RexxNet

COLLABORATORS

	TITLE : RexxNet		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 8, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

RexxNet

1.1 "

Documentation for

RexxNet v1.3

A Client and Server for Sending/Receiving
ARexx Messages Over A Network

Written by Michael Wood

1.2 what is rexxnet

What Is RexxNet?

Well, RexxNet is really two programs. The first program is called rexxnetd, and it is the "Server" program. The other program is called rexxnet and it is the "Client" program.

When these two programs are run on two different Amigas connected to a network, they allow ARexx messages to be passed between them as if the ARexx messages were being passed between ports on the same Amiga.

```
rexxnetd - The "Server" rexxnetd, accepts connections from a "Client" Amiga.
           The data packets it processes are interpreted as being ARexx
           messages destined to a specific message port on the "Server" Amiga.

rexxnet  - The "Client" rexxnet sends ARexx messages it receives at its port,
           over to the "Server" Amiga.
```

Normally you can only send ARexx messages between applications on the same Amiga...

```
+------------------------------------------------------------------------+
|                                                                        |
|     Amiga "A"                                                          |
|                                                                        |
|                                                                        |
| +-----------------+                                +-----------------+ |
| |                 |                                |                 | |
| | Application "A" |                                | Application "B" | |
| |                 |                                |                 | |
| |                 |<--------ARexx Messages-------->|                 | |
| |                 |                                |                 | |
| |                 |                                |                 | |
| +-----------------+                                +-----------------+ |
|                                                                        |
|                                                                        |
+------------------------------------------------------------------------+
```

Using rexxnetd and rexxnet you can send ARexx messages between applications on two or more Amigas.

```
+----------------------------------+            ↩
   +---------------------------------+
|                                  |          |  ↩
                                     |
|      Amiga "A"                   |          |                Amiga "B"  ↩
           |
|                                  |          |  ↩
                                     |
|                                  |          |  ↩
                                    |
| +-------------+                  |          |                  ↩
   +-------------+ |
| |             |  +----------+    |          |    +----------+  |  ↩
               | |
| | Application +->+ rexxnet  +->+ |          | +<-+ rexxnet  +<-| Application ↩
    | |
| |             |  +----------+  | |  TCP/IP  | |  +----------+  |  ↩
               | |
| |     "A"     |                +-+<-------->+-+                |     "B"  ↩
       | |
| |             |  +----------+  | |          | |  +----------+  |  ↩
               | |
| |             +<-+ rexxnetd +<-+ |          | +->+ rexxnetd +->|  ↩
               | |
| |             |  +----------+    |          |    +----------+  |  ↩
               | |
| +-------------+                  |          |                  ↩
   +-------------+ |
|                                  |          |  ↩
                                    |
|                                  |          |  ↩
                                    |
+----------------------------------+           ↩
   +----------------------------------+
```

1.3 quick start

Quick Start

For those who don’t want to read my documentation, which I worked so hard on...

You can quickly test RexxNet by performing the following:

- Make sure you have two Amigas running on a network

- Run "Install-RexxNet" on both machines to set things up

- From the WorkBench on Amiga #1:

    - Run rexxnetd (double-click on it)

- From the WorkBench on Amiga #2:

    - Edit the ToolTypes of rexxnet:

        - Set the "HOST=" to the network name of Amiga #1

        - Set the "PORT=" to an application ARexx port name on Amiga #1

        - Save your ToolType changes

```
        - Run rexxnet on Amiga #2 (double-click on it)

        - You can now send ARexx messages to Amiga #1’s ARexx port from
          Amiga #2.
                 Examples Usages
                        - You can terminate the programs from the WorkBench by  ↩
                            selecting
          the menu items "Quit rexxnetd" and "Quit rexxnet:<portname>" from
          the WorkBench menus.
```

1.4 hardware|software requirements

Hardware/Software Requirements

TCP/IP Software:

Version 1.3 of RexxNet requires TCP/IP transport software be installed and running before executing the RexxNet programs. RexxNet has been tested on the following TCP/IP packages.

AS225r2
Canadian Protype Replicas (compatible with AS225r2)
AmiTCP 4.3

Hardware:

This software has been tested on A3000’s and A4000’s. Various 68040 40Mhz cards were used on the testing. They all appear to work.

Amiga OS:

I’ve only tested the software under OS version 3.1.

Compatibility with Amiga programs:

I’ve tested RexxNet with the following Amiga programs through their ARexx port interfaces:

ARexx
TurboText 2.0
Professional Page 4.1
Directory Opus 4.12

I admit that a thorough job of testing on different configurations has not been done. If you come up with a configuration problem, let me know. I might attempt to fix it.

1.5 installation

Installation

* For AS225, replace <net> with "inet:".
For AmiTCP, replace <net> with "AmiTCP:".

The installation consists mainly of copying the files rexxnetd and rexxnet to your harddrive and modifying a TCP/IP file. The placement of these files is really up to you but most "Clients" go into the "inet:c" or "AmiTCP/bin" assigned location and most "Servers" go into the "<net>:serv" assigned location.

To Install, Perform One Of These:

A) Use the supplied "Install-RexxNet" program and follow the directions. (This is the BEST method)

B) Install Manually

- First you have to determine which version of the executables you want to install. There are 4 versions of each file. There versions for AS225 and AmiTCP, and there are versions for the kind of CPU you are running.

  If you are using AS225, the executables you need are in the "Binaries/AS225" directory.

  Within this directory are files named:

  ```
  rexxnetd_as225.000 - Server for 68000 CPUs
  rexxnetd_as225.020 - Server for 68020+ CPUs
  rexxnet_as225.000  - Client for 68000 CPUs
  rexxnet_as225.020  - Client for 68020+ CPUs
  ```

  Likewise if you are using AmiTCP, the executables are in "Binaries/AmiTCP".

  Within this directory are files named:

  ```
  rexxnetd_amitcp.000 - Server for 68000 CPUs
  rexxnetd_amitcp.020 - Server for 68020+ CPUs
  rexxnet_amitcp.000  - Client for 68000 CPUs
  rexxnet_amitcp.020  - Client for 68020+ CPUs
  ```

- Copy the Client rexxnet_xxx.0x0 to "inet:c/" for AS225 or "AmiTCP:bin/" for AmiTCP and rename it to just "rexxnet".

- Copy the Server rexxnetd_xxx.0x0 to "<net>:serv/" and rename it to just "rexxnetd".

- Edit the "<net>:db/services" file.

  You must edit this file to provide a mapping between the named service and a port number for the rexxnetd "Server". I chose the number 2001 because I liked it and it was high enough on the list not to interfere with anything else. You or your system administrator may pick another number.

  Add an entry into the "services" file like this:

  ```
  rexxnetd    2001/tcp
  ```

This is what my "inet:db/services" file looks like:

```
+---------------------------------------+
|                                       |
|  echo          7/udp                  |
|  echo          7/tcp                  |
|  discard       9/udp                  |
|  discard       9/tcp                  |
|  systat       11/tcp                  |
|  daytime      13/tcp                  |
|  netstat      15/tcp                  |
|  chargen      19/udp                  |
|  chargen      19/tcp                  |
|  ftp-data     20/tcp                  |
|  ftp          21/tcp                  |
|  telnet       23/tcp                  |
|  smtp         25/tcp                  |
|  time         37/tcp                  |
|  time         37/udp                  |
|  whois        43/tcp                  |
|  domain       53/udp                  |
|  domain       53/tcp                  |
|  sunrpc      111/udp                  |
|  sunrpc      111/tcp                  |
|  tftp         69/udp   tftp           |
|  finger       79/tcp                  |
|  nntp        119/tcp                  |
|  exec        512/tcp                  |
|  login       513/tcp                  |
|  shell       514/tcp                  |
|  printer     515/tcp                  |
|  syslog      514/udp                  |
|  talk        517/udp                  |
|  ntalk       518/udp                  |
|  router      520/udp     routed route |
|  timed       525/udp                  |
|  infranet   1025/udp                  |
|  rexxnetd   2001/tcp                  |
|  netser     3000/tcp                  |
|                                       |
+---------------------------------------+
```

- Optional: You may want to configure your Amiga to run rexxnetd during
  bootup. It doesn’t affect performance and it only uses up
  just a few kilobytes of memory.

Two effective methods are:

- You could include the following line in your "s:user-startup" file ↩
  after
  the TCP/IP startup is completed:

```
"run <>nil: inet:serv/rexxnetd"
```

OR

```
                "run <>nil: AmiTCP:serv/rexxnetd"


          - Or you could copy the icon file "Binaries/AS225/rexxnetd.project.info"
            to "System:WBStartup/" and rename it "rexxnetd.info". After copying
            the file over, select the icon and then select the WorkBench menu:
            "Icons->Information". Modify the default tool to be the path to where
            you placed the file "rexxnetd".

            (Make sure "DONOTWAIT=TRUE" is in the ToolTypes)

          Note:
          -----
            You can run the RexxNet programs anytime after the TCP/IP startup
            has completed. It doesn't have to be during the startup.
```

1.6 running rexxnet

```
                Running RexxNet

    You have to run rexxnetd  on one Amiga and rexxnet on another Amiga before
    you can send messages between them.

    rexxnetd: (The Server)
    ---------
        This program can be ran from either the CLI or from WorkBench.

        Running this program allows an Amiga to be used as a "Server" for
        receiving ARexx messages over a network.

        Messages arriving here from across the network, are converted into ARexx
        messages and are sent to the appropriate ARexx message port.

        The CLI arguments to rexxnetd are:

            "rexxnetd <-s> <-l>"

            -s  (OPTIONAL)

                This flag tells rexxnetd NOT to create its own window for
                output, but to use StandardOutput (usually the CLI the program
                was started from).

            -l  (OPTIONAL)

                This flag tells rexxnetd  to send errors to the error
                logger used by TCP/IP.

            -v  (OPTIONAL)

                This flag tells rexxnetd  to verify the user name and
                password before letting a Client log onto rexxnetd .
                If an incorrect password is received, the connection is broken
                and a message is displayed.
```

AS225 Version:
If this option is requested but the host that attempts to log into you is in the file "inet:db/hosts.equiv", their name and password are
NOT requested. This is because the names in this file are considered
trusted hosts.

The WorkBench ToolType arguments are:

"STDOUT"

Same as -s from the CLI.

"LOGERRORS"

Same as -l from the CLI.

"VERIFY"

Same as -v from the CLI.

You can terminate rexxnetd by either of these methods:

- By sending it a Control-C.
- By selecting the menu item "Quit rexxnetd" from WorkBench’s menu.
- By sending the ARexx message "RND_Quit" to rexxnetd’s port.

  You can do this from the CLI by typing:

```
rx "address 'REXXNETD' 'RND_Quit'"
```

When you terminate rexxnetd, it will signal any "Client" rexxnet programs that are connected to it, that their "Server" is going away. The rexxnet programs will then self-terminate.

This version of the "Server" allows a maximum of 16 "Client" connections at any one time.

rexxnet: (The Client)

This program can also be ran from either a CLI or WorkBench.

This program creates a public ARexx message port of the name supplied to it, on the Amiga it’s ran from. Any ARexx messages sent to this port are sent across the network to the remote "Server". From there the messages
are routed to the correct ARexx message port.

The CLI arguments to rexxnet are:

```
"rexxnet [host] [remote port name] <-m> <-s> <-l>"
```

host (REQUIRED)

This is the host name of the remote Amiga to send ARexx messages to. This can be either dotted decimal notation or the aliased name ↩
.

remote port name (REQUIRED)

This is the public, named ARexx message port on the remote Amiga to send messages to. If the remote portname already exists on this Amiga, you must prefix the remote port name with a "@" character. rexxnet creates a message port of this name on the Amiga it's ran on.

-m (OPTIONAL)

(mirror) This flag signifies that any ARexx messages sent to the remote message port, are also sent to the message port of the same name on the Amiga rexxnet is ran from. (confusing huh? I'll explain with more detail in the
Example Usages
section).

A sticky problem with "Message Mirroring" is that both messages sent out may return a result. And they may both be different! What rexxnet does is return to you the result from the local message only. If the mirroring flag is not set, you get the result from the remote message.

-s (OPTIONAL)

This flag specifies NOT use rexxnet's window for user messages. Instead send the output to StandardOutput (usually the CLI the program was started from).

-l (OPTIONAL)

This flag tells rexxnet to send errors to the error logger used by TCP/IP.

The WorkBench ToolType arguments are:

"HOST=<name>" - REQUIRED

Same as [host] from the CLI.

"PORT=<port>" - REQUIRED

Same as [remote port name] from the CLI.

"MIRROR" - OPTIONAL

Same as -m from the CLI.

"STDOUT" - OPTIONAL

Same as -s from the CLI.

"LOGERRORS" - OPTIONAL

Same as -l from the CLI.

An advantage of using the WorkBench icons to launch rexxnet is that you could have an icon for each "Amiga/ARexx" combination that you want to make a connection to.

Example Usage:

```
"rexxnet amiga004 @TURBOTEXT"
```

If the connection was successful you should see the message:

```
"rexxnet: Connected To Host:amiga004 Port:TURBOTEXT"
```

If successful, this would create a port named @TURBOTEXT on the Amiga rexxnet was started from. Any messages sent to @TURBOTEXT would be routed to Amiga "amiga004". From there the message is sent to the port TURBOTEXT.

(The leading "@" is stripped off at the destination).

When an ARexx message is sent to rexxnet’s port, it packs up the message and sends it over the network to the destination Amiga. The message will act the same as when it is sent to an application on the same Amiga. All result codes are returned properly along with any result strings (rm_Result1 and rm_Result2 for you ARexx programming gurus).

You can terminate rexxnet by either of these methods:

- By sending it a Control-C.
- By terminating the ’Server’ it’s connected to.
- By selecting the menu item "Quit rexxnet:<portname>" from WorkBench’s menu.
- By sending the ARexx message "RN_Quit" to rexxnet’s communication port. rexxnet creates an ARexx message port named "REXXNET.<portname>".
  If rexxnet was called with the portname "@TURBOTEXT", the communication port name is "REXXNET.@TURBOTEXT".

  You can terminate rexxnet under these parameters by typing the following from the CLI.

  ```
  rx "address ’REXXNET.@TURBOTEXT’ ’RN_Quit’"
  ```

Note:

You MUST run the "Server" program "rexxnetd" on one Amiga before running the program "rexxnet" on the other Amiga.

1.7 possible problems

Possible Problems/Limitations

Possible Problems:

These are some of the warning messages from the programs and why you might encounter them.

From rexxnetd :

- "rexxnetd: Can’t Run Without An AmiTCP Compatible Network."
- "rexxnetd: Can’t Run Without An AS225 Compatible Network."
  You are trying to run an AS225 version of RexxNet on an AmiTCP network or, you are trying to run an AmiTCP version of RexxNet on an AS225 network. Rerun the "Install-RexxNet" installer to get the appropriate version.

- "rexxnetd: Rexxnetd Is Already Running."
  You can only have one copy of rexxnetd running at any one time. This is not a limitation. You only need one copy running at any one time.

- "rexxnetd: Can’t Run Without A Network."
  RexxNet requires a network. I look for the existence of either of these public ports to see if a network is up: "INET Exchange" or "AMITCP".

- "rexxnetd: Failed To Open Required Libraries."
- "rexxnetd: Couldn’t Open socket.library." (AS225)
- "rexxnetd: Couldn’t Open bsdsocket.library." (AmiTCP)
- "rexxnetd: New Socket Failed To Open bsdsocket.library" (AmiTCP)
  One or more of the required libraries either don’t exist or are of insufficient version. Required libraries are:

  ```
  graphics.library V37
  intuition.library V37
  utility.library V37
  workbench.library V37
  rexxsyslib.library (any)
  socket.library (any) (AS225 and compatibles)
  bsdsocket.library V3 (AmiTCP)
  ```

- "rexxnetd: Couldn’t Open Server Task Port."
  I failed to create a named, public, ARexx message port. This might happen if a naming conflict is occurring. The names I use for ARexx message ports for each "Client" that logs on are: "REXXNETD.n" where "n" is a number and "REXXNETDS.n" and "n" again is a number. This could happen if this program failed to clean up its ports from previously being run.

- "rexxnetd: Failed To Bind Socket."
  The "Server" failed to "bind" itself to to its address. This could be caused if there is not a correct entry for rexxnetd in the

"<net>:db/services" file.

- "rexxnetd: Failed To Get Local Host Info."
  There needs to be an entry for the name/address of your computer in the "<net>:db/hosts" file. This means it's missing or you have the wrong host file.

- "rexxnetd: Failed To Get Server Service By Name."
  An entry for rexxnetd was not found in the "<net>:db/services" file. Read the installation section or use the Installer to edit the file.

- "rexxnetd: An Unknown Host Tried To Connect..."
  A connection was attempted to this "Server" but the remote host was not listed as a trusted host. Trusted hosts are listed in the "<net>/db/hosts" file.

- "rexxnetd: Received Signal To Quit."
  This is not an error. This message means that rexxnetd received either a CONTROL-C or the ARexx message "RND_Quit".

From rexxnet:

- When I attempt to log unto the Server I'm presented with a requester asking for my name and password.
  This is because the Server was started up with the -v (VERIFY) option. If the Server is running under AS225, it also means that you are not considered a 'trusted host'.

- The Server doesn't accept my name or password.
  (Assuming you are typing it corrrectly...)
  AmiTCP:
  Your host name might not have the 'allow' keyword in the "inet.access" file, or the password stored on the Server is not what you think it is.
  AS225:
  The password stored on the Server is not what you think it is.

- "rexxnet: Can't Run Without An AmiTCP Compatible Network."
- "rexxnet: Can't Run Without An AS225 Compatible Network."
  You are trying to run an AS225 version of RexxNet on an AmiTCP network or, you are trying to run an AmiTCP version of RexxNet on an AS225 network. Rerun the "Install-RexxNet" installer to get the appropriate version.

- "rexxnet: Connected To Host:xxx Port:xxx."
  This isn't an error. It means the connection worked.

- "rexxnet: Server Full, Cannot Accept Additional Connections."
  The registered version of the Server can handle up to 16 connections at any one time. In order to make an additional connection, one of the previous connections must terminate first.

- "rexxnet: Unknown Message From Server."
  During the initial connection to the Server, some communication messages are

```
        passed back and forth. This message indicates that an unknown message was received
        from the Server. The Client will terminate if this occurs because it indicates an
        unathorized Server or an out-of-sync condition.

    - "rexxnet: Rexxnet Already Started 'Client' For This Port"
        This means that an instance of rexxnet has already been started
        to this machine for that ARexx message port.

    - "rexxnet: Failed To Allocate Memory For Reply Buffers"
        rexxnet allocates memory buffers for receiving ARexx messages. There
        is insufficient memory for those buffers.

    - "rexxnet: Failed To Allocate Memory For Message Buffers"
        rexxnet allocates memory buffers for sending ARexx messages. There
        is insufficient memory for those buffers.

    - "rexxnet: Attempting Connection..."
        Rexxnet is attempting to connect to the Server. Rexxnet will timeout
        after about 20 seconds. Another message will occur after this one
        which says whether the connection was successful or not.

    - "rexxnet: Failed To Connect To rexxnetd Server."
        This error occurs when rexxnet fails to connect to the "Server"
        rexxnetd. A more detailed error message should have been displayed
        before this one describing why.

    - "rexxnet: Server Disconnected."
        Not an error. This means that rexxnet received a message from its
        "Server" that the "Server" was terminating. If the "Server"
        terminates, any "Clients" connected to it are signaled to terminate
        also.

    - "rexxnet: Failed To Find Host."
        The "Server" host was not found in your "<net>:db/hosts" file.

    - "rexxnet: Access Denied To Server:xxx."
        The "Server" does not have your Amiga in its list of trusted hosts.
        You cannot connect until the "Server's" host file has been modified.

    - "rexxnet: Client Failed To Find Server's Service."
        An entry for rexxnetd was not found in your "<net>:db/services file.
        Read the installation section or use the Installer to edit the file.

    - "rexxnet: Timeout Waiting For Server."
        A timeout occured waiting for the Server to validate your access.
        Either the connection took too long or you do not have access
        rights to the Server.

    - "rexxnet: Failed To Open Required Libraries."
    - "rexxnet: Couldn't Open socket.library." (AS225)
    - "rexxnet: Couldn't Open bsdsocket.library." (AmiTCP)
    - "rexxnet: New Socket Failed To Open bsdsocket.library" (AmiTCP)
        One or more of the required libraries either don't exist or are of
        insufficient version. Required libraries are:
            graphics.library V37
            intuition.library V37
            utility.library V37
            workbench.library V37
            rexxsyslib.library (any)
            socket.library (any) (AS225 and compatibles)
            bsdsocket.library (3) (AmiTCP)

    Limitations:

        I hate limitations but RexxNet V1.3 has some. These are the ones I know
        about because of my design.

        - The Server can handle a maximum of 16 connections at any one time.

        - These fields from the original ARexx message structure have
          limitations or are not supported.

          RexxMsg->rm_TaskBlock - Is changed to contain the pointer to rexxnetd's
                                  task block (the Server).
          RexxMsg->rm_LibBase   - Is changed to contain the pointer to rexxnetd's
                                  rexxsyslib.library base.
          RexxMsg->rm_PassPort  - Is not supported because it is a local pointer.
          RexxMsg->rm_CommAddr  - Is limited to 31 chacters.
          RexxMsg->rm_FileExt   - Is limited to 7 chacters.
          RexxMsg->rm_Stdin     - Is set to NULL.
          RexxMsg->rm_Stdout    - Is set to NULL.
          RexxMsg->rm_avail     - Is not supported.

        As far as I know now, these are all of RexxNet's limitations. None of
        these limits have caused problems with any ARexx programs I've used
        to date. If you encounter a problem because of one of these, let me know.
```

1.8 example usages

```
            Example Usages

    Ok, here are some real world examples on how to use RexxNet.

    In these examples, I'm assuming two Amigas are currently connected to a
    network using TCP/IP transport software. The host names of these two Amigas
    are "amiga001" and "amiga002" (host names are specified in the
    "<net>:db/hosts" table).

    I'm also assuming that the server software rexxnetd is running on
    both Amigas and REXX is running on both Amigas.

    Scenario #1:
    -----------
    I want to send ARexx messages from "amiga001" to ProPage which is running on
    "amiga002".
             Here's How...
                Scenario #2:
```

I want to send ARexx messages from "amiga001" to TurboText which is running on "amiga002". TurboText is also running on "amiga001".

Here's How...

Scenario #3:

I want to send ARexx messages from "amiga001" to ProPage and TurboText which are running on "amiga002". TurboText is also running on "amiga001".

Here's How...

Scenario #4:

I want to execute an ARexx script on "amiga001" and have the script send commands to TURBOTEXT at "amiga002" and to TURBOTEXT on the local Amiga "amiga001". (This is called "Message Mirroring")

Here's How...

Scenario #5:

I want to send ARexx messages from "amiga001" to TurboText which is running on "amiga002". I also want to send ARexx messages from "amiga002" to TurboText which is running on "amiga001".

Here's How...

1.9 "

Heres How:

ProPage is not running on "amiga001". We can run rexxnet from a CLI on "amiga001" by typing:

```
"rexxnet amiga002 PPAGEAREXX"
```

From a CLI on "amiga001" you could send the following commands:

```
Create a new project
rx "address 'PPAGEAREXX' 'PPM_New'"

Set the doc name
rx "address 'PPAGEAREXX' 'PPM_SetDocName RexxNet'"

Create a new page
rx "address 'PPAGEAREXX' 'PPM_CreatePage 1 1 0 0 0'"

Set the magnification to 100%
rx "address 'PPAGEAREXX' 'PPM_SetMagMode 100'"

Create A Box
rx "address 'PPAGEAREXX' 'PPM_CreateBox .5 .5 4 2 0'"

Set Edit Mode
rx "address 'PPAGEAREXX' 'PPM_SetEdit 0'"

Enter Text into the box
rx "address 'PPAGEAREXX' 'PPM_InsertText' ''Testing RexxNet'''"
```

```
            End Edit Mode
            rx "address 'PPAGEAREXX' 'PPM_EndEdit'"
```

1.10 "

Heres How:

```
    Since TurboText is running on both Amigas, you have to run rexxnet with the
    "@" sign prefixed to the portname. From a CLI on "amiga001", type the ↩
       following:

            "rexxnet amiga002 @TURBOTEXT"

        From the CLI you could send the following commands:

            Create a new document
            rx "address '@TURBOTEXT' 'OpenDoc'"
```

1.11 "

Heres How:

```
    You can run as many rexxnet tasks as you have memory for. There is a limit
    on the number of socket connections that the Server rexxnetd will permit. You ↩
       can
    run both rexxnet's from the same CLI by using the following method:

            "run <> nil: rexxnet amiga002 @TURBOTEXT"
            "run <> nil: rexxnet amiga002 PPAGEAREXX"
```

1.12 "

Heres How:

```
    "Message Mirroring" is turned on by using the "-m" flag on rexxnet.
    The result strings returned are the results from the local TURBOTEXT.

            "rexxnet amiga002 @TURBOTEXT -m"

    Your ARexx script will now send its TURBOTEXT commands to both TurboText
    applications on both Amigas.
```

1.13 "

Heres How:

Since TurboText is running on both Amigas, you have to run rexxnet with the "@" sign prefixed to the portname. From a CLI on "amiga001", type the following:

```
"rexxnet amiga002 @TURBOTEXT"
```

From a CLI on "amiga002", type the following:

```
"rexxnet amiga001 @TURBOTEXT"
```

If you send a message to port "@TURBOTEXT" from either Amiga, that command will be sent to the other Amiga.

1.14 "

Design

These programs were written using the "standard" socket interface for TCP/IP communications. There are currently two "standards" on the Amiga for the socket interface; the "socket.library" that is compatible with AS225 and the "bsdsocket.library" which is compatible with AmiTCP.

Design Requirements:

A requirement of this design was to create a TCP/IP Client and Server that would allow ARexx messages to be sent across a network. The Client and Server should fully support all aspects of ARexx messages. This includes sending and receiving messages of the maximum size allowed by ARexx.

My second design goal was to make this occur as quickly as possible.

Another design goal was to allow "Message Mirroring". Message Mirroring (my terminology), means to allow a message to be sent across a network to a remote application of a specific portname (example: "PortA") and then send the message again to the application of the same name ("PortA") on the local Amiga where it originated. Why this requirement? On the applications that we produce at work, this feature allows us to send the same message to a remote Amiga and to the Amiga we are working on, from one application. This feature allows us to control two Amigas and keep them in sync.

For example:

Lets say we have a text file on a network. We could write an ARexx script that would tell TurboText or ProPage or whatever, to load the file, manipulate it, and print it. With RexxNet running, in Message Mirroring mode, the exact same operations would appear on both Amigas. You could use it as a teaching tool to show how certain operations could be performed. It also allows you to control a remote Amiga and see the results of your operations on your Amiga.

Program Limits:

The current version has these built in limitations (I hate limits).

- A maximum of 16 connections are permitted to the Server. This seemed like a good compromise between usability and memory usage.
- RexxNet will not pass the rm_PassPort field in the RexxMsg structure because it is a local address pointer. This value is set to NULL at the destination.
- RexxNet will not pass the rm_Stdin field in the RexxMsg structure because it is a local address pointer. This value is set to NULL at the destination.
- RexxNet will not pass the rm_Stdout field in the RexxMsg structure because it is a local address pointer. This value is set to NULL at the destination.
- RexxNet will not pass the rm_avail field in the RexxMsg structure because I could never find something that used it.
- RexxNet will only pass the first 31 characters of the string pointed to by the rm_CommAddr field in the RexxMsg structure.
- RexxNet will only pass the first 7 characters of the string pointed to by the rm_FileExt field in the RexxMsg structure.

These programs have been tested with Enforcer and Mungwall. No respectful Amiga programmer should do elsewise with their code.

1.15 registering

Registering

The RexxNet Package is "Giftware". A gift price of $7.00 U.S. would be nice. Please support what shareware Amiga programmers are left with a few dollars.

If you do regsiter, I'll make sure that any bugs fixes or improvements are quickly sent off to you. E-mail support is also available.

You can send your gift to:

Mike Wood
2828 168th Street SE
Bothell, WA 98012
USA

mikew@atl.com

Please state what method you prefer in receiving the program.

```
    I can E-mail the upgrade. (preferred)
    I can send a disk thru regular mail.

    Even if you don't register, you're still welcome to e-mail me with your ↩
       comments.
```

1.16 legal stuff

```
Legal Stuff

    You are not allowed to sell RexxNet for profit, except for reasonable
    copying/media charges. You are not allowed to use RexxNet in or as part of
    a commercial release without prior written approval. This program may be
    copied to BBS's and included in freely distributable software packages
    (such as Fred Fish or AmiNet disks).

    "TurboText"        - is a registered trademark of Martin Taillefer and Oxxi.

    "ProPage"          - is a registered trademark of Gold Disk.

    "Directory Opus" - is a registered trademark of InovaTronics Inc.

    "AS225r2"          - is a registered trademark of Amiga Technologies
                         (I guess???)

    "AmiTCP/IP"        - is a registered trademark of NSDi - Network Solutions
                         Development Inc, Finland.
```

1.17 the author

```
The Author

    I'm an Amiga programmer since 1987 and I've been doing it as my source of
    income since 1990 (It's great to get paid for having fun). I was a commercial
    Amiga Developer until CBM went out of business. I'm mostly into writing ↩
       medical
    software for the Amiga. I'm the author of the TeleRadiology software that
    was distributed through National Diagnostic Imaging, Inc. (There are some
    TeleRadiology units out there in hospitals and doctor's homes that are based
    upon an Amiga 3000 and the A2410 Display Card!)

    I also write some software for OS/2 (the next best thing).

    * TeleRadiology is the digitizing and transmission of medical images
      (X-Ray, CT, Ultrasound, etc) from one site to another.

    My home address is:

        Michael Wood
        2828 168th St. SE
        Bothell, WA 98012
        USA
```

I can be reached by E-Mail at this address:

mikew@atl.com

Many thanks to Boyd Edmondson and Adam Levin-Delson for their help and suggestions. Thanks go to Matt Weatherford for his networking help and testing.

Also, thanks go to Jarno Tapio Rajahalme and Tomi Ollila for their help in my "porting problems" with AmiTCP.

Thanks go to Allan Purtle for his suggestions on security and CPR compatibility.

1.18 history

History

Version	Date	Description
1.0	10/01/95	First "working" version
1.1	12/02/95	Added support for AmiTCP
1.2	01/16/96	Reduced packet sizes. It is more compatible now with 3rd party programs that use ARexx.
1.3	02/26/96	Added user name and password protection.