ARB02

COLLABORATORS

	TITLE : ARB02		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 27, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

ARB02

1.1 ARexx For Beginners - Article 2 - Program Elements & Names

```
                AREXX FOR BEGINNERS

ARTICLE 2 - PROGRAM ELEMENTS & NAMES

BY FRANK BUNTON

 COPYRIGHT © FRANK P. BUNTON 1995-1998


                 Elements Of A Program

                 Comments

                 Program Names

                 Program Name Suffixes
                === End of Text ===
```

1.2 ARexx For Beginners - Article 2 - Elements Of A Program

```
                ELEMENTS OF A PROGRAM

There are three main elements to a program:-

   TOKENS
   CLAUSES
   EXPRESSIONS

There is a lot of jargon associated with these elements and their sub
categories which may only confuse the beginner. It is not really necessary
to learn it all at this stage. However, those of you who wish to know
more about this jargon can double click on the above highlighted items
and weave your way through the interlaced cross references!
```

One problem with manuals is that they try to cover a lot of information right at the start, much of which you will not use for a long time if at all. In these articles I will get into actual programming fairly quickly and explain it as we go along. I always feel that this is the best way to learn.

So... at this stage I will only explain about one of the TOKENS called

COMMENTS
as we need to know about these early on in our programming career.

It is also important to understand how ARexx programs can be
named
so
I will cover this also in this article.

=== End of Text ===

1.3 ARexx For Beginners - Article 2 - Comments

COMMENTS

Every ARexx program MUST start with a comment line. If you leave it out you will get an error message "Program not found".

A comment must always start with /* and end with */. for example:-

```
  /* This is a comment */
```

If you have a very small script and don't particularly want a first line comment as such, you can abbreviate it to:-

```
  /**/
```

Usually the first line comment tells the name of the program and what it does. For example:-

```
  /* VCBBCheck.rexx - Tell Virus_Checker to check boot blocks */
```

Or you could be more fancy and use:-

```
  /***************************************
   **          VCBBCheck.rexx           **
   ** This program  tells Virus_Checker **
   ** to check the boot blocks of disks **
   ***************************************/
```

If the comments go beyond one line of text, as the above does, then the closing */ can be at the end of the last line of comments (as illustrated above) rather than having /* and */ at the start and end of each line.

The above is O.K. at the start of a program. However, if you do this sort of thing in the middle of a program:-

```
  .... lots of programming

  /* this is a comment
     that goes over
     a lot of lines  */

  .... lots more programming
```

then it makes it a lot harder to see exactly which text lines are comments and which are program. I much prefer to have every text line of comments start and end with the comment indicators, as in:-

```
  /* this is a comment */
  /* that goes over    */
  /* a lot of lines    */
```

It is all a matter of choice.

A comment can be on a line of its own or at the end of a line containing instructions, as in:-

```
  say 'hello' /* this displays hello */
```

A comment is a message telling us humans something about the program as a whole or a part of the program. Apart from looking for the first line comment, ARexx completely ignores comments. The program would work quite well without any but the first line comment.

The main use of comments is to either tell some other user what the program is all about and/or how it works or to remind you, the writer, of what it does. It may sound silly to have to remind yourself but you would be surprised at how often you can come back to a program some months after it was written and wonder what various lines in it do.

Put in enough comments to make it understandable but do not over comment. Comments take up memory and disk space.

=== End of Text ===

1.4 ARexx For Beginners - Article 2 - Program Names

PROGRAM NAMES

Apart from the use of the
suffixes
, you can call your programs any name
that you like within the normal rules for naming AmigaDOS files. After all, they are stored just like any other file!

If you stick to the alpha characters (A-Z, a-z) and the numerics (0-9) and the underscore (_) and the hyphen (-) and have no more than 30 characters (including the suffix, e.g. ".rexx") then you won't go far wrong!

One character that you should not use in any part of the name before the suffix is the full stop "." For example, you should not use names like this:-

```
  My.Program.rexx
```

This is because ARexx will consider everything after the first . as the suffix!!

Note that the space can be used but is best avoided and the underscore used instead. For example:-

```
  My Program.rexx
```

is best replaced with:-

```
  My_Program.rexx
```

This is because file and disk names with spaces in them often have to be enclosed in quotation marks to make them valid and people often forget to include the quotation marks.

=== End of Text ===

1.5 ARexx For Beginners - Article 2 - Program Name Suffixes

PROGRAM NAME SUFFIXES

It is normal to include a suffix at the end of the program name. For example, in the name:-

```
  My_Program.rexx
```

The ".rexx" is the suffix.

The suffix used will depend on whether the program has been designed to be launched (run) from AmigaDOS or from some other program.

As we will come to learn, ARexx programs can also be written to be used from within many other programs. These programs will often have a menu item such as "ARexx Script" or "ARexx Macro" or some other wording. Selecting this menu item will allow you to run an ARexx script written especially for that program.

If the program does not have such a menu item then it might have some other means of launching ARexx programs. For example, the Multiview/Amigaguide programs allows an ARexx program to be launched by clicking text provided that the correct linkage commands are written into the text file.

If you want to read the program that has just been run, then click here. It is on the disk in the same directory as this article and has the name:-

```
  TestProgram.guide
```

Note that its suffix is ".guide".

Most programs that have ARexx capabilities will have their own special suffixes. Some examples are:-

AmigaDOS	.rexx
Art Department Professional	.adpro
Cygnus Ed Professional	.ced
Ed	.ed
Multiview/Amigaguide	.guide
PowerPacker	.pprx

As there are now hundreds of programs with ARexx support, it is impossible to list them all here. Read the ARexx section in your program’s manual to find the appropriate suffix.

If you use the correct suffix when naming the program file, then the program should recognise its own ARexx scripts even if you do not include the suffix when you call the program by its file name. For example, AmigaDOS will recognise the script called:-

```
  My_Script.rexx
```

even if you only use:-

```
  RX My_Script
```

when you run the program.

The ARexx program I used above:-

```
  Testprogram.guide
```

was started from within this text file by referring only to:-

```
  Testprogram
```

In fact, by using the correct suffixes, you can have the same root name for scripts used by different programs. For example, the scripts:-

```
  My_Script.rexx
  My_Script.adpro
  My_Script.ced
  My_Script.ed
  My_Script.guide
  My_Script.pprx
```

could all be stored in your assigned REXX: directory and could be called by the relevant programs by using only:-

```
  My_Script
```

There will be no confusion. Each program will look only for scripts with its own suffix.

Although a program will probably work without the correct suffix, it is best to always use the suffix partly because it is easy to recognise an ARexx script and the program it is supposed to be used with by its suffix.

Another good reason to use the suffix is that, if there are two files in the same directory with the same name except that one has the suffix and the other doesn't, as in:-

```
    My_Program
    My_Program.rexx
```

then you will NOT be able to use RX to run the file without the suffix as, in situations like this, RX will always go for the file name with the suffix. For example, if the above two program files exist and you use:-

```
  > RX My_Program
```

then the file "My_Program.rexx" will be used!

ARexx will only use a name without the suffix if it cannot first find the same name with the suffix.

=== End of Text ===