

Flamp Quick Start

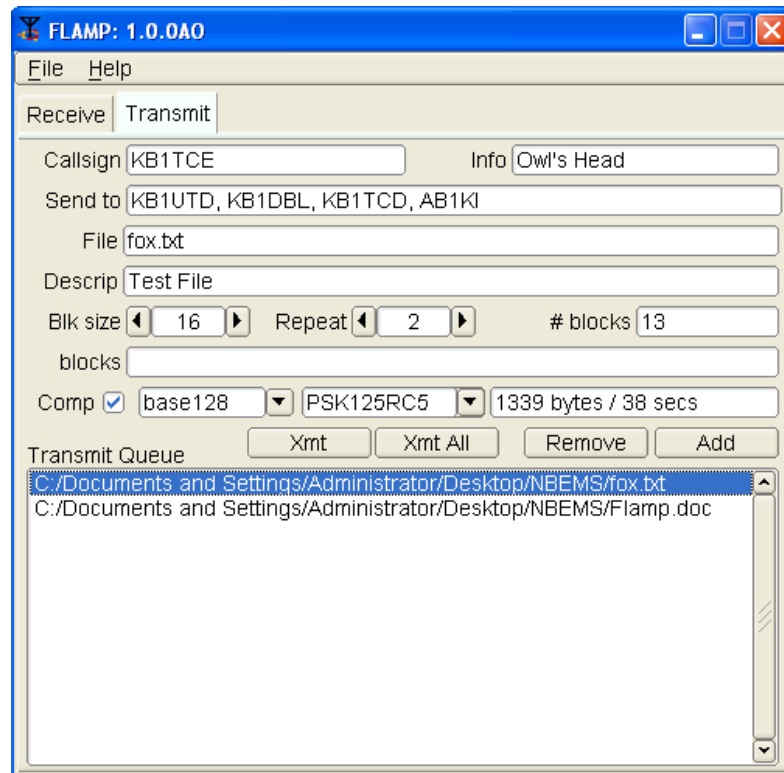
This is based on the current (as of 10/08/2012) alpha version of flamp, 1.0.0AO. Details will change as the program matures.

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 w1hkj.com/alpha/flamp. Install the software as you would any of the NBEMS applications.

Transmit Procedure

When you open flamp you will see a screen with two tabs, **Receive** and **Transmit**. We'll start with the Transmit screen as shown below.



Start by filling in your **Call Sign** and desired **Information**. 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.

As noted above, 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.

Flamp will repeat the transmission from 1 to 10 times. Select the appropriate number based on transmission mode (i.e. mode performance), propagation conditions, permissible transmit time (e.g. repeater time out), etc.

The file identification is based on the date and time of the last file modification. Therefore, additional transmission sequences may be performed if necessary as long as the file is not changed. If the file is modified, Flamp will treat the file as a new file.

To send a file, it has to be added to the queue. Click **Add** and then browse for the file. 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).

Flamp provides some additional information. When you select the compression/no compression option and the mode (in this case PSK125RC5), you will see the file size, number of blocks and estimated transmission time. This is useful for determining tradeoffs 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 header overhead.

In the example figure, the small fox.txt file will take 38 seconds to send twice with the selected mode and parameters. The larger Flamp.doc file is 12.2 kB and would take 5 minutes, 48 seconds by itself using the selected parameters.

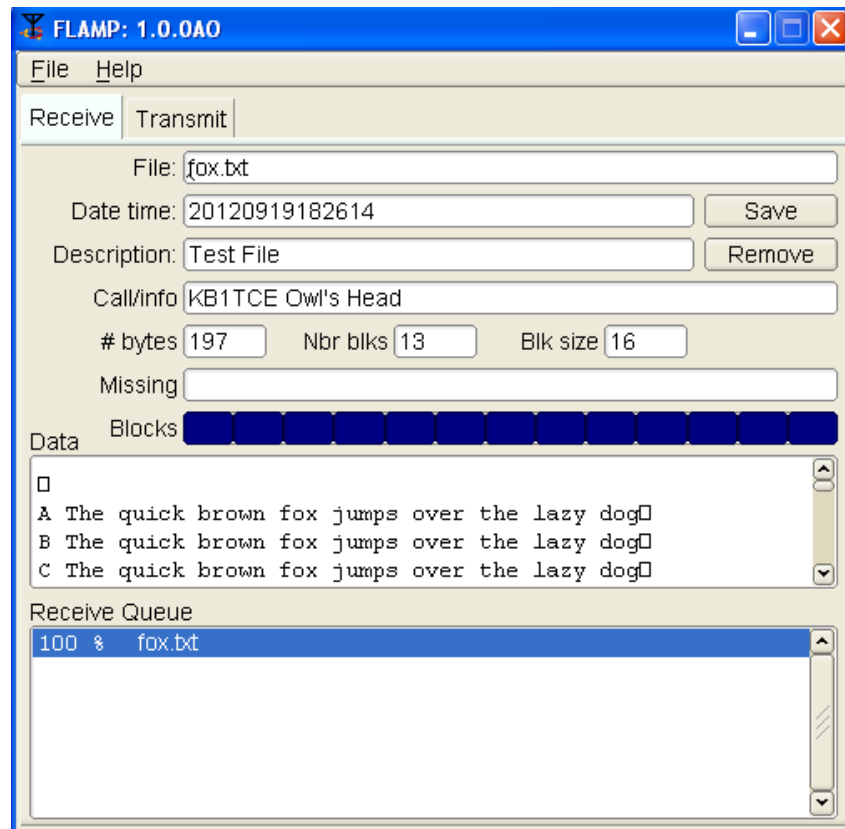
When the file has been added to the **Transmit Queue**, you may add a file description in the **Descrip** line. This could be just a basic description of the file or have handling instructions. If sending just one file, you must highlight the file in the queue to transmit. This will place the file name 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 Queue**. Highlighting is not necessary.

Before transmitting, an flamp-compatible version of fldigi must be running with the mode selected and frequency clear.

The **blocks** line will be discussed later. This is used for filling in missing blocks without a full repeat of the file transmission.

Receive Procedure

The receiving stations must have flamp open with fldigi running. Fldigi will not “wake up” flamp as it does with, for example, flmsg.



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.

You will also see block numbers showing up in the **Missing** line. When the complete file is received, that line should be blank as shown in the capture.

Received text files will appear in complete form and readable in the **Data** panel.

The **Send to** information does not appear on the flamp screen. It will show in the fldigi RX panel as shown below:

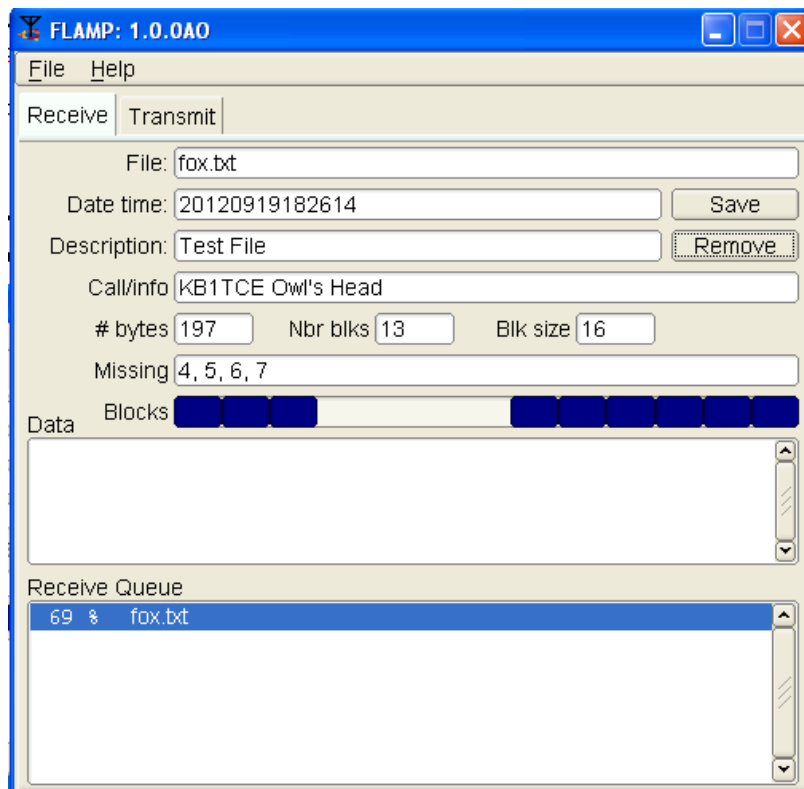
KB1UTD, KB1DBL, KB1TCD, AB1KI DE KB1TCE
 <PROG 13 C0C7>FLAMP 1.0.0AO
 <FILE 22 FB17>20120919182614:fox.txt
 <ID 17 1EFA>KB1TCE Owl's Head
Etc.

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 name
Windows	C:\Documents & Settings\Usr\flamp\
Linux	\home\user\.flamp\
OS X	\home\user\flamp\

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 capture below shows a partially completed file. Four blocks were missed in the first pass and they will (hopefully) be filled in during the next iteration(s).

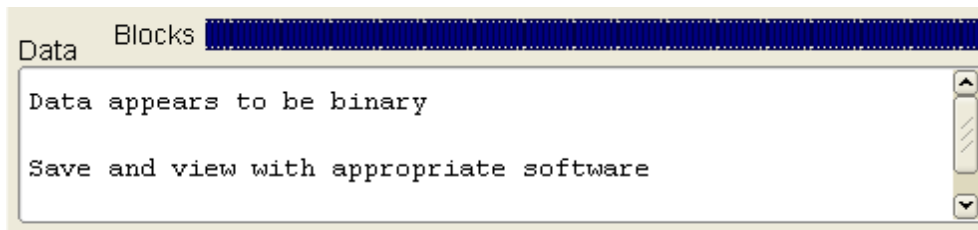


Note that the blocks with errors show up in the **Missing** line. In this case, blocks 4, 5, 6 and 7 were received with errors. The **Receive Queue** panel will show the percent received. Nothing will appear in the **Data** panel.

Non-Text Files

Flamp will send any file format. For example, ICS forms from Flmsg may be sent as well as spreadsheets in csv format or even binary files such as the Flamp.doc file as shown in the Transmit capture.

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



Sending Missing Blocks

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.

There is no automation for requesting a partial block transfer. This should be accomplished using keyboard-to-keyboard chat (HF) or voice (VHF/UHF). After a one-to-many transfer the sending station should request verification and missing block information from the intended recipients. The sender then would compile the combined list of missed blocks for retransmission.

For the example with 4 missing blocks, the **blocks** line would be filled in as shown below. The blocks are delineated in a comma-separated list. You may also use any non-numeric character such as space, colon, semicolon, etc.

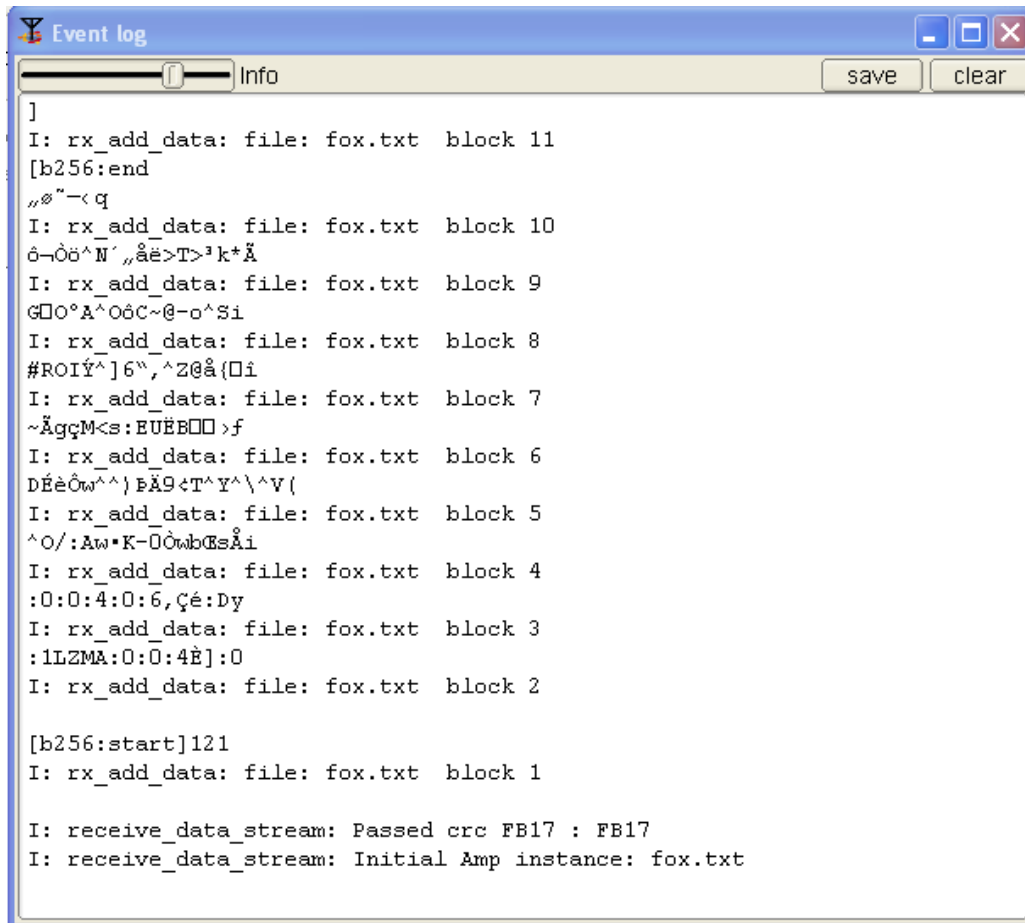
A screenshot of a file transmission configuration window. It contains several input fields and buttons. The 'File' field is set to 'fox.txt'. The 'Descrip' field is set to 'Test File'. The 'Blk size' field has a value of 16, with left and right arrow buttons. The 'Repeat' field has a value of 2, with left and right arrow buttons. The '# blocks' field is set to 13. The 'blocks' field contains the text '4,5,6,7'. The 'Comp' field has a checked checkbox, a dropdown menu set to 'base128', another dropdown menu set to 'PSK125RC5', and a text field showing '651 bytes / 19 secs'. At the bottom, there is a 'Transmit Queue' label and four buttons: 'Xmt', 'Xmt All', 'Remove', and 'Add'.

Event Log

An Event Log has been added to Flamp for debug 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.



Example

The text file used in the examples consists of 26 iterations of “The quick brown fox....” as shown below:

A The quick brown fox jumps over the lazy dog
B The quick brown fox jumps over the lazy dog
|
Z 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. In this example, the **Send to** is the default “QST.”

QST QST de kb1tce Owl's Head

```
<FLAMP 7 0D85>1.0.0AE
<ID 17 3325>kb1tce Owl's Head
<FILE 22 B49B>20120910221518:fox.txt
<SIZE 14 FD5D>{B49B}161 6 32
<DATA 40 32EF>{B49B:1}[b256:start]121
:1LZMA:0:0:4-]:0
<DATA 40 5379>{B49B:2}:0:0:4:0:6---:Dy=/:Aw-K-0-wb-s-i
<DATA 40 66E8>{B49B:3}D---w=)--9-T===(~-g-M<s:EU-B----
<DATA 40 B02F>{B49B:4}#ROI-=6---=@- {---G-----A=-C~@-o=i
<DATA 40 0EB0>{B49B:5}----->T>-k*-----q
[b256:end
<DATA 9 319D>{B49B:6}]
<CNTL 5 9016>{EOF}

<FILE 22 B49B>20120910221518:fox.txt
<SIZE 14 FD5D>{B49B}161 6 32
<DATA 40 32EF>{B49B:1}[b256:start]121
:1LZMA:0:0:4-]:0
<DATA 40 5379>{B49B:2}:0:0:4:0:6---:Dy=/:Aw-K-0-wb-s-i
<DATA 40 66E8>{B49B:3}D---w=)--9-T===(~-g-M<s:EU-B----
<DATA 40 B02F>{B49B:4}#ROI-=6---=@- {---G-----A=-C~@-o=i
<DATA 40 0EB0>{B49B:5}----->T>-k*-----q
[b256:end
<DATA 9 319D>{B49B:6}]
<CNTL 5 9016>{EOF}

<CNTL 5 301A>{EOT}
```

QST QST de kb1tce Owl's Head