ARB01

COLLABORATORS

	TITLE : ARB01	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 27, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

ARB01

1.1 ARexx For Beginners - Article 1 - Getting Started

```
                AREXX FOR BEGINNERS

ARTICLE 1 - GETTING STARTED - AN INTRODUCTION TO AREXX

BY FRANK BUNTON

 COPYRIGHT © FRANK P. BUNTON 1995-1998
```



```
               === End of Text ===
```

1.2 ARexx For Beginners - Article 1 - What Is ARexx?

```
                WHAT IS AREXX?

- "AREXX" is a programming language.
```

- "AREXX" is the Amiga version of the IBM language "Rexx".
- "AREXX" is an
                Interpreted language
                .

The ARexx manual says:-

   "... it can act as a hub through which applications - even those
     created by different companies - can exchange data and commands".

Thus it can be used to tell one program to do things like pass commands
or data to another program, or to receive commands or data from another
program. This is known as Inter Process Communication.

A lot of people only use AREXX for writing small programs to communicate
with other programs. However, AREXX can be used for much more than
this. It can be used for writing small or large programs which are
fully functional in their own right.

As we progress through the articles on this disk you will gain a
much better appreciation of the power of AREXX.

=== End of Text ===

1.3 ARexx For Beginners - Article 1 - Message Ports

                MESSAGE PORTS

To participate in IPC, a program must have been written with a "message
port" called an "ARexx Port". A port is a software interface that
can be accessed by other programs. In other words, it is a small
module of the program code that is written to send signals to, and
receive signals from, other programs.

The program reading these text files has an ARexx port named
either:-

  "MULTIVIEW.x"   or   "AMIGAGUIDE.x"

depending on which program you are using. I will talk about "Multiview"
in the following notes, but if you are using "Amigaguide" then just
substitute "AMIGAGUIDE" for "MULTIVIEW" in the following
notes.

The "x" shown above is a number depending in how many versions of
Multiview are running at the same time. If one is running, it will
be:-

  "MULTIVIEW.1"

If two are running, the the second one will be:-

  "MULTIVIEW.2"

Thus you can talk to "Multiview" by using the ARexx instruction "Address", as in:-

```
ADDRESS 'MULTIVIEW.1'
```

and all future messages are interchanged with the "Multiview" program until a different ARexx port is accessed with another "Address" instruction.

If you use the virus checker program called "Virus_Checker" you will find that it has an ARexx port named "Virus_Checker" (the best virus checker - written by John Veldthuis of New Zealand). Thus you can talk to "Virus_Checker" by using the ARexx instruction "Address" with:-

```
ADDRESS 'Virus_Checker'
```

Most programs with ARexx ports give their ARexx port the same name as the program itself or a similar name. For example, the directory utility program "DirWork" (the best directory utility - by Chris Hames of Victoria) has an ARexx port named "DIRWORK_1" and "Directory Opus" has an ARexx port named "dopus_rexx". Having port names the same as or similar to the program name saves confusion and prevents the possibility of two programs having the same ARexx port name! AmigaDOS commands can be accessed through a port named "COMMAND", while ARexx itself has a port named "REXX".

As a very quick example of one process talking to another, open a Shell or CLI window (if you haven't got one open already) and, position it so that it covers the Multiview or amigaguide window. Now enter at the CLI prompt:-

```
> RX "ADDRESS 'MULTIVIEW.1' ; 'WINDOWTOBACK'"
```

or, if using amigaguide

```
> RX "ADDRESS 'AMIGAGUIDE.1' ; 'WINDOWTOBACK'"
```

(Note - Click on CLI/Shell now as it talks about how to use the Shell/CLI window. I will assume, in all these ARexx articles, that you are quite familiar with the usage of Shell/CLI.)

The Multiview program should now send its window to the back of any other windows in the screen. To bring it to the front again, use:-

```
> RX "ADDRESS 'MULTIVIEW.1' ; 'WINDOWTOFRONT'"
```

or, if using amigaguide

```
> RX "ADDRESS 'AMIGAGUIDE.1' ; 'WINDOWTOFRONT'"
```

The CLI process has sent an ARexx message to Multiview or Amigaguide telling it to send its window to the back and then bring it to the front.

If this doesn't work, then:-

```
- read through the part of this article that talks about what is

                needed to use ARexx
                , carry out everything that it says,
  then come back to the above lines and try again.

- check if any other Multiview windows are open. You may need to use a
  different number suffix.


=== End of Text ===
```

1.4 ARexx For Beginners - Article 1 - Programs

```
PROGRAMS

As mentioned above, ARexx can also be used to create small programs
to perform repetitive tasks or to customise the Amiga working environment
in a more powerful way than can be done with AmigaDOS scripts. It
can even be used to write replacements for some AmigaDOS commands!
(but they will not work as efficiently).

As an example of a more complex ARexx project, I keep our User Group's
membership on the "Pen Pal" data base program. However, Pen Pal is
a commercial copyright program so I cannot give copies of the actual
program to other committee members. But I can do an ASCII dump (text
only) to disk of all the data. This is not very readable as such
so I have written an ARexx program that can read the ASCII file and
format the details as required. A member's name can be entered and
the program searches the file for the name then displays all that
member's details, just like a proper data base program (well....
almost!!). It can also list committee members only, or all members
or, as a future improvement, members who have Amigas or Commodores,
etc. etc.

ARexx is certainly not as powerful a programming language as other
interpreted languages such as "Amos" or "CanDo" but it has its uses
and is quite powerful in the things that it is good at doing.


=== End of Text ===
```

1.5 ARexx For Beginners - Article 1 - Interpreted Languages

```
WHAT IS AN INTERPRETED LANGUAGE?

With computer languages, Machine Code is the computer's native language.
To us humans it is just a set of numbers. These days nobody does
any serious programming in Machine Code, or Machine Language although,
believe it or not, in the very early days of computers they did!
If someone now says that they program in Machine Language they will
```

almost certainly mean that they use "Assembler Language".

A "Compiled" or "Assembled" program is one that is translated into
Machine Code by a "Compiler" program or an "Assembly" program. However,
this is a once only translation. Once completed, the translated version
of the program is saved to disk and, when it is reloaded for use,
the computer sees it as a Machine Code program. The original coding
is not needed by the program user.

An "Interpreted" language, such a Basic or ARexx, is loaded into
memory in the format in which it was written, i.e. untranslated.
To use it an interpreter program must first be loaded into memory.
Each line of program code has to be interpreted into Machine Code
before it can be carried out. Once that instruction is completed,
the next instruction is interpreted then carried out. Each time the
program is used, the interpretation must be done all over again.
This interpretation "on the fly" is the reason that interpreted languages
are so much slower in their execution than compiled or assembled
languages.

With Basic, the "AmigaBasic" program is used both as an editor to
create the program and as an interpreter to carry out the interpretation
as the program is running. "Amos" and "CanDo" are somewhat similar
in this respect.

With ARexx, the program can be written with any text editor or word
processor. The interpreter program it uses is "RexxMast", which is
usually stored in the System directory of your boot disk, and its
associated library file "rexsyslib.library".

=== End of Text ===

1.6 ARexx For Beginners - Article 1 - What Do I Need?

WHAT DO I NEED TO USE AREXX?

AmigaDOS Knowledge

Text Editor

ARexx Software & Files
=== End of Text ===

1.7 ARexx For Beginners - Article 1 - What Do I Need? - AmigaDOS Knowledge

AMIGADOS KNOWLEDGE

Firstly, you should be able to open a Shell or CLI window and enter
AmigaDOS commands. It is not necessary to be any where near expert

in this but a limited general understanding is needed. I include
in this an ability to be able to edit your Startup-Sequence file
so that you can make the necessary additions to the "Assigns" and
"Path" if they are not already there.

I would recommend to you my disk AMIGADOS FOR BEGINNERS which treats
AmigaDOS and CLI/Shell in the same manner as this disk treats
AREXX.

=== End of Text ===

1.8 ARexx For Beginners - Article 1 - What Do I Need? - Text Editor

TEXT EDITOR

Secondly, you should be able to use a text editor. You could use
"Ed", "Edit" or "MEmacs" (on the Workbench disk), one of the Public Domain
ones (there a few on C.H.U.G. Disk 16) or one of the commercial ones
such as CygnusEd. You could also use a Word Processor provided that
it is capable of saving its files as "Text Only" or ASCII. It is
essential that it does not save all the extra information relating
to text and document formatting.

=== End of Text ===

1.9 ARexx For Beginners - Article 1 - What Do I Need? - Software

AREXX SOFTWARE AND FILES

Thirdly, you will need the necessary software. If you have Workbench
1.x then ARexx must be obtained from a retailer. It is a copyright
commercial program and so is NOT in the Public Domain and cannot
be provided with this disk. At time of writing, it will cost you
in the vicinity of $90.

In Workbench 2+ it is provided as part of the package. Commodore-Amiga
now seem to think that it is more useful than AmigaBasic which was
supplied with Workbench 1.x but was removed from the Workbench 2+
packages.

The files needed on your boot disk are:-

Directory	File
Libs	RexSupport.library
Libs	Rexsyslib.library
Libs	Mathieedoubbas.library
System	RexxMast
System	RexxMast.info
RexxC	Various utilities

The maths library is not part of ARexx itself but should be on all
standard workbench disks. It is essential for the running of
ARexx.

=== End of Text ===

1.10 ARexx For Beginners - Article 1 - Starting ARexx

STARTING AREXX

To automatically make ARexx active, your startup-sequence file should
include:-

```
ASSIGN REXX: S:
PATH Sys:system sys:rexxc ADD
RexxMast >nil:
```

(Get hold of a copy of the disk AMIGADOS FOR BEGINNERS for a full
descriptions of the "Assign" and "Path" commands.)

The assignment of REXX: is made because the RX command (which launches
an ARexx program - see Article 3) and other applications will look
for ARexx programs in the logical device "REXX:". It is usually assigned
to S: as this is the natural home for script files. However, if you
prefer it, you could put ARexx programs in any directory that you
like.

I much prefer to have ARexx scripts in a separate directory from
other scripts as it is easier to find them. I have therefore created
a directory called "ARexxS" for all my ARexx scripts and I use this
assignment in my startup-sequence file:-

```
assign REXX: Sys:ARexxS
```

The PATH command additions are there so that these directories will
be automatically searched for command names. The directory "Sys:System"
normally holds the "RexxMast" program and the directory "Sys:RexxC"
normally holds the special ARexx utility commands (see Article 3).

The last of the above three command lines, i.e.:-

```
RexxMast >nil:
```

can be omitted from the Startup-Sequence and used manually in a CLI/Shell
window whenever it is wanted, or you could double click on the RexxMast
icon. In Workbench 2+ this third line could be removed from the
Startup-Sequence and be replaced by dragging the RexxMast icon into the
WBStartup drawer.

This last command runs the program "RexxMast" which is essential
to any ARexx program as it is the
Interpreter Program
. Once run,

or "launched", it stays "resident" in memory waiting for an ARexx
program to use it.

I will always assume, in these articles, that all the above have
been done. This means that ARexx instructions or scripts can be entered
directly at a CLI/Shell.

=== End of Text ===