rexxstate

COLLABORATORS

	TITLE : rexxstate		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		December 18, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

rexxstate

1.1 rexxstate.doc

```
--overview--

close_console()

console_open()

end()

halt()

new()

open_console()

resume()

running()

shutdown()

start()

suspend()

suspended()

trace_off()

trace_on()

tracing()
```

1.2 rexxstate.m/--overview--

```
                 PURPOSE
  To control the global aspects of the ARexx system.

   OVERVIEW
  ARexx  is  an  interpreter program which has the ability to trace,
  start,  pause,  and  cancel  the  ARexx scripts that it runs. This
  object allows you to control these global ARexx processes:

  - whether ARexx is running or not (RexxMast / RXC)

                running()
                ,
                start()
                ,
                shutdown()
                  - whether ARexx scripts are suspended or not

                suspended()
                ,
                suspend()
                ,
                resume()
                  - whether the ARexx console is open or not (TCO/TCC)

                console_open()
                ,
                open_console()
                ,
                close_console()
                  - whether ARexx is running in trace mode or not (TS/TE).

                tracing()
                ,
                trace_on()
                ,
                trace_off()
                  There  is  also  a method to immediately and irreversibly halt  ↩
                      all
  running ARexx programs, which simulates the HI command.

   NOTE
  The  state  of  ARexx  is global, and this class respects that, it
  shows  and sets this global ARexx state, it does not hold seperate
  states  in  individual  instances of the class, so therefore using
  more than one instance does not make sense, and is unneccessary.

  This class is based on the guts of commands in SYS:RexxC/

  Still todo is RXSET, RX is done by other/sendrexx.m and
  WaitForPort / RXLIB aren't all that useful?!
```

1.3 rexxstate.m/close_console

```
                 NAME
rexxstate.close_console() -- close the ARexx trace console.

 SYNOPSIS
succeeded := close_console()

 FUNCTION
Shuts the ARexx trace and debugging console, like the TCC command.

 RESULT
succeeded - TRUE if the console is now shut, otherwise FALSE.

 SEE ALSO

              console_open()
              ,
              open_console()
```

1.4 rexxstate.m/console_open

```
                 NAME
rexxstate.console_open() -- check if ARexx trace console is open.

 SYNOPSIS
open := console_open()

 FUNCTION
Checks if the ARexx trace and debugging console is open or not.

 RESULT
open - TRUE if the console is open, otherwise FALSE.

 SEE ALSO

              open_console()
              ,
              close_console()
```

1.5 rexxstate.m/end

```
                 NAME
rexxstate.end() -- Destructor.

 SYNOPSIS
end()

 FUNCTION
Closes resources used by an instance of the rexxstate class.
```

```
 SEE ALSO

             new()
```

1.6 rexxstate.m/halt

```
 NAME
rexxstate.halt() -- end all currently running ARexx programs.

 SYNOPSIS
halt()

 FUNCTION
Halts  all  currently  running  ARexx programs and ends them. When
this returns, all programs are most likely to be in the process of
ending. This does not shut down ARexx.
```

1.7 rexxstate.m/new

```
                 NAME
rexxstate.new() -- Constructor.

 SYNOPSIS
new()

 FUNCTION
Initialises an instance of the rexxstate class. Raises "LIB" if it
cannot open 'rexxsyslib.library' version 33 or better.

 SEE ALSO

             end()
```

1.8 rexxstate.m/open_console

```
                 NAME
rexxstate.open_console() -- open the ARexx trace console.

 SYNOPSIS
succeeded := open_console()

 FUNCTION
Opens the ARexx trace and debugging console, like the TCO command.

 RESULT
succeeded - TRUE if the console is now open, otherwise FALSE.

 SEE ALSO
```

```
                console_open()
                ,
                close_console()
```

1.9 rexxstate.m/resume

```
                 NAME
  rexxstate.resume() -- resume running ARexx programs.

   SYNOPSIS
  success := resume()

   FUNCTION
  Lets ARexx program execution continue.

   RESULT
  success - TRUE if ARexx programs are now running, otherwise FALSE.

   SEE ALSO

                suspended()
                ,
                suspend()
```

1.10 rexxstate.m/running

```
                 NAME
  rexxstate.running() -- test if ARexx is running.

   SYNOPSIS
  running := running()

   FUNCTION
  Determines whether ARexx is running or not.

   RESULT
  running - TRUE if RexxMast is running, FALSE otherwise.

   SEE ALSO

                start()
                ,
                shutdown()
```

1.11 rexxstate.m/shutdown

```
                 NAME
  rexxstate.shutdown() -- shut down ARexx.

   SYNOPSIS
  succeeded := shutdown()

   FUNCTION
  Shuts down ARexx.

   RESULT
  succeeded - TRUE if ARexx stops running, FALSE otherwise.

   SEE ALSO

              start()
              ,
              running()
```

1.12 rexxstate.m/start

```
                 NAME
  rexxstate.start() -- start up ARexx.

   SYNOPSIS
  succeeded := start()

   FUNCTION
  Starts up the RexxMast program, which starts the ARexx system.

   RESULT
  succeeded - TRUE if RexxMast is now running, FALSE otherwise.

   SEE ALSO

              shutdown()
              ,
              running()
```

1.13 rexxstate.m/suspend

```
                 NAME
  rexxstate.suspend() -- suspend the running of ARexx programs.

   SYNOPSIS
  success := suspend()

   FUNCTION
  Puts all running ARexx programs on hold.
```

```
 RESULT
success - TRUE if ARexx programs are now suspended, otherwise FALSE.

 SEE ALSO

             suspended()
             ,
             resume()
```

1.14 rexxstate.m/suspended

```
                NAME
rexxstate.suspended() -- check if ARexx programs are suspended.

 SYNOPSIS
suspended := suspended()

 FUNCTION
Checks if ARexx is currently running its programs or not.

 RESULT
suspended - TRUE if ARexx is suspending programs, otherwise FALSE.

 SEE ALSO

             suspend()
             ,
             resume()
```

1.15 rexxstate.m/trace_off

```
                NAME
rexxstate.trace_off() -- turn ARexx trace mode off.

 SYNOPSIS
success := trace_off()

 FUNCTION
Turns ARexx trace mode off.

 RESULT
success - TRUE if ARexx is now out of trace mode, otherwise FALSE.

 SEE ALSO

             tracing()
             ,
             trace_on()
```

1.16 rexxstate.m/trace_on

```
                 NAME
rexxstate.trace_on() -- turn ARexx trace mode on.

 SYNOPSIS
success := trace_on()

 FUNCTION
Turns  ARexx  trace  mode  on.  Program execution flow will now be
listed  to  the  trace  console (if it is open) or to the standard
output of the programs in question.

 RESULT
success - TRUE if ARexx is now in trace mode, otherwise FALSE.

 SEE ALSO

              tracing()
              ,
              trace_off()
```

1.17 rexxstate.m/tracing

```
                 NAME
rexxstate.tracing() -- check if ARexx trace mode is on.

 SYNOPSIS
tracing := tracing()

 FUNCTION
Checks if ARexx is in trace mode or not.

 RESULT
tracing - TRUE if ARexx is in trace mode, otherwise FALSE.

 SEE ALSO

              trace_on()
              ,
              trace_off()
```