

# **TransNibAm**

Matt Francis

Copyright © 1993 Matt Francis

---

**COLLABORATORS**

	<i>TITLE :</i> TransNibAm		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Matt Francis	October 27, 2024	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>TransNibAm</b>	<b>1</b>
1.1	Contents . . . . .	1
1.2	credits . . . . .	1
1.3	requirments . . . . .	1
1.4	introduction . . . . .	1
1.5	Using TransNibAm . . . . .	2
1.6	Send Mode . . . . .	2
1.7	Receive Mode . . . . .	3
1.8	signals . . . . .	3
1.9	acknowledgments . . . . .	3
1.10	history . . . . .	3

---

# Chapter 1

# TransNibAm

## 1.1 Contents

TRANSNIBAM V3.11

[Credits](#)

[Requirments](#)

[Introduction](#)

[Using TransNibAm](#)

[Acknowledgments](#)

[History](#)

## 1.2 credits

TRANSNIBAM V3.11

Created by Matt Francis

ReqTools by Nico François

GUI created with GadToolsBox V2.0c

Comments, suggestions, bug reports etc. to

[m.p.francis@newcastle.ac.uk](mailto:m.p.francis@newcastle.ac.uk)

## 1.3 requirments

TransNibAm requires Kickstart V37 (release 2.04) or above. Also required is Nico François' ReqTools library V38 or above (supplied).

## 1.4 introduction

TransNibAm is the Amiga front-end for the TransNib V1.00 protocol (please read the TransNib docs if you haven't already).

TransNibAm is totally OS-friendly, and will comfortably multi-task alongside the best of them. Its window can be opened on any public screen, so you could operate it from something like Directory Opus should you wish. The parallel port hardware is

---

locked properly before a transfer commences, and unlocked as soon as the transfer finishes. Any other OS-compliant tasks trying to use the parallel port while TransNibAm is transferring won't be granted access to it. Likewise, if another program is using the parallel port (sending something to a printer for example), TransNibAm will inform you that it can't lock the port, and you can try again once the port is free.

## 1.5 Using TransNibAm

TransNibAm can be started from a Shell or the Workbench. The Shell particulars follow:

Format: TRANSNIBAM [[SENDDIR] <name of default send directory>]

[[RECVDIR] <name of default receive directory>]

[BUFFSIZE <buffer size in bytes>]

[PUBSCREEN <public screen name>]

Template: SENDDIR,RECVDIR,BUFFSIZE/N/K,PUBSCREEN/K

SENDDIR - You can set this to the directory containing the files you wish to send.

Default: the current directory.

RECVDIR - You can set this to the directory where you want received files to be saved.

Default: the current directory.

BUFFSIZE - TransNibAm conducts all transfers in blocks. The size of each block is determined by this option.

Default: 1024 bytes.

PUBSCREEN - Set this to the name of the public screen you want TransNibAm to open on.

Default: the default public screen (normally Workbench).

When started from the Workbench, TransNibAm accepts tool-types whose names are identical to its Shell arguments.

TransNibAm's window has a number of fields. The Status field indicates what TransNibAm is currently up to, and will read "Ready" when a transfer is not taking place. The File field will display the name of the filename currently being sent or received. The Bytes field shows how many bytes of the current file remain to be sent or received during a transfer, and the field after it shows the actual length of the current file.

TransNibAm's single menu is self-explanatory.

TransNibAm has two modes of operation, **Send** and **Receive**.

Sending TransNibAm a **CTRL-D** will cause it to pop up. TransNibAm can be killed with a **CTRL-C**.

## 1.6 Send Mode

In this mode, you select one or more files to send. Before the transfer begins, you will be asked if you want to "Proceed" or "Cancel". Make sure the target machine is ready to receive before clicking on "Proceed".

If any problems occur (for example, a file cannot be opened or there is a read error), the whole operation will be aborted and TransNibAm will revert to its "Ready" state.

## 1.7 Receive Mode

In this mode, you select the destination directory and click on "Proceed" when the sending machine is ready. After that, everything will happen automatically until the transfer finishes.

If any problems occur (for example, a file cannot be opened or there is a write error), the whole operation will be aborted and TransNibAm will revert to its "Ready" state.

## 1.8 signals

If you started TransNibAm synchronously from a Shell (i.e. you didn't use the Run command) you can send it a CTRL-C or CTRL-D signal just by pressing those keys. If you started the program using the Run command, or you are in another Shell window, use the Status command to find TransNibAm's process number and then use the Break command to send it a signal.

If you started TransNibAm from the Workbench, you will have to find its process number some other way or use a task monitor program (e.g. TaskX or TaskE) to send it a signal.

## 1.9 acknowledgments

Thanks to the following people:

Nico François for the ReqTools library, which makes life much more pleasant.

Jan van den Baard for GadToolsBox, which was used to design TransNibAm's GUI.

CygnusSoft for CygnusEd, the fastest, most stable and best text editor for the Amiga (and, from what I've seen, for any computer!). And it still works without problems on my A1200!

Commodore for brilliant computers which the competition just can't match.

## 1.10 history

V3.11

Modified to use new-look menus.

V3.10

Restructured the GUI event handling code. Changed the arguments and GUI slightly. Made several incidental changes. Added ID port feature to prevent two copies of TransNibAm from running at once.

V3.00

At last, a custom GUI (thanks to GadToolsBox). ReqTools is still in there, handling the requesters, but now TransNibAm has its own nice window, complete with status boxes and menus. I also made this version totally system-friendly (the other versions didn't allocate the parallel port, and disabled multitasking while transferring).

V2.00

After a few intermediate versions (which I didn't keep track of) I optimised the transfer routines and changed a few things round. Still uses the ReqTools library for the interface though.

V1.00

This version was written using nothing but ReqTools functions for the GUI. TransNib was still a little experimental in those days, but the system worked well enough.

---