

# FLAMP Users Manual

2.2

Generated by Doxygen 1.9.1



---

<b>1 FLAMP Users Manual - Version 2.2</b>	<b>1</b>
1.1 FLAMP Description	1
1.2 Initial Configuration	2
1.3 Configuration Scripts	2
1.4 Transmit Procedure	3
1.5 Transmit Mode Options	4
1.6 Transmit Interval Time Warning	6
1.7 Mode Synchronization: FLDIGI to/from FLAMP	6
1.8 Receive Procedure	7
1.9 Auto Save Received Files	9
1.10 Requesting Missing Blocks/Header	9
1.11 Sending Missing Blocks/Header Information	10
1.11.1 Single File Fills	11
1.11.2 Transmit Queue Fills	11
1.12 Allowable File Type Transfers	11
1.13 Relaying Data Between Stations (Digipeater)	11
1.13.1 Relay Operation	11
1.13.2 Save and Restore Relay Data	13
1.14 Transfer Receive Queue Item to the Transmit Queue.	13
1.15 Event Scheduler	14
1.15.1 Timed Event Scheduler	14
1.15.2 Event Time Format	15
1.15.3 Queue Load list	15
1.15.4 Hamcast Multimode	16
1.16 Debug Log	17
1.17 Command Line Time Table Switch	17
1.17.1 File/Table Generation	18
1.18 Command Line ARQ/XML_IO Server/Port Switches	20
1.19 Repeaters	21
1.19.1 Permission from the owner and tone filtering	21
1.19.2 Repeater timeouts and resets	21
1.19.3 Optional	21
1.19.4 Transmit delay	22
1.19.5 Setting RsID sensitivity level.	22
1.19.6 Using RsID	22
1.19.7 Multi path signals.	22
1.19.8 Over deviation	22
1.20 FLDIGI Version Issues	22
1.20.1 RSID Sensitivity Settings	22
1.20.2 8PSK Modem Availability	23
1.21 FLAMP Version 2.2.10+ Differences	23

---

<b>2 Configure and Queue File Scripting</b>	<b>25</b>
2.1 List of commands	25
2.2 Script Command Format	26
2.3 Script Class Types	27
2.4 Operating System Script Differences	28
2.5 Script Execution	28
2.6 Script Command Descriptions	29
2.6.1 FLAMP_CONFIG File ID Tag	29
2.6.2 '#' Pound sign.	29
2.6.3 Auto load Queue	29
2.6.4 Base Encoding	29
2.6.5 Transmit block size.	29
2.6.6 Callsign of the transmitting station	29
2.6.7 Callsign of the receiving stations	29
2.6.8 Clear Missing	30
2.6.9 Clear the transmit queue	30
2.6.10 Compression	30
2.6.11 Events, Forever	30
2.6.12 Events Times	30
2.6.13 Events Types	30
2.6.14 Enable/Disable Timed Events	31
2.6.15 Enable/Disable Events	31
2.6.16 Add file to TX Queue	31
2.6.17 Setting/Enabling/Disabling Hamcast Modems	31
2.6.18 Enable/Disable Hamcast Events	31
2.6.19 Header Transmit Repeat	32
2.6.20 Header Modem	32
2.6.21 Header Modem Enable/Disable	32
2.6.22 Information Field	32
2.6.23 Inhibit header modem on fills	32
2.6.24 Interval Timer	32
2.6.25 Load from TX directory	32
2.6.26 Modem	33
2.6.27 Path	33
2.6.28 Enable/Disable AMP protocol	33
2.6.29 Load Queue List	33
2.6.30 Reset/Clear Configure settings.	33
2.6.31 Receive interval time (seconds).	33
2.6.32 FLDIGI and FLAMP modem sync	34
2.6.33 Transmit interval time (minutes).	34
2.6.34 Transmit on Report	34
2.6.35 Unprotocol (plain text) Markers	34

---

2.6.36 File Delete Warning . . . . .	34
2.6.37 Data Transmit Repeat . . . . .	34
2.7 Script Error Reporting . . . . .	35
2.8 Example Program Level Script (PLS) . . . . .	35
2.9 Example Queue Level Script (QLS) . . . . .	36



## Chapter 1

# FLAMP Users Manual - Version 2.2



### 1.1 FLAMP Description

**Notice:** This version of FLAMP is not compatible with version 1.x.x

**FLAMP** is a program for AMP or Amateur Multicast Protocol. An FLAMP session will transmit one or more files with one or more iterations of the transmission. Each file is broken into blocks, each of which has a check sum. The receiving station saves the blocks that pass check sum. Successive transmissions will fill in the missing blocks provided that the new blocks pass the check sum. After the transmission sequence, the entire file is assembled and may be saved. “Fills” may be provided by retransmitting the entire file or by the sending station only sending the missing blocks. Start by downloading the current version of FLAMP from [wlhkj's website](#). Install the software as you would any of the NBEMS applications. You should also uninstall any older or development versions.

## 1.2 Initial Configuration

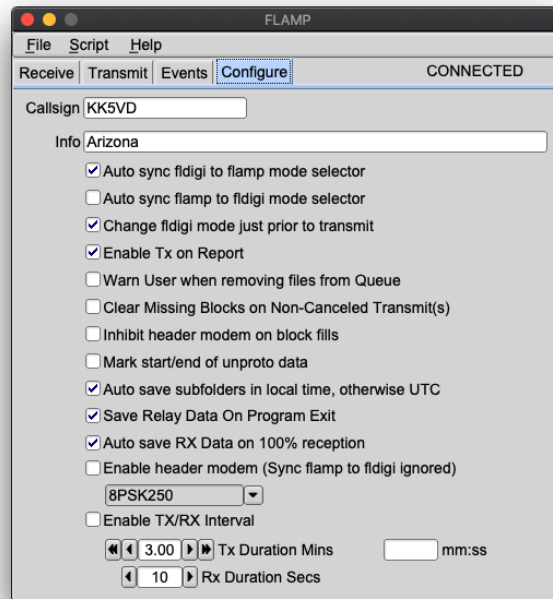


Figure 1.1 Configure Panel

When you open FLAMP you will see a screen with four tabs, Receive, Transmit, Events and Configure. Click on the Configure tab and fill in your call sign and relevant information as shown. There are other options which will be covered later.

## 1.3 Configuration Scripts

See [Configure and Queue File Scripting](#) for full details.



## 1.4 Transmit Procedure

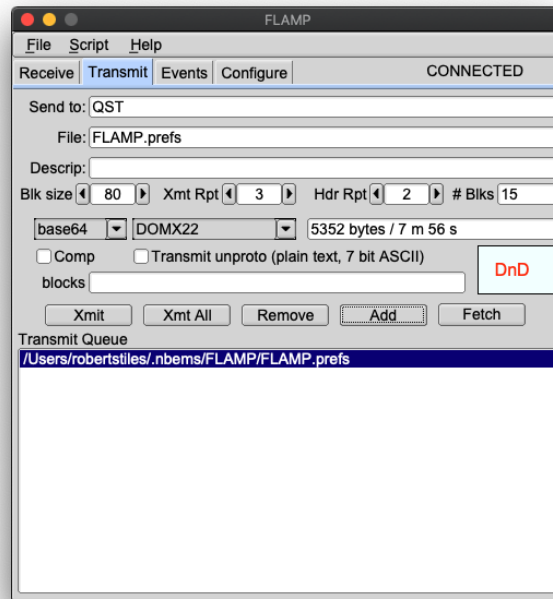


Figure 1.2 Transmit Panel

FLAMP divides a message file into blocks, each of which will be checked for errors when received. You can select the block size with the Blk size selector. The range is from 16 to 2048 in increments of 16.

FLAMP will repeat the transmission from 1 to 10 times with an option of repeating the header data independently. Select the appropriate number based on transmission mode (i.e. mode performance) and propagation conditions.

File identification is derived from multiple sources of information. These include date and time of the last file modification, compression, base encoding, and block size. Therefore, additional transmission sequences may be performed if necessary as long as these settings are not changed. If these settings are modified, FLAMP will treat the file as a new transmit queue item.

On the "Send to" line you may add any specific recipients for the file. The default is "QST" (general call to all amateur radio operators) which may not be appropriate in some countries.

To send a file, it has to be added to the queue. Click Add and then browse for the file or you can drag and drop the file from your desktop to the DnD icon (one or more files). FLAMP will handle any file format.

You have the option to compress the file with the Comp check box and select the level of encoding: base-64, base-128, or base-256. A warning dialog will be displayed if the compression is not selected and a file is added to the transmit queue that requires compression (images, all files with high bit set bytes, etc). Compression, base encoding, and description fields are a per file configurable item.

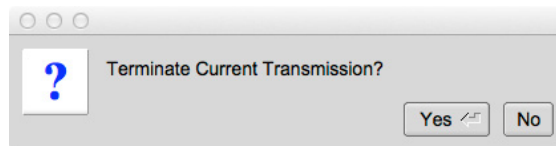
**Note:** In some circumstances depending on the contents of a file. FLAMP will automatically enable compression. Disabling compression on these files(s) may have unintended effects between FLAMP and FLDIGI. It is strongly suggested that compression remain enabled in these cases. The blocks line and Fetch button will be discussed later.

FLAMP provides some additional information. When you select the compression/no compression option and the mode you will see the file size, number of blocks and estimated transmission time. This is useful for determining

trade offs in terms of mode, repeats, compression and block size. Please note that the file size is the total transfer size (in bytes) and includes the AMP protocol overhead.

**Note:** When Tx/Rx Interval is enabled transmit time does not include the time between transmit periods.

When the file has been added to the Transmit Queue, you may add a file description in the Description line. This could be just a basic description of the file or have handling instructions. If sending just one file, you must highlight the file in the queue and press the 'Xmit' button to transmit. This will place the file name and description in the corresponding lines. If you have more than one file in the queue, you may transmit all of the files by clicking 'Xmt All'. Highlighting is not necessary.

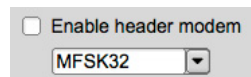


**Figure 1.3 Terminate Transmit Warning**

The file transfer can be terminated by selecting the cancel button. Once selecting cancel the user has the option to cancel or continue with the transfer termination by selecting "Yes" or "No" in the presented dialog box. Termination resets FLDIGI internal buffers and returns to a RX state.

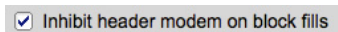
## 1.5 Transmit Mode Options

Located on the Configure panel.



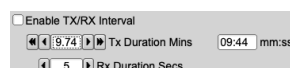
**Figure 1.4 Header Modem**

Enable Header Modem will allow the Critical Header data to be transmitted using slower more robust mode. The modem is user selectable. FLDIGI's Tx/Rx RSID will be activated prior to transmission.



**Figure 1.5 Inhibit Header Modem**

Inhibits header modem when block fills are being transmitted. Header modem use is resumed after the Fetch text field is cleared. If a "PREAMBLE" report is received this feature is ignored until after the block fill transmission is completed. At which time the internal flag is reset until another "PREAMBLE" report is received.



**Figure 1.6 Interval Timer**

Enable TX/RX Interval time allows the segmentation of data to be transmitted at specific intervals. Used for transmitting data over repeater systems that incorporate limited/restricted transmit time frames. The actual transmit time may be less than the selected time by 10-15% depending on the modem and block size used.

Change fldigi mode just prior to transmit

**Figure 1.7 Change Modem on Transmit**

Enabling TX/RX Interval requires Change FLDIGI mode just prior to transmit activation. This ensure the transmit duration is correct for the FLAMP selected modem (software controlled). This feature is user controlled when Enable TX/RX interval is disabled.

Clear Missing Blocks on Non-Canceled Transmit(s)

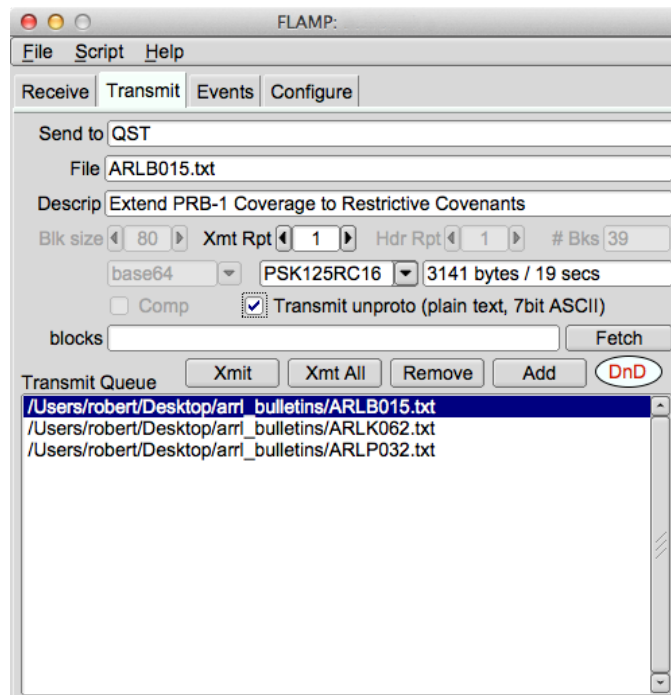
**Figure 1.8 Clear Missing Blocks**

Enabling Clear Missing Blocks on Transmit clears the Fetch text field of all missing blocks after a fill transmission has been conducted. Disabled leaves the Fetch text field unaltered.

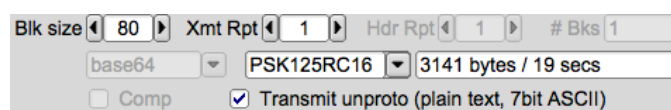
Transmit unproto (plain text, 7bit ASCII)

**Figure 1.9 Enable Unproto**

Enable Transmit Unproto allows for the transmission of plain text without FLAMP control codes and checksum values. This mode restricts the data to 7 bit ASCII. Any file that contains control characters and/or binary (upper bit set) will be removed and the remaining 7 bit ASCII data will be transmitted. FLAMP will not be able to receive this data as it lacks the necessary control information.



**Figure 1.10 Unproto Widget Deactivation**



**Figure 1.11 Unproto With Interval Enabled**

Enabling unproto disables (grayed out) some widgets in the Transmit panel.

If the unproto message contains embedded ARQ commands a character within the command will be substituted to prevent FLDIGI from processing them. A notice will be appended to the transmitted message indicating the change.

```
NOTICE: Command Character Substitution!
<_md> <-> ' _ ' = ' c '
```

## 1.6 Transmit Interval Time Warning

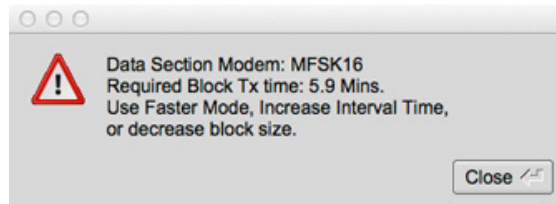


Figure 1.12 Block Tx Time Warning

When using Transmit/Receive Interval option there is a possibility of a block segment transmit time will exceed the interval transmit time. If this condition occurs a warning dialog box will be displayed listing the options required to correct the condition. FLAMP will not transmit until this condition is resolved.

## 1.7 Mode Synchronization: FLDIGI to/from FLAMP

FLAMP and FLDIGI have the capability to synchronize modes, i.e. you only have to set the mode on one or the other and the second will follow. This feature may also be disabled so that the mode being used with FLDIGI at a given moment can be independent of the mode being used for FLAMP file transfers.

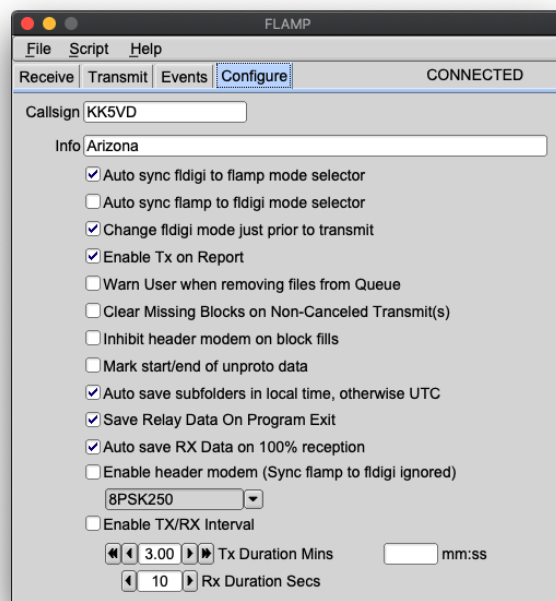


Figure 1.13 Configure Panel

On the Configure screen note the three top options as shown. The first option auto syncs FLDIGI to the mode selected on FLAMP. The second syncs FLAMP to the mode selected on FLDIGI. These two options are not mutually exclusive: if the first two options are selected, the mode selection is bilateral.

The third option will override the current FLDIGI mode (e.g. a keyboard mode such as Olivia 8/500) with the mode selected with FLAMP for file transfers (e.g. one of the MT63 modes). The mode change will take place at the time of transmission.

If none of the boxes is checked, there will be no synchronization.

Before transmitting, an FLAMP compatible version of FLDIGI must be running with the mode selected and frequency clear.

## 1.8 Receive Procedure

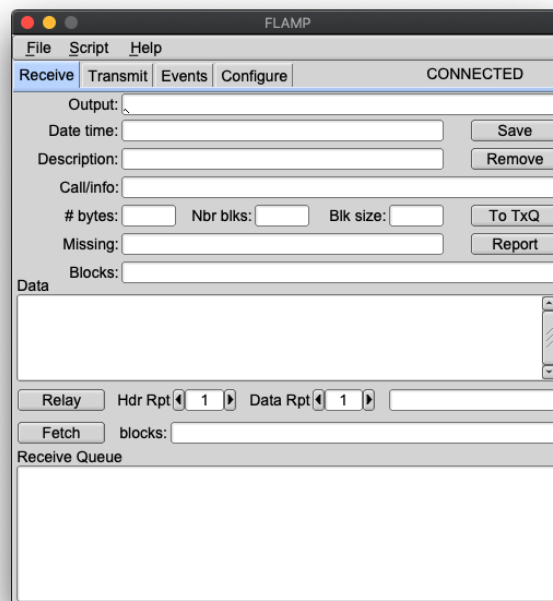


Figure 1.14 Receive Panel

The receiving stations must have FLAMP open with FLDIGI running. FLDIGI will not “wake up” FLAMP as it does with, for example, FLMSG. However, more recent versions of FLDIGI sports the option of auto executing FLAMP and other NBEMS tools. See FLDIGI user manual for details.

Note: Starting with version 2.2.10, FLAMP can be executed independently of FLDIGI and a connection will be establish once both programs are running.

The receive operation is hands-off. As the blocks are received the file information is filled in (file name, date/time, etc.) and the successfully received blocks will be depicted as dark blue rectangles on the progress bar. The blocks are positional. That is, a missing block will simply be a white space where the block would be if received correctly. On subsequent receptions that block will fill in when received correctly. The percent complete on the respective line also shows the state of each file being received.

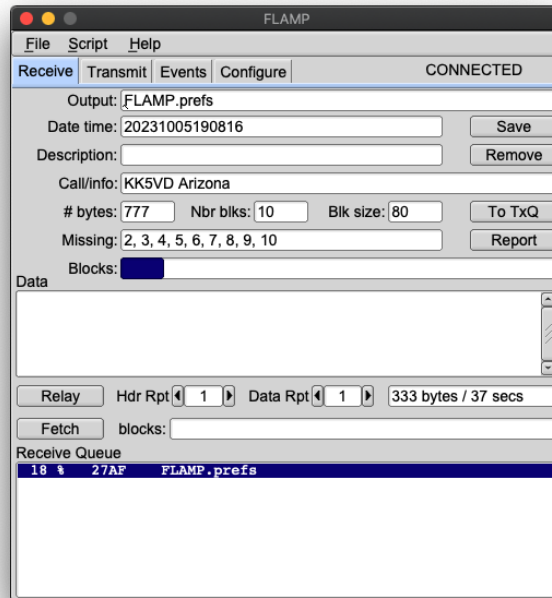


Figure 1.15 Receive Panel

You will also see block numbers showing up in the Missing line. When the complete file is received, that line should be blank. Received text files will appear in complete form and readable in the Data panel.

To save a file, click Save with the file highlighted. This will place the file in the rx folder in the default directory:

Operating System	Folder/Directory
Windows †	C:\Documents & Settings\User\flamp
Windows †	C:\Users\User_login_name\flamp.files
Linux	/home/user/.flamp
Macintosh	/Users/User_Login_Name/.flamp

**Note:** † Depending on Windows version or install configuration.

The received file size is the actual file size, i.e. the actual bytes being transported (per repeat) less all of the Amp header information.

The Send to information does not appear on the FLAMP screen. It will be seen in the FLDIGI RX panel as shown below:

Data segments encoded in base64.

```
NOBODY DE SOMCAL
<PROG 18 E9CE>{0EE2}FLAMP x.x.x
<FILE 28 13C7>{0EE2}20130323070339:Fox.txt
<ID 41 EEBC>{0EE2}SOMCAL Tesla's Lab Colorado Springs
<SIZE 14 2F49>{0EE2}221 5 48
```

```
<DESC 15 856E>{0EE2}Lazy Dogs
<DATA 56 B927>{0EE2:1}[b64:start]AUxaTUEAAAggXQAAAAQAEAxB/TEW1lGk3J5mK
<DATA 56 BAFA>{0EE2:2}pYEOyC+U2nmNwaGCvVfVmqBX/av7rhbtp4H4xoY39GsVdk5W
<DATA 56 24A2>{0EE2:3}ZmH3Jm/CULrCd1B39rhWJUUCcE0jXPg113yWPIV2q4fQDgaz
<DATA 56 87D0>{0EE2:4}FXD/IqYKt7x07GamZsVFZORzk/TLgX4vBXqTc86KG07kCFhJ
<DATA 37 8435>{0EE2:5}Wpmo3mbURumL0Xj8cIE[b64:end]
<CNTL 10 8E8D>{0EE2:EOF}
<CNTL 10 2E81>{0EE2:EOT}
NOBODY DE SOMCAL K
```

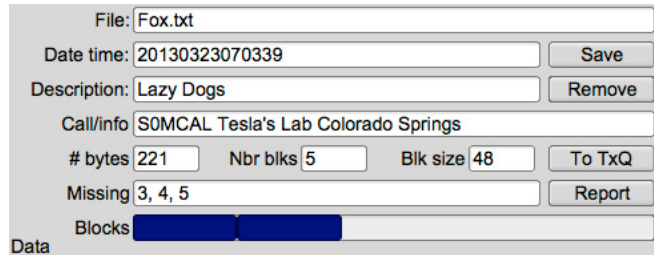


Figure 1.16 Receive Panel (Cropped)

This image shows a partially completed file. Multiple blocks were missed in the first pass and they will (hopefully) be filled in during a repeated transmission.

Note that the blocks with errors show up in the Missing line. In this case, blocks 3,4,5 were received with errors. The Receive Queue panel will show the percent received. Nothing will appear in the Data panel.

## 1.9 Auto Save Received Files



Figure 1.17 Configure Panel - Auto Save Option

When selected, all files that are received 100% are save in FLAMP's "rx/" directory automatically. The saved file will be place in a sub-directory using a UTC date signature "rx/<DD\_MM\_YYYY\_UTC>". This date corresponds to when the file was saved.

## 1.10 Requesting Missing Blocks/Header

If a transmission sequence has been completed and one or more stations have missing blocks or header information, the receiving operators may request the fill-ins using the Report button.

The Report button adds a text stream to the FLDIGI Tx buffer, but does not start the FLDIGI transmit processing (unless this option is enabled in the config panel, "Enable TX on Report"). The recipient sends the Tx buffer using the T/R command when it is his or her turn to make the report. The report format in FLDIGI transmit panel will look like this:

```
DE SOMCAL File : Fox.txt
<MISSING 12 8793>{0EE2}1 2 3
DE SOMCAL K
```

In the event the header data is missing:

Missing file name (<FILE tag).

```
DE SOMCAL File : Unassigned
<MISSING 14 FB42>{0EE2}PREAMBLE
DE SOMCAL K
```

Missing file block count size or Station ID (<SIZE or <ID tags).

```
DE SOMCAL File : Fox.txt
<MISSING 14 FB42>{0EE2}PREAMBLE
DE SOMCAL K
```

If there were no missing data, a report of receipt confirmation will be sent. For our example it would be of the form:

```
DE SOMCAL File : Fox.txt
<MISSING 15 107D>{0EE2}CONFIRMED
DE SOMCAL K
```

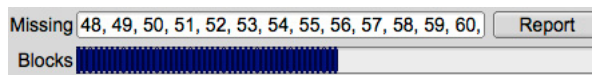
The important word is "CONFIRMED." The "<MISSING" is the line descriptor. The "15" the length of the data block "{0EE2}CONFIRMED." The "107D" is the checksum associated with the data block, and the "0EE2" is the checksum of the file being processes (may be multiple files being reported). This is necessary to ensure that there are no false reports received.

Enable Tx on Report

**Figure 1.18 Transmit on Report**

Enable Tx on Report located in the Configuration panel allows the user to initiate transmit from FLAMP when mouse clicking on the "Report" button. When disabled transmit function must be initiated from FLDIGI's T/R button.

## 1.11 Sending Missing Blocks/Header Information



**Figure 1.19 Blocks Received Progress**

For small files, another transmission of the complete file should suffice for fill-ins. In the case of a large file where only a few blocks are missing, FLAMP permits the sending station to just transmit the missing blocks.

When the sender has received all of the block missing reports he or she then retrieves the reports from FLDIGI by pressing the Fetch button. Any missing reports for the selected transmit queue item will be display in the blocks field.

FLAMP parses all of the received data from one or more reporting stations. It combines them, discards duplicates and sorts the missing blocks numerically. It then updates the "blocks" entry control with the requested blocks. Missing header information assigns '0' to the missing block list, triggering a retransmission of the header information.



### 1.11.1 Single File Fills

Single file fills are executed by numerical data content in the 'blocks' field in combination with 'Xmit' button selection.

### 1.11.2 Transmit Queue Fills

Sending fills for all queued item(s) is accomplished by holding the 'Shift Key' down and selecting 'Xmt All' button. Queued item(s) with missing reports will be transmitted, all others will be skipped. Pressing 'Xmt All' without the 'Shift Key' transmits the entire content of the queue.

## 1.12 Allowable File Type Transfers

FLAMP will send any file format. For example, ICS forms from FLMSG may be sent as well as spreadsheets in CSV format or binary data such as image files.

In the case of binary files, a message will be displayed upon receipt as shown below.

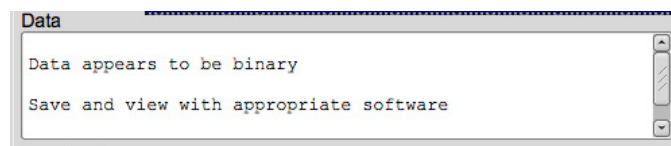


Figure 1.20 Received binary

## 1.13 Relaying Data Between Stations (Digipeater)



Figure 1.21 Relaying Data (Digipeater)

### 1.13.1 Relay Operation

Under less than stellar propagation conditions there exists the possibility of stations being unable to receive the data completely. This feature allows a receiving station to retransmit verbatim copies of the data to other receiving stations in the attempt to fill missing header/data sections. As with transmitting fills from the TX queue the Relay function acts on the missing reports from other receiving stations. Pressing the Fetch button will transfer a list of missing blocks associated with the selected received queue item into the relay blocks field. This field is user editable; block numbers can be entered directly without the need to press the Fetch button. If the Blocks field is blank, the entire file will be retransmitted. It is not necessary for the received file to contain 100% of the data. Only checksum verified data is available for retransmission.

To enter missing block numbers manually, the following numerical values are used to indicate what part of the data is to be relayed. Missing header data is represented by the number '0'. Block numbers 1 to n (the maximum number of blocks for the given file) represent the <DATA 34 2E43>{hash:block number n} AMP-2 tagged data.

Retransmission of the data format is limited to how it was received. The user has no control over the blocks size, compression, base conversion, etc. In effect it's a manually initiated digipeater. Interval Timer can be enabled/disabled and time duration altered to the users requirements. Header Modem is not available for relay transmits. The modem used to transmit the data is the same as the one selected in the transmit panel.

Once FLAMP has exited or the files are removed from the receive queue, digipeat option for those files are lost. See: "Save and Restore Relay Data" before removing any file(s) from the receive panel.

Two counters are provided for the repeated transmission of multiple 'n' copies of either the header or data sections. Estimated transmit time is displayed to the right of "Data Rpt" counter.

Status	Hash	File Name
100 %	7E9C	EMCOM.txt
100 %	2836	SGARN.htm
100 %	21F6	K.txt
100 %	FD4B	SGARN2.txt
100 %	6407	X.txt
0 %	0F3C	Unassigned
75 %	8F5D	P.txt

Figure 1.22 Receive Queue List

When sending fills with "unassigned" file names, it might be difficult to differentiate among them. The second column of the receive queue list contains a hexadecimal number. This number is the hash value assigned to every file transmitted. A match can be found by scrolling/searching in the FLDIGI receive window for MISSING reports. The hash value is the hexadecimal number enclosed in the curly brackets "{}" and displayed in bold text. See below.

#### Examples:

```
<MISSING 12 8793>{8F5D}1 2 3
<SIZE 14 2F49>{8F5D}, ...
<DATA 56 B927>{8F5D:1}, ...
```

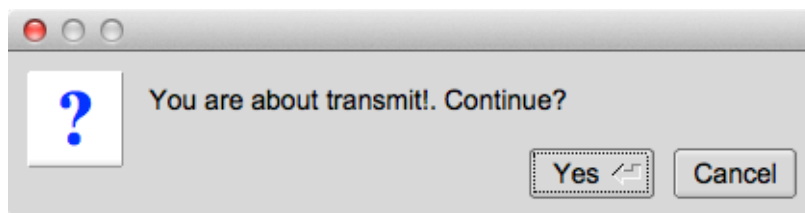


Figure 1.23 Relay Transmit Warning

Pressing the Relay button brings up a dialog box warning of an impending transmit. Pressing 'Yes' starts the transmit process.



Figure 1.24 Canceling relay Transmit

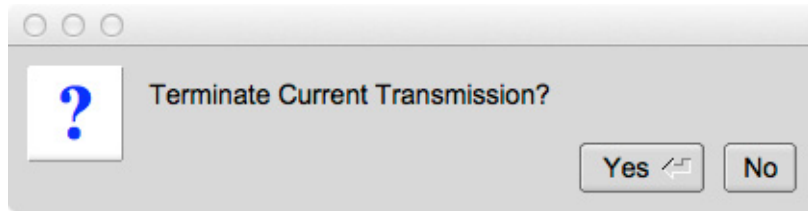


Figure 1.25 Cancel Warning

While FLDIGI is transmitting the "Relay" button changes to "Cancel". Pressing the "Cancel" button immediately terminates the transmission on validation via a Warning Dialog box.

### 1.13.2 Save and Restore Relay Data

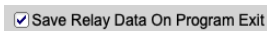


Figure 1.26 Configure Panel - Relay Save On Exit Checkbox

Selecting this checkbox will cause FLAMP on program exit to save all AMP encoded copies of RX data listed in the receive panel.

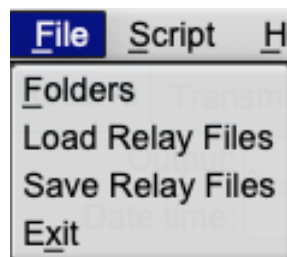


Figure 1.27 Relay Load/Save Menu Items

Loading and saving relay (AMP encoded) files are possible at anytime using the options under the "File" menu.

The file name format for the saved files starts with the HASH value of the file as received originally, and followed by either "Unassigned" or the "file name" if the relevant header data was received.

Example Saved Relay file names: 3F28\_Net\_Message.txt, AFE1\_Unassigned

## 1.14 Transfer Receive Queue Item to the Transmit Queue.

1. Select the receive queue item.
2. Click on "To TxQ" button.

The content of the file, file name, description, and block size will be transferred to the TxQ after the user is requested to save the file. Software will only allow 100 percent completed files to be transferred. Mouse clicking on the Transmit tab reveals the transferred queue item. Retransmission of the file in it's complete form must be conducted (no missing blocks fills). Use of this option does not allow for digipeater functionality. For digipeat operations see [Relaying Data Between Stations \(Digipeater\)](#)

## 1.15 Event Scheduler

### 1.15.1 Timed Event Scheduler

FLAMP can transmit the queued files on a specified interval, on a timed schedule or continuously. This is configured via the Events tab.

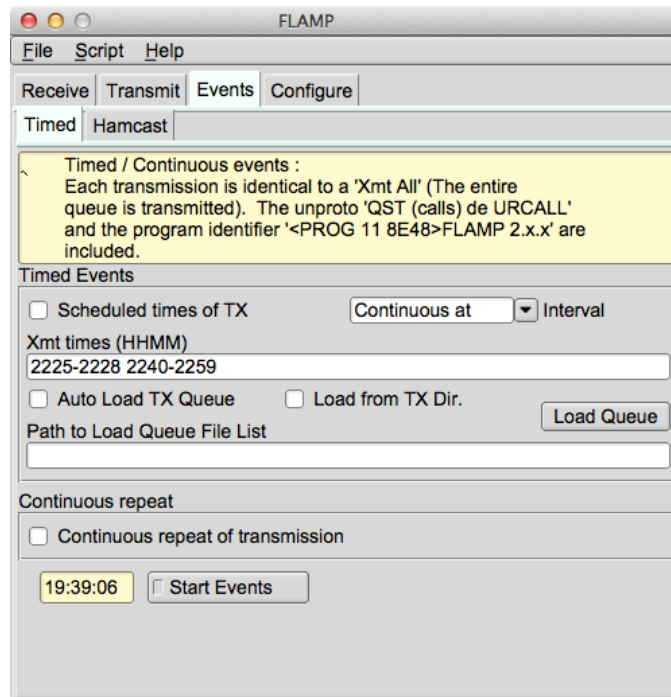


Figure 1.28 Event Panel

Timed Events can be scheduled to occur on specified minute / hour values via the "Retransmit interval" combo as shown:

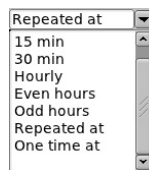


Figure 1.29 Event Time Selection

When executing the "One time at" event schedule, the time entries in the Xmt time (HHMM) field are removed as each event time is executed. When all times are exhausted the Start/Stop Events button is disabled (Stopped).

When "Repeated at" is selected then the event times are as delineated in the Xmt times (HHMM) control block. Other options include various repeat intervals (5, 15, 30 minutes; hourly, even hours, odd hours) and one time repeat at a specified time.

"Continuous at" event transmits the content of the transmit queue within a given window of time. Entering the start time followed by a hyphen and the end time (0100-0200, no spaces between numbers and the hyphen). The end time is the last time the queue will be sent to FLDIGI, not the end of transmission. Transmit will continue until the buffer in FLDIGI has been processed.

Selecting Auto Load TX Queue on "Continuous at" Event checkbox has two functions. First is to automatically load the transmit queue (at program execution) with the files listed in a load queue file. The path information to the queue

file can either be dragged from a desktop icon into the 'Path to Load Queue File List' text input box or typed in by hand. The second is to load/reload the same files into to TX queue by replacement when a 'Continuous at' event occurs. Thus keeping the queue updated in the event these file have changed.

Pressing 'Load Queue' button will load the files listed in the queue list by replacement.

Continuous repeat works exactly as you would expect. The timer is based on 1-second intervals, which will result in a ~1 second silent period between transmissions.

**Note:** Software will prevent the reloading of the Transmit queue via the Event panel if FLDIGI/FLAMP is transmitting. It's possible to miss a later event if the previous data is still being transmitted.

### 1.15.2 Event Time Format

HHMM<space>HHMM<space>,... or HHMM<comma>HHMM<comma>,...

Example:

1030 1230  
or  
1030,1230

When using 'Continuous at' Event

HHMM-HHMM<space>HHMM-HHMM<space>,... or HHMM-HHMM<comma>HHMM-HHMM<comma>,...

To the left of the hyphen is the start time, to the right is the end time. No spaces between the numbers and the hyphen.

Example:

1030-1230 1245-1300  
or  
1030-1230,1245-1300

### 1.15.3 Queue Load list

See [Configure and Queue File Scripting](#) for full details.

### 1.15.4 Hamcast Multimode

The primary purpose for hamcast resided in the fact that a particular modem may function better under the conditions at the receiving end. Another reason might be that the receiving station software/hardware may not support all of the available modems that FLDIGI supports. Selecting various modems increases the possibility of a successfully reception by as many stations as possible.

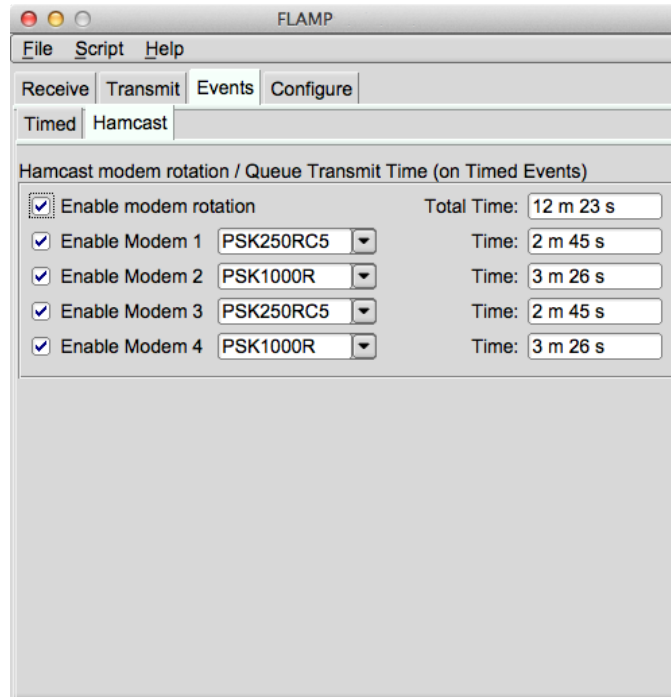


Figure 1.30 Event Panel

Hamcast activation facilitates the transmission of queued files using a rotating modems selected by the user. To illustrate this effect a 'Continuous at' is scheduled to occur between the hours of 0533 to 0539 hrs UTC. On the triggering of this event "Modem 1" (if enabled) will be selected and used to transmit the contents of the transmit queue. Once completed the second modem will then be selected to transmit the queue again. This cycling continues through all of the enabled modems until the event end is reached.

On multiple single events each enabled modem is assigned to each event in sequence.

Example:

If you have multiple entries in the Xmt times text field. Such as, 1000 1010 1035 1500 1510.

The queue content will be transmitted using the modem listed in the below table in association to the event time. Assuming all 4 hamcast modems are enabled.

Event time	Modem Used
1000	Hamcast Modem 1
1010	Hamcast Modem 2
1035	Hamcast Modem 3
1500	Hamcast Modem 4
1510	Hamcast Modem 1

The time information to the right of the modems indicates the amount of time each modem takes to transmit the entire transmit queue. Total time indicates the amount of time it takes to transmit the entire queue times the number of selected modems. This information might be useful in setting up the timed events in the Events Timed panel.

Times listed are the total transmit time for the selected modem(s) only. Overall times are effected by interval timer, header modem, RSID, video text, carrier, and CW ID's generated by FLDIGI and/or FLAMP.

**Note 1:** Deselecting all modems automatically disables "Enable modem rotation". To Enable "Enable Modem rotation" one or more "Enable modem n" must be selected first.

**Note 2:** Hamcast only operates from timed events. Manual transmission of the queue content defaults to the Transmit panel modem and/or selected header modem if enabled.

**Note 3:** Interval timer and header modem are available with hamcast operations.

**Note 4:** Events will not start unless there is a file in the transmit queue.

## 1.16 Debug Log

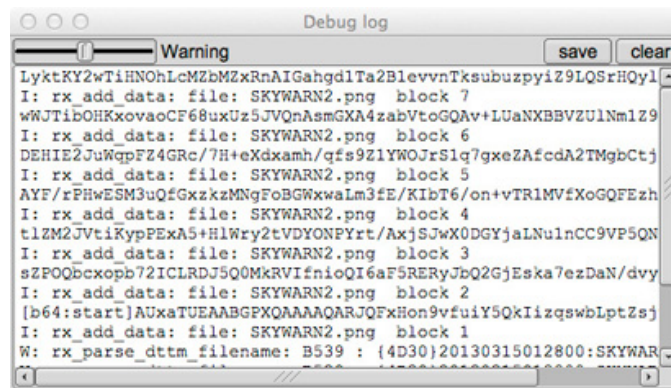


Figure 1.31 Debug Log

A Debug Log has been added to FLAMP for debugging purposes. It will capture events such as failed headers, data blocks etc. To access the Event Log, click Help – Events on the FLAMP screen. The slider may be used to select 5 levels of event capture: Quiet, Error, Warning, Info (default) and Debug. The log may be saved. The saved log will be placed in the FLAMP folder.

## 1.17 Command Line Time Table Switch

### FOR DEVELOPERS ONLY

Command line switch `--time-table` repurposing FLAMP for creating mode time tables in 'C/C++' languages formatted syntax.

**Note 1:** In order to use this FLDIGI must have the XMLRPC command 'main.get\_char\_timing' available.

**Note 2:** Ensure RSID, CWID, TONE transmit, Video Text, etc. are disabled in FLDIGI, and Interval Timer, Header Modem are disabled in FLAMP. The waterfall in FLDIGI will be disabled to allow for high speed processing and will pause during the measurement process.

Two primary functions are available.

- Selecting 'XMIT'  
on the Transmit panel will initiate a time table generation starting from the current selected mode and continue to the end of the available modes. Selecting a file in the transmit queue will be required for the operation to commence.
- Selecting 'XMT ALL'  
Initiates a table generation of all modes available to FLAMP.

During the measurement process the software will print to the Debug Log the following information.

```
D: create_tx_table: Table Gen filename time_table-01.txt
D: create_tx_table: Processing Modem PSK1000RC2
D: create_tx_table: Mode PSK1000RC2 overhead=2.048000
D: create_tx_table: Proc Char = 0, 0 :
D: create_tx_table: Character '0 (0)' Transmit time in seconds=0.011000
D: create_tx_table: Proc Char = 1, 1 :
D: create_tx_table: Character '1 (1)' Transmit time in seconds=0.011000
D: create_tx_table: Proc Char = 2, 2 :
D: create_tx_table: Character '2 (2)' Transmit time in seconds=0.011000
D: create_tx_table: Proc Char = 3, 3 :
D: create_tx_table: Character '3 (3)' Transmit time in seconds=0.011000
D: create_tx_table: Proc Char = 4, 4 :
D: create_tx_table: Character '4 (4)' Transmit time in seconds=0.011000
D: create_tx_table: Proc Char = 5, 5 :
D: create_tx_table: Character '5 (5)' Transmit time in seconds=0.011000
D: create_tx_table: Proc Char = 6, 6 :
D: create_tx_table: Character '6 (6)' Transmit time in seconds=0.011000
, ...
```

The initial first\_measurement and second\_measurement measurements are necessary to determine the data segment from the preamble, postamble, and interleave transmit times.

Each transmit encompasses:

$$\text{total\_time} = \text{overhead} + (\text{character}_1 \text{ time} + \text{character}_2 \text{ time} + \dots + \text{character}_n \text{ time})$$

The process is time consuming, expect the complete mode table generation to take 1 or more hours.

### 1.17.1 File/Table Generation

Measurements are stored in a 'C/C++' data structure and saved in a text file.

File names are internally created in a numbered sequence 'time\_table-[00-99].txt'.

Example: time\_table-01.txt

and will be located in the current working directory.

Stopping and starting the process creates another time table file to prevent the previous file from being over written. A maximum of 100 files are possible. If this number is exceeded deletion of these file(s) will be required.

**File contents:**



```

typedef struct {
    char *mode_name;
    float scale;
    float overhead;
    float table[256];
} MODE_TIME_TABLE;

MODE_TIME_TABLE mode_time_table[] = {
    {
        (char *) "DOMX22", 1.0, 1.346757,
        {
            0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,
            0.000000, 0.000000, 0.139320, 0.139320, 0.139320, 0.092880, 0.139320, 0.139320,
            0.139320, 0.139320, 0.139320, 0.139320, 0.139320, 0.139320, 0.139320, 0.139320,
            ,...
            0.139320, 0.139320, 0.139320, 0.139320, 0.139320, 0.139320, 0.139320, 0.139320,
            0.139320, 0.139320, 0.139320, 0.139320, 0.139320, 0.139320, 0.139320, 0.000000
        }
    }, ...
};

```

#### 'C/C++' Structure are as follows:

- char \*mode\_name;  
The mode name for the table.
- float scale;  
used to adjust the overall table when calculating transmit time.
- float overhead;  
This is the time the mode takes to transmit preamble and postamble (in seconds).
- float table[256];  
The table of data used to calculate the transmit time. Each entry is the amount of time it take to transmit a single byte of information (in seconds)

After the table has been integrated into the time\_table.cxx source code and recompiled. Re-executing FLAMP without --time-table command line switch enables a transmit time indicator when the Debug Log window is set to 'Debug' level. Monitoring the transmit times can be used to determine the accuracy of the table.

The following output should be seen after a transmit period:

```
D: ztimer: TX time [mm:ss]: 01:22 [m:82 c:82.591 r:0.992840] PSK1000RC2
```

In the above example a file of 9460 bytes was send using modem PSK1000RC2. The first number [mm:ss] is based on the function time() (located in ztimer() function) which has a resolution of one second. Thus, an error of at least  $\pm 0.9$  of a second are possible.

#### Time Measurement Values:

- 'm:'  
Measured value in seconds representation of [mm:ss].
- 'c:'  
Calculated value based on the table contents.
- 'r:'  
The ratio assigned to 'float scale;' in the C/C++ structure MODE\_TIME\_TABLE.

Note: Unless the value of 'm:' is significantly larger then 'c:' it's advised the value of 'float scale;' not be altered. Since it does not scale well when transmitting shorter or longer character lengths. However, if you must because of the amount of error. Selecting a file with a near 4 minute transmit period is preferred. Because, the whole intent of this table is to ensure we don't excede the timeout period of Repeaters when [Interval timer](#) is used.

## 1.18 Command Line ARQ/XML\_IO Server/Port Switches

User selectable ARQ/XML\_IO Server/Port Command line switches. In the event the user requires a second or more paired FLDIGI/FLAMP process for concurrent operation on another Transmitter/Receiver. User will need to assign a common server address/port number between each paired FLDIGI/FLAMP programs. It is recommend setting the PORT parameters only for each ARQ/XML\_IO interprocess communication switch.

```
--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

--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
```

This bash script is an example to dedicate a FLDIGI/FLAMP pair for VHF work. User will be required to set the Audio in/out ports in the FLDIGI configuration panel to the radio designated for VHF work.

### For OS' supporting the BASH command line program.

```
Cut and Paste then Modify (if needed)
----- Start - Dont include this line
#!/bin/bash

CMDS="--title VHF --arq-server-address 127.0.0.1 --XMLRPC-server-port 7363"

FLDIGI $CMDS &
sleep 4
FLAMP $CMDS &

----- End - Dont include this line
```

### Example

The text file used in the examples consists of 40 iterations of "The quick brown fox,..." as shown below:

```
1. The quick brown fox jumps over the lazy dog
2. The quick brown fox jumps over the lazy dog
...
40. The quick brown fox jumps over the lazy dog
```

A copy of the FLDIGI receive panel with the information entered as shown in the screen captures is below.

```
NOBODY DE SOMCAL
<PROG 18 E9CE>{0EE2}FLAMP x.x.x
<FILE 28 13C7>{0EE2}20130323070339:Fox.txt
<ID 41 EEBC>{0EE2}SOMCAL Tesla's Lab Colorado Springs
<SIZE 14 2F49>{0EE2}221 5 48
<DESC 15 856E>{0EE2}Lazy Dogs
<DATA 56 B927>{0EE2:1} [b64:start]AUxaTUEAAAggXQAAAAQAEAxB/TEWllgk3J5mK
<DATA 56 Bafa>{0EE2:2}pYEOyC+U2nmNwaGCVvFVmqBX/av7rhbtp4H4xoY39GsVdk5W
<DATA 56 24A2>{0EE2:3}ZmH3Jm/CULrCd1B39rhWJUUCcE0jXPg113yWPIV2q4fQDgaz
<DATA 56 87D0>{0EE2:4}FXD/IqYKt7x07GamZsVFZORzk/TLgX4vBXqTc86KGO7kCFhJ
<DATA 37 8435>{0EE2:5}Wpmo3mbURumL0Xj8cIE [b64:end]
<CNTL 10 8E8D>{0EE2:EOF}
<CNTL 10 2E81>{0EE2:EOT}
NOBODY DE SOMCAL K
```

## 1.19 Repeaters

The following guide lines should be employed to increase the likely hood of receiving data correctly over a repeater system.

When transmitting data over a repeater one has to contend with a few issues that are not present in simplex communications.

- Repeater timeouts and resets.
- Transmit delay.
- Over deviation
- Other issues not related to repeaters
- Setting RsID sensitivity level.
- Using RsID
- Multi path signals.
- Permission of the repeater owner

### 1.19.1 Permission from the owner and tone filtering

1. The repeater may be for voice use only.
2. The repeater might employ tone filtering to prevent DTMF and side tone from passing to the transmitter. Digital communications works better if this is turned off.
3. Owner may only allow certain times for digital use.
4. Above all, be respectful of the owners wishes.

### 1.19.2 Repeater timeouts and resets

From the FLAMP configuration panel.

1. Enable Interval Timer.
2. Select a receive duration suitable for the repeater in use. There is a minimal amount of time the repeater takes to reset for transmission to reoccur. Listen to the repeater while voice communications are occurring and do some time measurements. Add 1 or 2 additional seconds as a buffer.
3. Select a transmit duration that does not exceed the repeater timeout period. Setting to one minute less is recommended due to the possible use of RSID, CWID, Video Text, and 'Pre-Signal Tone' overhead generated by FLDIGI.

### 1.19.3 Optional

Enable Header modem.

### 1.19.4 Transmit delay

From the FLDIGI configuration panel.

Menu: Configure(menu)->IDs(menu)->ID(tab)->RsID(tab).

Adjust the 'Pre-Signal Tone' to a duration long enough to overcome the transmit delay of the repeater. You will have to experiment until the right value is known.

Not setting this correctly could cause the truncation of the beginning part of your transmission. Resulting in missed RSID's or data.

### 1.19.5 Setting RsID sensitivity level.

From the FLDIGI configuration panel.

Menu: Configure(menu)->IDs(menu)->ID(tab)->RsID(tab).

Set the RxID 'Sensitivity' slider above 'Pre-Signal Tone' to the left until the value of '2' is displayed. See Using RsID.

### 1.19.6 Using RsID

FLDIGI's RsID decoder uses a probability model to allow detection of the RsID signal with varying degrees of sensitivity and false detections. The greater the sensitivity, the more probable that a code is decoded incorrectly.

### 1.19.7 Multi path signals.

Not a repeater issue but effects digital communications.

Since the repeater is at a fixed location, some of the multi-path issues can be negated by using a directed element (Yagi, Quad, etc) over a ground plane.

### 1.19.8 Over deviation

Repeaters are notoriously finicky about the deviation of your FM signal. Monitor your audio output level. See the owners manual for your radio on how to verify proper modulation level.

## 1.20 FLDIGI Version Issues

### 1.20.1 RSID Sensitivity Settings

On 3.21.77 (or later) versions of FLDIGI the naming convention has been replaced from numerical values to low, medium, and high error rate settings. If this is the case, select medium or a level the conditions dictate.

### 1.20.2 8PSK Modem Availability

FLDIGI 3.22.04 and 3.22.05 contain 8PSK1333, 8PSK1333F and 8PSK1200 modems that are no longer present in 3.22.06 and later releases. The timing tables are still available in FLAMP 2.22.01+ and as a selectable option when transmitting. However, these modems will only be available in FLAMP if the stated versions are used.

## 1.21 FLAMP Version 2.2.10+ Differences



Figure 1.32 Connect State

1) FLAMP can be run independently of FLDIGI. Once both programs are running a connection will be established and a message in the upper right hand side of FLAMP's dialog box will either indicate a "CONNECTED" or "NOT CONNECTED" state.

2) XMLRPC command that's used to receive a modem list from FLDIGI has changed. The pre-filtered list in FLAMP has been removed and as a result, only the available 8 bit modems in the current FLDIGI version are supported.



## Chapter 2

# Configure and Queue File Scripting

### 2.1 List of commands

- [AUTO LOAD QUEUE](#)
- [BASE ‡](#)
- [BLOCKS](#)
- [CALLFROM](#)
- [CALLTO ‡](#)
- [CLEAR MISSING](#)
- [CLEAR TXQ †](#)
- [COMP ‡](#)
- [EVENT FOREVER](#)
- [EVENT](#)
- [EVENT TIMED](#)
- [EVENT TIMES](#)
- [EVENT TYPE](#)
- [FILE †](#)
- [FLAMP\\_CONFIG †](#)
- [HAMCAST MODEM](#)
- [HAMCAST](#)
- [HDR REPEAT](#)
- [HEADER MODEM](#)
- [INFO](#)
- [INHIBIT HEADER](#)
- [INTERVAL](#)
- [LOAD TXDIR](#)





This command has spaces in it's parameter list, requiring quotes.

```
PATH: "/directory path"
```

This command has both commas and spaces in it's parameter list, requiring quotes.

```
FILE: "weather_report.txt", "Report, Weekly Weather"
```

All parameter can be enclosed in quotes.

```
CALLTO: "WX"
```

```
FILE: "weather_report.txt", "Weekly Weather Report"
```

## 2.3 Script Class Types

There are two classes of script files, 'Program Level Script (PLS)' and the 'Queue Level Script (QLS)'. The 'Program' and 'Queue' level scripts can be executed from the 'Script'->'Execute Config Script' menu item. 'Program Level' scripts are responsible for configuring parameters of the program. All commands are available, See [List of commands](#).

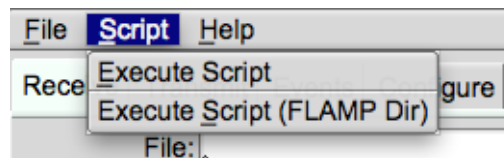


Figure 2.1 Executing PL Scripts

The 'Queue Level Scripts (QLS)' uses a subset of commands listed here ([List of commands](#)) and marked with † or ‡ Notes. This file is placed and executed in the event->timed panel.

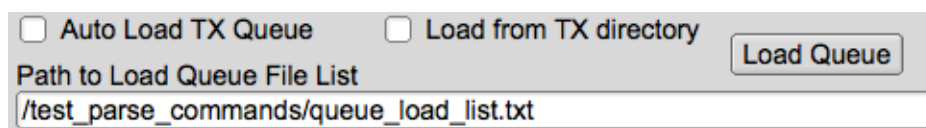


Figure 2.2 Executing QL Scripts

Both class files can be executed in either location. However, if a PL script is executed in the event timed panel, warnings/errors will be issued in the debug log.

Some commands are used to modify the attributes of a queued file. In order to set these attributes the commands noted with ‡ should be called prior to issuing the [FILE](#) command. See [List of commands](#).

Script files are expected to have the .txt file extension.

## 2.4 Operating System Script Differences

The following commands have specific parameter difference for each of the three operations systems.

[PATH](#), [FILE](#), and [QUEUE FILEPATH](#).

[PATH](#) Parameter:

- **Linux:**  
The path structure for accessing directories are  
/Root\_Drive/Directory/Directory/,...
- **Macintosh:**  
Has a case insensitive file system (unless the disk drive is formatted for a case sensitive access). The path structure for accessing directories are  
/Root\_Drive/Directory/Directory/,...
- **Windows:** The primary difference is the use of a disk drive letter and a backslash path separator in place of a fore-slash.  
Example: C:\Directory\Directory\,...  
However, the Win32 Cross Compiler used to create the executable performs translations if the fore-slash is used.  
Example: C:/Directory/Directory/,...  
Produces the same results.

[FILE](#) Parameter:

The file name must contain filename.ext format with no [PATH](#) data. Depending on OS case sensitivity might apply. By having the [PATH](#) as a separate entry. Porting scripts to the other OS's is easier to maintain.

[QUEUE FILEPATH](#) Parameter:

Since this is a single entry the path and filename are combined into one parameter. The format rules are specified above.

## 2.5 Script Execution

To execute a configure script. Select 'Script'->'Execute Script' menu item. This is bring up a open file dialog box allowing for navigating to the desired script file. Select the file in question and press the open button. This executes the script file. See [Error Reporting](#) in the event of errors.

'Script'->'Execute Script (FLAMP Dir)' menu item reset the initial directory path back to the default starting location.

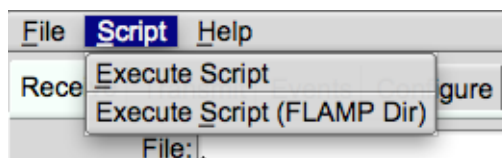


Figure 2.3 Executing PL Scripts

## 2.6 Script Command Descriptions

Available list of command and description of their use:

### 2.6.1 FLAMP\_CONFIG File ID Tag

Required file identifier. Must be located on the first line of the configuration and queue load list.

### 2.6.2 '#' Pound sign.

Any text to the right of the '#' (pound sign) is considered a remark statement.

### 2.6.3 Auto load Queue

During triggered events reload the queue with the contents specified in either the source queue list file or from FLAMP's 'tx' directory.

**Format:** AUTO LOAD QUEUE:<ON|OFF>

### 2.6.4 Base Encoding

Used to transmit data using a method to keep the data values in a specific range. File Attribute Command.

**Format:** BASE:<64|128|256>

### 2.6.5 Transmit block size.

Used to adjust the transmit block size. Normally adjust to the conditions of the band. Poor conditions requires smaller values. The value range is between 16 and 2048 in increments of 16 (16,32,48,...,2048).

**Format:** BLOCKS:<value>

### 2.6.6 Callsign of the transmitting station

Assign the callfrom call. Any text are allowed. Follow local rules on it's use (<callfrom> de <callto>).

**Format:** CALLFROM:<callsign>

### 2.6.7 Callsign of the receiving stations

Assign the callto call. Any text are allowed. Follow local rules on it's use (<callfrom> de <callto>). File Attribute Command.

**Format:** CALLTO:<callsign>

### 2.6.8 Clear Missing

After a uninterrupted transmit fill. The contents of the missing blocks list is cleared.

**Format:** CLEAR MISSING : <ON|OFF>

### 2.6.9 Clear the transmit queue

Clears the transmit queue of all files. No parameters used. **Caution:** No warning issued.

**Format:** CLEAR TXQ :

### 2.6.10 Compression

Enables/Disables Compression use on a file by file basis. Compression may or may not be employed. The file size of before and after compression determines if it's used or not.

**Format:** COMP : <ON|OFF>

### 2.6.11 Events, Forever

Enables/Disables events to occur without the restriction of time. Requires EVENTS to be enable. File Attribute Command.

**Format:** EVENT FOREVER : <ON|OFF>

### 2.6.12 Events Times

Events will occur at time HHMM or between HHMM-HHMM. HHMM-HHMM requires Event type "Continuous at".

**Format:** EVENT TIMES : <HHMM|HHMM-HHMM>

### 2.6.13 Events Types

Event types are listed below. The event type requires enclosed "".

**Format:** EVENT TYPE : "<event\_type>"

Event Type List:

- "5 min"  
Transmit every 5 minutes
- "15 min"  
Transmit every 15 minutes
- "30 min"  
Transmit every 30 minutes

- "Hourly"  
Transmit every hour
- "Even hours"  
Transmit every even hour
- "Odd hours"  
Transmit every odd hour
- "Repeated at"  
Transmit at specific time intervals repeatedly
- "One time at"  
Transmit at specific time intervals once
- "Continuous at"  
Transmit between specific time intervals repeatedly

### 2.6.14 Enable/Disable Timed Events

Enable/Disable Timed Events.

**Format:** EVENT TIMED : <ON|OFF>

### 2.6.15 Enable/Disable Events

Enable/Disable all event types. When enabled, data is sent to FLDIGI for transmission when the event condition is reached. Sets the Start/Stop event button in the event panel.

**Format:** EVENT : <ON|OFF>

### 2.6.16 Add file to TX Queue

Add a file to the transmit queue list. Do not include a path to the file, file name and/or extension only. Use the [PATH](#) command to indicated the directory location of the file(s). The second parameter is a description of the file and fills the AMP protocol DESC tag field. Enclosed "" are required if the description contains either a comma or space(s). See [Path](#) and [Operating System Script Differences](#) for formatting/use.

**Format:** FILE : <filename.ext>, <"description of the file">

### 2.6.17 Setting/Enabling/Disabling Hamcast Modems

Enable/Disable specific <pos> hamcast modem and/or set the modem type for that position.

**Format:** HAMCAST MODEM : <pos>, <ON|OFF|MODEM\_ID\_STRING>

### 2.6.18 Enable/Disable Hamcast Events

Enable/Disable hamcast modem rotation events.

**Format:** HAMCAST : <ON|OFF>

### 2.6.19 Header Transmit Repeat

The number of time the header section of the AMP protocol is repeatedly transmitted. Repeat <count> times.

**Format:** HDR REPEAT:<count>

### 2.6.20 Header Modem

The ID string used to set the modem type when modem header is enabled.

**Format:** HEADER MODEM:<modem\_id\_string>

### 2.6.21 Header Modem Enable/Disable

When enabled (on) transmit the header data using the head modem.

**Format:** HEADER MODEM:<ON|OFF>

### 2.6.22 Information Field

Use to set the AMP protocol tag INFO. Can contain any data (text). "" must be used if the text contains either commas or spaces. Normally set to the transmitters QTH and locator.

**Format:** INFO:<"City, State AA00aa">

### 2.6.23 Inhibit header modem on fills

Enable/Disable Inhibit header modem on fills.

**Format:** INHIBIT HEADER:<ON|OFF>

### 2.6.24 Interval Timer

Enable/Disable Interval Timer.

**Format:** INTERVAL:<ON|OFF>

### 2.6.25 Load from TX directory

When auto load queue on event is enable. Setting this option to on cause the files to be loaded from FLAMP's tx directory. When disable the file are loaded the queue list.

**Format:** LOAD TXDIR:<ON|OFF>

### 2.6.26 Modem

Sets the primary modem for non-header/hamcast transmit's

**Format:** MODEM:<modem\_id\_string>

### 2.6.27 Path

Set the path to all file loads. See [FILE](#) and [Operating System Script Differences](#) for formatting.

**Format:** PATH:<directory\_path>

### 2.6.28 Enable/Disable AMP protocol

Enabling proto (on) transmits the file using AMP-2 specification. Disabling allows the data to be transmitted in plain text (7 bit ASCII). File Attribute Command.

**Format:** PROTO:<ON|OFF>

### 2.6.29 Load Queue List

Sets the file name and path of the load queue list. See [Operating System Script Differences](#) for formatting.

**Format:** QUEUE FILEPATH:</directory\_path/filename.ext>

### 2.6.30 Reset/Clear Configure settings.

Resets/Clears configuration items.

**Format:** RESET:<ALL|PARTIAL>

- "ALL"  
Clears all attributes
- "PARTIAL"  
Clears all attributes with the exception of CALLFROM and INFO fields

### 2.6.31 Receive interval time (seconds).

Designed for repeater use. This is the time the transmitter is turned off allowing the repeater to reset. See [INTERVAL](#) and [TX INTERVAL](#). Allowable range 1-120.

**Format:** RX INTERVAL:<time\_in\_seconds>

### 2.6.32 FLDIGI and FLAMP modem sync

Used to synchronize the modem selected between FLDIGI and FLAMP.

**Format:** SYNC WITH:<direction>, <ON|OFF>

Sync List:

- "FLAMP"  
Sync FLDIGI's modem to FLAMP's
- "FLDIGI"  
Sync FLAMP's modem to FLDIGI's
- "PRIOR"  
Set FLDIGI's modem to FLAMP's just prior to sending transmit data.

### 2.6.33 Transmit interval time (minutes).

Designed for repeater use. This is the time the transmitter is turned on. See [INTERVAL](#) and [RX INTERVAL](#). Value range 1.0-8.0 in 0.05 increments (3 seconds).

**Format:** TX INTERVAL:<time\_in\_minutes>

### 2.6.34 Transmit on Report

Directly enable transmission when Missing Block reports are sent to FLDIGI.

**Format:** TX REPORT:<ON|OFF>

### 2.6.35 Unprotocol (plain text) Markers

Add --- start --- to the beginning and --- end --- to the end of plain text transmissions. Allows for easy determinations of content over the airwaves.

**Format:** UNPROTO MARKERS:<ON|OFF>

### 2.6.36 File Delete Warning

Enable/Disable Warning message when deleting transmit queue items from the transmit panel. Note available when using [CLEAR TXQ](#) script command.

**Format:** WARN USER:<ON|OFF>

### 2.6.37 Data Transmit Repeat

The number of time the data section of the AMP protocol is repeatedly transmitted. Repeat <count> times.

**Format:** XMIT REPEAT:<count>



## 2.7 Script Error Reporting

If errors are found at the completion of script parsing. This dialog box will be displayed.



Figure 2.4 Script Error Dialog Box

For a detailed review, open the debug log under the 'Help' menu. This example indicates the path to the files was not found. This [PATH](#) error is responsible for the [FILE](#) errors.

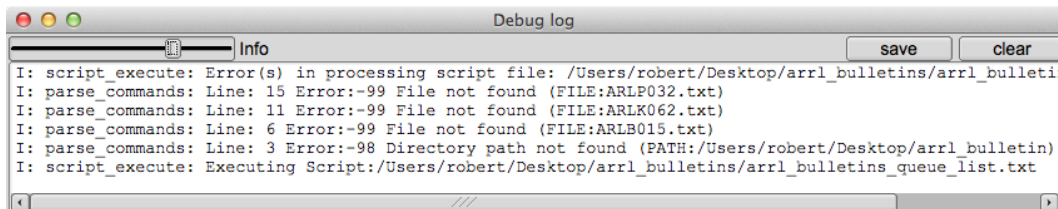


Figure 2.5 Debug Log Script Error(s)

The information in the debug log provides a line number of where the error occurred. Take note of this when making corrections to the script file.

### NOTE:

Although errors have occurred, all commands that are syntax correct have been processed and the appropriate settings applied.

## 2.8 Example Program Level Script (PLS)

Using this example requires editing to properly execute on your system. See [Operating System Script Differences](#). Not all commands are used in this example.

```
FLAMP_CONFIG # Required file identifier. Must be located on the first line.
# Remark lines start with the character `#'.
RESET:ALL # Disable all Event and Configuration panel settings.
UNPROTO MARKERS:ON # Enable unprotocol markers (pain text separators)
EVENT TIMES:1425-1431 # The time the event(s) are triggered settings.
LOAD TXDIR:ON # Enable/Disable file loads from the 'tx' directory on events
QUEUE FILEPATH:"/directory_path/queue_load_list.txt" # Path to the QLS File List
BLOCKS:80 # Set the size of the transmit blocks
MODEM:BPSK63 # Set the the transmit mode to x
CALLFROM:KK5VD # Assign the callfrom call.
INFO:"Madison, AL EM64or" # Set the information field in the configuration panel
SYNC WITH:PRIOR,ON # Sync FLDIGI's modem to FLAMP's
CLEAR TXQ: # Clear the transmit queue
PATH:"/Root_drive/directory_path/" # Directory path to the pending FILE load commands
PROTO:OFF # Set the protocol use to the pending FILE commands
```

```

CALLTO:"WX" # Set the CALLTO field to the pending FILE commands.
FILE:"wx_report.txt", "Weekly Weather Report" # Load a file into the transmit queue
PROTO:ON # Set the protocol use to the pending FILE commands
BASE:64 # Set the base conversion to the pending FILE commands
CALLTO:"SPACE WX" # Set the CALLTO field to the pending FILE commands.
FILE:"space_wx.txt", "Weekly Space Weather Report" # Load a file into the transmit queue
AUTO LOAD QUEUE:ON # During events reload queue with the contents specified in either
MODEM:BPSK31 # Transmitting modem.
COMP:ON # Enable/Disable Compression.
BASE:64 # Set Conversion base.
BLOCKS:64 # Block size.
PROTO:ON # Enable/Disable Protocol use.
XMIT REPEAT:2 # Number of times to repeat data content.
HDR REPEAT:3 # Number of times to repeat header content.
EVENT TYPE:"Continuous at" # Example: 'Continuous at'.
EVENT:ON # Enable/Disable Events.
EVENT TIMES:1425-1431 # The time the event(s) are triggered.
AUTO LOAD QUEUE:ON # During events reload queue with the contents specified.
INTERVAL:ON # Enable/Disable Interval Timer.
TX INTERVAL:8 # Interval transmit time in minutes
RX INTERVAL:5 # Interval receive time in seconds.
HEADER MODEM:ON # Enable/Disable Header modem use.
HEADER MODEM:BPSK63 # Modem used for Header Modem.
HAMCAST MODEM:1,ON # Enable/disable Modem 1 use for Hamcast
HAMCAST MODEM:1,BPSK63 # Modem used for Hamcast in position 1
HAMCAST MODEM:2,ON # Enable/disable Modem 2 use for Hamcast
HAMCAST MODEM:2,THOR100 # Modem used for Hamcast in position 2
HAMCAST MODEM:3,ON # Enable/disable Modem 3 use for Hamcast
HAMCAST MODEM:3,THOR50 # Modem used for Hamcast in position 3
HAMCAST MODEM:4,ON # Enable/disable Modem 4 use for Hamcast
HAMCAST MODEM:4,BPSK31 # Modem used for Hamcast in position 4
HAMCAST:ON # Enable/Disable Hamcast.
TX REPORT:ON # Enable/Disable Transmit on report.
WARN USER:ON # Enable/Disable User warning on transmit queue file removal.
CLEAR MISSING:ON # Enable/Disable Clear missing of fills
INHIBIT HEADER:ON # Enable/Disable Inhibit header modem on fills
EVENT FOREVER:ON # Enable events until disabled.

```

## 2.9 Example Queue Level Script (QLS)

Using this example requires editing to properly execute on your system. See [Operating System Script Differences](#).

```

FLAMP_CONFIG
CLEAR TXQ:
PATH:"C:/directory_path/"
PROTO:OFF
BLOCKS:128
TX INTERVAL:2.75
RX INTERVAL:5
MODEM:8PSK250F
CALLTO:"WX"
FILE:"weather_report.txt", "Weekly Weather Report"
PROTO:ON
BASE:64
COMP:ON
CALLTO:"SPACE WX"
FILE:"space_weather_report.txt", "Weekly Space Weather Report"
PROTO:ON
COMP:OFF
CALLTO:"ARRL/FCC"
FILE:"arrl_fcc_ruling.txt", "Ruling on Encryption Use"

```