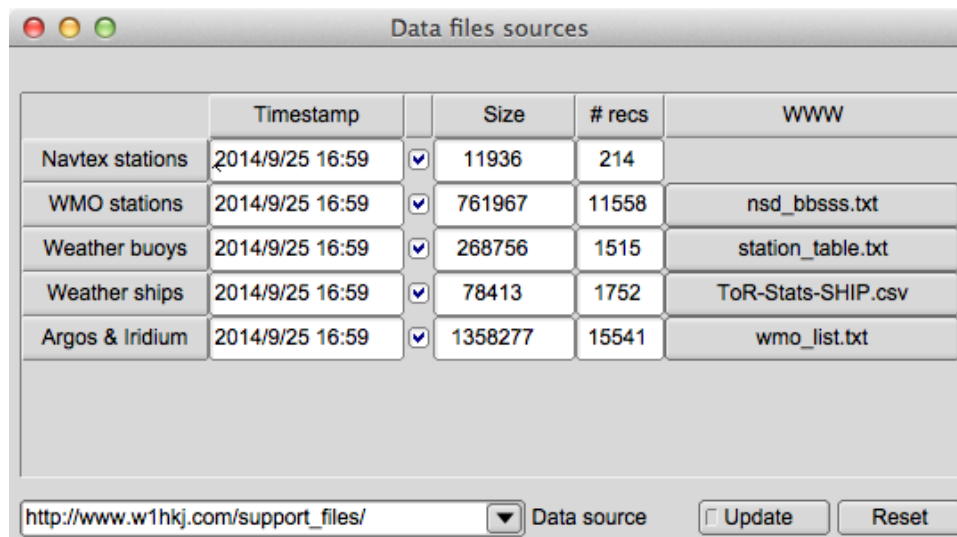


Figure 6.92: Downloading and updating data files used for various decodings

#### 6.21.4.1 Outputs

Each message is a set of key-value pairs, associated to:

Description	Measurement
Wind direction	315 degrees
Wind speed	8 m/s (Anemometer)
Temperature	0.0 deg C
Dew point temperature	-3.8 deg C
Station pressure	1006 hPa
Sea level pressure	1030 hPa

#### 6.21.4.2 Output formats

##### 6.21.4.2.1 ADIF

When the option is ticked, SYNOP weather reports are saved in a new ADIF record, in the default logfile.

The report itself is saved with field `<NOTES:xx>`. The length can be of several hundredth of chars, and accordingly to the ADIF standard, new-lines can appear in each report.

Example of `<NOTES>` field in an ADIF record:

```
Header
ICAO indicator=LOWM
Identification and location
Land station observation
Land observations
Precipitations=Precipitation omitted, no precipitation
Station type=Manned station. 7WW omitted, not significant
Cloud base=600 to 1000 m
Visibility=15 km
Cloud cover=7/8
```

##### 6.21.4.2.2 KML

SYNOP reports can generate [KML](#) files for fixed stations, mobile weather ships, buoys etc... that is, every observation which can be associated to a set of coordinates. Several reports of the same station can be aggregated in a single place mark. Mobile stations have their path drawn, linking all coordinates spotted for a given station with an unique name.

KML data are also saved in a display-independent format in `<description>` tags, and thus can easily be reused by other software.

#### 6.21.5 Command-line SYNOP decoder program : `synop_tool`

The decoder comes with `synop_tool`, a command-line tool able to decode input text files and generate the same output files as `fldigi`. It is intended as a development for checking internal consistency, and accuracy of SYNOP decoding without the complexity of the graphical interface.

It is accessible in the directory `fldigi/src/synop-src` and must be built with the command `make`.

Its command-line options are displayed with the option `-help`:

```
[fldigi/src/synop-src]$ ./synop_tool --help
```

Valid options are:

```

data_dir # Where the data files are stored. For example /usr/local/share/flidigi/
kml_dir  # Out put directory of created KML files.
load_dir # Input directory of KML files loaded at startup.
dbg      # Verbose mode.
usage    # Print this message.
test     # Decoding preceded by an internal test.
matrix   # KML output in aggregated matrices (See KML documentation).
regex    # text output of regular expressions, not decoded reports.
version  # Prints version number.
help     # Prints this messages

```

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.22 UTF-8 support

Fidigi supports the UTF-8 character set in all of it's labels and widgets (controls). The Rx and Tx text panels accept both keyboard and text stream that is in UTF-8 format. Transmission of UTF-8 characters is supported by these modes:

- PSK, PSKR, and Multi-Channel PSK/PSKR
- Olivia - must enable 8 bit extended characters
- MT63 - must enable 8 bit extended characters
- MFSK - all baudrates supported

For example, the following Russian and annotated texts can be sent and received:

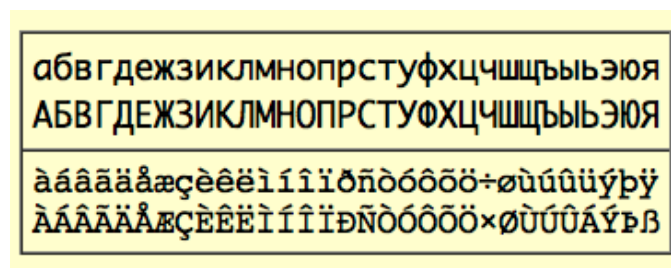


Figure 6.93: UTF-8 Encoded Characters

UTF-8 characters are represented by 2 bytes which means that a single character will take longer to transmit. Transmission speed will be further reduced on modes like PSK where the character to bit stream conversion has been optimized for the English language.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.23 Using the FLdigi Rt. Channel for transmitter PTT

This interface is designed to take advantage of the Right Speaker tone output in FLdigi for actuating the PTT. It is especially useful for interfacing a small notepad with minimal I/O capability to a radio. It has been my experience that complete ground isolation between a laptop and the radio (including eliminating any kind of rig control) provides the minimum QRM to the receiver from the laptop. The Headphone and Mic connections are the only computer connections necessary.

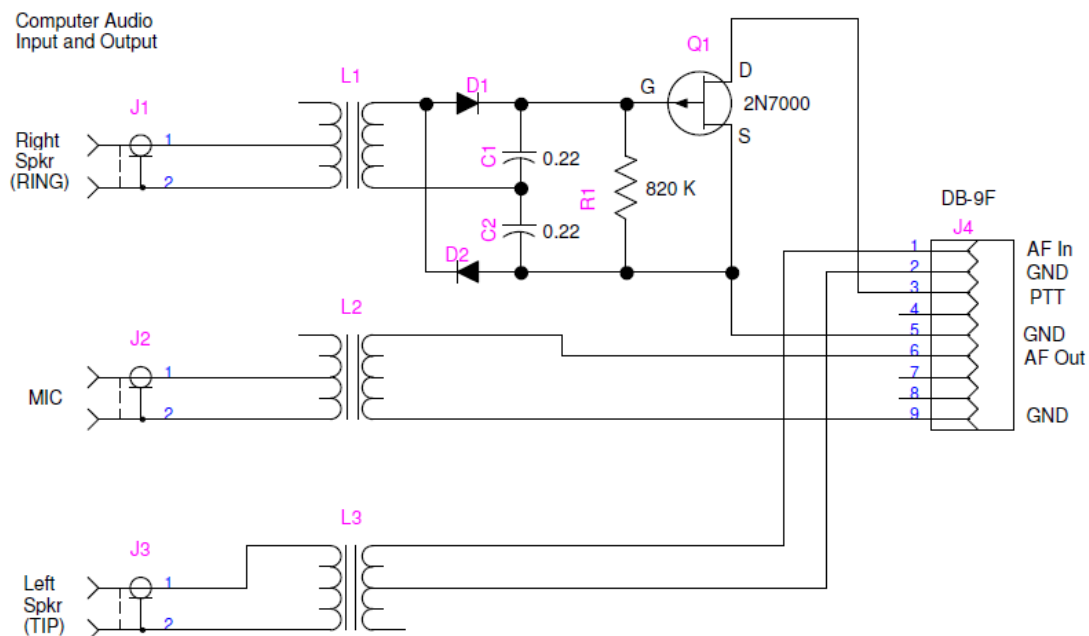
The tone output is coupled through a transformer to maintain complete ground isolation between the computer and the radio. There is no external power required for this interface and an FET is used to provide the PTT output. I use transformers in both the audio input and output paths to complete the isolation.

Note that two of the transformers are wired to the single stereo jack from the Speaker output of the computer. The normal Left Speaker Output (TIP) is used to provide the modulating audio for the radio transmit input while the Right Speaker Output (RING) provides the tone output that drives the PTT circuit. The SHELL of the connector provides the return path for both signals.

Audio from the radio is supplied to the MIC input on the computer for demodulation by the FLdigi program.

Somewhat higher output from the computer is required to reliably operate the PTT circuit and the audio balance control in the computer is used to reduce the output to the radio to the level required for normal modulation. Since the right speaker tone output is not the same audio used to modulate the radio, no signal distortion is caused by the clipping action of the diode rectifiers used in the PTT circuit.

To configure FLdigi to use this interface go to (Configure/Rig Control/Hardware PTT) and check only the (PTT tone on right audio channel) box.



L1, L2, L3 600:600 Ohm CT Audio Transformers  
 D1, D2 1N4148 Small Signal Diode  
 J1 and J3 are a single Stereo Tip-Ring-Sleeve socket  
 J2 can be Stereo or Mono socket

Set FLdigi to send a 1 kHz tone on the  
 Right Speaker output channel for PTT

W5ZIT  
 Jun 2011

Figure 6.94: W5ZIT Interface

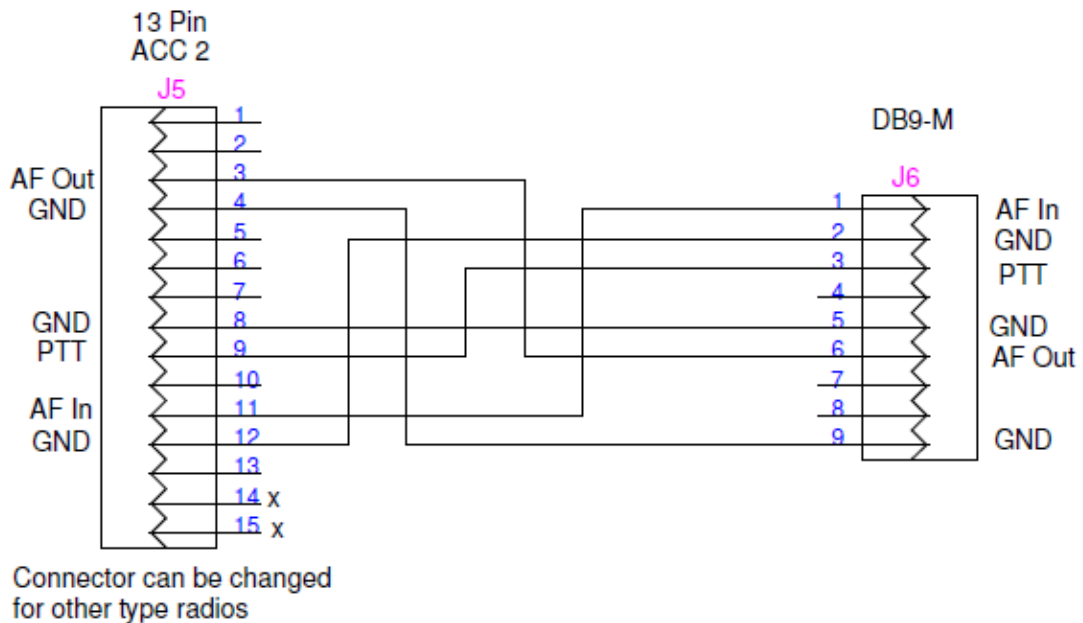


Figure 6.95: TS-2000 ACC-2 to DB9 Adapter

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.24 Working Logs

### 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

Everytime you start or stop fldigi that event is recorded in a daily log file. The daily log is named as:

fldigYYYYMMDD.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 time stamped. 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): o ur 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"

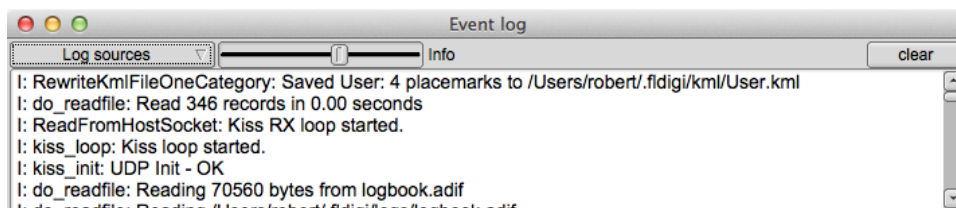


Figure 6.96: Event Log

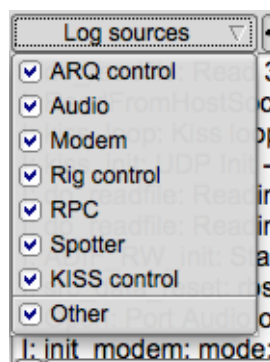


Figure 6.97: Event Log Menu

There are six levels of event logging with increasing depth of reports:

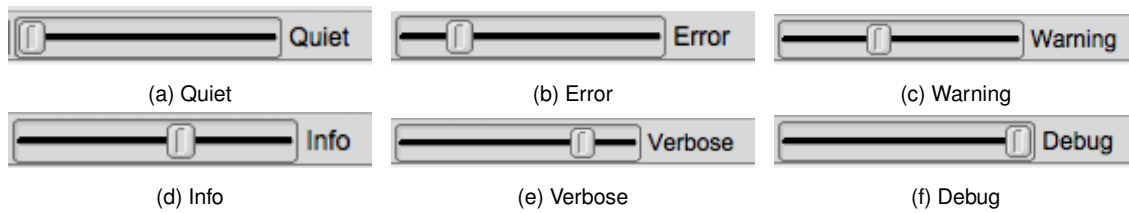


Figure 6.98: Event Logging Levels

The default level for logging events is "warning."

The event log show 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 psk reporter 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.

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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 6.25 WWV transmit mode

The WWV mode is normally used to measure the offset of the sound card oscillator. (see [WWV ppm measurement](#)).

It can also be used to transmit time tick signals similar to the format that is used by WWV. The WWV modem sends a time tick which is a 200 msec wide pulse at a once per second rate when the T/R button is enabled. This pulse is shaped with a 4 msec raised cosine shape on the leading and trailing edges to reduce key clicks. The accuracy of the transmitted time tick is solely dependent on the accuracy with which the WWV ppm measurement was performed.

The purpose of the WWV time tick transmission is to allow other stations to calibrate their sound cards against your calibrated system. This can be used to align all systems in a VHF/UHF net for example. It is only necessary for one of the net members to be able to calibrate his or her sound card against WWV. The other's would then be calibrated by proxy use of the WWV time tick transmit mode. This can even be used in the case where no member has access to a HF transceiver. The "master" station would set the Rx and Tx ppm settings to zero. It would then transmit the time tick signal for the other stations to calibrate their sound cards against the master sound card. Having all of the stations calibrated in this way will insure that the modem decoders will give maximum performance. Here is an example of an [advanced macro](#) that will send a CW announcement, 2 minutes of time ticks and end with another CW announcement.

```
<MODEM:CW>
<!GOFREQ:1000>
<!WPM:24>
QRZ QRZ de <MYCALL> <MYCALL>
2 minute time tick cal run follows
<IDLE:2>
<!MODEM:WWV><!IDLE:120>
<!MODEM:CW><!IDLE:2>
end of time tick run
de <MYCALL> k
<TX><RX>
```

[Return to Top of Page](#)  
[Return to Main Page](#)



# Chapter 7

## Developers

- [Build Information](#)
- [Google Maps](#)
- [Installing from Source](#)
- [FLDIGI KISS Hardware Commands](#)
- [Parse UALR](#)
- [Pseudo FSK](#)
- [Rig Xml How to](#)
- [ualr telnet](#)
- [Xmlrpc Control](#)

### 7.1 Build Information

#### 7.1.1 Build Info

The following is obtained by executing "fldigi --version"

```
Build information:  
built :
```

```
Fri Dec 25 15:57:13 CST 2009 by dave@linux-dev on i686-pc-linux-gnu for  
i686-pc-linux-gnu
```

```
configure flags: '--without-asciidoc' '--enable-optimizations=native'
```

```
compiler : gcc version 4.3.3 (Ubuntu 4.3.3-5ubuntu4)
```

```
compiler flags : -I$(srcdir) -I$(srcdir)/include  
-I$(srcdir)/irrxml -I$(srcdir)/fileselector -pthread  
-I/usr/local/include -I/usr/local/include -I/usr/include/freetype2  
-D_THREAD_SAFE -D_REENTRANT -D_REENTRANT -I/usr/local/include  
-I/usr/include/libpng12 -I/usr/local/include -pipe -Wall -fexceptions  
-O2 -ffast-math -finline-functions -fomit-frame-pointer -march=native  
-mfpmath=sse -DNDEBUG
```

```
linker flags : -L/usr/local/lib -lportaudio -lm  
-lpthread -L/usr/local/lib -lfltk_images -lpng -lz -ljpeg -lfltk -lXft  
-lpthread -ldl -lm -lXext -lX11 -lX11 -lsndfile -lsamplerate  
-lpulse-simple -lpulse -L/usr/local/lib -lhamlib -lm -lpng12  
-L/usr/local/lib -lxmlrpc_server_abyss++ -lxmlrpc_server++  
-lxmlrpc_server_abyss -lxmlrpc_server -lxmlrpc_abyss -lpthread
```

```
-lxmlrpc++ -lxmlrpc -lxmlrpc_util -lxmlrpc_xmlparse -lxmlrpc_xmltok
-ldl -lrt -lpthread

libraries : FLTK 1.3.2
libsamplerate 0.1.4
libsndfile 1.0.17
PortAudio 19
PulseAudio 0.9.14
Hamlib 1.2.10
XMLRPC-C 1.06.31

Runtime information:
system : Linux
linux-dev 2.6.28-17-generic #58-Ubuntu SMP Tue Dec 1 18:57:07 UTC 2009
i686

libraries : libsamplerate-0.1.4 (c) 2002-2008 Erik de Castro Lopo
libsndfile-1.0.17

PortAudio V19-devel (built May 25 2009 06:36:24) 1899
Pulseaudio 0.9.14

Hamlib version 1.2.10
```

[Return to Top of Page](#)

[Return to Main Page](#)

## 7.2 Google Maps

```
snip ----- copy the following to ~/.fldigi/scripts/map.pl

#!/usr/bin/perl

# Author: Stelios Bounanos, MOGLD
# Date: 20080625

use warnings;
use strict;
use Getopt::Std;

our $VERSION = "0.3141";
our %opts = ( "e" => 0, "m" => 1, "z" => 4);

cmdline();
open(STDOUT, '>', "/dev/null");

my $loc = exists($opts{'l'}) ? $opts{'l'} : $ENV{'FLDIGI_LOG_LOCATOR'};
die "Invalid locator\n" unless ((defined($loc) && length($loc) =~ /[2-6]/));

my $label = exists($opts{'t'}) ? $opts{'t'} : $ENV{'FLDIGI_LOG_CALL'};
$label = $loc if (!defined($label) || $label eq "");

my ($lon, $lat) = map { sprintf("%+.6f", $_) } mtoll($loc);
if ($opts{'m'}) {
    my $url = "http://maps.google.com/maps?q=${lat},${lon}(${label})&t=p&z=${opts{'z'}}";
    # $url =~ s/([()])/sprintf("%02X", ord($1))/ge; # encode some chars
    exec("xdg-open", $url);
    die "Could not exec xdg-open: $!\n";
}

exit(0) unless ($opts{'e'});
my $kml = (exists($ENV{'TMPDIR'}) ? $ENV{'TMPDIR'} : "/tmp") .
"/" . $loc . ".kml";
open(KML, '>', $kml) or die "Could not write $kml: $!\n";
print KML <<EOF
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
  <Placemark>
```

```

    <name>$label</name>
    <description>
        $label
        $loc
    </description>
    <Point>
        <coordinates>$lon,$lat,0</coordinates>
    </Point>
</Placemark>
</kml>
EOF
;
close(KML);

#####

sub cmdline
{
    $Getopt::Std::STANDARD_HELP_VERSION = 1;
    my $old_warn_handler = $SIG{__WARN__};
    $SIG{__WARN__} = sub { die $_[0]; };
    getopts('t:l:mz:e', \%opts);
    $SIG{__WARN__} = $old_warn_handler;
}

# Convert a 2, 4, or 6-character Maidenhead locator string
# to decimal degrees. Return a (longitude, latitude) pair.
sub mtoll
{
    my $len = length($_[0]);
    $_[0] .= join("", ("A", "A", "0", "0", "A", "A")[$len .. 5]) if ($len < 6);
    $_[0] = uc($_[0]);
    die "Invalid locator\n" unless ($_[0] =~ /[A-R]{2}\d{2}[A-X]{2}/);

    my @digits = split(//, $_[0]);
    my ($lon, $lat) = (-180, -90);

    $lon += (ord($digits[0]) - ord('A')) * 20 +
            (ord($digits[2]) - ord('0')) * 2 +
            (ord($digits[4]) - ord('A') + 0.5) / 12;
    $lat += (ord($digits[1]) - ord('A')) * 10 +
            (ord($digits[3]) - ord('0')) +
            (ord($digits[5]) - ord('A') + 0.5) / 24;

    return ($lon, $lat);
}

sub HELP_MESSAGE
{
    print <<EOF

Usage: $0 [-OPTIONS [-MORE_OPTIONS]] [--] [PROGRAM_ARG1 ...]

The following single-character options are accepted:

    -t LABEL  Use LABEL as the marker label
               The default is \${FLDIGI_LOG_CALL}

    -l LOC    Place marker at IARU locator LOC
               The default is \${FLDIGI_LOG_LOCATOR}

    -m        Show in Google Maps (default)
    -z        Zoom level (Google Maps only)

    -e        Write a Google Earth kml file in
               \${TMPDIR}/LOC.kml
EOF
;
}

snip-----

```

[Return to Top of Page](#)  
[Return to Main Page](#)

## 7.3 Installing from Source

The developers recommend that you build either fldigi from source or install from the repository associated with your distribution. The repository may not be immediately available for the most current version number. In that case you can try installing the binary distribution. Keep in mind that the version numbers of the dependent shared libraries must match those on the machine used to create the binary.

The source code for fldigi is very large and has a number of dependencies that must be satisfied before a successful compile. If you are not familiar with compiling and linking source code you should probably practice on a simpler package before treading these waters. Please refer to the following web site for information on building for Linux, Windows and OS X.

[Fldigi WIKI - build instructions](#)

[Return to Top of Page](#)  
[Return to Main Page](#)

## 7.4 FLDIGI KISS Hardware Commands

### Custom Frame Extensions

These commands are NON-STANDARD extension(s) to the KISS interface specifications and designed to control/configure aspects of FLDIGI not normally found in Hardware TNC's. Undefined kiss frame types 6 and 7 are used to identify these custom frames.

Definition	Kiss frame ID
HARDWARE	6
RAW	7

External program which conform to the official KISS I/O specifications are expected to operate normally.

UDP/IP Connectionless communications are used to transfer information between FLDIGI and the host program. See [Configure ARQ/KISS I/O](#) and [Command Line Switches](#) for specific address and port number assignments.

### 7.4.1 Set Commands

Command <parameters(s)>	Use
<b>BCHN:</b> <ON OFF>	Busy Channel On/Off



<b>BCHNS:</b> <0-999>	Busy Channel Wait Duration (seconds)
<b>CSMA:</b> <ON OFF>	Enable/Disable CSMA. Used on a shared frequency
<b>IBCHN:</b> <N>	Set Inhibit busy channel duration to N seconds
<b>IBCHN:</b> 0	Resets temporary duration to default setting.
<b>IBCHN:</b> S	Inhibit busy channel temporarily. 'S' (character) (default 5 seconds)
<b>KISSRAW:</b> <ON OFF ONLY>	Pass non HDLC encoded data to the transmitter.

ON = HDLC and Unaltered data.

OFF = HDLC data only.

ONLY = Unaltered data only.

Command	Use
<b>KISSRCRCM:</b> <NONE SMACK CCITT XOR FCS>	Enable KISS frame checksum type
<b>KPSATT:</b> <value>	Set the fractional ratio gain value (1/value)
<b>KPSQL:</b> <ON OFF>	OFF=Histogram Mode, ON=User set level
<b>KPSQLS:</b> <0-100>	Set squelch level (percent)
<b>MODEM:</b> <modem_id_string>	Set modem type. Example MODEM:PSK63RC32
<b>RSIDBCAST:</b> <ON OFF>	Enable/disable RSID broadcast state change (default: OFF)
<b>RSIDRX:</b> <ON OFF>	Enable/disable RX RSID
<b>RSIDTX:</b> <ON OFF>	Enable/disable TX RSID
<b>SQL:</b> <ON OFF>	SQL On/Off
<b>SQLS:</b> <0-100>	Set SQL Level (percent)
<b>TRXSBCAST:</b> <ON OFF>	Enable/disable TX/RX state change broadcast (default: OFF)
<b>TXBEBCAST:</b> <ON OFF>	Enable/disable TX buffer empty broadcast (default: OFF)
<b>WFF:</b> <integer value>	Move TXRX cursor to frequency on waterfall.

NONE = No CRC used

SMACK = Enable CRC use for Host <-> TNC ASYNC links

CCITT = Enable CCITT CRC

FCS = Enable FCS CRC

XOR = Enable BPQ's XOR CRC

Command <parameters(s)>	Use
<b>RSIDM:</b> <BANDPASS MODEM>,<ACTIVE NOTIFY>	Set the RSID mode to,...

BANDPASS = Search entire bandpass for RSID signals

MODEM = Search Modem bandpass (+/- 200HZ) for RSID signals

ACTIVE = When RSID received, move frequency transmit/receive cursor to location.

NOTIFY = Report RSID when detected (Notify only).

## 7.4.2 Query Commands

Command / Return Status	Use
<b>BCHN:</b> FLDIGI returns BCHN:<ON OFF>	Busy Channel State On/Off Query
<b>BCHNS:</b> FLDIGI returns BCHNS:<0-999>	Busy Channel Wait Duration Query (seconds)
<b>BUSY:</b> FLDIGI returns BUSY:<Y N>	Return the state of the squelch, if there is a signal present
<b>CSMA:</b> FLDIGI returns CSMA:<ON OFF>	CSMA State ON/OFF
<b>FLSTAT:</b> FLDIGI returns FLSTAT:<INIT OK>,<HH:MM:SS>	Start up State and runtime. Returns OK and after initial Query

<b>IBCHN:</b> FLDIGI returns IBCHN:<SECONDS>	Inhibit busy channel durations in seconds.
<b>KISSRCM:</b> FLDIGI returns CRCMODE:<NONE SMACK>,<CCITT XOR FCS>	Host <-> TNC KISS CRC Types
<b>KPSATT:</b> FLDIGI returns KPSATT:<value>	Fractional value of KPSQL ratio gain figure (1/value)
<b>KPSQL:</b> FLDIGI returns KPSQL:<ON OFF>	OFF=Histogram Mode, ON=User set level
<b>KPSQLP:</b> FLDIGI returns KPSQLP <0-100>	Current Power Level
<b>KPSQLS:</b> FLDIGI returns KPSQLS:<0-100>	Squelch set level Query (percent)
<b>MODEM:</b> FLDIGI returns MODEM:<Modem ID String>	Current Modem
<b>MODEMBW:</b> FLDIGI returns MODEMBW:<Bandwidth in Hz>	Current Modem Bandwidth
<b>MODEML:</b> FLDIGI returns MODEML:Modem1,Modem2,...	A List of comma delimited modem ID strings.
<b>RSIDM:</b> FLDIGI returns RSIDM:<BANDPASS MO↔ DEM>,<ACTIVE NOTIFY>	Return current RSID Mode
<b>RSIDRX:</b> FLDIGI returns RSIDRX:<ON OFF>	RX RSID ON?
<b>RSIDTX:</b> FLDIGI returns RSIDTX:<ON OFF>	TX RSID ON?
<b>SQL:</b> FLDIGI returns SQL:<ON OFF>	SQL On/Off
<b>SQLP:</b> FLDIGI returns SQLP:<0-100>	Current Symbol Quality Level (Currently Not returning valid data)
<b>SQLS:</b> FLDIGI returns SQLS:<0-100>	Set SQL Level Query (percent)
<b>TNC:</b> FLDIGI returns TNC:FLDIGI <Version Number>	Returns the version number of FLDIGI
<b>TRXS:</b> FLDIGI returns TRXS:<RX TX>	Returns the current RX/TX state of FLDIGI
<b>TXBUF:</b> FLDIGI returns TXBUF:<count>	Return the number of byte in the transmit queue
<b>WFBW:</b> FLDIGI returns WFBW:<LOWER HZ>,<UPPER HZ>	Return the active waterfall bandwidth
<b>WFF:</b> FLDIGI returns WFF:<integer value>	Current modem position on waterfall (center)

#### 7.4.2.1 Query Commands (FLDIGI to HOST)

Command	Use
<b>HOST:</b>	Host returns HOST:<program_name> <version>

#### Note:

Not required

#### 7.4.3 Broadcast Status Messages (FLDIGI to HOST)

Status <parameters(s)>	Use
<b>RSIDN:</b> NEW_WF_OFFSET,NEW_MODEM,OLD_↔ WF_OFFSET,OLD_MODEM,<ACTIVE NOTIFY U↔ SER>	RSID NOTICE
<b>TRXS:</b> <RX TX>	Transmitted during a state change between RX/TX or TX/RX.
<b>TXBE:</b>	Broadcast on emptied transmit buffer.

```

NEW_WF_OFFSET = 0-4000
NEW_MODEM     = Modem ID string of newly switched modem.
OLD_WF_OFFSET = 0-4000
OLD_MODEM     = Modem ID string of modem prior to switching.
ACTIVE        = Indicating TXRX cursor changed.
NOTIFY        = Notice only.
USER          = Report User or commanded via KISS interface modem change.

```

### 7.4.4 Format of Hardware Commands

FLDIGI will only except KISS frames with a PORT ID of '0' (zero). The hardware frame is a kiss frame type 6.

```
MODEM:PSK125RC16
FEND,PORT(0)|HRDW(6),'M','O','D','E','M',':','P','S','K','1','2','5','R','C','1','6',FEND
KISS HARDWARE FRAME="MODEM:PSK63RC32"
KISS HARDWARE FRAME="RSIDTX:ON"
KISS HARDWARE FRAME="RSIDRX:OFF"
```

### 7.4.5 Format of RAW Data

To enable RAW use, issue hardware command **KISSRAW**:<ON|OFF|ONLY>. The encode of the raw data is the same as the encoding of normal kiss data. The host program is responsible for the proper protocol data syntax.

#### Format:

Dn=8 bit data (byte)

```
KISS RAW FRAME=D1,D2,D3,...
FEND,PORT(0)|RAW(7),D1,D2,D3,...,FEND
```

#### Notes:

1. The transmit buffer has a ten minute timer association. In the event data has been retained in the transmit buffer for more then 10 minutes without being transmitted in part or whole the buffer is cleared. This will prevent a build up of data and a subsequent transmit for an extended period once the frequency is clear. Consider all packets in the transmit queue lost. If enabled, TXBE will be issued when this occurs.
2. Both ARQ and KISS IO ports are active for the reception of data. However, only the selected port will pass data to the higher functions and the other port will not buffer the data for later use. Reason: see note 1.
3. By keeping a complete ARQ (RAW) packet in one KISS frame both HDLC and ARQ (RAW) data can coexist. This allows for sequential data transmission without interleaving data at the byte level.

## 7.5 Parse UALR

A simple parser to create a formatted console output for fldigi's <EXEC> macro:

```
snip-----
#include <ctime>
#include <cstdio>
#include <cstdlib>
#include <unistd.h>
#include <string>
#include <iostream>
#include <fstream>

using namespace std;
using std::cout;
using std::cin;

int main(int argc, char *argv[])
{
    size_t pos = 0, pos2 = 0, pos3 = 0, pos4 = 0, pos5 = 0;
    string commandline = "";
    string name = "";
```

```

string qth = "";
string answer = "";
char c = cin.get();

while (!cin.eof()) {
    commandline += c;
    c = cin.get();
}

if (commandline.find("No match found") != string::npos)
    goto noresponse;

pos = commandline.find(", ");

if (pos == string::npos)
    goto noresponse;

pos += 2;
pos2 = commandline.find("\n", pos);

if (pos2 == string::npos)
    goto noresponse;

name = commandline.substr(pos, pos2 - pos);
pos3 = name.find(32);

if (pos3 != string::npos)
    name = name.substr(0, pos3);

for (size_t i = 1; i < name.length(); i++)
    name[i] = tolower(name[i]);

answer = "$NAME";
answer.append(name);

pos4 = commandline.find(", ", pos2);
pos4 = commandline.rfind( "\n", pos4);
pos4 += 1;
pos5 = commandline.find("\n", pos4);

qth = commandline.substr(pos4, pos5 - pos4);

answer.append("$QTH");
answer.append(qth);

cout <<< answer.c_str();

return 0;

noresponse:;

    cout <<< "$NAME?$QTH?";

    return 0;
}
snip-----

```

Save the above as "parseUALR.cxx" and then compile and link as follows:

```
g++ parseUALR.cxx -o parseUALR
```

Copy the "parseUALR" executable to a directory on your shell exec PATH.

[Return to Top of Page](#)

[Return to Main Page](#)

## 7.6 Pseudo FSK

Using the FLdigi Pseudo FSK (Rt. Channel) function to key a transmitter

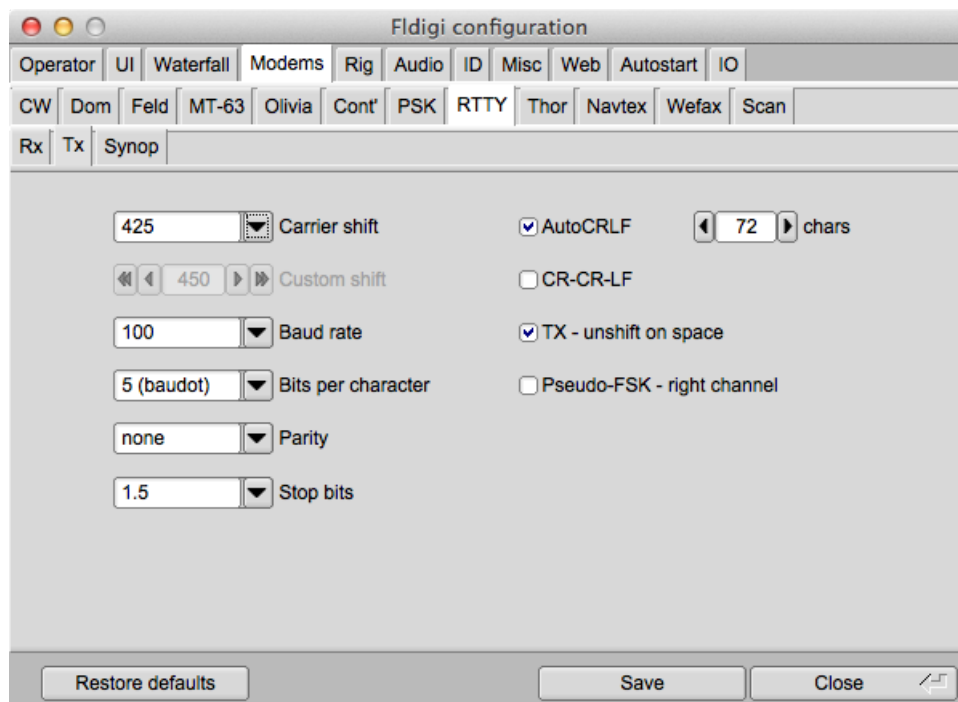


Figure 7.1: RTTY TX Configure

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.

## FULL WAVE DIODE VOLTAGE DOUBLER

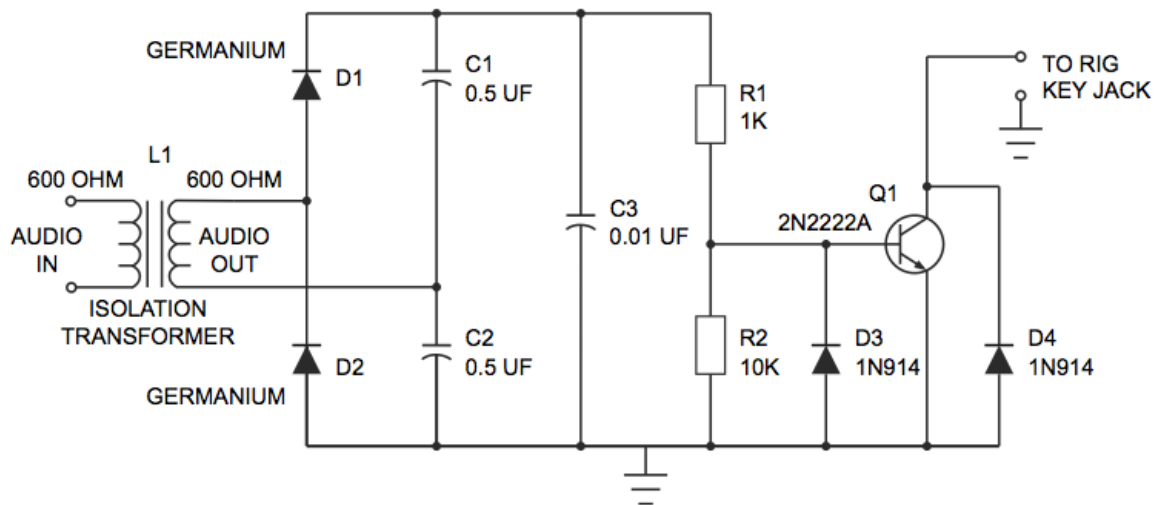


Figure 7.2: CW Keying Circuit

### 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.

[Return to Top of Page](#)

[Return to Main Page](#)

## 7.7 Rig Xml How to

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: [xml archives](#). 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 feedback [at] w1hkj [dot] 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:

<!--

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

```
<RIGDEF>
...
</RIGDEF>
```

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

```
<RIG>Icom 746 PRO</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:

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

The text within the tag pair

```
<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:

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

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

```
<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. These values are loaded from the xml file. If a value is changed on the configuration tab it is saved in the progdefaults.xml file if the configuration is saved. On a subsequent start of fldigi the saved parameters will override the ones in the rig definition file.

| xml tag                       | parameter          | description  |
|-------------------------------|--------------------|--|
| <TIMEOUT>TT</TIMEOUT>         | TT in milliseconds | serial port timeout                                    |
| <RETRIES>NN</RETRIES>         | NN integer         | number of times CAT string is resent                   |
| <WRITE_DELAY>TT</WRITE_DELAY> | TT in milliseconds | wait time after sending normal command to xcvr         |
| <INIT_DELAY>IT</INIT_DELAY>   | IT in milliseconds | wait time after sending init string to xcvr            |
| <BAUDRATE>BAUD</BAUDRATE>     | BAUD integer       | 1200, 2400, 4800, 9600, 19200, 38400 ...               |
| <STOPBITS>B</STOPBITS>        | B integer          | 1 or 2   |
| <RTSCTS>BOOL</RTSCTS>         | BOOL true, false   | h/w handshake used for data flow control               |
| <RTSPLUS>BOOL</RTSPLUS>       | BOOL true, false   | set RTS signal line to +12 V (default -12 V)           |
| <RTSPPT>BOOL</RTSPPT>         | BOOL true, false   | toggle RTS signal line for PTT                         |
| <DTRPLUS>BOOL</DTRPLUS>       | BOOL true, false   | set DTR signal line to + 12 V (default -12 V)          |
| <DTRPTT>BOOL</DTRPTT>         | BOOL true, false   | toggle DTR signal line for PTT                         |
| <CMDPTT>BOOL</CMDPTT>         | BOOL true, false   | use command string for PTT (not supported by all xcvs) |

|  |                      |   |
|--|----------------------|---|
| <ECHO>BOOL</ECHO>                      | BOOL true, false     | xcvr/interface echoes all chars (typical of CI-V interface) |
| <WAIT_FOR_DEVICE>nnn</WAIT_FOR_DEVICE> | nnnn in milliseconds | used for correct startup of K9MJ CI-V router                |
| <VSP>BOOL</VSP>                        | BOOL true/false      | serial port is a virtual device                             |

The transceiver modes are specified within the <MODES></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>FM</SYMBOL><BYTE>34</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>AM</SYMBOL><BYTE>35</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>FSK</SYMBOL><BYTE>36</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>CW-R</SYMBOL><BYTE>37</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>FSK-R</SYMBOL><BYTE>39</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></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></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>
  <ELEMENT><SYMBOL>1000</SYMBOL><BYTE>14</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>1100</SYMBOL><BYTE>15</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>1200</SYMBOL><BYTE>16</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>1300</SYMBOL><BYTE>17</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>1400</SYMBOL><BYTE>18</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>1500</SYMBOL><BYTE>19</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>1600</SYMBOL><BYTE>20</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>1700</SYMBOL><BYTE>21</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>1800</SYMBOL><BYTE>22</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>1900</SYMBOL><BYTE>23</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2000</SYMBOL><BYTE>24</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2100</SYMBOL><BYTE>25</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2200</SYMBOL><BYTE>26</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2300</SYMBOL><BYTE>27</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2400</SYMBOL><BYTE>28</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2500</SYMBOL><BYTE>29</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2600</SYMBOL><BYTE>30</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2700</SYMBOL><BYTE>31</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2800</SYMBOL><BYTE>32</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>2900</SYMBOL><BYTE>33</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>3000</SYMBOL><BYTE>34</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>3100</SYMBOL><BYTE>35</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>3200</SYMBOL><BYTE>36</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>3300</SYMBOL><BYTE>37</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>3400</SYMBOL><BYTE>38</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>3500</SYMBOL><BYTE>39</BYTE></ELEMENT>
  <ELEMENT><SYMBOL>3600</SYMBOL><BYTE>40</BYTE></ELEMENT>
</BANDWIDTHS>

```

If the bandwidth data stream is unique for send and receive data streams then they are specified separately with a `<BW-CMD></BW-CMD>` tag pair for data sent to the transceiver, and a `<BW-REPLY></BW-REPLY>` tag pair for data returned to the computer. The number and symbol name for these must match.

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>

```

## Commands and Replies

CAT command strings are defined within a `<COMMAND>...</COMMAND>` block. Each block structure appears as:

```
<COMMAND>
  <SYMBOL>...</SYMBOL>  noun name of symbol
  <SIZE>...</SIZE>      number of bytes in CAT string
  <STRING>...</STRING>  string literal (optional)
  <DATA>
    <DTYPE>...</DTYPE>  type of data in data block
    <SIZE>...</SIZE>    number of bytes in data block
    <MAX>...</MAX>      maximum value of data value
    <MIN>...</MIN>      minimum value of data value
    <RESOL>...</RESOL>  resolution of data value
  </DATA>
  <STRING>...</STRING>  string literal (optional)
</COMMAND>
```

Here are three examples from the KX3.xml definition file

```
<COMMAND>
  <SYMBOL>INIT</SYMBOL>
  <SIZE>12</SIZE>
  <STRING>AI0;DT0;K31;</STRING>
</COMMAND>
```

```
<COMMAND>
  <SYMBOL>SETFREQ</SYMBOL>
  <SIZE>14</SIZE>
  <STRING>FA</STRING>
  <DATA>
    <DTYPE>DECIMAL</DTYPE>
    <SIZE>11</SIZE>
    <MAX>99999999999</MAX>
    <MIN>00000490000</MIN>
    <RESOL>1</RESOL>
  </DATA>
  <STRING>;</STRING>
</COMMAND>
```

```
<COMMAND>
  <SYMBOL>GETFREQ</SYMBOL>
  <SIZE>3</SIZE>
  <STRING>FA;</STRING>
  <INFO>FREQ</INFO>
</COMMAND>
```

A command string may have a corresponding reply string sent by the transceiver. The SYMBOL noun name for each command - reply pair is the same.

Fldigi can parse and decode message returned from the transceiver that define 4 aspects of the transceiver operation:

|      |  |
|------|--|
| OK   | data accepted by the transceiver             |
| BAD  | data rejected by the transceiver             |
| MODE | current operating mode of the transceiver    |
| BW   | current bandwidth setting of the transceiver |

|      |  |
|------|--|
| FREQ | frequency of the active VFO (might be either A or B for example) |
|------|--|

These are all contained within multiple <REPLY></REPLY> tag pairs. The REPLY block structure is:

```
<REPLY>
  <SYMBOL>...</SYMBOL>      noun name of symbol
  <SIZE>...</SIZE>          number of bytes in reply string
  <BYTES>...</BYTES>       space separated hexadecimal values
  <BYTE>...</BYTE>         single hexadecimal value
  <STRING>...</STRING>     returned character string
</REPLY>
```

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:

```
<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:

```
<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 or  
DECIMAL**

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

```
<REPLY>
  <SYMBOL>MODE</SYMBOL>
  <SIZE>8</SIZE>
  <BYTES>FE FE E0 66</BYTES>
  <BYTE>04</BYTE>
  <DATA>
    <DTYPE>BINARY</DTYPE>
    <SIZE>1</SIZE>
  </DATA>
  <FILL>1</FILL>
  <BYTE>FD</BYTE>
</REPLY>
```

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

This is an example for the reply to a frequency query that is returned by the Elecraft KX3. The corresponding query command structure is shown above.

```
<REPLY>
  <SYMBOL>FREQ</SYMBOL>
  <SIZE>14</SIZE>
  <STRING>FA</STRING>
  <DATA>
    <DTYPE>DECIMAL</DTYPE>
    <SIZE>11</SIZE>
    <MAX>99999999999</MAX>
    <MIN>00001500000</MIN>
    <RESOL>1</RESOL>
  </DATA>
  <STRING>;</STRING>
</REPLY>
```

[Return to Top of Page](#)

[Return to Main Page](#)

## 7.8 ualr telnet

```
snip ----- copy the following to ~/.fldigi/scripts/ualr-telnet.pl

#!/usr/bin/perl

# Author: Stelios Bounanos, MOGLD
# Date: 20090103
#
# ualr-telnet is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 3 of the License, or
# (at your option) any later version.
#
# ualr-telnet1 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 General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.
# -----

use strict;
use warnings;

die "Usage: $0 CALLSIGN\n" unless (@ARGV == 1);

use Net::Telnet ();

sub error { print "\$NAME?\$QTH?\n"; exit(1); }
my $t = new Net::Telnet( Host => "callsign.ualr.edu", Port => 2000, Timeout => 10,
    errmode => \&error );

$t->open();
$t->waitfor('/LOOKUP>.*$/');
$t->print($ARGV[0]);

$_ = $t->getline();      # blank line
$_ = $t->getline();      # call
error() if (m/No match found/);

$_ = $t->getline();      # name
chomp; s/\.+,\s+//; s/\s.+$//;
print "\$NAME$_";
$_ = $t->getline();      # addr
$_ = $t->getline();      # qth
chomp;
```

```

$_ =~ ", ";
$_ = $`;

print "\$QTH$_\n";

$t->waitfor('/LOOKUP>.*$/');
$t->print("bye");

snip-----

```

[Return to Top of Page](#)

[Return to Main Page](#)

## 7.9 Xmlrpc Control

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 tar ball 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

### 7.9.1 XML Command Symbol Interpretation

| Symbol | Interpretation |
|--------|----------------|
| 6      | bytes          |
| A      | array          |
| b      | boolean        |
| d      | double         |
| i      | integer        |
| n      | nil            |
| s      | string         |
| S      | struct         |

### 7.9.2 Table of XML Commands

| Method Name           | Sig (ret:arg) | Description   |
|-----------------------|---------------|---|
| fldigi.config_dir     | s:n           | Returns the name of the configuration directory                                       |
| fldigi.list           | A:n           | Returns the list of methods   |
| fldigi.name           | s:n           | Returns the program name  |
| fldigi.name_version   | s:n           | Returns the program name and version  |
| fldigi.terminate      | n:i           | Terminates fldigi. "i" is bitmask specifying data to save: 0=options; 1=log; 2=macros |
| fldigi.version        | s:n           | Returns the program version as a string   |
| fldigi.version_struct | S:n           | Returns the program version as a struct   |
| io.enable_arq         | n:n           | Switch to ARQ I/O   |
| io.enable_kiss        | n:n           | Switch to KISS I/O  |
| io.in_use             | s:n           | Returns the IO port in use (ARQ/KISS).  |
| log.clear             | n:n           | Clears the contents of the log fields   |
| log.get_az            | s:n           | Returns the AZ field contents   |
| log.get_band          | s:n           | Returns the current band name   |
| log.get_call          | s:n           | Returns the Call field contents   |
| log.get_country       | s:n           | Returns the Country field contents  |
| log.get_exchange      | s:n           | Returns the contest exchange field contents   |
| log.get_frequency     | s:n           | Returns the Frequency field contents  |
| log.get_locator       | s:n           | Returns the Locator field contents  |
| log.get_name          | s:n           | Returns the Name field contents   |

|                            |     |   |
|----------------------------|-----|---|
| log.get_notes              | s:n | Returns the Notes field contents  |
| log.get_province           | s:n | Returns the Province field contents   |
| log.get_qth                | s:n | Returns the QTH field contents  |
| log.get_rst_in             | s:n | Returns the RST(r) field contents   |
| log.get_rst_out            | s:n | Returns the RST(s) field contents   |
| log.get_serial_number      | s:n | Returns the serial number field contents  |
| log.get_serial_number_sent | s:n | Returns the serial number (sent) field contents   |
| log.get_state              | s:n | Returns the State field contents  |
| log.get_time_off           | s:n | Returns the Time-Off field contents   |
| log.get_time_on            | s:n | Returns the Time-On field contents  |
| log.set_call               | n:s | Sets the Call field contents  |
| log.set_exchange           | n:s | Sets the contest exchange field contents  |
| log.set_locator            | n:s | Sets the Locator field contents   |
| log.set_name               | n:s | Sets the Name field contents  |
| log.set_qth                | n:s | Sets the QTH field contents   |
| log.set_rst_in             | n:s | Sets the RST(r) field contents  |
| log.set_rst_out            | n:s | Sets the RST(s) field contents  |
| log.set_serial_number      | n:s | Sets the serial number field contents   |
| main.abort                 | n:n | Aborts a transmit or tune   |
| main.get_afc               | b:n | Returns the AFC state   |
| main.get_char_rates        | s:n | Returns table of char rates.  |
| main.get_char_timing       | n:6 | Input: value of character. Returns transmit duration for specified character (samples:sample rate). |
| main.get_frequency         | d:n | Returns the RF carrier frequency  |
| main.get_lock              | b:n | Returns the Transmit Lock state   |
| main.get_max_macro_id      | i:n | Returns the maximum macro ID number   |
| main.get_reverse           | b:n | Returns the Reverse Sideband state  |
| main.get_rsid              | b:n | Returns the RSID state  |
| main.get_txid              | b:n | Returns the TxRSID state  |
| main.get_squelch           | b:n | Returns the squelch state   |
| main.get_squelch_level     | d:n | Returns the squelch level   |
| main.get_status1           | s:n | Returns the contents of the first status field (typically s/n)                                      |
| main.get_status2           | s:n | Returns the contents of the second status field   |
| main.get_trx_state         | s:n | Returns T/R state   |
| main.get_trx_status        | s:n | Returns transmit/tune/receive status  |
| main.get_tx_timing         | n:6 | Returns transmit duration for test string (samples:sample rate:secs).                               |
| main.get_wf_sideband       | s:n | Returns the current waterfall sideband  |

|                            |     |  |
|----------------------------|-----|--|
| main.inc_frequency         | d:d | Increments the RF carrier frequency. Returns the new value |
| main.inc_squelch_level     | d:d | Increments the squelch level. Returns the new level        |
| main.run_macro             | n:i | Runs a macro   |
| main.rx                    | n:n | Receives   |
| main.rx_only               | n:n | Disables Tx.   |
| main.rx_tx                 | n:n | Sets normal Rx/Tx switching.                               |
| main.set_afc               | b:b | Sets the AFC state. Returns the old state                  |
| main.set_frequency         | d:d | Sets the RF carrier frequency. Returns the old value       |
| main.set_lock              | b:b | Sets the Transmit Lock state. Returns the old state        |
| main.set_reverse           | b:b | Sets the Reverse Sideband state. Returns the old state     |
| main.set_rsid              | b:b | Sets the RSID state. Returns the old state                 |
| mmain.set_txid             | b:b | Sets the TxRSID state. Returns the old state               |
| main.set_squelch           | b:b | Sets the squelch state. Returns the old state              |
| main.set_squelch_level     | d:d | Sets the squelch level. Returns the old level              |
| main.set_wf_sideband       | n:s | Sets the waterfall sideband to USB or LSB                  |
| main.toggle_afc            | b:n | Toggles the AFC state. Returns the new state               |
| main.toggle_lock           | b:n | Toggles the Transmit Lock state. Returns the new state     |
| main.toggle_reverse        | b:n | Toggles the Reverse Sideband state. Returns the new state  |
| main.toggle_rsid           | b:n | Toggles the RSID state. Returns the new state              |
| main.toggle_txid           | b:n | Toggles the TxRSID state. Returns the new state            |
| main.toggle_squelch        | b:n | Toggles the squelch state. Returns the new state           |
| main.tune                  | n:n | Tunes  |
| main.tx                    | n:n | Transmits  |
| modem.get_afc_search_range | i:n | Returns the modem AFC search range                         |
| modem.get_bandwidth        | i:n | Returns the modem bandwidth                                |
| modem.get_carrier          | i:n | Returns the modem carrier frequency                        |
| modem.get_id               | i:n | Returns the ID of the current modem                        |
| modem.get_max_id           | i:n | Returns the maximum modem ID number                        |
| modem.get_name             | s:n | Returns the name of the current modem                      |



|                            |     |   |
|----------------------------|-----|---|
| modem.get_names            | A:n | Returns all modem names   |
| modem.get_quality          | d:n | Returns the modem signal quality in the range [0:100]                                     |
| modem.inc_afc_search_range | n:i | Increments the modem AFC search range. Returns the new value                              |
| modem.inc_bandwidth        | n:i | Increments the modem bandwidth. Returns the new value                                     |
| modem.inc_carrier          | i:i | Increments the modem carrier frequency. Returns the new carrier                           |
| modem.olivia.get_bandwidth | i:n | Returns the Olivia bandwidth  |
| modem.olivia.get_tones     | i:n | Returns the Olivia tones  |
| modem.olivia.set_bandwidth | n:i | Sets the Olivia bandwidth   |
| modem.olivia.set_tones     | n:i | Sets the Olivia tones   |
| modem.search_down          | n:n | Searches downward in frequency  |
| modem.search_up            | n:n | Searches upward in frequency  |
| modem.set_afc_search_range | n:i | Sets the modem AFC search range. Returns the old value                                    |
| modem.set_bandwidth        | n:i | Sets the modem bandwidth. Returns the old value   |
| modem.set_by_id            | i:i | Sets the current modem. Returns old ID  |
| modem.set_by_name          | s:s | Sets the current modem. Returns old name  |
| modem.set_carrier          | i:i | Sets modem carrier. Returns old carrier   |
| navtex.get_message         | s:i | Returns next Navtex/SitorB message with a max delay in seconds.. Empty string if timeout. |
| navtex.send_message        | s:s | Send a Navtex/SitorB message. Returns an empty string if OK otherwise an error message.   |
| rig.get_bandwidth          | s:n | Returns the name of the current transceiver bandwidth                                     |
| rig.get_bandwidths         | A:n | Returns the list of available rig bandwidths  |
| rig.get_mode               | s:n | Returns the name of the current transceiver mode  |
| rig.get_modes              | A:n | Returns the list of available rig modes   |
| rig.get_name               | s:n | Returns the rig name previously set via rig.set_name                                      |
| rig.get_notch              | s:n | Reports a notch filter frequency based on WF action                                       |
| rig.release_control        | n:n | Switches rig control to previous setting  |
| rig.set_bandwidth          | n:s | Selects a bandwidth previously added by rig.set_bandwidths                                |
| rig.set_bandwidths         | n:A | Sets the list of available rig bandwidths   |
| rig.set_frequency          | d:d | Sets the RF carrier frequency. Returns the old value                                      |

|                         |     |   |
|-------------------------|-----|---|
| rig.set_mode            | n:s | Selects a mode previously added by rig.set_modes  |
| rig.set_modes           | n:A | Sets the list of available rig modes  |
| rig.set_name            | n:s | Sets the rig name for xmlrpc rig  |
| rig.set_pwrmeter        | n:i | Sets the power meter returns null.  |
| rig.set_smeter          | n:i | Sets the smeter returns null.   |
| rig.take_control        | n:n | Switches rig control to XML-RPC   |
| rx.get_data             | 6:n | Returns all RX data received since last query.  |
| rxtx.get_data           | 6:n | Returns all RXTX combined data since last query.  |
| spot.get_auto           | b:n | Returns the autospotter state   |
| spot.pskrep.get_count   | i:n | Returns the number of callsigns spotted in the current session                            |
| spot.set_auto           | n:b | Sets the autospotter state. Returns the old state   |
| spot.toggle_auto        | n:b | Toggles the autospotter state. Returns the new state                                      |
| text.add_tx             | n:s | Adds a string to the TX text widget   |
| text.add_tx_bytes       | n:6 | Adds a byte string to the TX text widget  |
| text.clear_rx           | n:n | Clears the RX text widget   |
| text.clear_tx           | n:n | Clears the TX text widget   |
| text.get_rx             | 6:i | Returns a range of characters (start, length) from the RX text widget                     |
| text.get_rx_length      | i:n | Returns the number of characters in the RX widget   |
| tx.get_data             | 6:n | Returns all TX data transmitted since last query.   |
| wefax.end_reception     | s:n | End Wefax image reception   |
| wefax.get_received_file | s:i | Waits for next received fax file, returns its name with a delay. Empty string if timeout. |
| wefax.send_file         | s:i | Send file. returns an empty string if OK otherwise an error message.                      |
| wefax.set_adif_log      | s:b | Set/reset logging to received/transmit images to ADIF log file                            |
| wefax.set_max_lines     | s:i | Set maximum lines for fax image reception   |
| wefax.set_tx_abort_flag | s:n | Cancels Wefax image transmission  |
| wefax.skip_apt          | s:n | Skip APT during Wefax reception   |
| wefax.skip_phasing      | s:n | Skip phasing during Wefax reception   |
| wefax.state_string      | s:n | Returns Wefax engine state (tx and rx) for information.                                   |

Deprecated methods:

| Method Name             | Sig | Resolution               |
|-------------------------|-----|--------------------------|
| log.get_sideband        | s:n | use main.get_wf_sideband |
| main.get_rig_bandwidth  | s:n | use rig.get_bandwidth    |
| main.get_rig_bandwidths | n:A | use rig.get_bandwidths   |

|                         |     |  |
|-------------------------|-----|--|
| main.get_rig_mode       | s:n | use rig.get_mode                             |
| main.get_rig_modes      | A:n | use rig.get_modes                            |
| main.get_sideband       | s:n | use main.get_wf_sideband and/or rig.get_mode |
| main.rsid               | n:n | use main.{get,set,toggle}_rsid               |
| main.set_rig_bandwidth  | n:s | use rig.set_bandwidth                        |
| main.set_rig_bandwidths | n:A | use rig.set_bandwidths                       |
| main.set_rig_frequency  | d:d | use rig.set_frequency                        |
| main.set_rig_mode       | n:s | use rig.set_mode                             |
| main.set_rig_modes      | n:A | use rig.set_modes                            |
| main.set_rig_name       | n:s | use rig.set_name                             |
| main.set_sideband       | n:s | use main.set_wf_sideband and/or rig.set_mode |

### 7.9.3 Minimized WF Window

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:

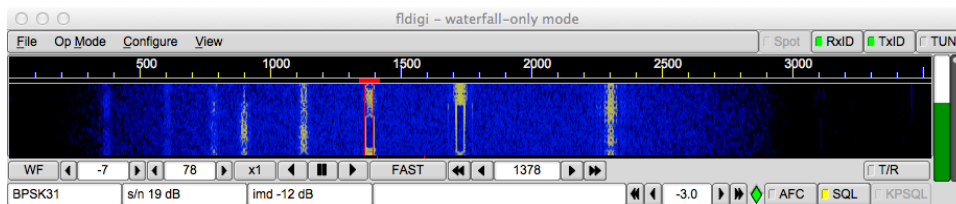


Figure 7.3: Simplified WF Window Display

The documentation for the external control program will provide additional information if this user interface is desired.

[Return to Top of Page](#)

[Return to Main Page](#)



## Chapter 8

# Installing fldigi on a thumb drive

Installing fldigi (and any of the other NBEMS flxxxx applications) on a thumb drive is very simple. Create a folder on the thumb drive named FL\_APPS. Then copy the fldigi executable to that folder. Note that folder FL\_APPS may reside on any hard drive, or removable drive (including a rewritable CD/DVD) on your system. Copy the executable for flarq, flrig, flmsg, flamp, etc to that same folder if you want to have a complete NBEMS installation on that device:

- fldigi.exe
- flarq.exe
- flmsg.exe
- flamp.exe
- flrig.exe

The key to using the applications thus installed is to execute the program from that folder. This is best accomplished by opening the FL\_APPS folder using the system files explorer. Then start the application by double clicking on it's icon.

You will be guided through the new install wizard the first time you do this and fldigi will place all of it's configuration folders in the same folder from which it was started. For example, if the thumb drive were on an XP system and the drive was system device "H:"

- H:\FL\_APPS\fldigi.exe
- H:\FL\_APPS\fldigi.files
- H:\FL\_APPS\fldigi.files\fldigi\_def.xml
- etc ...

If this removable drive installation is to be used on several computers or you expect that the hardware configuration might change then you should follow these steps each and every time that you exit the program. This will insure that fldigi will restart correctly.

- select sound card as "File I/O only"
- deselect all transceiver control (be sure to Initialize any changes)
- note: you do not have to disable xmlrpc rig control
- save the configuration

You will follow a similar procedure when you shut down fldigi. The purpose is to remove all configuration items which are hardware specific. Reconfiguring the sound card and transceiver control only takes a few steps the next time fldigi is executed.

[Return to Main Page](#)

## Chapter 9

# Configure Scripting

### 9.1 Configure Scripting

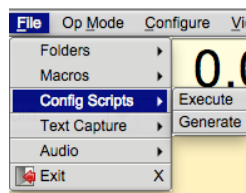


Figure 9.1: Scripts

#### WARNING:

If you do anything with this option, ensure FLDIGI is configured to your preferences. Then generate a restore script. In doing so, any alterations performed below can be returned to it's intended state.

#### NOTICE:

Because of different hardware and Operating Systems it's advisable to avoid commands that are hardware/OS dependent when creating standard configuration scripts.

#### Configure scripting uses:

- Configuring FLDigi to support a specific net operation.
- Restore user data when issues with FLDigi requires the deletion or reset of fldigi.prefs and fldigi\_def.xml files.

#### Net Operations

A script can be created for setting the rig frequency, waterfall offset, modem, rig mode, and custom macros. See [Configure Script Commands](#) for other available options.

#### Configuration Reset

After reset, a configure script can be executed to restore **User Data**. The commands available do not support a full restore. Some option may need manual entry. See [Configure Script Commands](#) for limits.

### 9.1.1 Executing Scripts

From FLDigi's main window select menu item File->Config Scripts->Execute. This will bring up a file browser. Search and select the file in question, then press [Open]. At this point the script will execute.

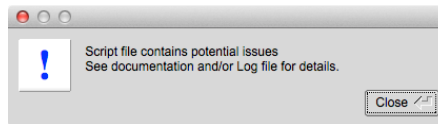


Figure 9.2: Warning Report Dialog Box

If problems are encountered the above dialog box is used to inform the user. To ascertain the extent of the issue(s), select menu item Help->Event Log. The log entries will contain the reason and line number for each reported error.

Some issues reported are the result of preset data that is specific to a particular operating system. These can be safely ignored until configured to operate on your system.

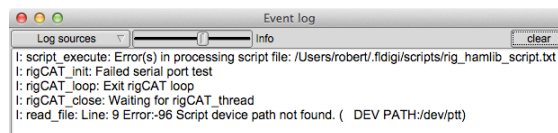


Figure 9.3: Error Report Log Entry

Some configuration options will require FLDigi restart. Ensure configuration is saved prior to restarting.

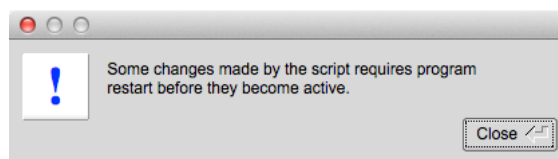


Figure 9.4: Restart FLDIGI notice

#### NOTE:

Some script commands are specific to operating system and hardware combination. If your system does not support these commands, they will be ignored. The easiest way to determine what your system supports is to generate a script and note its content.

### 9.1.2 Generating Scripts

To generate a script the operator selects FLDigi's menu items File->Config Scripts->Generate. A file save dialog box is opened for the user to select a name and where the file is saved. Pressing [Save] initiates the process.

To create custom scripts there are two ways of doing this. With a Text editor (see [Scripting Syntax](#)) or changing the content/settings of FLDigi prior to the script generation.



If the latter is used, you are advised to generate a script prior to changing any of FLDigi's settings. Just in case restoration is needed.

Prior to editing any script file, take the time to read the relevant sections of this document. See [Scripting Syntax](#) and [Example Script](#)

**NOTE:**

Generated script contents are specific to the operating system and hardware available on the host machine. If a device is not supported on your system, script code will not be generated for it.

## 9.2 Configure Script Commands

### FLDIGI

FREQ | MODE | WFHZ | RXID | TXID | SPOT | REV | AFC | LOCK | SQL | KPSQL |  
MODEM

### OPERATOR

CALLSIGN | NAME | QTH | LOC | ANT

### AUDIO DEVICE

OSS | OSS DEV | PA | PA PLAYBACK | PA CAPTURE | PUA | PUA SERVER

### AUDIO SETTINGS

CAPTURE | PLAYBACK | CONVERTER | RX PPM | TX PPM | TX OFFSET

### AUDIO RT CHANNEL

MONO AUDIO | MODEM LR | REV LR | PTT RT CHAN | QSK RT CHAN | FSK RT CHAN

### AUDIO WAVE.SRATE

### RIG HRDWR PTT

PTT RT CHAN | SERIAL PORT | DEVICE | RTS | DTR | RTSV | DTRV | START PTT DELAY  
| END PTT DELAY | UHROUTER | PARALLEL | INIT

### RIGCAT

STATE | DESC FILE | DEV PATH | RETRIES | RETRY INT | WDELAY | IDELAY | BRATE  
| SBITS | ECHO | TOGGLE RTS PTT | TOGGLE DTR PTT | RESTORE | PTT COMMAND |  
RTS 12V | DTR 12V | HRDWR FC | VSP | INIT

## HAMLIB

```
STATE | RIG | DEV PATH | RETRIES | RETRY INT | WDELAY | PWDELAY | BRATE |
SBITS | SBAND | PTT COMMAND | DTR 12V | RTS 12V | HRDWR FC | SFTWR FC | ADV
CONF | INIT
```

## XMLRPC RC

```
STATE | BWDELAY | INIT
```

## IO

```
LOCK | PORT | CSMA | KISS | KISS.IPA | KISS.IOPN | KISS.OPN | KISS.DP | KISS.BUSY
| KISS.CONT | KISS.ATTEN | ARQ | ARQ.IPA | ARQ.IOPN | XMLRPC | XMLRPC.IPA |
XMLRPC.IOPN
```

## MISC NBEMS

```
TIMEOUT | OPEN BRWSR | STATE | OPEN MSG | OPEN FLMSG | PATH
```

## ID

```
RSID.NOTIFY | RSID.SRCH BP | RSID.MARK PREV | RSID.DETECTOR | RSID.ALRT DIALOG
| RSID.TX FREQ LOCK | RSID.FREQ CHANGE | RSID.ALLOW ERRORS | RSID.SQL OPEN |
RSID.PRETONE | RSID.END XMT ID | VIDEO.ID MODE | VIDEO.VIDEO TXT | VIDEO.TEXT INPUT
| VIDEO.SMALL FONT | VIDEO.500HZ | VIDEO.WIDTH LIMIT | VIDEO.CHAR ROW | CW.TX CALL
| CW.DETECTOR | CW.SPEED |
```

## MACRO

## 9.3 Scripting Syntax

The script syntax is formatted into a multi tier system using two variations. The first is a structured syntax. The second uses a hierarchy system employing dot '.' notation. The command text used closely mimics the GUI configuration panel tabs hierarchy and contents to easily discern the connection between commands and the effected settings.

### 9.3.1 Structured Commands

Each command ends with a colon (':'). This separates the command section with the paramater section. A command may or may not contain a paramater. Structured statements are terminated using "END:" statement.

```
<TOP_TIER>:
<SUB_TIER>:<Parameter>
<SUB_TIER>:<No Parameter>
END:
```

### 9.3.2 Dot Notation Commands

Each command ends with a colon (':'). This separates the command section with the parameter section. A command may or may not contain a parameter. The inclusion of a period ('.') within the command section denotes the separation of upper and lower tier segments.

```
<TOP_TIER>.<SUB_TIER>,...:<Parameter>  
<TOP_TIER>.<SUB_TIER>,...:<No Parameter>
```

### 9.3.3 Parameters

Parameter content varies on the command used. The various parameter types are:

- `Integers` (whole numbers, ie 0, 10, 100, 234).
- `Real Numbers` (floating point numbers, ie 1.2, 10.3).
- `Characters Strings` (one or more words, "Madison, AL", "Toney, AL").

### 9.3.4 Script Files

The script file name is expected to contain a .txt extension. The editing of these files must utilize a true text editor<sup>1</sup>. Preferably one that understands the various line ending sequences used between the different Operating Systems.

- The first line in the script file must contain "**FLDIGI\_CONFIG**" tag. This is used to verify the "text" file is a configuration script for FLDIGI. If it's not present the file will not be processed.
- One command : parameter pair per line.
- Command indentations with spaces or tabs are not required (cosmetic).
- Parameters are separated by a comma ','.
- Parameters regardless of type can be encapsulated in quotes (" "). However, if a parameter contains spaces or commas it must be encapsulated.
- One or more commands per file.
- One or more empty lines per file..
- Any text entry after the '#' pound sign is considered a 'comment' and will be ignored.

**Note: 1**

Windows: Notepad++, Geany  
Macintosh: TextWrangler  
Linux: Geany, Gedit, Pluma

[http://en.wikipedia.org/wiki/Category:Windows\\_text\\_editors](http://en.wikipedia.org/wiki/Category:Windows_text_editors)  
[http://en.wikipedia.org/wiki/Category:Linux\\_text\\_editors](http://en.wikipedia.org/wiki/Category:Linux_text_editors)

## 9.4 FLDigi Commands

See [Operating Controls and Displays](#)

### 9.4.1 Frequency Control

Sets the RF Carrier frequency (effects rig control).

FREQ:<Real Number>

<Real Number> values are entered directly or as exponential notation. The following frequencies represent the same frequency value.

Frequency Value = Parameter Syntax

- 14.070 MHz = 14.070e+06
- 14070.00 KHz = 14070.0e+03 - Using this exponent magnitude represents the scale of the frequency display
- 14070000 Hz = 14070000 - Directly in Hz

### 9.4.2 Mode

Sets the rig modulation mode (ie. USB, LSB, FM, etc). The actual values are dependent on the rig and the interface used.

MODE:<Character String>

### 9.4.3 Waterfall Offset (Hz)

Sets the audio tone offset in the waterfall.

WFHZ:<Integer>

Valid values range from 0 - 4000 +/- 50% of the bandwidth needed for the selected modem.

### 9.4.4 RxID

Enable or disable the use of RSID reception.

RXID:<ENABLE | DISABLE>

### 9.4.5 TxID

Enable or disable the use of RSID transmission.

TXID:<ENABLE | DISABLE>

### 9.4.6 Spot

Enable or disable the use of the PSKreporter mechanism.

SPOT:<ENABLE | DISABLE>

### 9.4.7 Rev

Enable or disable the use of upper and lower tone reversal. Allowing modems like MFSK to receive/transmit LSB signal while the rig is in USB. RSID will not decode when this option is selected and not all modems are supported.

REV:<ENABLE | DISABLE>

### 9.4.8 AFC

Enable or disable the use of Automatic Frequency Control.

AFC:<ENABLE | DISABLE>

### 9.4.9 Lk

Enable or disable the use of frequency lock. The behavior of this feature can be altered depending on the settings of the [RSID / Video / ID Configuration](#) panel.

LOCK:<ENABLE | DISABLE>

### 9.4.10 SQL

Enable or disable the use of the squelch.

SQL:<ENABLE | DISABLE>

### 9.4.11 KPSQL

Enable or disable the use of the KISS power squelch. This option is only available when KISS IO interface option is selected.

See [Configure ARQ/KISS I/O](#)

KPSQL:<ENABLE | DISABLE>

### 9.4.12 MODEM Control

Sets the desired modem to receive and transmit from. Some modems will not be available if KISS IO interface option is selected. See [Mode Table](#) under 'mode' column for proper modem parameter syntax.

MODEM:<Character String>

### 9.4.13 Examples

| Structured                           | Dot Notation           |
|--------------------------------------|------------------------|
| FLDIGI :<br>FREQ:14.070e+06<br>END : | FLDIGI.FREQ:3853.0e+03 |
| FLDIGI :<br>MODE:DIG<br>END :        | FLDIGI.MODE:DIG        |

|                                      |                          |
|--------------------------------------|--------------------------|
| FLDIGI :<br>WFHZ : 1500<br>END :     | FLDIGI . WFHZ : 700      |
| FLDIGI :<br>RXID : ENABLE<br>END :   | FLDIGI . RXID : DISABLE  |
| FLDIGI :<br>TXID : DISABLE<br>END :  | FLDIGI . TXID : DISABLE  |
| FLDIGI :<br>SPOT : DISABLE<br>END :  | FLDIGI . SPOT : ENABLE   |
| FLDIGI :<br>REV : DISABLE<br>END :   | FLDIGI . REV : ENABLE    |
| FLDIGI :<br>AFC : DISABLE<br>END :   | FLDIGI . AFC : DISABLE   |
| FLDIGI :<br>LOCK : DISABLE<br>END :  | FLDIGI . LOCK : ENABLE   |
| FLDIGI :<br>SQL : DISABLE<br>END :   | FLDIGI . SQL : ENABLE    |
| FLDIGI :<br>KPSQL : DISABLE<br>END : | FLDIGI . KPSQL : DISABLE |
| FLDIGI :<br>MODEM : BPSK31<br>END :  | FLDIGI . MODEM : BPSK63  |

## 9.5 Operator Commands

See [Configure Operator](#)

### 9.5.1 Callsign:

OPERATOR.CALLSIGN:<Region Specific>

### 9.5.2 Name:

OPERATOR.NAME:<Your Name>

### 9.5.3 QTH:

OPERATOR.QTH:<City, State, Country, etc>

### 9.5.4 Locator:

Maidenhead Grid Locator. A series of alpha numerical characters representing a position on earth.

OPERATOR.LOC:<AAnnaann>

A: Uppercase characters  
 n: Numerical data (0-9)  
 a: Lowercase characters

Additional Information: <http://www.arrl.org/grid-squares>, or [http://en.wikipedia.org/wiki/Maidenhead\\_Locator\\_System](http://en.wikipedia.org/wiki/Maidenhead_Locator_System)

### 9.5.5 Operators Antenna Description

OPERATOR.ANT:<Antenna type, height, etc>

### 9.5.6 Examples

| Structured  | Dot Notation                         |
|---|--------------------------------------|
| OPERATOR:<br>CALLSIGN:KK5VD<br>END:               | OPERATOR.CALLSIGN:w1hkj              |
| OPERATOR:<br>NAME:Robert<br>END:                  | OPERATOR.NAME:Dave                   |
| OPERATOR:<br>QTH:"Madison, AL"<br>END:            | OPERATOR.QTH:"Toney, AL"             |
| OPERATOR:<br>LOC:EM64pr<br>END:                   | OPERATOR.LOC:EM64                    |
| OPERATOR:<br>ANT:"Inverted 'V' @22FT AGL"<br>END: | OPERATOR.ANT:"15 EL LP @ 200 FT AGL" |

## 9.6 Audio Device Commands

See [Sound Card Configuration](#)

### 9.6.1 OSS

Enable or disable OSS. This option is not available on all operating systems.

OSS:<ENABLE | DISABLE>

### 9.6.2 Device:

OSS DEV:<Character String>

Parameter:/dev/path

### 9.6.3 PortAudio

Enable or disable Port Audio. This option may or may not be available to all operating systems.

PA:<ENABLE | DISABLE>

### 9.6.4 Playback:

The device name is specific to the hardware installed.

PA PLAYBACK:<Integer>,<Character String>

Parameter:Menu Index, Device Name

**Note:**

Parameter Menu Index is specific to the fldigi/hardware combination. If the device name is available on other systems it will be set to the first occurrence of that device name.

### 9.6.5 Capture:

The device name is specific to the hardware installed.

PA CAPTURE:<Integer>,<Character String>

**Note:**

Parameter Menu Index is specific to the fldigi/hardware combination. If the device name is available on other systems it will be set to the first occurrence of that device name.

Parameter:Menu Index, Device Name

### 9.6.6 PulseAudio

Enable or disable Pulse Audio. This option is not available on all operating systems.

PUA:<ENABLE | DISABLE>

### 9.6.7 Server string:

PUA SERVER:<Character String>

Parameter: See Note: † below.

### 9.6.8 Examples

| Structured  | Dot Notation                                   |
|---|--|
| AUDIO DEVICE:<br>OSS:ENABLE<br>END:                     | AUDIO DEVICE.OSS:ENABLE                        |
| AUDIO DEVICE:<br>OSS DEV:/dev/path<br>END:              | AUDIO DEVICE.OSS DEV:/dev/path                 |
| AUDIO DEVICE:<br>PA:ENABLE<br>END:                      | AUDIO DEVICE.PA:ENABLE                         |
| AUDIO DEVICE:<br>PA CAPTURE:2,"USB Audio CODEC"<br>END: | AUDIO DEVICE.PA CAPTURE:2,"USB<br>Audio CODEC" |



|  |  |
|--|--|
| AUDIO DEVICE:<br>PA PLAYBACK:3,"USB Audio CODEC"<br>END: | AUDIO DEVICE.PA CAPTURE:3,"USB<br>Audio CODEC" |
| AUDIO DEVICE:<br>PUA:ENABLE<br>END:                      | AUDIO DEVICE.PUA:ENABLE                        |
| AUDIO DEVICE:<br>PUA SERVER:†<br>END:                    | AUDIO DEVICE.PUA SERVER:†                      |

Note: † Pulse Audio Server Strings

Pulse audio server string information can be found here:

<http://www.freedesktop.org/wiki/Software/PulseAudio/Documentation/User/ServerStrings/>

## 9.7 Audio Settings

See [Sound Card Configuration](#)

### 9.7.1 Capture Audio Sample Rate

Set the input signal sample rate. The sample rate availability is dependent on the ADC hardware.

Auto, Native, 8000,...

CAPTURE:<Character String>

### 9.7.2 Playback Audio Sample Rate

Set the output signal sample rate. The sample rate availability is dependent on the DAC hardware.

Auto, Native, 8000,...

PLAYBACK:<Character String>

### 9.7.3 Converter

CONVERTER:<Character String> Best, Medium, Fastest, Linear

### 9.7.4 Receive Audio PPM Correction

RX PPM:<Integer> negative, 0, positive values

### 9.7.5 Transmit Audio PPM Correction

TX PPM:<Integer> negative, 0, positive values

### 9.7.6 Transmit Audio Hertz Offset Correction

TX OFFSET:<Integer> negative, 0, positive values (Hertz)

### 9.7.7 Examples

| Structured                                  | Dot Notation                       |
|---|------------------------------------|
| AUDIO SETTINGS:<br>CAPTURE:48000<br>END:    | AUDIO SETTINGS.CAPTURE:Auto        |
| AUDIO SETTINGS:<br>PLAYBACK:8000<br>END:    | AUDIO SETTINGS.PLAYBACK:Native     |
| AUDIO SETTINGS:<br>CONVERTER:"Best"<br>END: | AUDIO SETTINGS.CONVERTER:"Fastest" |
| AUDIO SETTINGS:<br>RX PPM:17<br>END:        | AUDIO SETTINGS.RX PPM:17           |
| AUDIO SETTINGS:<br>TX PPM:17<br>END:        | AUDIO SETTINGS.TX PPM:17           |
| AUDIO SETTINGS:<br>TX OFFSET:17<br>END:     | AUDIO SETTINGS.TX OFFSET:17        |

## 9.8 Audio Right Channel Commands

See [Right Channel Audio Output](#)

### 9.8.1 Mono audio output

Enable or disable monophonic signal output.

MONO AUDIO:<ENABLE | DISABLE>

### 9.8.2 Modem signal on left and right channels

Enable or disable audio signal to both left and right channels

MODEM LR:<ENABLE | DISABLE>

### 9.8.3 Reverse Left/Right channels

Enable or disable audio signal reversal on Left/Right channels.

REV LR:<ENABLE | DISABLE>

### 9.8.4 PTT tone on right audio channel

Enable or disable PTT use by supplying a signal on the right channel.

PTT RT CHAN:<ENABLE | DISABLE>

### 9.8.5 CW QSK signal on right channel

QSK RT CHAN:<ENABLE | DISABLE>

### 9.8.6 Pseudo-FSK on right audio channel

FSK RT CHAN:<ENABLE | DISABLE>

### 9.8.7 Examples

| Structured                                       | Dot Notation                            |
|--|---|
| AUDIO RT CHANNEL:<br>MONO AUDIO:DISABLE<br>END:  | AUDIO RT CHANNEL.MONO AUDIO:DISABLE     |
| AUDIO RT CHANNEL:<br>MODEM LR:DISABLE<br>END:    | AUDIO RT CHANNEL.MODEM LR:DISABLE       |
| AUDIO RT CHANNEL:<br>CONVERTER:"Best"<br>END:    | AUDIO RT<br>CHANNEL.CONVERTER:"Fastest" |
| AUDIO RT CHANNEL:<br>REV LR:DISABLE<br>END:      | AUDIO RT CHANNEL.REV LR:DISABLE         |
| AUDIO RT CHANNEL:<br>PTT RT CHAN:DISABLE<br>END: | AUDIO RT CHANNEL.PTT RT<br>CHAN:DISABLE |
| AUDIO RT CHANNEL:<br>QSK RT CHAN:DISABLE<br>END: | AUDIO RT CHANNEL.QSK RT<br>CHAN:DISABLE |
| AUDIO RT CHANNEL:<br>FSK RT CHAN:DISABLE<br>END: | AUDIO RT CHANNEL.FSK RT<br>CHAN:DISABLE |

## 9.9 Wav write sample rate

See [WAV File Sample Rate](#)

Sets the record wave file sample rate.

AUDIO WAVE.SRATE:<Integer>

Available Parameters: 8000, 11025, 16000, 22050, 24000, 44100, 48000

| Structured                         | Dot Notation           |
|------------------------------------|------------------------|
| AUDIO WAVE:<br>SRATE:16000<br>END: | AUDIO WAVE.SRATE:16000 |

## 9.10 Hardware PTT Option Commands

See [Rig Configuration](#)

**9.10.1 PTT tone on right audio channel**

PTT RT CHAN:<ENABLE | DISABLE>

**9.10.2 Use separate serial port PTT**

SERIAL PORT:<ENABLE | DISABLE>

**9.10.3 Device:**

DEVICE:<Character String>

Parameter: A device path (/dev/ptt)

**9.10.4 Use RTS**

RTS:<ENABLE | DISABLE>

**9.10.5 Use DTR**

DTR:<ENABLE | DISABLE>

**9.10.6 RTS = +V**

RTSV:<ENABLE | DISABLE>

**9.10.7 DTR = +V**

DTRV:<ENABLE | DISABLE>

**9.10.8 PTT delays valid for all CAT/PTT types**

START PTT DELAY:<Integer>

**9.10.9 PTT delays valid for all CAT/PTT types**

END PTT DELAY:<Integer>

**9.10.10 Use uHRouter PTT**

UHROUTER:<ENABLE | DISABLE>

**9.10.11 Use parallel port PTT**

PARALLEL:<ENABLE | DISABLE>

**9.10.12 Initialize Changes**

INIT:

## 9.10.13 Examples

| Structured                                    | Dot Notation                      |
|---|-----------------------------------|
| RIG HRDWR PTT:<br>PTT RT CHAN:DISABLE<br>END: | RIG HRDWR PTT.PTT RT CHAN:DISABLE |
| RIG HRDWR PTT:<br>SERIAL PORT:DISABLE<br>END: | RIG HRDWR PTT.SERIAL PORT:DISABLE |
| RIG HRDWR PTT:<br>DEVICE:/dev/ptt<br>END:     | RIG HRDWR PTT.DEVICE:COM1         |
| RIG HRDWR PTT:<br>RTS:DISABLE<br>END:         | RIG HRDWR PTT.RTS:DISABLE         |
| RIG HRDWR PTT:<br>RTSV:DISABLE<br>END:        | RIG HRDWR PTT.RTSV:DISABLE        |
| RIG HRDWR PTT:<br>DTR:DISABLE<br>END:         | RIG HRDWR PTT.DTR:DISABLE         |
| RIG HRDWR PTT:<br>DTRV:DISABLE<br>END:        | RIG HRDWR PTT.DTRV:DISABLE        |
| RIG HRDWR PTT:<br>PARALLEL:DISABLE<br>END:    | RIG HRDWR PTT.PARALLEL:DISABLE    |
| RIG HRDWR PTT:<br>UHROUTER:DISABLE<br>END:    | RIG HRDWR PTT.UHROUTER:DISABLE    |
| RIG HRDWR PTT:<br>START PTT DELAY:1<br>END:   | RIG HRDWR PTT.START PTT DELAY:1   |
| RIG HRDWR PTT:<br>END PTT DELAY:1<br>END:     | RIG HRDWR PTT.END PTT DELAY:1     |
| RIG HRDWR PTT:<br>INIT:<br>END:               | RIG HRDWR PTT.INIT:               |

## 9.11 RIGCAT Commands

See [RigCAT control](#)

## 9.11.1 Use RigCAT

STATE:<ENABLE | DISABLE>

## 9.11.2 Rig description file:

DESC FILE:<Character String>

Parameter:/directory/path/file.xml

### 9.11.3 Device:

DEV PATH:<Character String>

Parameter:/dev/path

### 9.11.4 Retries

RETRIES:<integer>

### 9.11.5 Retry interval (ms)

RETRY INT:<integer>

### 9.11.6 Write delay (ms)

WDELAY:<integer>

### 9.11.7 Init delay (ms)

IDELAY:<integer>

### 9.11.8 Baud rate:

BRATE:<integer>

### 9.11.9 Stopbits

SBITS:<integer>

### 9.11.10 Commands are echoed

ECHO:<ENABLE | DISABLE>

### 9.11.11 Toggle RTS for PTT

TOGGLE RTS PTT:<ENABLE | DISABLE>

### 9.11.12 Toggle DTR for PTT

TOGGLE DTR PTT:<ENABLE | DISABLE>

### 9.11.13 Restore Settings on Close

RESTORE:<ENABLE | DISABLE>

### 9.11.14 CAT command for PTT

PTT COMMAND:<ENABLE | DISABLE>

**9.11.15 RTS +12 v**

RTS 12V:<ENABLE | DISABLE>

**9.11.16 DTR +12 v**

DTR 12V:<ENABLE | DISABLE>

**9.11.17 RTS/CTS flow control**

HRDWR FC:<ENABLE | DISABLE>

**9.11.18 VSP Enable**

VSP:<ENABLE | DISABLE>

**9.11.19 Initialize Changes**

INIT:

**9.11.20 Examples**

| Structured   | Dot Notation                                       |
|--|--|
| RIGCAT:<br>STATE:ENABLE<br>END:                                | RIG HRDWR PTT.STATE:ENABLE                         |
| RIGCAT:<br>DEV<br>PATH:/dev/tty.usbserial-RT071600<br>END:     | RIGCAT.DEV PATH:COM1                               |
| RIGCAT:<br>DESC<br>FILE:/Users/.fldigi/rigs/FT-817.xml<br>END: | RIGCAT.DESC<br>FILE:/Users/.fldigi/rigs/FT-817.xml |
| RIGCAT:<br>RETRIES:2<br>END:                                   | RIGCAT.RETRIES:2                                   |
| RIGCAT:<br>RETRY INT:500<br>END:                               | RIGCAT.RETRY INT:500                               |
| RIGCAT:<br>WDELAY:5<br>END:                                    | RIGCAT.WDELAY:5                                    |
| RIGCAT:<br>IDELAY:0<br>END:                                    | RIGCAT.IDELAY:0                                    |
| RIGCAT:<br>BRATE:4800<br>END:                                  | RIGCAT.BRATE:4800                                  |

|   |                               |
|---|-------------------------------|
| RIGCAT:<br>SBITS:1<br>END:                | RIGCAT.SBITS:1                |
| RIGCAT:<br>ECHO:DISABLE<br>END:           | RIGCAT.ECHO:DISABLE           |
| RIGCAT:<br>TOGGLE RTS PTT:DISABLE<br>END: | RIGCAT.TOGGLE RTS PTT:DISABLE |
| RIGCAT:<br>RESTORE:DISABLE<br>END:        | RIGCAT.RESTORE:DISABLE        |
| RIGCAT:<br>PTT COMMAND:DISABLE<br>END:    | RIGCAT.PTT COMMAND:DISABLE    |
| RIGCAT:<br>TOGGLE DTR PTT<br>END:         | RIGCAT.TOGGLE DTR PTT         |
| RIGCAT:<br>TOGGLE RTS PTT:DISABLE<br>END: | RIGCAT.TOGGLE RTS PTT:DISABLE |
| RIGCAT:<br>RTS 12V:DISABLE<br>END:        | RIGCAT.RTS 12V:DISABLE        |
| RIGCAT:<br>DTR 12V:DISABLE<br>END:        | RIGCAT.DTR 12V:DISABLE        |
| RIGCAT:<br>HRDWR FC:DISABLE<br>END:       | RIGCAT.HRDWR FC:DISABLE       |
| RIGCAT:<br>VSP:DISABLE<br>END:            | RIGCAT.VSP:DISABLE            |
| RIGCAT:<br>INIT:<br>END:                  | RIGCAT.INIT:                  |

## 9.12 HAMLIB Commands

See [Hamlib CAT control](#)

### 9.12.1 Use Hamlib

Enable to disable Hamlib Rig Control interface.

STATE:<ENABLE | DISABLE>

### 9.12.2 Rig:

Set the rig to control. The list of Rigs are derived from internal list provided by HAMLIB.

RIG:<Character String>

Parameter: HAMLIB Device Names



### 9.12.3 Device:

The device path for controlling the Rig.

DEV PATH:<Character String>

Parameter:Device path

### 9.12.4 Retries

The number of tries at sending commands to the Rig

RETRIES:<Integer>

### 9.12.5 Retry Interval (ms)

RETRY INT:<Integer>

### 9.12.6 Write delay (ms)

WDELAY:<Integer>

### 9.12.7 Post write delay (ms)

PWDELAY:<Integer>

### 9.12.8 Baud rate:

BRATE:<Integer>

### 9.12.9 Stopbits

SBITS:<Integer>

### 9.12.10 Sideband:

SBAND:<Character String>

### 9.12.11 PTT via Hamlib command

PTT COMMAND:<ENABLE | DISABLE>

### 9.12.12 DTR +12

DTR 12V:<ENABLE | DISABLE>

### 9.12.13 RTS +12

RTS 12V:<ENABLE | DISABLE>

**9.12.14 RTS/CTS flow control**

HRDWR FC:<ENABLE | DISABLE>

**9.12.15 XON/XOFF flow control**

SFTWR FC:<ENABLE | DISABLE>

**9.12.16 Advanced configuration:**

ADV CONF:<Character String>

**9.12.17 Initialize Changes**

INIT:

Parameters: None

**9.12.18 Examples**

| Structured   | Dot Notation                      |
|--|-----------------------------------|
| HAMLIB:<br>STATE:DISABLE<br>END:                           | HAMLIB.STATE:DISABLE              |
| HAMLIB:<br>RIG:"Yaesu FT-817 (Beta)"<br>END:               | HAMLIB.RIG:"Yaesu FT-817 (Beta) " |
| HAMLIB:<br>DEV<br>PATH:/dev/tty.usbserial-RT071600<br>END: | HAMLIB.DEV PATH:COM1              |
| HAMLIB:<br>RETRIES:2<br>END:                               | HAMLIB.RETRIES:2                  |
| HAMLIB:<br>RETRY INT:10<br>END:                            | HAMLIB.RETRY INT:10               |
| HAMLIB:<br>WDELAY:0<br>END:                                | HAMLIB.WDELAY:0                   |
| HAMLIB:<br>PWDELAY:5<br>END:                               | HAMLIB.PWDELAY:5                  |
| HAMLIB:<br>BRATE:4800<br>END:                              | HAMLIB.BRATE:4800                 |
| HAMLIB:<br>SBITS:1<br>END:                                 | HAMLIB.SBITS:1                    |

|  |                            |
|--|----------------------------|
| HAMLIB:<br>SBAND:"Rig mode"<br>END:    | HAMLIB.SBAND:"Rig mode"    |
| HAMLIB:<br>PTT COMMAND:DISABLE<br>END: | HAMLIB.PTT COMMAND:DISABLE |
| HAMLIB:<br>DTR 12V:DISABLE<br>END:     | HAMLIB.DTR 12V:DISABLE     |
| HAMLIB:<br>RTS 12V:DISABLE<br>END:     | HAMLIB.RTS 12V:DISABLE     |
| HAMLIB:<br>HRDWR FC:DISABLE<br>END:    | HAMLIB.HRDWR FC:DISABLE    |
| HAMLIB:<br>SFTWR FC:DISABLE<br>END:    | HAMLIB.SFTWR FC:DISABLE    |
| HAMLIB:<br>ADV CONF:<br>END:           | HAMLIB.ADV CONF:           |
| HAMLIB:<br>INIT:<br>END:               | HAMLIB.INIT:               |

## 9.13 XMLRPC Rig Control Commands

See [Xml-Rpc CAT](#)

### 9.13.1 Use XML-RPC program

STATE:<ENABLE | DISABLE>

### 9.13.2 Mode/BW delay

BWDELAY:<Real Number>

### 9.13.3 Initialize Changes

INIT:

Parameters: None

### 9.13.4 Examples

| Structured                          | Dot Notation            |
|-------------------------------------|-------------------------|
| XMLRPC RC:<br>STATE:DISABLE<br>END: | XMLRPC RC.STATE:DISABLE |

|                                   |                       |
|-----------------------------------|-----------------------|
| XMLRPC RC:<br>BWDELAY:0.5<br>END: | XMLRPC RC.BWDELAY:0.5 |
| XMLRPC RC:<br>INIT:<br>END:       | XMLRPC RC.INIT:       |

## 9.14 IO

See [Configure ARQ/KISS I/O](#)

### 9.14.1 Lock

Used to lock the panel from user GUI access. Has no effect on script commands.

LOCK:<ENABLE | DISABLE>

### 9.14.2 Enable ARQ or KISS IO Access

Select the active IO port for data operation between programs.

PORT:<Character String>

Parameters:

ARQ: Used to control the FL Suite of auxiliary programs.

KISS: KISS protocol over UDP/IP interface

### 9.14.3 AX25 Decode

Enable or disable Ax25 decoding. Only available when the KISS IO port is active.

AX25D:<ENABLE | DISABLE>

### 9.14.4 Enable CSMA

Enable or disable Carrier Sense Multiple Access. Only available when the KISS IO port is active.

CSMA:<ENABLE | DISABLE>

### 9.14.5 KISS Sub Command

KISS related sub commands.

#### 9.14.5.1 IP Address

IP address for program to program communication

IPA:<Character String>

#### 9.14.5.2 I/O

IP address I/O port number for program to program communication

IOPN:<Integer>

#### 9.14.5.3 O

IP address Output port number for program to program communication. Required when FLDigi and auxiliary programs reside on the same host.

OPN:<Integer>

#### 9.14.5.4 Dual Port

Enable or disable the use of dual port io.

DP:<ENABLE | DISABLE>

#### 9.14.5.5 Enable Busy Channel

Enable or disable busy channel. Allows for the cooperative use of the frequency.

BUSY:<ENABLE | DISABLE>

#### 9.14.5.6 Continue After (sec)

The number of seconds the frequency must be clean clear before transmissions resume. Enabled by busy channel activation.

CONT:<Integer>

#### 9.14.5.7 KPSQL Attenuation

Controls the gain sensitivity of the KISS Power Squelch (KPSQL). The attenuation is measure in fractions. If the value is '4' then the gain is reduced to 1/4th. Maximum gain (sensitivity) requires a value of 1.

ATTEN:<Integer>

### 9.14.6 ARQ Sub Command

ARQ sub commands

#### 9.14.6.1 IP Address

IP address for program to program communication

IPA:<Character String>

#### 9.14.6.2 Port

IP address I/O port number for program to program communication

IOPN:<Integer>

### 9.14.7 XMLRPC Sub Command

#### 9.14.7.1 IP Address

IP address for program to program communication

IPA:<Character String>

#### 9.14.7.2 Port

IP address I/O port number for program to program communication

IOPN:<Integer>

### 9.14.8 Examples

| Structured                   | Dot Notation     |
|------------------------------|------------------|
| IO:<br>LOCK:ENABLE<br>END:   | IO.LOCK:ENABLE   |
| IO:<br>PORT:ARQ<br>END:      | IO.PORT:ARQ      |
| IO:<br>AX25D:DISABLE<br>END: | IO.AX25D:DISABLE |
| IO:<br>CSMA:DISABLE<br>END:  | IO.CSMA:DISABLE  |

| Structured                                    | Dot Notation          |
|---|-----------------------|
| IO:<br>KISS:<br>IPA:127.0.0.1<br>END:<br>END: | IO.KISS.IPA:127.0.0.1 |
| IO:<br>KISS:<br>IOPN:7342<br>END:<br>END:     | IO.KISS.IOPN:7342     |
| IO:<br>KISS:<br>OPN:7343<br>END:<br>END:      | IO.KISS.OPN:7343      |
| IO:<br>KISS:<br>DP:DISABLE<br>END:<br>END:    | IO.KISS.DP:DISABLE    |

|  |                      |
|--|----------------------|
| IO:<br>KISS:<br>BUSY:DISABLE<br>END:<br>END: | IO.KISS.BUSY:DISABLE |
| IO:<br>KISS:<br>CONT:3<br>END:<br>END:       | IO.KISS.CONT:3       |
| IO:<br>KISS:<br>ATTEN:2<br>END:<br>END:      | IO.KISS.ATTEN:2      |

| Structured                                   | Dot Notation         |
|--|----------------------|
| IO:<br>ARQ:<br>IPA:127.0.0.1<br>END:<br>END: | IO.ARQ.IPA:127.0.0.1 |
| IO:<br>ARQ:<br>IOPN:7322<br>END:<br>END:     | IO.ARQ.IOPN:7322     |

| Structured                                      | Dot Notation            |
|---|-------------------------|
| IO:<br>XMLRPC:<br>IPA:127.0.0.1<br>END:<br>END: | IO.XMLRPC.IPA:127.0.0.1 |
| IO:<br>XMLRPC:<br>IOPN:7362<br>END:<br>END:     | IO.XMLRPC.IOPN:7362     |

## 9.15 Misc NBEMS Commands

See [NBEMS \(flmsg / flwrap\) interface](#)

### 9.15.1 Enable

NBEMS data file interface

STATE:<ENABLE | DISABLE>

### 9.15.2 Open message folder

OPEN MSG:<ENABLE | DISABLE>

### 9.15.3 Open with flmsg

OPEN FLMSG:<ENABLE | DISABLE>

### 9.15.4 FLMsg executable path

PATH:<Character String>

Parameter: Directory and file name to the executable file location.

### 9.15.5 Open in browser

OPEN BRWSR:<ENABLE | DISABLE>

### 9.15.6 Timeout (secs)

TIMEOUT:<Character String>

### 9.15.7 Examples

| Structured  | Dot Notation                             |
|---|--|
| MISC NBEMS :<br>STATE:DISABLE<br>END :                    | MISC NBEMS.STATE:DISABLE                 |
| MISC NBEMS :<br>OPEN MSG:DISABLE<br>END :                 | MISC NBEMS.OPEN MSG:DISABLE              |
| MISC NBEMS :<br>OPEN FLMSG:ENABLE<br>END :                | MISC NBEMS.OPEN FLMSG:ENABLE             |
| MISC NBEMS :<br>OPEN BRWSR:DISABLE<br>END :               | MISC NBEMS.OPEN BRWSR:DISABLE            |
| MISC NBEMS :<br>TIMEOUT:4.0<br>END :                      | MISC NBEMS.TIMEOUT:4.0                   |
| MISC NBEMS :<br>PATH:c:/applications/nbems/flmsg<br>END : | MISC<br>PATH:c:/applications/nbems/flmsg |

## 9.16 ID

See [RSID / Video / ID Configuration](#)

### 9.16.1 RsID



#### 9.16.1.1 Notify only

Notify operator of RSID reception, disabling waterfall, frequency, and modem changes.

NOTIFY:<ENABLE | DISABLE>

#### 9.16.1.2 Searches passband

Search the entire band pass for RSID signatures (Enabled). Otherwise search center frequency +/- 200Hz

SRCH BP:<ENABLE | DISABLE>

#### 9.16.1.3 Mark prev freq/mode

Annotate on receive panel previous frequency and modem on RSID reception.

MARK PREV:<ENABLE | DISABLE>

#### 9.16.1.4 Disables detector

Ignore RSID tone sequence.

DETECTOR:<ENABLE | DISABLE>

#### 9.16.1.5 Disable alert dialog

Display Dialog box indicating reception of RSID tone sequence (Enabled).

ALRT DIALOG:<ENABLE | DISABLE>

#### 9.16.1.6 Retain tx freq lock

Retain frequency/waterfall transmit position on modem switches (enabled).

TX FREQ LOCK:<ENABLE | DISABLE>

#### 9.16.1.7 Disable freq change

Do not change frequency when RSID tones received.

FREQ CHANGE:<ENABLE | DISABLE>

#### 9.16.1.8 Allow errors

Adjust RSID reception accuracy at the expense of sensitivity.

ALLOW ERRORS:<Character String>

Parameters: Low, Medium, High

#### 9.16.1.9 Squelch open (sec)

On RSID reception disable squelch for n seconds.

SQL OPEN:<Integer>

Parameters: 0 - 20

#### 9.16.1.10 Pre-Signal Tone Seconds

On RSID reception disable squelch for n seconds.

PRETONE:<Real Number>

Parameters: 0.0 - 10.0 in 0.1/1.0 increments

#### 9.16.1.11 End of xmt ID

Transmit an RSID tone sequence at the end of the data transmission.

END XMT ID:<ENABLE | DISABLE>

### 9.16.2 Video

#### 9.16.2.1 Transmit mode ID

Enable/Disable the transmission of video text modem name.

ID MODE:<ENABLE | DISABLE>

#### 9.16.2.2 Transmit video text

Enable/Disable the transmission of custom video text.

VIDEO TXT:<ENABLE | DISABLE>

#### 9.16.2.3 Custom Video Text Input

Enable/Disable the transmission of custom video text.

TEXT INPUT:<Character String>

#### 9.16.2.4 Use small font

Enable/Disable the of small font for video text.

SMALL FONT:<ENABLE | DISABLE>

#### 9.16.2.5 500 Hz limit

Enable/Disable 500 Hz video width limit

500HZ:<ENABLE | DISABLE>

#### 9.16.2.6 Mode width limit

Enable/Disable video width limit based on modem bandwidth use.

WIDTH LIMIT:<ENABLE | DISABLE>

#### 9.16.2.7 Chars/Row:

Maximum number of video text character per "line"

CHAR ROW:<Integer>

Parameters: 1 - 8 Characters

### 9.16.3 CW

#### 9.16.3.1 Transmit callsign

Enable/Disable post callsign CW ID

TX CALL:<ENABLE | DISABLE>

#### 9.16.3.2 Speed (WPM):

Enable/Disable post callsign CW ID transmit speed

SPEED:<Integer>

Parameters: 15 - 40 WPM

#### NOTE:

Check the specifications imposed by your country on the maximum speed allowing for CW identifying.

### 9.16.4 Examples

| Structured  | Dot Notation                |
|---|-----------------------------|
| ID:<br>RSID:<br>NOTIFY:DISABLE<br>END:<br>END:      | ID.RSID.NOTIFY:DISABLE      |
| ID:<br>RSID:<br>SRCH BP:DISABLE<br>END:<br>END:     | ID.RSID.SRCH BP:DISABLE     |
| ID:<br>RSID:<br>MARK PREV:DISABLE<br>END:<br>END:   | ID.RSID.MARK PREV:DISABLE   |
| ID:<br>RSID:<br>DETECTOR:DISABLE<br>END:<br>END:    | ID.RSID.DETECTOR:DISABLE    |
| ID:<br>RSID:<br>ALRT DIALOG:DISABLE<br>END:<br>END: | ID.RSID.ALRT DIALOG:DISABLE |

|  |                              |
|--|------------------------------|
| ID:<br>RSID:<br>TX FREQ LOCK:DISABLE<br>END:<br>END: | ID.RSID.TX FREQ LOCK:DISABLE |
| ID:<br>RSID:<br>FREQ CHANGE:DISABLE<br>END:<br>END:  | ID.RSID.FREQ CHANGE:DISABLE  |
| ID:<br>RSID:<br>ALLOW ERRORS:Medium<br>END:<br>END:  | ID.RSID.ALLOW ERRORS:Medium  |
| ID:<br>RSID:<br>SQL OPEN:5<br>END:<br>END:           | ID.RSID.SQL OPEN:5           |
| ID:<br>RSID:<br>PRETONE:0.000<br>END:<br>END:        | ID.RSID.PRETONE:0.000        |
| ID:<br>RSID:<br>END XMT ID:DISABLE<br>END:<br>END:   | ID.RSID.END XMT ID:DISABLE   |

| Structured   | Dot Notation               |
|--|----------------------------|
| ID:<br>VIDEO:<br>ID MODE:DISABLE<br>END:<br>END:   | ID.VIDEO.ID MODE:DISABLE   |
| ID:<br>VIDEO:<br>VIDEO TXT:DISABLE<br>END:<br>END: | ID.VIDEO.VIDEO TXT:DISABLE |
| ID:<br>VIDEO:<br>TEXT INPUT:CQ<br>END:<br>END:     | ID.VIDEO.TEXT INPUT:CQ     |
| ID:<br>VIDEO:<br>SMALL FONT:ENABLE<br>END:<br>END: | ID.VIDEO.SMALL FONT:ENABLE |

|   |                             |
|---|-----------------------------|
| ID:<br>VIDEO:<br>500HZ:ENABLE<br>END:<br>END:       | ID.VIDEO.500HZ:ENABLE       |
| ID:<br>VIDEO:<br>WIDTH LIMIT:ENABLE<br>END:<br>END: | ID.VIDEO.WIDTH LIMIT:ENABLE |
| ID:<br>VIDEO:<br>CHAR ROW:8<br>END:<br>END:         | ID.VIDEO.CHAR ROW:8         |

| Structured                                    | Dot Notation          |
|---|-----------------------|
| ID:<br>CW:<br>TX CALL:DISABLE<br>END:<br>END: | ID.CW.TX CALL:DISABLE |
| ID:<br>CW:<br>SPEED:18<br>END:<br>END:        | ID.CW.SPEED:18        |

## 9.17 Load Macro Command

See [Macros](#)

The Macro command is a structured command only, dot notation is not supported.

```
MACRO, row, column, label
<Macro Content>
END:
```

Parameters:

- row - This is the macro bar number seen to the right of the macro buttons (1-4)
- column - The physical macro button position from left to right (1-12).
- label - The macro buttons label. See [Macro Display Symbols](#) for special label glyphs.

No syntax checking is performed for the macro content. User responsibility.

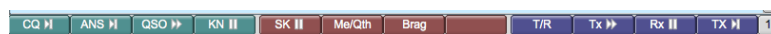


Figure 9.5: Macro Bar

The following example sets the first button in the first macro bar as seen above.

```
MACRO:1,1,"CQ @>|"
<TXRSID:on><TX>
cq cq cq de <MYCALL> <MYCALL> <MYCALL> pse k
```

```
<RX>
END:
```

## 9.18 Example Script

### NOTE:

This example script may report errors if used as is. Modification is required to meet your particular setup.

```
FLDIGI_CONFIG
# Fldigi Generated Config Script
# Created: Wed Jan 21 15:44:14 2015

FLDIGI:
  FREQ:1.407000000e+07
  MODE:USB
  WFHZ:1500
  RXID:DISABLE
  TXID:DISABLE
  SPOT:ENABLE
  REV:DISABLE
  AFC:DISABLE
  LOCK:DISABLE
  SQL:DISABLE
  KPSQL:DISABLE
  MODEM:BPSK31
END:
OPERATOR:
  CALLSIGN:kk5vd
  QTH:Robert
  NAME:"Madison, AL"
  LOC:EM64or
  ANT:"Invert 'V' @ 22FT AGL"
END:
AUDIO DEVICE:
  PA:ENABLE
  PA CAPTURE:2,"USB Audio CODEC"
  PA PLAYBACK:3,"USB Audio CODEC"
END:
AUDIO SETTINGS:
  CAPTURE:Auto
  PLAYBACK:Auto
  CONVERTER:"Medium Sinc Interpolator"
  RX PPM:17
  TX PPM:17
  TX OFFSET:0
END:
AUDIO RT CHANNEL:
  MONO AUDIO:DISABLE
  MODEM LR:DISABLE
  REV LR:DISABLE
  PTT RT CHAN:DISABLE
  QSK RT CHAN:DISABLE
  FSK RT CHAN:DISABLE
END:
AUDIO WAVE:
  SRATE:16000
END:
RIG HRDWR PTT:
  PTT RT CHAN:DISABLE
  SERIAL PORT:DISABLE
  DEVICE:/dev/ptt
  RTS:DISABLE
  RTSV:DISABLE
  DTR:DISABLE
  DTRV:DISABLE
```

```
PARALLEL:DISABLE
UHROUTER:DISABLE
START PTT DELAY:1
END PTT DELAY:1
INIT:
END:
RIGCAT:
STATE:ENABLE
DEV_PATH:/dev/tty.usbserial-RT071600
DESC_FILE:/Users/.fldigi/rigs/FT-817.xml
RETRIES:2
RETRY INT:500
WDELAY:5
IDELAY:0
BRATE:4800
SBITS:1
ECHO:DISABLE
TOGGLE RTS PTT:DISABLE
RESTORE:DISABLE
PTT_COMMAND:DISABLE
TOGGLE DTR PTT:DISABLE
TOGGLE RTS PTT:DISABLE
RTS 12V:DISABLE
DTR 12V:DISABLE
HRDWR FC:DISABLE
VSP:DISABLE
INIT:
END:
HAMLIB:
STATE:DISABLE
RIG:"Yaesu FT-817 (Beta) "
DEV_PATH:/dev/tty.usbserial-RT071600
RETRIES:2
RETRY INT:10
WDELAY:0
PWDELAY:5
BRATE:4800
SBITS:1
SBAND:"Rig mode"
PTT_COMMAND:DISABLE
DTR 12V:DISABLE
RTS 12V:DISABLE
HRDWR FC:DISABLE
SFTWR FC:DISABLE
ADV CONF:
INIT:
END:
XMLRPC RC:
STATE:DISABLE
BWDELAY:0.500
INIT:
END:
IO:
LOCK:ENABLE
PORT:ARQ
AX25D:DISABLE
CSMA:DISABLE
KISS:
  IPA:127.0.0.1
  IOPN:7342
  OPN:7343
  DP:DISABLE
  BUSY:DISABLE
  CONT:3
  ATTN:2
END:
ARQ:
  IPA:127.0.0.1
  IOPN:7322
END:
XMLRPC:
  IPA:127.0.0.1
  IOPN:7362
```

```
END:
END:
MISC NBEMS:
  STATE:ENABLE
  OPEN MSG:DISABLE
  OPEN FLMSG:ENABLE
  PATH:/Applications/flmsg-2.0.6AB.app/Contents/MacOS/flmsg
  OPEN BRWSR:ENABLE
  TIMEOUT:2.000
END:
ID:
  RSID:
    NOTIFY:DISABLE
    SRCH BP:DISABLE
    MARK PREV:ENABLE
    DETECTOR:DISABLE
    ALRT DIALOG:DISABLE
    TX FREQ LOCK:DISABLE
    FREQ CHANGE:DISABLE
    ALLOW ERRORS:Medium
    SQL OPEN:5
    PRETONE:0.000
    END XMT ID:DISABLE
  VIDEO:
    ID MODE:DISABLE
    VIDEO TXT:DISABLE
    TEXT INPUT:CQ
    SMALL FONT:ENABLE
    500HZ:ENABLE
    WIDTH LIMIT:ENABLE
    CHAR ROW:8
  END:
  CW:
    TX CALL:DISABLE
    SPEED:18
  END:
END:

MACRO:1,1,"CQ"
<TXRSID:on><TX>
cq cq cq de <MYCALL> <MYCALL> <MYCALL>/qrp pse k
<RX>
END:

MACRO:1,2,"ANSWER"
<TX><CALL> <CALL> de <MYCALL> <MYCALL> <MYCALL> kn
<RX>
END:
```

[Return to Top of Page](#)

[Return to Main Page](#)



# Chapter 10

## Command Line Switches

### 10.1 Linux Terminal

Usage:

```
fldigi [option...]
```

fldigi options:

```
--home-dir DIRECTORY
  Set the home directory to full pathname of DIRECTORY
  fldigi will put the file stores
  .fldigi.files, and .nbems.files
  in this directory
  The default is: /Users/robert/

--config-dir DIRECTORY
  Look for configuration files in DIRECTORY
  The default is: /Users/robert/.fldigi/

--enable-io-port <1|2> ARQ=1 KISS=2
  Select the active IO Port
  The default is: 1

--kiss-server-address HOSTNAME
  Set the KISS UDP server address
  The default is: 127.0.0.1

--kiss-server-port-io I/O PORT
  Set the KISS UDP server I/O port
  The default is: 7342

--kiss-server-port-o Output PORT
  Set the KISS UDP server output port
  The default is: 7343

--kiss-server-dual-port Dual Port Use (0=disable / 1=enable)
  Set the KISS UDP server dual port flag
  The default is: 0

--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

--flmsg-dir DIRECTORY
  Look for flmsg files in DIRECTORY
  The default is

--auto-dir DIRECTORY
  Look for wrap_auto_file files in DIRECTORY
```

The default is WRAP/auto/

```
--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

--cpu-speed-test
  Perform the CPU speed test, show results in the event log
  and possibly change options.

--noise
  Unhide controls for noise tests

--wfall-only
  Hide all controls but the waterfall

--debug-level LEVEL
  Set the event log verbosity

--debug-pskmail
  Enable logging for pskmail / arq events

--debug-audio
  Enable logging for sound-card events

--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: ArialMT:12

## 10.2 Windows Icon Properties

Setting command line parameters on all versions of Windows is accomplished by modifying the parameters of fldigi's desktop icon. Open the icon by right clicking on the icon. This is an example of modifying the font size:

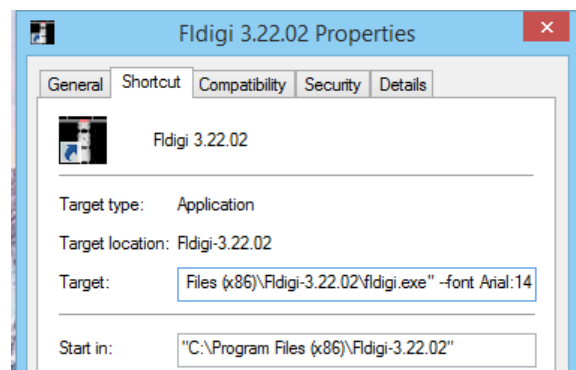


Figure 10.1: Windows 8.1 Properties

The before and after for changing the font size:

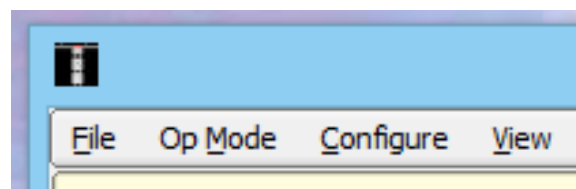


Figure 10.2: Windows 8.1 Original Font

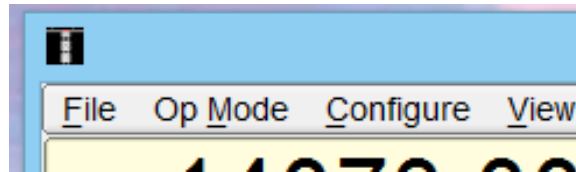


Figure 10.3: Windows 8.1 Larger Font

Note the space between the location of the executable and the command line parameter. Multiple parameters can be entered. Each parameter must be exactly as shown in the list and separated from its neighbors by at least 1 space. If the parameter has spaces in the value then it must be enclosed in quotations. For example:

```
-home-dir "C:\Documents and Settings\dave\fldigi-vhf.files"
```

For `-home-dir` I would also recommend changing the name of the launcher icon.

### 10.3 Mint Launcher Properties

Linux users may also change the command line parameters. Here is an example of modifying the fldigi icon that was added to a Mint-17 launch bar:

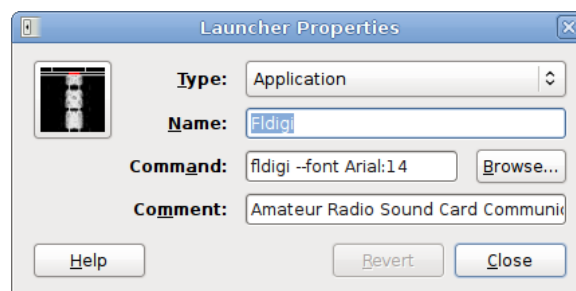


Figure 10.4: Mint Launch Bar Icon Properties

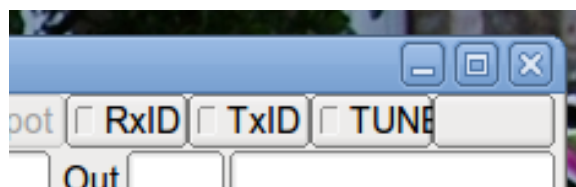


Figure 10.5: Mint Larger Font

Changing the font or font size can result in some loss of text on a control label. The size of a control and its label space do not change when the font is modified on the command line.

[Return to Top of Page](#)  
[Return to Main Page](#)



# Chapter 11

## License(s)

### 11.1 Copyright

Copyright ©

- 2006 through 2014 Dave Freese, W1HKJ
- 2007, 2008, 2009 Stelios Bounanos, M0GLD
- 2007, 2008, 2009 Leigh Klotz Jr., WA5ZNU
- 2007, 2008, 2009 Joe Veldhuis, N8FQ
- 2008, 2009 Stephane Fillod, F8CFE
- 2009 John Douyere, VK2ETA
- 2013 Remi Chateauneu, F4ECW
- 2013, 2014 Robert Stiles, KK5VD
- 2014, John Phelps - KL4YFD

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. 51 Franklin Street, Fifth Floor Boston, MA 02110-1301 USA.

[Return to Top of Page](#)  
[Return to Main Page](#)





## Chapter 12

# 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
- Stéphane Fillod - F8CFE
- John Douyere - VK2ETA
- Joe Veldhuis - N8FQ
- Chris Sylvain - KB3CS
- Gary Robinson - WB8ROL
- Stefan Fendt - DO2SMF
- John Phelps - KL4YFD
- Andrej Lajovic - S57LN
- Robert Stiles - KK5VD

Localization files:

|            |         |                     |        |
|------------|---------|---------------------|--------|
| Español    | Spanish | Pavel Milanes Costa | CO7WT  |
|            |         | Christian W. Correa | HK4QWC |
| Deutsch    | German  | Marc Richter        | DF2MR  |
| Français   | French  | Stéphane Fillod     | F8CFE  |
| Italiano   | Italian | Pierfrancesco Caci  | IK5PVX |
| Język      | Polish  | Roman Bagiński      | SP4JEU |
| Nederlands | Dutch   | Peter van der Post  | PA1POS |

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
- 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 book on digital signal processing on the internet. (<http://www.dspguide.com>)

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 John Green's public domain FFT code ([FFT for RISC for MAC](#)). The rewrite is in C++ and is implemented as a C++ template. 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:

| Call Name     | Call Name    | Call Name   | Call Name      |
|---------------|--------------|-------------|----------------|
| 4Z5ST Boris   | K3GAU David  | KU1T Zibi   | VA3DB Dianne   |
| AA0HW Chuck   | K4XTT Victor | KV9U Rick   | VE3IXI Dave    |
| AC7JN Dave    | K6KAR Kirk   | N0NB Nate   | VK2TMG Brett   |
| CT1DRB David  | K7BRK Chris  | N2AMG Rick  | VK4BDJ David   |
| CX7BF Walter  | K4RE Brian   | N4UM Tim    | W3NR Ed        |
| DF4OR Ekki    | K9AO Rick    | N4ZNV Mike  | W4ROS Ross     |
| DK1JBE Tom    | KB3FN Lynn   | N6WFL Jason | W6JVE Jim      |
| DL6XAZ Fred   | KD0AR Mike   | N8FQ Joe    | WA3VPZ Marshal |
| DL8FCL Walter | KD4O Phil    | NN8B Don    | WA4SXZ Rich    |

---

|             |             |                |              |
|-------------|-------------|----------------|--------------|
| G0UZP Paul  | KD8DKT Mike | NT1G Skip      | WB8ROL Gary  |
| G3TDJ Andy  | KE3Y Travis | OZ4KK Erik     | WD4FDW Steve |
| G6CKR Roger | KH6TY Skip  | PA0R Rein      | WD4FNY Bill  |
| G8SQH David | KL7NA Rob   | PA3GWH Richard | WU9Q Bob     |

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 chews. 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.

[Return to Top of Page](#)

[Return to Main Page](#)



## Chapter 13

# Using BPQ32 with FLDIGI

### 13.1 General Information, BPQ32

Setting up FLDIGI to use BPQ32 require the use of the KISS I/O port. see [Configure ARQ/KISS I/O](#) page for instructions.

BPQ32 is beyond the scope of this document. The following links are available for detailed information about the program and it's use with FLDIGI.

Documentation:

[BPQ32 Documents](#)

and for a specific reference to FLDIGI (search for keyword FLDigi)

[BPQ32 Documents \(FLDIGI\)](#)

Installation files:

[Program and Support Data](#)

**Note:**

The Macintosh Binary located on John's website are for PowerPC systems. Binaries for Intel systems are available on request by contacting Robert, KK5VD [kk5vd\(at\)yahoo\(dot\)com](mailto:kk5vd(at)yahoo(dot)com) or via John's email listed below.

BPQ32 is copyrighted © work of John Wiseman, G8BPQ. Any inquires about the software can be directed to:

[john\(dot\)wiseman\(at\)cantab\(dot\)net](mailto:john(dot)wiseman(at)cantab(dot)net)

[Return to Top of Page](#)

[Return to Main Page](#)

