

# **AmiSlate**

Jeremy Friesner

**COLLABORATORS**

	<i>TITLE :</i> AmiSlate		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Jeremy Friesner	February 14, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>AmiSlate</b>	<b>1</b>
1.1	Contents . . . . .	1
1.2	Installation . . . . .	1
1.3	credits . . . . .	2
1.4	How to reach me . . . . .	2
1.5	disclaimer . . . . .	3
1.6	AmiSlate is DonationWare . . . . .	3
1.7	requirements . . . . .	4
1.8	introduction . . . . .	5
1.9	Using AmiSlate . . . . .	5
1.10	keys . . . . .	6
1.11	The AmiSlate GUI . . . . .	6
1.12	The Toolbar . . . . .	7
1.13	The Palette . . . . .	7
1.14	The Chat Lines . . . . .	8
1.15	Serving Suggestion . . . . .	8
1.16	Rexx Scripts included with AmiSlate . . . . .	9
1.17	arexx . . . . .	10
1.18	Starting AmiSlate . . . . .	11
1.19	REXXMENU#/K . . . . .	11
1.20	OUTQUEUE/K . . . . .	12
1.21	REXXOUTPUT/K . . . . .	12
1.22	Workbench Screen . . . . .	12
1.23	Specify a Public Screen . . . . .	12
1.24	Specify window height . . . . .	13
1.25	Specify Window width . . . . .	13
1.26	Specify window X-coordinate . . . . .	13
1.27	Specify window Y-coordinate . . . . .	13
1.28	AmiSlate menus . . . . .	13
1.29	Project Menu . . . . .	14

---

---

1.30 TCP Menu . . . . .	14
1.31 Script Menu . . . . .	14
1.32 Options Menu . . . . .	15
1.33 Safe Flood Fills . . . . .	16
1.34 Not-So-Safe Flood Fills . . . . .	16
1.35 The Rexx menu . . . . .	16
1.36 Thanks . . . . .	17
1.37 What's new, what's old, what's changed . . . . .	18
1.38 What's Next? . . . . .	20
1.39 Known Bugs and Other Problems . . . . .	21

---

# Chapter 1

## AmiSlate

### 1.1 Contents

AmiSlate V1.3 by [Jeremy Friesner](#)

AmiSlate is a program that allows you to draw pictures or play graphical type games with a friend over an AmiTCP/Internet link.

**Disclaimer** Don't blame me!

**Distribution** AmiSlate is DonationWare (tm)

**Requirements** What do I need to run this program?

**Introduction** What does AmiSlate do?

**Installation** How do I set AmiSlate up?

**Using AmiSlate** How to run AmiSlate

**Credits** Where it's due

**Acknowledgments** Thanks to...

**ARexx** AmiSlate supports ARexx!!

**Included Scripts** Games and Stuff

**History** Bug fixes and enhancements

**Future** What next?

**Known Problems** Bugs! Aack!

### 1.2 Installation

To install AmiSlate, either run the supplied Installer script, or follow the directions below. Note that if you are upgrading from AmiSlate 1.0 or 1.1, you will need to manually edit your inetd.conf file to change the word "wait" on AmiSlate's line to "nowait". The Installer script isn't smart enough to do this for you.

If you want to run AmiSlate without using its TCP connection capability, then installation is simple. You will only need to do steps 1, 4 and 5,

below.

If you want to connect to another person, on the other hand, then Installation is a bit harder. However, follow the steps below and you should be OK. (Note: this is what works for me under AmiTCP 3.0B2-- other versions of AmiTCP may vary slightly, so if you find a discrepancy, please tell me about it!)

1) Copy the AmiSlate executable to where you want it to be. I put mine in amitcp:bin, so if you put it elsewhere, substitute that directory wherever you see amitcp:bin, below.

2) Add the following line to the end of your amitcp:db/services file:

```
AmiSlate 2955/tcp
```

3) Add the following line to the end of your amitcp:db/inetd.conf file:

```
AmiSlate stream tcp nowait root amitcp:bin/AmiSlate
```

4) Copy the SlateRexx directory to wherever it is you want to keep it.

5) Copy the SlateScripts directory to wherever you want to keep it.

6) Edit your s:User-Startup (or whatever file you do all of your ASSIGNs in at system bootup) to assign the logical drive SlateRexx: to the directory where you are keeping your AmiSlate ARexx scripts.

Also, assign the logical drive SlateScripts: to the directory where you want to keep your AmiSlate scripts (the binary recordings).

For example, I keep my AmiSlate ARexx scripts in the directory dh0:AmiSlate/SlateRexx, and binary scripts in dh0:AmiSlate/SlateScripts, so I added the lines:

```
assign SlateRexx: dh0:AmiSlate/SlateRexx
```

```
assign SlateScripts: dh0:AmiSlate/SlateScripts
```

to my s:user-startup file.

7) Re-start your computer (just to be sure of things), and try it out!

## 1.3 credits

AmiSlate V1.3

Created by [Jeremy Friesner](#)

ARexx port created with ARexxBox by Michael Balzer

Compiled with DICE C by Matt Dillon

## 1.4 How to reach me

Here are some ways to get in touch with me:

by EMail: [jfriesne@ucsd.edu](mailto:jfriesne@ucsd.edu)

by SMail: Jeremy Friesner,

7581 Knollwood Place

Dublin, CA 94568

---

## 1.5 disclaimer

This software comes with no warranty, either expressed or implied.

The **author** is in no way responsible for any damage or loss that may occur due to direct or indirect usage of this software. Use this software entirely at your own risk.

## 1.6 AmiSlate is DonationWare

AmiSlate is DonationWare. I've put a lot of time into it to make it as fun and useful as possible, so if you find AmiSlate to your liking and use it often, please consider sending **me** a \$5 or \$10 donation, and in return I will send you future upgrades directly and give your suggestions preferred treatment.

However, if you can't afford that or for some other reason don't want to send money, that's okay also. Just send me email telling me that you're using it, and list any suggestions that you have for improving it. :-)

Furthermore, if you have any cool ARexx scripts that you've written for AmiSlate that you think should be included in the AmiSlate archive, please send me those.

Permission is given to include this program in a public archive (such as a BBS, FTP site or PD library) providing that all parts of the original distribution are kept intact. These are as follows:

Listing of archive 'AmiSlate.lha':

Original Packed Ratio Date Time Name

```
-----
2273 564 75.1% 05-Aug-95 19:53:34 AmiSlate.info
91276 44752 50.9% 05-Aug-95 19:53:26 +AmiSlate
1934 1266 34.5% 05-Aug-95 19:53:26 +AmiSlate.info
2228 516 76.8% 05-Aug-95 19:53:26 +Doc.info
43260 16464 61.9% 05-Aug-95 19:53:26 +AmiSlate.guide
504 283 43.8% 05-Aug-95 19:53:28 +AmiSlate.guide.info
38248 11540 69.8% 05-Aug-95 19:53:28 +AmiSlateRexx.guide
504 284 43.6% 05-Aug-95 19:53:28 +AmiSlateRexx.guide.info
2273 562 75.2% 05-Aug-95 19:53:32 +ExampleRexx.info
1692 785 53.6% 05-Aug-95 19:53:32 +boxme.rexx
2051 921 55.0% 05-Aug-95 19:53:32 +constants.rexx
309 196 36.5% 05-Aug-95 19:53:34 +easyreq.rexx
176 132 25.0% 05-Aug-95 19:53:34 +getpixel.rexx
```

```

566 309 45.4% 05-Aug-95 19:53:34 +raster.rexx
378 232 38.6% 05-Aug-95 19:53:34 +remote.rexx
446 266 40.3% 05-Aug-95 19:53:34 +remotestringreq.rexx
2521 996 60.4% 05-Aug-95 19:53:32 +wait.rexx
5997 1914 68.0% 05-Aug-95 19:53:28 +Install_AmiSlate
676 334 50.5% 05-Aug-95 19:53:28 +Install_AmiSlate.info
1128 664 41.1% 05-Aug-95 19:53:26 +README
509 279 45.1% 05-Aug-95 19:53:26 +README.info
2273 561 75.3% 05-Aug-95 19:53:32 +SlateRexx.info
39079 9221 76.4% 05-Aug-95 19:53:30 +chess.rexx
1678 802 52.2% 05-Aug-95 19:53:30 +coords.rexx
1499 747 50.1% 05-Aug-95 19:53:30 +diamond.rexx
9210 3062 66.7% 05-Aug-95 19:53:30 +displaytext.rexx
888 496 44.1% 05-Aug-95 19:53:30 +explode.rexx
1055 558 47.1% 05-Aug-95 19:53:30 +explode2.rexx
1329 715 46.2% 05-Aug-95 19:53:32 +flower.rexx
1820 849 53.3% 05-Aug-95 19:53:30 +ghostcircle.rexx
1945 853 56.1% 05-Aug-95 19:53:30 +qix.rexx
1108 601 45.7% 05-Aug-95 19:53:30 +rainbowpen.rexx
5528 1883 65.9% 05-Aug-95 19:53:30 +simple.slatefont
3444 1565 54.5% 05-Aug-95 19:53:30 +surround.rexx
15772 3823 75.7% 05-Aug-95 19:53:30 +tictactoe.rexx
2228 514 76.9% 05-Aug-95 19:53:32 +SlateScripts.info
12416 6531 47.3% 05-Aug-95 19:53:32 +Welcome.script

```

```

-----
300221 116040 61.3% 05-Aug-95 19:53:58 37 files

```

No charge may be made for this program, other than a reasonable copying fee, and/or the price of the media.

## 1.7 requirements

AmiSlate requires an Amiga running Kickstart V37 (WorkBench 2.04) or higher to operate.

AmiTCP 3.0b or higher, an Internet connection, and a friend are not required, but life is definitely more fun if you have them :)

AmiSlate opens the following libraries and devices:

Library Minimum Version #

```
-----
```

intuition.library 37

bsdsocket.library 2 (AmiSlate will run without this)

iffparse.library 37 (AmiSlate will run without this)

graphics.library 37

gadtools.library 36

asl.library 37

icon.library 33

Device

-----

timer.device

## 1.8 introduction

Please read the [History](#) section for information on changes and bug-fixes.

AmiSlate is a program that lets you and a friend paint together on the same canvas, using two different Amigas that are connected by AmiTCP. It also allows recording and playback of your paintings, and has extensive ARexx support that allows AmiSlate to be used as a base for simple network-aware [games](#) and other things.

AmiSlate lets two communicating sessions be on screens of different sizes and depths; colors transmitted will be mapped to the closest color available on the remote client's palette. However, AmiSlate demands that the window size of both sessions be the same; when one client resizes its window, the other client will resize its window as well. Thus, if your AmiSlate window seems to be resizing itself spontaneously, blame your friend on the other end of the connection. ;-)

## 1.9 Using AmiSlate

[The GUI](#) Where do I click?

[Startup options](#) Command line arguments & ToolTypes

[Menu options](#) Explanation of cryptic menus

[Keyboard Shortcuts](#) For people too lazy to use the mouse

[Serving Suggestion](#) Hints

---

## 1.10 keys

There are keyboard equivalents in AmiSlate for most options available in the menus; those items with keyboard equivalents have the equivalent listed next to the item.

Most keys, if pressed, will transmit a character to the "chat line" at the bottom of the window.

There are a few "special" keys, however. To control palette selection, (at least until I put in a real palette requester) AmiSlate uses the DeluxePaint style of palette control keys, on the numeric keypad. The keys are as follows:

keypad 1 : Decrement Blue in selected color

keypad 2 : Increment Blue in selected color

keypad 3 : Remove all Blue from selected color

keypad 4 : Decrement Green in selected color

keypad 5 : Increment Green in selected color

keypad 6 : Remove all Green from selected color

keypad 7 : Decrement Red in selected color

keypad 8 : Increment Red in selected color

keypad 9 : Remove all Red from selected color

Note that if you have "Lock Palettes" selected, then palette changes will also affect the remote user's screen. Otherwise, they won't.

## 1.11 The AmiSlate GUI

When started without any arguments, AmiSlate opens a requester to allow you to select what type of screen you wish to draw on. You may use this requester to pick either the WorkBench screen or your favorite type of Custom Screen. Note that at the moment, you can only use a Public Screen by specifying it as a Tooltype or command line argument. After you've selected a screen type, AmiSlate will open a window to draw in. Most of the window is taken up by the drawing area, but there are several areas around the right and bottom edges which have different uses:

**The Toolbar** - On the upper portion of the righthand side

**The Palette** - Below the Toolbar

**The Chat Lines** - Along the bottom of the window

**The Menus** - In the traditional Amiga menu spot :)

## 1.12 The Toolbar

The Toolbar consists of eight little icons that run along the right side of the window. Any resemblance these have to Deluxe Paint icons, is of course complete and total coincidence ;-). Seriously, if you have ever used Deluxe Paint or any other half decent paint program, then these are probably self explanatory, but here's the rundown anyway:

Starting from the upper left:

The Dot Tool : Allows you to draw dotted lines by dragging the mouse.

The Pen Tool : Allows you to draw connected lines by dragging the mouse.

The Line Tool : Allows you to draw straight lines by... oh, you know how.

The Circle Tool : Allows you to draw filled or empty circles and ellipses by dragging the mouse from the center of the circle until both radii are where you want them. Clicking on the upper left portion of this icon produces empty circles, while the lower right produces filled ones.

The Square Tool : Allows you to draw filled or empty rectangles by dragging from one corner of the rectangle to the other. Select Filled or unfilled just as with the circle tool.

The Polygon Tool: This allows you to draw shapes out of straight lines. (Sorry, no filled polygons yet!) Click once or draw out a line to start the polygon; you'll notice a little square appear on the polygon tool's icon. This indicates that a polygon draw is in progress. Click or drag out the next line of the polygon, and continue repeating this for as long as you want. When you are done drawing lines, either connect the last line to the original point of the polygon, or click on a Toolbar icon to reset the Polygon tool.

The Flood Tool : Does a "Flood Fill" of the current color at whatever location in the drawing you click on.

The Clear Tool : Clears the drawing window.

Note that these tools can have their effects redefined by an ARexx script; if you want their original functions back, quit the ARexx script or **abort** it.

## 1.13 The Palette

The Palette lets you choose the color you wish to draw with. The current drawing color is shown as "depressed" in the palette array, and the others are shown as "raised." The neat thing about the palette is that it automatically reconfigures itself in accordance with your window and palette size. Well, at least I think it's neat :-)

## 1.14 The Chat Lines

The Chat lines allow you to converse more easily with the person you are connected to over a TCP link. They operate rather simply; If you press a key, it goes into the local chat line (the lower one), and the letter pressed will be sent to the other person's computer as well, where it will show up on the remote chat line (the upper one). The same thing goes in reverse for him/her. The delete and backspace keys can be used to remove boo-boos, but text that has scrolled off the screen cannot be recovered. Resizing or clearing the window will clear the chat lines.

## 1.15 Serving Suggestion

AmiSlate is designed to run as well as possible on a variety of screen sizes and depths. It is also designed to be able to connect between two users who are not running the same screen type. To do this, AmiSlate uses a palette matching scheme where colors are transmitted as 12-bit RGB values and the receiving computer uses the color with the "best fit". This will work, but may lead to less than the best results if the displays differ too greatly. (Specifically, two different colors on a deep display might match to the same color on a more shallow one, causing confusion to the user of the lesser palette, and possibly confusing the flood fill tool as well)

To minimize color matching problems, it is best to run AmiSlate on a screen as much like your peer's as possible. When you receive an AmiSlate connection request, you will be given some information about your partner's screen at the bottom of the screen mode requester. Try to use a screen with the same depth as your partner. Furthermore, the color matching scheme will work best if you and your partner **Lock Palettes** . If you are using the same Screen Depth, this will guarantee a one to one, exact palette match.

AmiSlate is designed to handle many operations simultaneously; it is possible to have both users drawing at once, and it is even possible to have the users drawing at the same time as ARexx scripts are drawing. In practice, however, such heavy loading of AmiSlate may lead to problems. If you are experiencing problems with AmiSlate during "heavy drawing", try easing off a little to let AmiSlate catch its breath. If you find a problem that is reproducible, email **me** about it so that I can try to fix it.

## 1.16 Rexx Scripts included with AmiSlate

There are several ARexx scripts included with AmiSlate; I intend to add to the collection as AmiSlate grows.

To launch any of these, use the "Execute ARexx script" option in the Rexx menu--manually launching them with "rx" or another program may cause them to malfunction.

Also make sure you have your SlateRexx: directory set up as per the **Installation** instructions, and that these scripts are in the SlateRexx: directory.

Note: For games that involve two players, only one person should execute an ARexx script themselves. The ARexx scripts know how to start another ARexx instantiation on the remote machine.

(After asking the remote user's permission, of course)

Chess - My first "big" ARexx project for AmiSlate. Mediates a game of chess that is played between two people. If a network connection is active when you run this script, the Chess program will allow you to play over the connection with a remote opponent, otherwise it will let you play with both players on the same Amiga. The person who initiates the script is always white.

To play, just click and drag a piece where you want it to go. Castling is done by dragging the king onto the rook you want to castle with.

For now, CheckMate isn't detected (it would make things even slower than they are!). Oh yeah, En Passant isn't implemented either, because I'm still a bit unsure as to what that is myself ;).

Oh yeah, the person who initiates the chess game can control whether or not the chess pieces will be filled or line drawings. Click on the "dot" tool for line drawings, anywhere else for filled ones.

TicTacToe - A cute little TicTacToe game for one or two computers.

Autodetects a connection for two-computer mode.

BoxMe - Just a dorky little demo of the interactive capabilities of AmiSlate ARexx; Draws a box around the current mouse position whenever you press or let go of the mouse button.

RainbowPen - Cycles the current Foreground pen through the user's palette so that when you draw, you get a neat "rainbow"

effect.

Flower - A line "orbits" around the center of the drawing area to make a cute little design.

Qix - Draws an animated line that bounces around your screen. The line uses XOR mode, so your drawings will not be affected.

For extra fun, run two or three of these at once, on one or two computers. Watch AmiSlate beg for mercy ;-)

Diamond - Remember those "make curves out of straight lines" projects that you dazzled all of your friends with in the third grade? Well, this draws four of them to form a diamond in the user's current pen color.

Explode - Draws 300 lines worth of violent explosion on your window :)

Explode2 - Draws a little explosion whenever you click or let go of the mouse button.

GhostCircle - A little XOR circle follows your mouse pointer around the canvas. This script demonstrates mouse polling via the WaitEvent timeout feature.

DisplayText - Draws vector text on the canvas for you. When run, you are presented with a box represents the area the text will occupy. You can move it around, or hold down the left mouse button and move the mouse to resize it. (Caution: go a bit slow here, or the script might not catch your mouse clicks. Watch the title bar to be sure). Then press return when it's positioned how you want it. Then type your text into the requester that appears. The text will be previewed for you. You can then choose whether or not you want to keep it.

Coords - Places a continually updating display of the current mouse coordinates and button status in AmiSlate's title bar.

Surround - A simple script to play a game of Snake/Surround/Tron/whatever with the arrow keys. You can play by yourself or two people can play, each by starting it on his/her own machine.

## 1.17 arexx

AmiSlate contains extensive support for ARexx scripting. AmiSlate's ARexx port is designed to run concurrently with the user's input, rather than "taking over" control from the user. This means that your ARexx scripts can be highly interactive, even down to the level of interpreting

---

mouse clicks from the user!

Documentation on ARexx commands is included in this archive, in the file AmiSlateRexx.guide. Also included are several example/test scripts in the ExampleRexx directory, and even a few modest [games and applications](#) in the SlateRexx directory.

## 1.18 Starting AmiSlate

AmiSlate can be started from either the Shell or the WorkBench. It supports a few command line arguments. Each of these arguments also has a ToolType equivalent. Look in the AmiSlate icon for examples of these ToolTypes--they are there, just commented out with parentheses.

If you specify no arguments on the command line, AmiSlate will try to find parameters in an icon with the same name as the executable.

Template: AmiSlate WBScreen/S, TOP/K/N, LEFT/K/N, WIDTH/K/N, HEIGHT/K/N, PUBSCREEN/K, REXXOUTPUT/K, REXXOUTPUT/K, OUTQUEUE/K/N REXXMENU1/K, REXXMENU2/K, {...}

**WBScreen/S** Opens AmiSlate window on Workbench screen

**TOP/K/N** Assigns default Y co-ordinate to window

**LEFT/K/N** Assigns default X co-ordinate to window

**WIDTH/K/N** Assigns default width to window

**HEIGHT/K/N** Assigns default height to window

**PUBSCREEN/K** Opens AmiSlate on given Public Screen

**REXXOUTPUT/K** Sends REXX scripts' output to given file

**OUTQUEUE/K/N** Sets size of Output Queue

**REXXMENU#/K** Assigns a script to a menu item (# = 0-9)

## 1.19 REXXMENU#/K

REXXMENU1 through REXXMENU0 allow you to keep your favorite ARexx scripts just a keypress or menu select away. Any ARexx script that you specify in one of these arguments will be put as a menu item in the REXX menu. The path and .rexx extension (if any) will be stripped from the name in the menu, but will be remembered by the program.

Usage Example: AmiSlate REXXMENU1=SlateRexx:Chess.rexx

---

## 1.20 OUTQUEUE/K

AmiSlate is designed never to block input from the user. (Well, at least as close to never as possible) However, it may be the case that the remote AmiSlate client is not yet able to accept data from the local client; in this case, AmiSlate puts the data generated by user commands into a local queue, and sends it as soon as the remote client is ready for it. Data is not lost until the queue is filled. This keyword allows you to set the size of the queue (the default is about two kilobytes).

To set the size of the queue, specify it in bytes, as shown. A large outqueue will keep AmiSlate from "locking up" during large data sends, like IFF transmission. A small queue will use less memory.

Usage Example: AmiSlate OUTQUEUE=12000

## 1.21 REXXOUTPUT/K

This keyword allows you to specify a destination for any console output that ARexx scripts run by AmiSlate might produce. It is useful for debugging your ARexx scripts.

Usage Example: AmiSlate REXXOUTPUT=ram:myrexx.out

or, more fun: AmiSlate REXXOUTPUT=speak:

## 1.22 Workbench Screen

This keyword causes AmiSlate to appear on the Workbench Screen.

Usage Example: AmiSlate WBScreen

## 1.23 Specify a Public Screen

This allows you to specify the name of a Public Screen upon which you want the AmiSlate window to appear. If a Public Screen with the given name does not exist, AmiSlate will fail with a "Couldn't lock public screen!" error.

Remember, with Public Screen names, alphabetical case counts!

Usage Example:

AmiSlate PUBSCREEN=MyPubScreen

---

## 1.24 Specify window height

This allows you to set the height of the AmiSlate window.

Usage Example:

AmiSlate HEIGHT=250

will make the AmiSlate window 250 pixels tall.

## 1.25 Specify Window width

This allows you to set the width of the AmiSlate window.

Usage Example:

AmiSlate WIDTH=350

will make the AmiSlate window 350 pixels wide.

## 1.26 Specify window X-coordinate

This allows you to set the horizontal position of the AmiSlate window.

Usage Example:

AmiSlate LEFT=60

will make the AmiSlate window appear 60 pixels from the left of the screen.

## 1.27 Specify window Y-coordinate

This allows you to set the vertical position of the AmiSlate window.

Usage Example:

AmiSlate TOP=60

will make the AmiSlate window appear 60 pixels from the top of the screen.

## 1.28 AmiSlate menus

AmiSlate has a few menus for you to play with...

**Project** - The usual

**TCP** - Network connection and disconnection

**Script** - Draw recording and playback

**Options** - Other random stuff

**Rexx** - Launch and abort ARexx scripts

---

## 1.29 Project Menu

Open IFF - Allows you to load an IFF-ILBM graphic file into AmiSlate.

The file can be up to 256 colors deep, and any size (although it may be cropped if it's too big for your window/screen. Note that loading an IFF can be slow, especially if it has lots of color changes (such as a photograph). Also note that the colors of the IFF are not guaranteed to be preserved 100% accurately.

Save IFF - Saves the drawing canvas to an IFF-ILBM file.

About - Opens a little info-window about **me** and the program.

Quit - Makes AmiSlate go away.

## 1.30 TCP Menu

This menu controls connecting and disconnecting to a remote Amiga that has AmiSlate installed.

Connect - Connect to a given host. Note that you don't need to specify the user's name at that host, just the computer's name.

Disconnect - Sort of the Internet equivalent of hanging up, I suppose.

This menu is only enabled if you have AmiTCP running when AmiSlate is started.

## 1.31 Script Menu

This menu controls the recording and playback of AmiSlate's proprietary "scripts". These are NOT ARexx scripts; if you don't believe me, try loading one into your text editor ;). For ARexx scripts, check out the **Rexx** menu.

Play Script - Select a previously recorded script for playback.

Record Script - Start recording drawing events to the file you specify.

To stop recording, select this item once again.

Append Scripts - If this item is checked, and the user chooses the name of an existing file as the output destination of the "Record Script" command, then AmiSlate will add the new data to the end of the old file. Otherwise, AmiSlate will overwrite the old file.

## 1.32 Options Menu

This menu controls options that didn't really belong to any of the other menus.

IFF Load - There are three subitems under this option. They are as follows:

Load Palette - When this is selected, **loading an IFF file** will cause the screen colors to change to those of the IFF file's palette. If this is not selected, the IFF's colors will be mapped into the current palette instead.

This option defaults to true for custom screens, and false for Workbench of public screens.

Expand Window - While this is selected, any IFF file that is loaded into AmiSlate that is bigger than the size of the drawing canvas will cause AmiSlate to try and enlarge its window to accommodate the IFF.

Protect GUI Pens - If this option is selected, pens 0, 1, 2, and 3 of AmiSlate's screen will not be overwritten during IFF palette loading or palette synchronization.

This will keep the window border and gadgets looking normal, at the possible expense of some color accuracy in the pictures loaded.

Safe Flood Fills - This option allows you to toggle between two different methods of transmitting flood fill information to a peer or file. If it is checked, AmiSlate will use **Safe Flood Fills**, otherwise AmiSlate will use **Not-So-Safe Flood Fills**.

Lock Palettes - This option is only accessible if you are connected to another person. Selecting this item will cause your peer's palette to be copied to your palette. If your palette is the same size as your partner's, this will guarantee that you are both using the same palette. If your palettes are not the same size, it will help, but a match is not guaranteed.

Furthermore, as long as this item is selected, any adjustments made by one of you to his/her palette will be transmitted to the other person, thus keeping the palettes "In Synch".

Resynch to Remote - If things get out of whack, you can select this and

your partners screen will be copied to yours, putting you back "in sync".

Reset Palette - If your palette has been modified, selecting this will cause the screen palette to revert to its original configuration.

### 1.33 Safe Flood Fills

In Safe Flood Fill mode, AmiSlate will transmit (or store) a flood fill as a number of horizontal lines. Because the shape of the flood fill is transmitted (or stored), this ensures that any slight inconsistencies between screens (i.e. a missing pixel on the flood boundary) will not lead to enormously different results after the flood. This method is necessarily slower and more disk-space hungry than [the alternative](#) , but is much more reliable, especially in [less than optimal conditions](#) .

### 1.34 Not-So-Safe Flood Fills

In Not-So-Safe Flood Fill mode, AmiSlate will transmit (or store) a flood fill operation as a single set of co-ordinates and a color.

The only error-checking done is to make sure that the color of the first pixel flooded over at the remote end is the same as the color of the first pixel flooded over locally. After that, the flood fill is left free to go its merry way. This can lead to some havoc if you aren't careful with it, but it is much less bandwidth and disk-space intensive than [the other way](#) .

If you wish to use this mode, be sure to read the [Serving Suggestion](#) .

### 1.35 The Rexx menu

This mode controls the launching and aborting of ARexx scripts. Note that AmiSlate runs these scripts asynchronously, so that it is quite possible to have multiple ARexx scripts running at once.

Doing this with scripts that make use of WaitEvent will most likely lead to undesirable results, because each script will only receive information about every other event. (Or every third event, if there are three, etc.)

However, if you are running only non-interactive scripts (explode,

flower, qix, etc.) you should be able to run several scripts simultaneously without problems.

If your ARexx scripts start to get out of hand, select Abort Rexx Scripts to get rid of them.

Execute Rexx Script - Allows you to select an ARexx script to run in conjunction with AmiSlate.

Abort Rexx Script - Causes AmiSlate to send a Quit message to any Rexx script with a pending WaitEvent() call, then to close its ARexx port for a brief period. This should cause most ARexx scripts to go away.

[Custom ARexx Command] - There are ten entries of this type in this menu.

They default to "Unused ARexx Slot", and in their default state cannot be selected. However, using the **REXXMENU#** startup argument, you can program them to execute ARexx scripts of your choice. Once the program has been started, these cannot be changed.

This menu is only enabled if you have ARexx running.

## 1.36 Thanks

Thanks go to the following people:

Matt Dillon for DICE.

Commodore for the nifty computer and O/S.

NSDi for AmiTCP, which made AmiSlate possible.

Pekka Pessi for his help with AmiTCP related problems.

Michael Balzer for ARexxBox, which made the ARexx port coding ten billion times easier.

Scott Ellis for the source code to AmiPop, a lovely TCP programming example.

Jayson Mondala and Brian Gunn for beta testing.

Tim Koch, for making life so bloody interesting... ;-)

my friends, and all the nice people on the Internet who helped me out with Amiga problems.

---

## 1.37 What's new, what's old, what's changed

("-" = new feature, "\*" = bug fix)

1.4 : (Released )

\* Fixed coords.rexx, which was broken.

1.3 : (Released 08/05/95)

- If AmiSlate is started with no arguments, it will read arguments in from the icon associated with the executable (if there is one). This means that you can have startup arguments for a remotely initiated AmiSlate session, by placing an icon with (the AmiSlate executable that you have specified in amitcp:db/inetd.conf)
  - AmiSlate will also look at the icon tooltypes to try to fill in any ARexx scripts for the ARexx menu that you didn't specify on the command line.
  - Added the option to abort from script playback, IFF Loading, and flood fills via the escape key, and the ability to quit AmiSlate during all of the above via the close box.
  - GhostCircle.rexx now prompts the user for circle radius, and is faster.
  - Added **DisplayText.rexx** to the included ARexx scripts directory.
  - Added **Surround.rexx** to the included ARexx scripts directory.
  - Added **Coords.rexx** to the included ARexx scripts directory.
  - Added the MouseX, MouseY, and Button result fields to the WaitEvent ARexx command.
  - Added new result fields to the WAITEVENT ARexx commands return record. Namely: LastKey, MouseX, MouseY, and Button are now returned.
  - Added the MOUSEMOVE event to WAITEVENT's repertoire.
  - Upgraded ARexx scripts to take advantage of the above two additions
  - Implemented a (hopefully) somewhat better palette arbitration screen: now if there is pen contention (i.e. trying to stuff a 32 color pic into an 8 color palette), the higher colors will not replace the lower ones; rather the pens will be set to the "average" of the lower and higher colors. If you think this is worse than the old scheme, let me know!
  - Most Rawkey events are now detected by WAITEVENT and dealt with as KEYPRESS. (i.e. you can now use the arrow keys, etc. with ARexx scripts)
  - \* Changed the behavior of GETPIXEL so as not to cause error messages.
  - \* New radio button choice of "update/install" to Installer script
  - \* Fixed the chess game so that it runs properly under a 4-color screen.
-

- \* Removed the 2000 pixel width limit on IFF pictures.
- \* ARexx-drawn circles are now clipped properly on the bottom of the canvas.
- \* The REXXMENU0 option now works.
- \* Fixed a bug that would cause crashes if you did a WAITEVENT 0.

1.2 : (Released 5/25/95)

- Now works with AmiTCP4.0! Actually, it always did, but the AmiSlate line in inetd.conf was slightly wrong. Changing the "wait" keyword to "nowait" fixes it.
  - Added (somewhat compressed) raster data transmission.
  - Added "Load IFF" and "Save IFF" features, along with more options in the Options menu (formerly the misc. menu) to support them.
  - Improved palette handling. Now loading a picture's palette and/or locking your palette to your partner's screen will assign colors to pens in the least disturbing manner possible.
  - Added four new ARexx commands, SetRaster, PutRasterPixels, LoadIFF, and SaveIFF.
  - Slightly improved fairness of task dispatching... AmiSlate instances running at low priority should no longer be completely taken over by "busy-waiting" ARexx scripts.
  - The timeout feature of the WaitEvent ARexx command is implemented.
  - There is now a nice Installer script included with the distribution.
  - The MouseMove queue is made shorter during Circle dragging to reduce needless "catching up" and make the tool more responsive.
  - added "GhostCircle" ARexx script
  - The ASL file requester will "remember" directories for IFF pics.
  - "Lock Palettes" now functions differently, and is always available regardless of any palette size.
  - When the Output Queue is completely full, AmiSlate will now switch to Synchronous transmission, instead of losing data.
  - AmiSlate's maximum window size is now set to the smaller of the two screen sizes in a connection, so there should be no more problems with one person sizing their window too large for the other person's screen. (Actually, there never were any problems with this, but it's "cleaner" now...)
  - \* Fixed a nasty bug that would cause the flood fill tool to sometimes go into an infinite recursion and eat all your memory.
  - \* The circle tool will no longer "seize up" when you drag the mouse off of the canvas. Rather, it will still resize in whatever
-

dimension it still has room for.

\* The circle ARexx command now reduces the size of circles that are too large to fit on the screen, instead of failing. This makes it consistent in operation with the other tools.

\* The "in use" dot on the Polygon tool now is drawn in the correct place regardless of the screen font.

\* The "minimum window height" value now depends on the title font size.

\* The output queueing was broken. Fixed.

1.1 : (Released 3/24/95)

- ARexx scripts now more informative if the user tries to run them manually from the shell.

- Implemented DisplayBeep, SetUserFColor, and GetPixel REXX commands.

- Diamond.rexx now draws more cleanly.

- Added Append Scripts option to script menu.

- Added Castling, Pawn promotion, error messages, and filled/line mode to the Chess game.

- Added Explode2.rexx to distribution.

- Added customizable quick-run entries to the REXX menu.

- Typing "AmiSlate ?" from the CLI now gives a command template

\* Added Diagonal win lines to the TicTacToe game, and corrected a bug that caused win lines not to be drawn for player X in one-client mode.

\* SetFColor and SetUserFColor now return the pen number in rc2 instead of rc. ARexx was giving error messages whenever a high pen number was returned.

\* The ScreenMode requester now handles large font sizes and 3.0 style ListViews somewhat better.

\* Fixed an IntuiMessage leak that caused tools to stop working after about 3 seconds under AmigaDos 3.x.

1.0 : (Released 03/11/95)

- First release. Undoubtedly full of bugs.

## 1.38 What's Next?

Note: These are things I'm thinking of implementing; Whether I actually implement them or not depends on how difficult they will be to implement and user response (both in the form of **communications** and **donations** ).

- Implement palette mapping better. The current method is rather ad hoc.

---

- Better control of screen selection
- More ARexx scripts
- . More little screen-drawing ditties (Moire, etc.)
- . 3D animated spinning vector cubes, just like those European guys have ;)
- . Anything neat sent to me by users :-)
- . A script that ties in with DigiView to give a periodically updated remote video display. If only I hadn't sold my DigiView...
- More ARexx commands as necessary (it's good to be the king!)
- Implement Background Color support (a la DPaint with right mousebutton)
- AS225 support
- DCC/Grapevine IRC interface
- An option for scrolling over a "virtual slate" instead of keeping the windows the same size
- Bug Fixes Galore
- Add an option to specify the path to "rx", rather than assuming it is in the system's search path.

### 1.39 Known Bugs and Other Problems

Here are the things that still don't work right with AmiSlate V1.0:

- Aborting ARexx scripts from the menu item can have a sllloooowww response time if a non-waiting ARexx script (like qix.rexx) is running. Try using the key-combination instead, for some reason it's faster.
  - Running multiple ARexx scripts from AmiSlate at once and then aborting all scripts can cause Enforcer hits, and even crashes. Be careful!
  - Background color pens/drawing are not implemented.
  - A few ARexx commands are not done. See AmiSlateRexx.guide for details.
  - You can select up to 8 bitplanes on any ScreenMode, not just up to the number of bitplanes that ScreenMode can actually do on your Amiga.
  - There is no en passant in the chess game.
  - An AmiSlate session that was started by inetd sometimes isn't able to connect to any other AmiSlate processes via the "Connect" menu item. Actually, it connects, but the connection is immediately closed (!?)
  - There is no accounting for aspect ratios of varying screenmodes.
  - Opening an IFF for the first time causes a 64 byte memory leak.
  - Resizing the window while in positioning mode for DisplayText.rexx causes garbage to be left on the screen. Don't do it.
-