MOOS/MOOS

COLLABORATORS			
	TITLE : MOOS/MOOS		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 1, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

MOOS/MOOS

1.1 main

1.2 1) Overview.

1)~Overview. :

MOOS -- Modular Object Oriented System

Into its implementation ARexx let users create a lot of scripts for using as external functions. MOOS is mainly a collection of ARexx script and external function libraries expecially studied to enhance the "Modular" concept given with the ARexx language. MOOS also shows a way to approach something similar to the OO programming style.

```
rexxMOOS.library
   \
    +- rexx_asl.library
    |       \
    |        +-- AslFileReq()
    |        +-- AslFontReq()
    |        +-- AslScreenReq()
    :
    +- rexx_dos.library
    |       \
    |        +-- AddPart()
    |        +-- Delay()
    |        +-- DeleteVar()
    |        +-- ExAll()
    |        +-- Examine()
    |        +-- Execute()
    |        +-- GetVar()
    |        +-- Info()
    |        +-- IsFileSystem()
    |        +-- IsInteractive()
    |        +-- MatchPattern()
    |        +-- MaxCli()
    |        +-- ReadArgs()
    |        +-- ReadFile()
    |        +-- SameDevice()
    |        +-- SameLock()
    |        +-- SetVar()
    |        +-- SplitPath()
    |        +-- WaitForChar()
    |        +-- WriteFile()
    :
    +- rexx_exec.library
    |       \
    |        +-- AvailMem()
    |        +-- FlushMem()
    |        +-- GetSysBase()
    |        +-- ReBoot()
    :
    +- rexx_icon.library
    |       \
    |        +-- BumpRevision()
    |        +-- DeleteDiskObj()
    |        +-- GetDiskObj()
    |        +-- PutDiskObj()
    |        |-- SetDiskObj()
    :
    +- rexx_intuition.library
    |       \
    |        +-- Beep()
    |        +-- GetPubScreen()
    |        +-- SetPubScreen()
    |        +-- WBenchScreen()
    :
    +- rexx_math.library
    |       \
    |        +-- abs()
    |        +-- acos()
```

```
|       +-- acosh()
|       +-- asin()
|       +-- asinh()
|       +-- atan()
|       +-- atan2()
|       +-- atanh()
|       +-- ceil()
|       +-- cos()
|       +-- cosh()
|       +-- cot()
|       +-- csc()
|       +-- exp()
|       +-- fact()
|       +-- floor()
|       +-- log()
|       +-- log10()
|       +-- nint()
|       +-- pow()
|       +-- rnd()
|       +-- sec()
|       +-- sin()
|       +-- sinh()
|       +-- sqrt()
|       +-- tan()
|       +-- tanh()
:
+- rexx_misc.library
|      \
|       +-- BumpRevision()
|       +-- GetMouse()
|       +-- GetUniqueID()
|       +-- Phonemes()
|       +-- ReadClip()
|       +-- Speak()
|       +-- WBInfo()
|       +-- WriteClip()
:
+- rexx_reqtools.library
|      \
|       +-- RqtEasyReq()
|       +-- RqtFileReq()
|       +-- RqtFontReq()
|       +-- RqtLongReq()
|       +-- RqtPaletteReq()
|       +-- RqtScreenReq()
|       +-- RqtStringReq()
:
+- rexx_stem.library
|      \
|       +-- StemCopy()
|       +-- StemInsert()
|       +-- StemRead()
|       +-- StemRemove()
|       +-- StemSearch()
|       +-- StemSort()
|       +-- StemWrite()
:
```

```
REXX:
  |
  +-- 'dos/AddBuffers'()
  +-- 'dos/FileNote'()
  +-- 'dos/MakeDir'()
  +-- 'dos/Protect'()
  +-- 'dos/Relabel'()
  :
```

1.3 2) Installation.

```
2)~Installation. :
```

To install correctly all parts of the MOOS package use the provided installer script (the installer utility, v43.3, is on Aminet).
In case of difficulty follow these steps:

1) If is present a drawer "Libs" into the distribuition copy its content (MOOS's libraries) into LIBS:

2) If is present a drawer named "ARexx" copy its content into REXX: taking care on making all the required subdirectories. There can be ARexx scripts and/or MOOS's modules stored into any subdirectories. All must be copied into the rigth place.

3) Create a new drawer "MOOS" to copy the documentation files.

1.4 3) Using MOOS within any ARexx script.

```
3)~Using~MOOS~within~any~ARexx~script. :
```

For using MOOS within any ARexx scripts, after its easy installation, is enough to add the following line into the user-startup:

```
RxLib rexxMOOS.library 100 -30
```

Another way is to add, into all ARexx scripts, the normal call to the function ADDLIB() with the following parameters:

```
CALL ADDLIB("rexxMOOS.library",100,-30)
```

The value of "100" for the priority is necessary to be sure that the main library (rexxMOOS.library) is also the first into the libraries list mantained by the ARexx interpreter.

1.5 4) External function libraries.

4)~External~function~libraries. :

```
        rexx_asl.library
        rexx_dos.library
        rexx_exec.library
        rexx_icon.library
        rexx_intuition.library
        rexx_math.library
        rexx_misc.library
        rexx_reqtools.library
        rexx_stem.library
```

Let's talk about the way ARexx manage a function call.

Refer to the ARexx documentation to find more information about the use of ARexx scripts as external functions and... (TO BE COMPLETED)

1.6 5) Stem & Compound symbols.

5)~Stem~&~Compound~symbols. :

Stem & Compound symbols.

Let's refresh our memory about stem and compound symbols as used into the ARexx language. Those entities are very usefull when we need to build a vector or a structure. Here is the general form to specify a compound symbol:

```
    stem.field.sub_field.sub2_field ...
```

There are no limits on how many fields and sub-fields we can define (excluding the system memory) and the complexity of any resulting structure, but ARexx has no built-in facility to take full advantage from the use of stem and compound symbols (refer to the ARexx documentation for details).

Many functions from the MOOS libraries output data into a vector or a structure. Some functions may require a vector as input data. A vector has the following form of representation:

```
    stem.pre_fields.n.post_fields
```

Both "pre_fields" and "post_fields" may be any sequence of fields and sub-fields whereas "n" is the index for the vector (integer). Into its simpliest form a vector is composed of a "stem." and one index:

```
    stem.n
```

The first convenction to specify, into the vector, the total number of entries uses the special field: "count"

```
    stem.count

    stem.0
    stem.1
    stem.2
      :
      :
      . The last entry has n == stem.count-1
```

The second convenction use the entry number "0" to store the total number whereas the first entry is the number "1".

```
    stem.0

    stem.1
    stem.2
      :
      :
      . The last entry has n == stem.0
```

Most functions into the MOOS package support both convenctions and the user can specify the index position and type using a marker:

"#" - By default all vectors use the field "count" to store the total number of entries. With this marker we are able to choose the position of the index into the compound symbol used as vector, eg:

```
        "stem."
            stem.count            == tot
            stem.n              n == 0..tot-1

        "stem.pre."
            stem.pre.count        == tot
            stem.pre.n          n == 0..tot-1

        "stem.#"
            stem.count            == tot
            stem.n              n == 0..tot-1

        "stem.#.post"
            stem.count            == tot
            stem.n.post         n == 0..tot-1

        "stem.pre.#.post"
            stem.pre.count        == tot
            stem.pre.n.post     n == 0..tot-1
```

("pre" == "pre_fields", "post" == "post_fields")

"*" - With this marker we are able to choose index's position into the vector using the 2nd convenction, eg:

```
        "stem.*"
            stem.0                == tot
            stem.n              n == 1..tot
```

```
                "stem.pre_fields.*"
                    stem.pre.0            == tot
                    stem.pre.n          n == 1..tot

                "stem.*.post"
                    stem.0                == tot
                    stem.n.post         n == 1..tot

                "stem.pre.*.post"
                    stem.pre.0            == tot
                    stem.pre.n.post     n == 1..tot

            ("pre" == "pre_fields", "post" == "post_fields")

        "**" - Always using the 2nd convenction, with this marker we are
               able to manipulate more complicated vectors, eg:

                "stem.**.post1"
                "stem.**.post2"
                      ::
                "stem.**.postN"

                    stem.0.post1          == tot1
                    stem.0.post2          == tot2
                         :
                    stem.0.postN          == totN

                    stem.n.post1        n == 1..tot1
                    stem.n.post2        n == 1..tot2
                         :
                    stem.n.postN        n == 1..totN

               Obviously only one "post_fields" sequence may be treated
               at a time.
```

1.7 6) Sharing variables and compound symbols.

```
6)~Sharing~variables~and~compound~symbols. :


    Stem & Compound symbols.

    This new feature isn't documented yet (I need more time).
    Read the two example scripts into the "Tests" drawer.
```

1.8 7) Using templates.

```
7)~Using~templates. :


    Most functions into the MOOS sub-libraries accept arguments into an
    options template. In this way is easy to add/change features to any
```

```
    function. To do this the main library, rexxMOOS, use the ReadArgs()
    routine included into the system dos.library. The following rules
    are to prevent errors during template's parsing:

        "One/K,Multi/K"

            "One"   - Strings that include spaces must be enclosed with
                      double quotes:

                        'One "String with spaces"'
                        'One "String with *"spaces*" and quotes"'

            "Multi" - The previous rule is valid also for keywords that
                      accept more than one value:

                        'Multi one two "th ree" four'


    Due to ReadArgs() is better to use only double quotes into any options
    template:

        AslFileReq(,,'TITLE "The title" OK Yes CANCEL "No no!"')

        tit = "The title"
        yes = "Take it"
        AslFileReq(,,'TITLE "'tit'" OK "'yes'" CANCEL "No no!"')

        tit = 'Strange *"title*"'
        AslFileReq(,,'TITLE "'title'" CANCEL No')


    Using the Shell there's a little trick:

        Rx "AslFileReq('','','?')"
        Rx "ExAll('','?')"
```

1.9 Acknokledgements

```
Acknokledgements :


    Thanks to all people that sent me suggestions and bug reports:

        Andreas Mixich      humpty@tomate.tng.oche.de
        Giorgos Stagakis    giorgos@hotmail.com
        Jim Hines           hines_j@iolinc.net
        Kai Norhausen       kn@tribal.line.org
        Tom Byrer           avi013k1@pn.nettuno.it
```

1.10 MOOS's History

```
MOOS's~History :


    $VER: MOOS 1.05 (24.03.97)


MOOS 1.05 (24.Mar.1997)
=======================

    Updated rexx_asl.(doc|guide)
    MOOS.guide almost re-written.
    New sub-library: rexx_icon.library
    One script A call a script B and it can read/write variables into A.
    All MOOS functions access the ARexx structures.
    Little changes into all test scripts.
    IsInteractive() is the 75th function into the MOOS libraries... :-)

    rexx_dos.library
    ~~~~~~~~~~~~~~~~
    v37.4 (24.Mar.1997) -- WaitForChar(), IsInteractive()
                           ReadFile() replaces ReadBlock()
                           WriteFile() replaces WriteBlock()
                           Updated and fixed the documentation files
                           ExAll() minor bug fixed for using under V37
                           Bug fixed using GetVar(,'Exists')
                           GetVar(,,default)

    rexx_icon.library
    ~~~~~~~~~~~~~~~~~
    v37.0 (22.Mar.1997) -- First public release
                           BumpRevision(), DeleteDiskObj()
                           GetDiskObj(), PutDiskObj(), SetDiskObj()

    rexx_math.library
    ~~~~~~~~~~~~~~~~~
    v37.2 (25.Mar.1997) -- rnd()

    rexx_misc.library
    ~~~~~~~~~~~~~~~~~
    v37.1 (24.Mar.1997) -- ReadClip(), WriteClip(), GetMouse()



MOOS 1.04 (25.Feb.1997)
=======================

    Little changes into all test scripts.
    Updated the documentation for every library.
    New sub-library: rexx_asl.library
    New sub-library: rexx_misc.library
    New sub-library: rexx_reqtools.library
    New functions under beta-testing for the rexx_intuition.library
    Introduced some private functions into the sub-libraries.
    First time using the utility.library and TagLists facilities.
    Updated all MOOS modules (REXX:dos/#?).
```

```
    'dos/AddBuffers'()
    ~~~~~~~~~~~~~~~~~~
    v37.0 (26.Feb.1997) -- Use the function Execute()

    rexxMOOS.library
    ~~~~~~~~~~~~~~~~
    v37.2 (12.Feb.1997) -- Added support for the new sub-libraries.
                           Templates may have 12 keywords (instead of 6).

    rexx_asl.library
    ~~~~~~~~~~~~~~~~
    v37.1 (25.Feb.1997) -- First public release.
                           Minor changes and bug fixed.
    v37.0 (16.Feb.1997) -- AslFileReq(), AslFontReq(), AslScreenReq()
                           Added and removed AslGlobals()
                           First release of functions template==>tags.

    rexx_dos.library
    ~~~~~~~~~~~~~~~~
    v37.3 (25.Feb.1997) -- IsFileSystem()
    v37.2 (16.Feb.1997) -- Execute() now works.
                           MaxCli(), SameDevice(), SameLock()

    rexx_exec.library
    ~~~~~~~~~~~~~~~~~
    v37.2 (16.Feb.1997) -- GetSysBase()

    rexx_intuition.library
    ~~~~~~~~~~~~~~~~~~~~~~
    v37.2 (26.Feb.1997) -- Minor bug fixed into GetPubScreen()

    rexx_misc.library
    ~~~~~~~~~~~~~~~~~
    v37.0 (15.Feb.1997) -- First public release.
                           GetUniqueID(), WBInfo(), Phonemes(), Speak()
                           Fixed bug using translator.library (V42).

    rexx_reqtools.library
    ~~~~~~~~~~~~~~~~~~~~~
    v38.0 (25.Feb.1997) -- First public release.
                           RqtEasyReq(), RqtFileReq(), RqtFontReq()
                           RqtLongReq(), RqtStringReq()
                           RqtPaletteReq(), RqtScreenReq()

    rexx_stem.library
    ~~~~~~~~~~~~~~~~~
    v37.2 (12.Feb.1997) -- RxStem_ReadStem(), internal private function.


MOOS 1.03 (16.Feb.1997)
=======================

    Internal release never arrived on Aminet.
```

```
MOOS 1.02 (10.Feb.1997)
=======================

    Enhanced stem and compound symbols management.
    All libraries access correctly the global data segment.
    Created the MOOS.guide

    rexxMOOS.library
    ~~~~~~~~~~~~~~~~
    v37.1 (06.Feb.1997) -- Fixed the global data section.
                           Functions may return a little string (R_STR).
                           Fixed LibInit() and LibExpunge().

    rexx_dos.library
    ~~~~~~~~~~~~~~~~
    v37.1 (07.Feb.1997) -- Support for the new stem formats.
                           Added the keyword "Fields/S" to ExAll()
                           Execute()

    rexx_exec.library
    ~~~~~~~~~~~~~~~~~
    v37.1 (08.Feb.1997) -- Support for the new stem formats.

    rexx_intuition.library
    ~~~~~~~~~~~~~~~~~~~~~~
    v37.1 (08.Feb.1997) -- Support for the new stem formats.

    rexx_math.library
    ~~~~~~~~~~~~~~~~~
    v37.1 (05.Feb.1997) -- New ftoa() (internal) conversion routine.
                           Changed all functions definitions.
                           The new library is smaller (half) and much faster.

    rexx_stem.library
    ~~~~~~~~~~~~~~~~~
    v37.1 (08.Feb.1997) -- Support for the new stem formats.



MOOS 1.01 (30.Jan.1997)
=======================

    Added a standard installation script.
    New sub-library: rexx_exec.library
    New sub-library: rexx_intuition.library
    New sub-library: rexx_math.library
    All libraries use the standard version.revision specification.
    New global history for the whole project (this file).
    Created the rexxlib.lib to move all private functions in.

    rexxMOOS.library
    ~~~~~~~~~~~~~~~~
    v37.0 (30.Jan.1997) -- External functions' table.
                           Functions may not set any result string.
                           Optimized the dispatch routine.

    rexx_dos.library
```

```
    ~~~~~~~~~~~~~~~~
    v37.0 (29.Jan.1997) -- Minor bug fixed into ReadArgs()
                           SetVar(), GetVar(), DeleteVar()
                           All functions read the RexxArg structure.
                           WriteBlock() has only "Append/S" as option.
                           Examine() and ExAll() return more informations.
                           ReadBlock() has only the "Del=Delete/S" option.
                           Info()

    rexx_exec.library
    ~~~~~~~~~~~~~~~~~
    v37.0 (29.Jan.1997) -- AvailMem(), Reboot(), FlushMem()

    rexx_intuition.library
    ~~~~~~~~~~~~~~~~~~~~~~
    v37.0 (31.Jan.1997) -- First public release.
                           GetPubScreen(), SetPubScreen()
                           WBenchScreen() replace OpenWB() and CloseWB()
                           ScreenToBack() and ScreenToFront() removed.
    v0.10 (15.Jan.1997) -- Internal release.
                           Beep(), ScreenToBack(), ScreenToFront()
                           CloseWB(), OpenWB()

    rexx_math.library
    ~~~~~~~~~~~~~~~~~
    v37.0 (30.Jan.1997) -- Improved the dispatch function.
                           acosh(), asinh(), atanh(), nint(), ceil(),
                           floor(), sec(), csc()
    v0.10 (15.Jan.1997) -- First internal release.
                           A special dispatch function.
                           abs(), acos(), asin(), atan(), atan2()
                           cos(), cosh(), cot(), exp(), log(), log10()
                           pow(), sin(), sinh(), sqrt(), tan(), tanh()

    rexx_stem.library
    ~~~~~~~~~~~~~~~~~
    v37.0 (29.Jan.1997) -- Enhanced the internal functions.



MOOS 1.00 (24.Jan.1997)
=======================

    rexxMOOS.library
    ~~~~~~~~~~~~~~~~
    v1.00 (24.Jan.1997) -- Bug fixed in the "RESULT" string.
                           Minor bug fixed into the internal ReadArgs()
                           Enhanced the main dispatch function.
                           Bug fixed in the argument checking section.
    v0.80 (07.Jan.1997) -- First internal release, based on the old
                           rexx_stem.library

    rexx_dos.library
    ~~~~~~~~~~~~~~~~
    v1.00 (23.Jan.1997) -- AddBuffers() replaced with 'dos/AddBuffers'()
    v0.83 (19.Jan.1997) -- First public release.
                           WriteBlock(), ReadBlock()
```

```
                          ExAll(), Examine(), ReadArgs()
    v0.82 (12.Jan.1997) -- MatchPattern()
    v0.81 (10.Jan.1997) -- AddPart(), SplitPath()
    v0.80 (09.Jan.1997) -- rexx_dos.library becomes a sub-library of
                           the new rexxMOOS.library
                           Delay(), AddBuffers()

    rexx_stem.library
    ~~~~~~~~~~~~~~~~
    v0.81 (11.Jan.1997) -- First public release.
                           StemRead(), StemWrite()
                           Added the option "Delete/S" to StemRead()
    v0.80 (08.Jan.1997) -- rexx_stem.library becomes a sub-library of
                           the new rexxMOOS.library
                           Changed all function names *Stem ==> Stem*
    v0.13 (26.Nov.1996) -- Added "Case/S" to SortStem().
                           First beta-release in AREXX and AREXX.AMY
                           RemoveStem()
    v0.12 (25.Nov.1996) -- CopyStem()
                           Added "NoDupes/S" to SortStem()
    v0.11 (24.Nov.1996) -- Improved InsertStem(), SearchStem()
                           SortStem()
    v0.10 (22.Nov.1996) -- InsertStem(), SearchStem()
                           Added the ReadArgs() support.
```

1.11 The author

```
The~author :


    pub/aminet/util/rexx/OLE.lha        (1994)
    pub/aminet/comm/fido/OBManager.lha  (1996)
    pub/aminet/util/rexx/MOOS.lha       (1997)


 __
 |(
e
        rocco@inmedia.it
        moos@freenet.hut.fi

 Rocco "the dwarf" Coluccelli - MOOS Development

                                                  __
 Student of Computer Science Engineering.        ///  2:332/403.61@fidonet
 Member of Amiga Group Italia, Bologna.         ///  2:335/533.9@fidonet
 ARexx programmer.                             ///  39:102/12.9@amiganet
 Amiga user since 1987.                  __    ///  17:100/1.2@oznet
                                        \\\  ///
                                         \\\///  rocco@inmedia.it
                                          \XX/  moos@freenet.hut.fi

    Amiga Group Italia - http://www.inmedia.it/Amiga/
```