MRBackup

COLLABORATORS

	TITLE : MRBackup		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		October 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

MRBackup

1.1   MRBackup's ARexx Interface

The Amiga's multitasking operating system is one of its distinguishing
features. The typical Amiga user is apt to be running several programs at
any given time. With the addition of ARexx (the Amiga implementation of
the Rexx language), programs equipped with an ARexx "port" can communicate,
share resources with one another or be operated under "remote control" of
an ARexx program.

        The MRBackup ARexx Port

        Using ARexx with MRBackup

        MRBackup's ARexx Commands

        Go to Top Level

1.2   the mrbackup arexx port

The MRBackup ARexx Port
MRBackup provides an ARexx interface which allows many of its operating
parameters and features to be accessed this way. It is possible to run
multiple"copies" of MRBackup. Thus, MRBackup creates a unique ARexx port
name for each instance of MRBackup that is run. You can determine the
ARexx port name by selecting the About item from MRBackup's Project menu.
The ARexx port name will always be of the form:

        MRB_#<number>

where <number> is the number assigned to a given instance of MRBackup.
Typically, with one copy of MRBackup running, the ARexx port name will be
MRB_#1.

1.3   using arexx with mrbackup

Using ARexx with MRBackup
This is not an ARexx tutorial. If you are unfamiliar with ARexx, you will
have to obtain appropriate documentation. ARexx is bundled with AmigaDOS
version 2.04 and beyond. It can also be purchased from

        William S. Hawes
        P.O. Box 308
        Maynard, MA 01754
        (617) 568-8695

MRBackup's ARexx implementation requires that the results option be
enabled, since many commands return a value. Include the following
statement in all of your MRBackup ARexx scripts:

    options results

Commands which don't have a specific return value will set the result
variable to either "OK" or "FAIL" to indicate success or failure.

Each MRBackup ARexx script must have a filename extension of ".mrbk" (e.g.
DailyBackup.mrbk). Here is an example script which manipulates MRBackup's
voice setting (on or off) and demonstrates its effects:

```
/* voice.mrbk */
    /* MRBackup: turn voice on and off. */

    signal on ERROR
    signal on BREAK_C

    /* Enable command results. */
    options results

    /* Make sure that MRBackup is running. */
    if ~(Show('P', 'MRB_#1')) then do
            say "You must run MRBackup first."
            exit 1
    end

    /* Select MRBackup's ARexx port. */
    address "MRB_#1"

    /* Bring MRBackup's screen to the front. */
    poptofront

    /* Inform the user as to what is going to happen. */
    'notealert "This test turns the voice option off and on."'

    /* Turn the voice capability off. */
    'setvoice "no"'
    /* Check the result of the previous command. */
    if result ~= "OK" then do
            say "I could not turn the voice option off!"
            exit 1
    end

    /* The following message should be suppressed. */
```

```
      'speak "You should not hear this message."'
      if result ~= "OK" then do
              say "Attempt to speak failed."
              exit 1
      end

      call Delay(50)

      /* Enable MRBackup's voice capability. */
      'setvoice "yes"'
      if result ~= "OK" then do
              say "I could not turn the voice option on!"
              exit 1
      end

      /* This time, the user should hear the message. */
      'speak "This message is being brought to you by Ay Rexx."'
      if result ~= "OK" then do
              say "Attempt to speak failed."
              exit 1
      end

      exit 0

      /*--- Control-C interrupts come here. ---*/

      break_c:

      say "*** Control-C recieved.  Stopped by user. ***"
      exit 5


      /*--- ARexx-detected errors come here. ---*/
      error:

      say "Error"
      exit 6

      /*--- End of script. ---*/
```

You will find a set of example ARexx scripts in the ARexx Scripts directory
on your MRBackup program diskette. Please refer to them when you need help
in creating your own MRBackup/ARexx applications.

1.4   mrbackup's arexx commands

MRBackup's ARexx Commands

This section details each of the ARexx commands supported by MRBackup. In
the following discussion, certain notation conventions are adopted:

· Command parameters (arguments) are often specified with enclosing angle
  brackets <>. The enclosed word or phrase connotes the type of value which
  should be substituted. For instance, a parameter denoted as <path> could
  take a value such as "DH0:Devs/Printers".

· Optional parameters are enclosed in square brackets []. In these cases,
  the command's behavior with the optional parameter specified is contrasted
  with its behavior when the optional parameter is given.

· Literal text values (e.g. "OK", "FAIL", etc.) are specified as quoted
  strings.

COMMAND: backup

RESULT: "OK"or "FAIL"

DESCRIPTION: This command starts a backup operation. Prior to issuing the
backup command, all backup parameters (filter specifications, compression
settings, etc.) should be set to their desired values.

COMMAND: getbackpath

RESULT: MRBackup's current Backup Path specification.

DESCRIPTION: This command obtains the current Backup Path specification
and returns it via the result variable.

COMMAND: getbfilterpath

RESULT: MRBackup's current Backup Filter specification.

DESCRIPTION: This command obtains the current Backup Filter specification
and returns it via the result variable.

COMMAND: getbufsize

RESULT: The current Buffer Size value.

DESCRIPTION: The getbufsize command obtains the current Buffer Size value
(expressed as a multiple of "K", where "K" = 1024) and returns it via the
result variable.

COMMAND: getcfilterpath

RESULT: The current Compression Filter specification.

DESCRIPTION: The getcfilterpath command obtains the current Compression
Filter specification and returns it via the result variable.

COMMAND: getdfilterpath

RESULT: The current Decompression Filter specification.

DESCRIPTION: The getdfilterpath command obtains the current Decompression
Filter specification and returns it via the result variable.

COMMAND: getcompression

RESULT: "None", "12-Bit", "13-Bit", "14-Bit", "15-Bit", "16-Bit"

DESCRIPTION: The getcompression command obtains the current Compression
code size setting and returns it via the result variable.

COMMAND: getdecompression

RESULT: "None", "12-Bit", "13-Bit", "14-Bit", "15-Bit", "16-Bit"

DESCRIPTION: The getcompression command obtains the current Deompression
code size setting and returns it via the result variable.

COMMAND: gethomepath

RESULT: MRBackup's Home Path specification.

DESCRIPTION: The gethomepath command obtains MRBackup's Home Path
specification and returns it via the result variable.

COMMAND: getlistpath

RESULT: MRBackup's Listing Path specification.

DESCRIPTION: The getlistpath command obtains MRBackup's Listing Path
specification and returns it via the result variable.

COMMAND: getlogpath

RESULT: MRBackup's Log Path specification.

DESCRIPTION: The getlogpath command obtains MRBackup's Log Path
specification and returns it via the result variable.

COMMAND: getrfilterpath

RESULT: the current Restore Filter filename

DESCRIPTION: This command retrieves the current Restore Filter filename
and returns it via the result variable

COMMAND: getformatting

RESULT: "None", "Quick", "Normal"

DESCRIPTION: The getformatting command obtains MRBackup's current
Formatting setting and returns it in the result variable.

COMMAND: gettestdate

RESULT: the current Test Date value

DESCRIPTION: The gettestdate command fetches the current Test Date value
(used for backup operations) and returns it in the ARexx result variable.
The date is formatted according to the current date format.

COMMAND: ignorecatalog <yes_or_no>

RESULT: none

DESCRIPTION: The ignorecatalog command, with the "YES" parameter,
instructs MRBackup to perform the next restore operation without reference
to a backup catalog. This implies that the full saveset will be restored
unless a restore filter is specified. This setting is "non-sticky". That
is, as soon as a restore is started and this value is used, ignorecatalog
reverts to the "NO" (require catalog) setting.

COMMAND: listing <yes_or_no>

RESULT: "OK" or "FAIL"

DESCRIPTION: The listing command enables or disables MRBackup's listing
output, depending upon the <yes_or_no> parameter which must be either "YES"
or "NO". Example:

    listing "YES"

COMMAND: loadprefs <filename>

RESULT: the current preferences filename

DESCRIPTION: The loadprefs command causes some or all of MRBackup's
operating parameters to be loaded from the specified <filename>. This file
must conform to the format of the MRBackup.init file delievered with
MRBackup Professional. The file may have been previously created with the
Preferences/Save... menu command or it may be created by an editor,
application or ARexx script. This method of setting MRBackup parameters is
much more convenient than using the individual "set" ARexx commands.

COMMAND: notealert <message>

RESULT: "OK" or "FAIL"

DESCRIPTION: The notealert command provides access to MRBackup's
informational requester. The text of <message> will be presented in a
requester. The user must click the requester's OK button before program
execution will proceed. The <message> string may contain embedded newline
(line feed) characters.

COMMAND: poptofront

RESULT: "OK" or "FAIL"

DESCRIPTION: The poptofront insures that MRBackup's screen is the
frontmost screen.

COMMAND: quit

RESULT: "OK" or "FAIL"

DESCRIPTION: The quit command instructs MRBackup to terminate. When used,
this must be the last command issued to MRBackup.

COMMAND: restore

RESULT: "OK" or "FAIL"

DESCRIPTION: The restore command instructs MRBackup to perform a file
restore operation according to MRBackup's current settings.

COMMAND: setarcbits <yes_or_no>

RESULT: "OK" or "FAIL"

DESCRIPTION: The setarcbits command instructs MRBackup to enable/disable
the setting of file archive bits during a backup operation. If the
<yes_or_no> value is "YES", archive bits will be set upon successful
completion of a backup.

COMMAND: setbackpath [<path>]

RESULT: "OK" or "FAIL"

DESCRIPTION: The setbackpath command instructs MRBackup to adopt a new
Backup Path specification. The <path> parameter, if supplied, must be the
name of a valid device, directory or filename (depending upon the current
backup mode). If <path> is not given, the user will be presented with
MRBackup's file requester.

COMMAND: setbackupmode <mode>

RESULT: the new backup mode setting

DESCRIPTION: The setbackupmode command sets the backup mode (media type)
to one of "AmigaDOS", "FastDisk" or "SCSITape".

COMMAND: setbfilterpath [<path>]

RESULT: new backup filter file name

DESCRIPTION: The setbfilterpath command instructs MRBackup to adopt a new
Backup Filter specification. The <path> parameter, if given, must be the
name of a valid text file containing MRBackup backup filter specifications.
If <path> is not given, the user will be presented with MRBackup's file
requester so that a file may be selected. Note that this command always
returns the backup filter file name in effect upon its return. To test for
failure, test the ARexx rc variable for a non-zero result.

COMMAND: setbufsize <value>

RESULT: new buffer size value

DESCRIPTION: The setbufsize command instructs MRBackup to use a new buffer
size for backup/restore operations. The <value> parameter is expected to
be a number expressed as a multiple of "K" (K = 1024). For example, a
<value> of 64 would result in 65536 bytes being allocated for MRBackup
buffering operations. The return value is always the buffer size (again,
in K) in effect upon return from this command. If an error is detected,
the ARexx rc variable will contain a non-zero error code.

COMMAND: setcfilterpath [<path>]

RESULT: new compression filter file name

DESCRIPTION: The setdfilterpath command instructs MRBackup to adopt a new
Compression Filter file name specification. If <path> is given, it must be
the name of an existing text file containing valid MRBackup filter
patterns. If <path> is omitted, the user will be presented with MRBackup's
file requester so that a file may be selected.

This command always returns the name of the Compression Filter file in
effect upon its return. To check for command failure, test the ARexx rc
variable for a non-zero value.

COMMAND: setcomment<comment_string>

RESULT: "OK"

DESCRIPTION: The setcomment command sets the save set comment for the next
backup to the value specified by the <comment_string> parameter. This
value should be 80 characters or less.

COMMAND: setcompression <code_size>

RESULT: "OK" or "FAIL"

DESCRIPTION: The setcompression command instructs MRBackup to use a new
compression code size for subsequent backups. The valid values for
<code_size> are:

    None, 12-Bit, 13-Bit, 14-Bit, 15-Bit, 16-Bit

COMMAND: setdecompression <code_size>

RESULT: "OK" or "FAIL"

DESCRIPTION: The setdecompression command instructs MRBackup to use a new
decompression code size limit for subsequent backups. The valid values for
<code_size> are the same as those for the setcompression command.

COMMAND: setdfilterpath [<path>]

RESULT: new decompression filter file name

DESCRIPTION: The setdfilterpath command instructs MRBackup to adopt a new
Decompression Filter file name specification. If <path> is given, it must
be the name of an existing text file containing valid MRBackup filter
patterns. If <path> is omitted, the user will be presented with MRBackup's
file requester so that a file may be selected.

This command always returns the name of the Decompression Filter file in
effect upon its return. To check for command failure, test the ARexx rc
variable for a non-zero value.

COMMAND: setformatting <format_option>

RESULT: same as getformatting

DESCRIPTION: The setformatting command tells MRBackup what type of floppy
disk formatting to employ when doing a backup to floppy disk (AmigaDOS mode
only). The <format_option> must be one of "None", "Quick" or "Normal".
The result will always be the formatting option in effect upon return from
this command. If a bad <format_option> is specified, the ARexx rc variable
will be set to a non-zero value.

COMMAND: sethomepath [<path>]

RESULT: new Home Path specification

DESCRIPTION: The sethomepath command instructs MRBackup to adopt a new
Home Path specification. If <path> is given, it must be a valid pathname
that satisfies the requirements for the Home Path. If <path> is not given,
the user is presented with MRBackup's file requester so that a new Home
Path may be selected.

This command always returns the name of the Home Path in effect upon its
return. To check for command failure, test the ARexx rc variable for a
non-zero value.

COMMAND: setinfogadget <message>

RESULT: "OK"

DESCRIPTION: This command provides a means for an ARexx script to place a
text <message> in MRBackup's Info gadget. Example:

```
    setinfogadget 'I am about to start the backup.'
```

COMMAND: setlistpath [<path>]

RESULT: new listing pathname

DESCRIPTION: The setlistpath command instructs MRBackup to adopt a new
Listing Path specification. If <path> is given, it must be a valid device
name (e.g. PRT:, PAR:, etc.) or file name. If <path> is not given, the
user will be presented with MRBackup's file requester to allow a selection.

The setlistpath command always returns the Listing Path specification in
effect upon its return. To test for an error, check the ARexx rc variable
for a non-zero value.

COMMAND: setlogpath [<path>]

RESULT: new log file name

DESCRIPTION: The setlogpath command instructs MRBackup to adopt a new Log
Path specification. If <path> is given, it must be a suitable device,
console or file specification for MRBackup's log messages. If <path> is
not given, the user will be presented with MRBackup's file requester to
allow a new selection.

The setlogpath command always returns the Log Path specification in effect
upon its return. To test for an error, check the ARexx rc variable for a
non-zero value.

COMMAND: setprefix <prefix_string>

RESULT: accepted prefix string

DESCRIPTION: The setprefix command allows an ARexx script to set the
Prefix string used to create backup volume names. The result is the actual
prefix string accepted, which may differ from <prefix_string> slightly if
illegal characters exist in <prefix_string> or if it is too long (maximum
of 20 characters).

COMMAND: setrfilterpath <pathname>

RESULT: the Restore Filter pathname

DESCRIPTION: This command sets the Restore Filter specification to
<pathname>. This allows for selective file restores under ARexx control.

COMMAND: ssettestdate [<date_string> >]

RESULT: new Test Date specification

DESCRIPTION: The settestdate command instructs MRBackup to adopt a new

Test Date specification. If <date_string> is given, MRBackup attempts to
convert it to an AmigaDOS date value (DateStamp). If <date_string> is not
given, MRBackup's date requester is activated to allow the user to enter
the new test date.

The result will always be the Test Date in effect upon return from this
command. If a <date_string> conversion error occurs, the ARexx rc variable
will be set to a non-zero result.

COMMAND: setvoice <yes_or_no>

RESULT: "OK" or "FAIL"

DESCRIPTION: The setvoice command enables or disables MRBackup's voice
capability, depending upon the value of the <yes_or_no> parameter. A "YES"
value enables voice, while a "NO" value disables it. Example:

    setvoice "NO"

COMMAND: speak <message>

RESULT: "OK" or "FAIL"

DESCRIPTION: The speak command requests MRBackup to utter the text
contained in the <message> parameter. The <message> is only spoken if
MRBackup's voice is enabled (see getvoice/setvoice). Since MRBackup uses
the Amiga's translator, you might want to experiment with certain sentences
and phrases which aren't handled correctly. For instance, "MRBackup"
sounds like "merbackup" while "M R backup" produces more desirable results.

COMMAND: splitfiles <yes_or_no>

RESULT: "OK" or "FAIL"

DESCRIPTION: The splitfiles command enables or disables MRBackup's
splitting of "big files", depending upon the setting of the <yes_or_no>
flag. A "YES" value enables file splitting while a "NO" value disables it.
Note that this setting is only relevant to the AmigaDOS backup mode.

COMMAND: testarcbits <yes_or_no>

RESULT: "OK" or "FAIL"

DESCRIPTION: The testarcbits instructs MRBackup to test or ignore file
archive bits during backup operations, depending upon the value of the
<yes_or_no> parameter. If <yes_or_no> is "YES", only files whose archive
bit is clear will be candidates for backup.

COMMAND: utilities

RESULT: "OK" or "FAIL"

DESCRIPTION: The utilities command activates MRBackup's utilities window.
Script execution will be suspended until the user closes the Utilities
window.

COMMAND: yesno <question>

RESULT: "YES" or "NO"

DESCRIPTION: The yesno command provides access to MRBackup's YES/NO
requester. The <question> parameter should be a string containing a
question. Script execution will be suspended until the user responds by
clicking either the YES or NO buttons in MRBackup's requester. Upon
return, the result variable will contain the user's response.