# dosrt

| COLLABORATORS | | | |
|---|---|---|---|
| | *TITLE* :<br>dosrt | | |
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | | September 19, 2022 | |

| REVISION HISTORY | | | |
|---|---|---|---|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

# Contents

# Chapter 1

# dosrt

## 1.1 main

DOSReqTools - GiftWare - ©1994 by BOTH Software
Freely Distributable

```
                  rtFontRequest

           rtPaletteRequest
                  rtPaletteRequest

           rtScreenModeRequest
                  rtScreenModeRequest
```

## 1.2 ti

TITLE

This section applies to the following requesters:

```
     rtGetLong
     rtEZRequest
     rtGetString
     rtFileRequest
     rtFontRequest
     rtPaletteRequest
     rtScreenModeRequest
```

TITLE is a Required argument and it is Positional.

This information appears in the top bar of the window. It should describe the kind of information the user is expected to provide.

```
    rtEZRequest TITLE "Installation Instructions"
```

The word TITLE does not have to be used as long as the text is presented as the first string following the name of the requester.

```
    rtEZRequest "Installation Instructions"
```

TITLE

There is no default value for TITLE. If you choose not to present any information in the window you must include a null string as the first string after the argument.

```
rtEZRequest "" "No Title Text Appears"
```

NOTITLE

The text you assign as TITLE will be truncated if the window is too narrow to render it.

## 1.3 bo

```
     BODY


This section applies to the following requesters:

      rtGetLong
      rtEZRequest
      rtGetString

 BODY

 This is the text that appears inside the window. Use it to give
 specific instructions to the user.

  rtEZRequest "Installation Instructions" BODY "Install for real?"

 The word BODY does not have to be used as long as the text is
 presented as the next string following TITLE.

 rtEZRequest "Installation Instructions" "Install for real?"

     BODY

DISPLAYING MORE THAN ONE LINE OF BODYTEXT

 The inclusion of a \n (backslash and the letter n) will force the
 text that follows to print on a new line. Here is an example:

     BODY "Support Shareware\nRegister today"

 Notice there are no spaces before or after the \n. If you intend to
 use many lines of BODY, each line should be set as an ENVironment
 variable to save space on the command line. Here's an example:

     SET title "Voter Instructions"
     SET line1 "Support Smith for County Registrar\n"
     SET line2 "Give til it hurts\n"
     SET line3 "Vote early and vote often"

     rtEZRequest "$title" "$line1 $line2 $line3" "_Continue"

     MULTILINE

 There is no default value for BODY. If you choose not to
 present any information, you must include a null string
 as the second string after the argument.

      rtEZRequest "Working...." "" "Continue|Help|Cancel"

     NOBODYTEXT
```

## 1.4 bu

```
     BUTTONTEXT
```

This section applies to the following requesters:

```
     rtGetLong
     rtEZRequest
     rtGetString
```

BUTTONTEXT is optional.

YOU may select how many buttons to show in the window. You may also define the text of each button. The method of defining buttons is the same as ASK. The BUTTONTEXT argument is is not required because there is a default value built in to ReqTools. If you choose to use BUTTONTEXT, the string must appear as the third string following the argument. The use of the keyword BUTTONTEXT is optional.

```
rtEZRequest "$title" "body" BUTTONTEXT "Copy|Delete|Cancel"
```

is the same as:

```
rtEZRequest "$title" "body" "Copy|Delete|Cancel"
```

The default value is "Ok|Cancel". If you don't call a BUTTONTEXT value, ReqTools will supply the default for you.

BUTTONTEXT_DEFAULT

You determine which button was pressed by setting a variable to the value of $RC - the system's Return Code. An example:

```
rtEZrequest "Which Button was pressed" "Press a button" "1|2|3|0"
SET Result $RC
rtEZrequest "Button Tester" "You Pressed button $Result ." "I Get It"
```

BUTTON_TESTER

Notice that $RC returns a number, not the button's text. Always place the variable assignment on the very next line following the call to a requester. $RC changes each time DOS does something significant.

Regardless of the text you assign to the buttons, the rightmost button always returns a zero. The leftmost button returns a one.

In the example above, four buttons were defined. The leftmost button returns a one (1), and the buttons to the right of it return one number more until the rightmost button is reached.

As a practical matter, you should not assign more that 19 buttons to any requester. If you do, the results could interfere with the ERROR VALUE returned by ReqTools.

KEYBOARD SHORTCUTS

Instead of making the user click on a button with the mouse, you can assign a keyboard shortcut to any button. Just prefix the button text with an underscore. Example:

```
    "_Ok|_Help|_Cancel" OR "Quie_t|Q_uilt|_Quit"
```

The underscore lets the user press the letter following it. Here are two interactive examples showing the difference between the two methods:

```
    With Shortcuts
    Without Shortcuts
```

Calling a Help File from a button

You can define a button to call an Amigaguide file to present detailed Help. The script should call the AmigaGuide file with it’s full path. You can also include a "document" within the file to present. Note: when calling Multiview, a "document" cannot be specified. Example:

```
    IF $Result EQ 2
     SYS:Utilities/AmigaGuide SYS:S/MyDocFile.guide help
    ENDIF
```

Where "help" is the NODE Help within the database MyDocFile.

## 1.5 in

INVISIBLE

This section applies to the following requesters:

```
     rtGetLong
     rtGetString
```

This argument will suppress the ECHOing of keystrokes to the screen. Use this argument to take in Passwords.

## 1.6 de

DEFAULT

This section applies to the following requesters:

```
     rtGetLong
     rtEZRequest
     rtGetString
```

The DEFAULT argument sends a string or number to be placed in the requester as a default value. If the user wants to use the default value, they just press return and the value is returned to the script.

rtGetLong deals with numbers, so the default must be a whole number with a value between -2,000,000 and +2,000,000. Don’t include the commas. Example:

```
    DEFAULT 125
```

rtGetString returns a string, so the default must be a string. Example:

```
    DEFAULT "Green"
```

rtEZRequest does not return anything but a button number. You can specify which button will be returned when the return key is pressed. Also see NORETURNKEY. Example:

```
    "Ok|Help|Cancel"
    DEFAULT 2
```

If the user does not use the mouse to click a button, but elects to press the return key, the button number 2 will be returned. This corresponds to the Help key in the example. (1|2|0)

In all instances, the DEFAULT argument needs to be called with it’s name and a value after it. It is a Keyword.

## 1.7 al

```
    ALLOWEMPTY
```

This section applies to the following requesters:

```
     rtGetLong
     rtGetString
     rtFileRequest
```

If the user does not select a value, and presses the return key this argument will force the value in $RC to be true (1). This is the same as clicking the leftmost button. You may use this value to make tests in your script.

## 1.8 ce

```
    CENTERTEXT
```

This section applies to the following requesters:

```
      rtGetLong
      rtEZRequest
      rtGetString
```

The text you include with the BODY argument can be center-justified with this argument. Here are two examples showing the difference:

```
     Center Text
     Left Justified Text
```

## 1.9 mn

```
     MIN
```

This section applies to the following requester:

```
      rtGetLong
```

Use this when you want to specify the  Minimum value than can be returned.

Usage:

```
  MIN <num>
```

## 1.10 mx

```
     MIN
```

This section applies to the following requester:

```
      rtGetLong
```

Use this when you want to specify the Maximum value that can be returned.

Usage:

```
  MAX <num>
```

## 1.11 wi

```
     WIDTH
```

WIDTH is not functional for any ReqTools requester.

## 1.12 no

```
    NORETURNKEY

 This section applies to the following requester:

      rtEZRequest

 This turns off the Return key as a positive response. The user
 must click a button with the mouse or use a keyboard shortcut
 to respond to the requester.
```

## 1.13 cl

```
    COLOR <num>

 This section applies to the following requester:

      rtPaletteRequest


 COLOR presets a color in the requester. Allowable values for COLOR
 are zero through the 15 on a 16 color workbench.
```

## 1.14 po

```
                     POSITION or POS

 This section applies to the following requesters:

      rtGetLong
      rtEZRequest
      rtGetString
      rtFileRequest
      rtFontRequest
      rtPaletteRequest
      rtScreenModeRequest

 You may use the name POSITION or POS interchangeably.
 With this argument, you may place the requester on several places on
 the screen. The options are:

  CENTERSCR    Centers the requester in the screen. (default)
  TOPLEFTSCR   Top left corner of the screen
  CENTERWIN    Centers the requester in the active window
  TOPLEFTWIN   Top left corner of the active window
  POINTER      Places the window so the pointer is over a button
               POINTER is the default if no other option is used.

 Here's an interactive example showing all five positioning statements:
     POSITION.DEMO

 The POSITION argument must be included:

 rtEZRequest "title" "bodytext" POSITION POINTER

 The POSITION argument may be modified by two other arguments:


                TOPOFFSET

                LEFTOFFSET
                 For more information, see those sections.
```

## 1.15 tp

```
    TOPOFFSET or TOP
    LEFTOFFSET or LEFT

This section applies to the following requesters:

     rtGetLong
     rtEZRequest
     rtGetString
     rtFileRequest
     rtFontRequest
     rtPaletteRequest
     rtScreenModeRequest

 TOPOFFSET modifies the POSITION of the window placement. Example:

 TOPOFFSET 100

 This moves the window down 100 pixels from the place POSITION put it.
 Use this Argument in conjunction with the POSITION argument.
 Example:

 rtPaletteRequest POSITION TOPLEFTSCR TOPOFFSET 50

 LEFTOFFSET can also be used on the same line to move the requester
 away from the place POSITION put it. Example:

 rtPaletteRequest POSITION CENTERSCR TOP 50 LEFT 150

 The numeric values following TOP and LEFT refer to an offset of the
 WINDOW, not the from the edge of the screen. Both positive and negative
 numbers are valid. As an example, place the requester in the center of
 the screen with POSITION CENTERSCR. If you modify the position with
 TOP 100, the requester will be moved DOWN 100 pixels. TOP -100 would
 move it up 100 pixels.

 Using the same example, LEFT 100 will move the requester 100 pixels
 to the right. LEFT -100 will move the requester 100 pixels to the left.
```

```
LEFT and TOP will not work with POINTER.
```

## 1.16 le

```
                    LEFTOFFSET or LEFT

This section applies to the following requesters:

     rtGetLong
     rtEZRequest
     rtGetString
     rtFileRequest
     rtFontRequest
     rtPaletteRequest
     rtScreenModeRequest

LEFTOFFSET and TOPOFFSET perform similar functions. For a discription,
see
               TOPOFFSET
               .
```

## 1.17 hi

```
                    HIGHLIGHTTEXT or HIGHLIGHT

This section applies to the following requesters:

     rtGetLong
     rtGetString

This Switch will highlight the BODY text.

    Without Highlight
    With Highlight


Also see
               Backfill
```

## 1.18 ba

```
                    BACKFILL

This section applies to the following requesters:

     rtGetLong
     rtGetString
```

```
 This Switch will change the appearance of the window..

     Without Backfill
     With Backfill
     With Backfill and Highlight

 Also see
                Highlight
```

## 1.19 pu

```
     PUBSCREEN

This section applies to the following requesters:

      rtGetLong
      rtEZRequest
      rtGetString
      rtFileRequest
      rtFontRequest
      rtPaletteRequest
      rtScreenModeRequest

 PUBSCREEN is a KEYWORD. If you want a requester to appear on a Public
 Screen, enter the Public Screen name. Example:

     PUBSCREEN WorkBench
```

## 1.20 to

```
     TO

 This section applies to the following requesters:

      rtGetLong
      rtGetString
      rtFileRequest

 These requesters gather information for you. To use this information
 within your script, use the TO Keyword. You must use this Keyword
 with these requesters or you won't be able to retrieve the
 information the user provided.

 TO must be followed by ENV:<filename>

 The information taken in by the requester is written to the file
 ENV:<filename>

 If you do not need the information to be used in the script, TO can send
 the data to a file anywhere you specify.
```

This is how you would use the TO Keyword:

```
rtGetString "Psych Test" "Your favorite Color?" "_Continue" TO ENV:color
```

The user types in a color and that string is stored in ENV:color

Color Test

To retrieve the variable from within your script, just reference the filename with a $ in front of it. In this case $color.

```
SET response "We think you are nuts because your favorite color is"
rtEZrequest "Psych Evaluation" "$Response $Color"
```

TO can also be used to send the output to a file or the ClipBoard.

## 1.21 dr

This section applies to the following requester:

rtFileRequest

DRAWER is Optional.

Use this is preload the requester with a DRAWER of your choice.

```
rtFileRequest "Suggested Directory" DRAWER "SYS:Prefs"
```

DRAWER

## 1.22 fi

This section applies to the following requesters:

rtFileRequest

FILE is Optional.

Use this to preload the requester with a FILEname of your choice.
You may use this in conjunction with DRAWER, as this example shows:

```
rtFileRequest "Select" DRAWER "SYS:Utilities" FILE "AmigaGuide"
```

FILE

## 1.23 pa

This section applies to the following requesters:

```
    rtFileRequest
```

Preloads the requester with a PATTERN. Perhaps you want to show only files that end with ".font". Make the pattern "#?.font".

```
    rtFileRequest "Select" DRAWER "FONTS:" PATTERN "#?.font"
```

PATTERN

## 1.24 nb

This section applies to the following requesters:

```
    rtFileRequest
    rtFontRequest
```

Normally, the requester remembers the directories it shows in a buffer. If you want to save some memory, you can turn off this feature. NOBUFFER will not remember the directories. This option will also increase the amount of time it takes to re-read directories.

## 1.25 mu

This section applies to the following requesters:

```
    rtFileRequest
```

This will enable the multiselection of files by holding the SHIFT key while selecting files with the Left Mouse Button. If you also want to be able to multiselect directories, use
SELECTDIRS
.

## 1.26 sd

This section applies to the following requesters:

```
    rtFileRequest
```

This is similar to MultiSelect (for files). SELECTDIRS lets you multiselect directories.

## 1.27 sv

This section applies to the following requesters:

```
    rtFileRequest
```

Use this when you allow the user to SAVE/DELETE a file or directory. This argument will disable double clicking, which automatically selects the file or directory you double click. This is a safety feature to prevent accidental SAVEing or DELETEing files and directories.

Also see
TO
.

## 1.28 nf

This section applies to the following requesters:

```
    rtFileRequest
```

When this argument is called, the user cannot select a file. Only directories are allowed to be selected.

## 1.29 pg

This section applies to the following requesters:

```
    rtFileRequest
```

Using this argument will put a gadget in the window that allows the user to enter a pattern. If PATTERN is also called, the selected pattern will be loaded into this gadget.

```
    rtFileRequest "See the Pattern Gadget" PATGAD

    PATGAD


    rtFileRequest "See the Pattern Gadget" PATTERN "rt#?" PATGAD

    PATTERN & PATGAD
```

## 1.30 hg

```
 This section applies to the following requester:

     rtFileRequest
     rtScreenModeRequest

 You may specify how "tall" the requester appears. HEIGHT must be
 followed by a number of pixels. This could be tricky if you don’t
 know the screen size. Try this example:

RTGetLong "Screen Size" "Screen Height?" MIN 200 MAX 900 TO ENV:size
rtFileRequest "Maximum Height is $size Pixels" HEIGHT "$size"
rtFileRequest "HEIGHT is 150 Pixels" HEIGHT 150

     HEIGHT
```

## 1.31 ok

```
 This section applies to the following requesters:

     rtFileRequest
     rtFontRequest
     rtScreenModeRequest

 This allows you to substitute the buttontext OK with another string
 up to six characters in length.

     rtFileRequest "New text for the OK button" OKTEXT "Good"

     OKTEXT
```

## 1.32 vo

```
                This section applies to the following requesters:


     rtFileRequest

 Turns the file requester into a volume/assign disk requester.
 VOLUMEREQUEST must be used with one of these three arguments:


               NOASSIGNS

               NODISKS

               ALLDISKS
               Usage:

 VOLUMEREQUEST NOASSIGNS
 VOLUMEREQUEST NODISKS
 VOLUMEREQUEST ALLDISKS
```

The requester can be used to get a device name ("DF0:", "DH1:",..) or an assign ("C:", "FONTS:",...) from the user.

```
  rtFileRequest "Select a Volume or ASSIGNment" VOLUMEREQUEST NODISKS
```

```
    VOLUMEREQUEST
```

## 1.33 na

This section applies to the following requesters:

```
    rtFileRequest
```

This argument will show only ASSIGNs in the list.

```
    rtFileRequest "Make your selection" VOLUMEREQUEST NOASSIGNS
```

```
    VOLUMEREQUEST - NOASSIGNS
```

## 1.34 nk

This section applies to the following requesters:

```
    rtFileRequest
```

Excludes devices from the requester. Must be used with VOLUMEREQUEST.

```
    VOLUMEREQUEST - NODISKS
```

## 1.35 bn

This section applies to the following requesters:

```
    rtFileRequest
```

Normally, rtFilerequest will return the selected file as a full path/filename. By invoking BASENAME, only the filename portion will be returned.

## 1.36 ak

This section applies to the following requesters:

```
    rtFileRequest
```

Shows ALL devices. The default behavior of the requester is to show only those devices which have valid disks inserted into them. So if you have no disk in drive DF0: it will not show up. Set this flag if you DO want these devices included.

ALLDISKS must be used with VOLUMEREQUEST. Example:

```
VOLUMEREQUEST "Select a Device" VOLUMEREQUEST ALLDISKS
```

```
    VOLUMEREQUEST - ALLDISKS
```

## 1.37 fw

This section applies to the following requester:

```
    rtFontRequest
```

Show only fixed width fonts.

## 1.38 cf

This section applies to the following requester:

```
    rtFontRequest
```

Also show color fonts.

See
CHANGEPALETTE
See
LEAVEPALETTE

## 1.39 cp

This section applies to the following requester:

```
    rtFontRequest
```

Use this argument with COLORFONTS.

CHANGEPALETTE will change the screen’s palette to that of the selected color font. The change will be temporary unless LEAVEPALETTE is also specified.

See

COLORFONTS

See

LEAVEPALETTE

## 1.40 lp

This section applies to the following requester:

rtFontRequest

This argument works in conjunction with COLORPALETTE and CHANGEPALETTE. When the rtPaletteRequest window is closed, the current screen palette will be kept.

See

CHANGEPALETTE

See

COLORFONTS

## 1.41 sc

This section applies to the following requester:

rtFontRequest

Allows fonts to be scaled when they don’t exist in the requested size. Works on Kickstart 2.0 and up,  only. Has no effect on 1.2/1.3.

## 1.42 st

This section applies to the following requester:

rtFontRequest

STYLE will add some gadgets to the window that allow the user to select BOLD UNDERLINE and ITALIC.

## 1.43 sh

```
 This section applies to the following requester:

     rtFontRequest

 The box at the bottom of the window that displays fonts can be re-
 sized with this Keyword. The default is 24 (pixels high). If you
 intend to look at larger fonts, increase this number. Example:

     SAMPLEHEIGHT 75
```

## 1.44 mt

```
 This section applies to the following requester:

     rtFontRequest

 This limits the size of displayed fonts.
```

## 1.45 mh

```
 This section applies to the following requester:

     rtFontRequest

 This limits the size of displayed fonts.
```

## 1.46 rtfl

```
               Useage: rtFileRequest

 TITLE/A,DRAWER,FILE,PATTERN=PAT/K,NOBUFFER/S,MULTISELECT/S,SELECTDIRS/S,
 SAVE/S,NOFILES/S,PATGAD/S,HEIGHT/N/K,OKTEXT/K,VOLUMEREQUEST/S,NOASSIGNS/S,
 NODISKS/S,ALLDISKS/S,ALLOWEMPTY/S,POSITION = POS/K,TOPOFFSET=TOP/N/K,
 LEFTOFFSET=LEFT/N/K,BASENAME/K,TO/K,PUBSCREEN/K

Arguments:


               TITLE
                text>- Requester window title.

               DRAWER
                text>- Preload requester with this drawer.

               FILE
                text>- Preload with DRAWER and this file.

               PATTERN
                text>- Use with PATGAD
               NOBUFFER
                     - Do _not_ use a buffer to remember dir. contents
                       for the next time the file requester is used.

               MULTISELECT
                     - Allow multiple files to be selected.

               SELECTDIRS
                     - Allow selecting of dirs as well as files.

               SAVE
                     - Set this if you are using the requester to save or
                       delete something. Double-clicking will be disabled
                       so it is harder to make a mistake or select a wrong file.

               NOFILES
                     - Select a directory rather than a file.

               PATGAD
                     - Add pattern gadget to the requester.

               HEIGHT
                     - Suggested height of requester window.

               OKTEXT
                text>- New text for "Ok" gadget, max 6 chars.

               VOLUMEREQUEST
                     - Turn the requester into a volume/assign requester.
                       It can be used to get a device name ("DF0:", "DH1:",..) or ↩
                           an
                      assign ("C:", "FONTS:",...) from the user.

               NOASSIGNS
                     - Do not include assigns in the list, only real devices.

               BASENAME
                     - Returns only the root filename, not the path/filename

               NODISKS
                     - Do not include devices, just show the assigns.

               ALLDISKS
                     - Show _all_ devices. Default behavior is to show only
                       those devices which have valid disks inserted into them.
                       So if you have no disk in drive DF0: it will not show up.
                       Set this flag if you do want these devices included.

               ALLOWEMPTY
                     - An empty file string will also be accepted
                       and returned.

               POSITION
                   <pos> - Open requester window at <pos>. Values are:
>or POS                     CENTERSCR , TOPLEFTSCR, CENTERWIN, TOPLEFTWIN, POINTER ↩
    .
```

```
                TOPOFFSET
                    <num> - Offset position relative to above.
>or TOP

                LEFTOFFSET
                    <num> - Offset position relative to above.
>orLEFT

                TO
                   <file> - Send result to <file>.

                PUBSCREEN
                  <screen>- Public screen name.
```

## DESCRIPTION

rtFileRequest allows the user to select a file(s) or directory
to be passed back to your script. This is an ideal solution to writing
installer scripts.

## ABOUT FILENAMES

When the user selects a filename, that name might have SPACE
CHARACTERS embedded in it. If so, the filename will be enclosed in
"double quotes". If no SPACEs are detected, quotes will not be used.

Any filename enclosed in quotes must be handled differently inside
your script. It is assumed you will want to set an ENVironment
variable to use the filename selected.

We recommend you use the TO keyword in all cases. If the user
declines to select a file or directory, IconX might pop up a small
shell window to report something you don't want to the user to see.

If the user has clicked on a file or directory and then cancels the
requester with the Close gadget or the Cancel button, the file "selected"
will be ECHOed to the IconX window. By sending the selection TO ENV,
the window will be suppressed.

Sometimes a failed DOS command in your script will be reported in a
shell window. Typical of these incidents are NO FILE TO DELETE, and
FILES COPIED. You can suppress many of these outputs by using the
QUIET keyword.

## Using MagicFileRequest (MFR)

If you have installed Steven Stuntz's MFR, many of the Keywords will be
ignored. Since MFR presents all ASSIGNS, DEVICES, etc. automatically,
it is useless to specify Keywords like VOLUMEREQUEST. rtFileRequest will
pass TITLE and other information into the MFR requester.

## 1.47 rtfo

```
                Usage : rtFontRequest

  TITLE/A,NOBUFFER/S,FIXEDWIDTH/S,COLORFONTS/S,CHANGEPALETTE/S,LEAVEPALETTE/S,
  SCALE/S,STYLE/S,HEIGHT/N/K,OKTEXT/K,SAMPLEHEIGHT/N/K,MINHEIGHT/N/K,
  MAXHEIGHT/N/K,POSITION=POS/K,TOPOFFSET=TOP/N/K,LEFFTOFFSET=LEFT/N/K,
  PUBSCREEN/K

Arguments:


                TITLE
                   <text> - Title text for requester window.

                NOBUFFER
                          - Do not buffer the font list for subsequent calls.

                FIXEDWIDTH
                          - Only show fixed-width fonts.

                COLORFONTS
                          - Show color fonts also.

                CHANGEPALETTE
                          - Change the screen’s palette to match that of
                            a selected color font.

                LEAVEPALETTE
                          - Leave the palette as it is when exiting rtFontRequest.
                            Useful in combination with CHANGEPALETTE.

                SCALE
                          - Allow fonts to be scaled when they don’t exist in the
                            requested size. (works on Kickstart 2.0 only, has no
                            effect on 1.2/1.3).

                STYLE
                          - Include gadgets to select the font’s style.

                HEIGHT
                    <num> - Height, in pixels, of font requester window.

                OKTEXT
                   <text> - Replacement text for the OK button. Max 6 chars.

                SAMPLEHEIGHT
                    <num> - Height of font sample display in pixels (default 24).

                MINHEIGHT
                    <num> - Minimum font size displayed.

                MAXHEIGHT
                    <num> - Maximum font size displayed.

                POSITION
```

```
                    <pos> - Open requester window at <pos>. Values are:
>or POS                     CENTERSCR, TOPLEFTSCR, CENTERWIN, TOPLEFTWIN, POINTER.

                TOPOFFSET
                    <num> - Offset position relative to above.
>or TOP

                LEFTOFFSET
                    <num> - Offset position relative to above.
>or LEFT

                PUBSCREEN
                 <screen> - Public screen name.
```

DESCRIPTION

rtFontRequest allows the user to view and select a font.
This requester does not change the fonts selected by FontPrefs.

We feel that this requester was never intended to be used from a script.
It was included in this package is the hope that someone may find a use
for it.

## 1.48 rtpr

```
                Useage: rtPaletteRequest

 TITLE,COLOR/N/K,POSITION=POS/K,TOPOFFSET=TOP/N/K,LEFTOFFSET=LEFT/N/K,
 PUBSCREEN/K

Arguments:


                TITLE
                  <text> - Title text for requester window.

                COLOR
                         - Preset a color register.

                POSITION
                   <pos> - Open requester window at <pos>. Values are:
>or POS                   CENTERSCR, TOPLEFTSCR, CENTERWIN, TOPLEFTWIN, POINTER.

                TOPOFFSET
                   <num> - Offset position relative to above.
>or TOP

                LEFTOFFSET
                   <num> - Offset position relative to above.
>or LEFT

                PUBSCREEN
                 <screen>- Public screen name.
```

```
DESCRIPTION

 rtPaletteRequest is a requester that allows the user to change
 the palette, although temporarily, of the Amiga. No permanent setting
 can be made. The effect of this requester is global.

 The single requester-specific argument, COLOR, presets a color in the
 requester. Allowable values for COLOR are zero through the 15 on a 16
 color workbench.
```

## 1.49 rtsm

```
                Usage: rtScreenModeRequest

 TITLE,OVERSCANGAD/S,AUTOSCROLLGAD/S,SIZEGADS/S,DEPTHGAD/S,
 NONSTDMODES/S,GUIMODES/S,HEIGHT/N/K,OKTEXT/K,MINWIDTH/N/K,
 MAXWIDTH/N/K,MINHEIGHT/N/K,MAXHEIGHT/N/K,MINDEPTH/N/K,
 MAXDEPTH/N/K,PROPERTYFLAGS/N/K,PROPERTYMASK/N/K,POSITION=POS/K,
 TOPOFFSET=TOP/N/K,LEFTOFFSET=LEFT/N/K,PUBSCREEN/K

Arguments:


                TITLE
                 <text> - Title text for requester window.
 OVERSCANGAD          - Add an overscan gadget to the requester.
 AUTOSCROLLGAD        - Add an autoscroll checkbox gadget to the requester.
 SIZEGADS             - Add width and height gadgets to the requester.
 DEPTHGAD             - Add a depth slider gadget to the requester.
 NONSTDMODES          - Include all modes. Unless set, rtScreenModeRequest
                        will exclude nonstandard modes. (presently HAM and
                        EHB (ExtraHalfBrite))
 GUIMODES             - Get screen mode to open a user interface screen in

                HEIGHT
                  <num> - Suggested height of screenmode requester window.

                OKTEXT
                 <text> - Replacement text for "OK" gadget, max 6 chars.
 MINWIDTH             - The minimum display width allowed.
 MAXWIDTH             - The maximum display width allowed.
 MINHEIGHT            - The minimum display height allowed.
 MAXHEIGHT            - The maximum display height allowed.
 MINDEPTH             - The minimum display depth allowed.
 MAXDEPTH             - The maximum display depth allowed.
 PROPERTYFLAGS        - A mode must have these property flags to be included.
 PROPERTYMASK         - Mask to apply to PropertyFlags to determine which bits
                        to consider.

                POSITION
                   <pos> - Open requester window at <pos>. Values are:
```

```
>or POS                  CENTERSCR, TOPLEFTSCR, CENTERWIN, TOPLEFTWIN, POINTER.

                TOPOFFSET
                   <num> - Offset position relative to above.
>or TOP

                LEFTOFFSET
                   <num> - Offset position relative to above.
>or LEFT

                PUBSCREEN
                 <screen>- Public screen name.
```

DESCRIPTION

rtScreenModeRequest gives the user the ability to alter the screen mode used by their Amiga. This requester is exceptionally complex and should be used only by advanced programmers.

We feel that this requester was never intended to be used from a script. It was included in this package is the hope that someone may find a use for it.

If you use this requester, the screen mode WILL be changed to the mode selected by the user. Unless you are very sure of how the Amiga handles screen modes, the results of using this requester could be unpredictible.

NOTE: Arguments specific to this requester are not explained in this manual. If you REALLY need to know how to use this requester, begin with the descriptions listed above. Then refer to Nico Francois' docs describing ReqTools.

## 1.50 rtez

```
                Usage : rtEZRequest

 TITLE/A,BODY/A,BUTTONTEXT,DEFAULT/N/K,CENTERTEXT/S,NORETURNKEY/S,
 POSITION=POS/K,TOPOFFSET=TOP/N/K,LEFTOFFSET=LEFT/N/K,PUBSCREEN/K
```

Arguments:

```
                TITLE
                  <text> - Title text for requester window.

                BODY
                  <text> - Body text for requester window.

                BUTTONTEXT
                  <text> - Button text. (default "Ok|Cancel")

                DEFAULT
                   <num> - Default response if [RETURN] key is
```

```
                         pressed without selection.

                CENTERTEXT
                         - Center body text in requester window.
>or CENTER

                NORETURNKEY
                         - Turn off [RETURN] key as positive response.

                POSITION
                   <pos> - Open requester window at <pos>. Values are:
>or POS                  CENTERSCR, TOPLEFTSCR, CENTERWIN, TOPLEFTWIN, POINTER.

                TOPOFFSET
                   <num> - Offset position relative to above.
>or TOP

                LEFTOFFSET
                   <num> - Offset position relative to above.
>or LEFT

                PUBSCREEN
                 <screen>- Public screen name.
```

DESCRIPTION

rtEZRequest is a general purpose advisory requestor. It’s function is to give the user of a choice of responses by clicking on buttons. The identity of the button clicked is available to your script through the system variable $RC.

## 1.51 rtgs

```
                Usage : rtGetString

 TITLE/A,BODY/A,BUTTONTEXT,DEFAULT/K,INVISIBLE/S,ALLOWEMPTY/S,
 CENTERTEXT=CENTER/S,HIGHLIGHTTEXT=HIGHLIGHT/S,BACKFILL/S,WIDTH/N/K,
 POSITION=POS/K,TOPOFFSET=TOP/N/K,LEFTOFFSET=LEFT/N/K,PUBSCREEN/K,TO/K

Arguments:


                TITLE
                <text> - Title text for requester window.

                BODY
                <text> - Body text for requester window.

                BUTTONTEXT
                <text> - Button text for requester window.

                DEFAULT
                <text> - Initial value.
                INVISIBLE
                       - Do not echo keypresses (for passwords)

                ALLOWEMPTY
                       - Empty string returns TRUE.

                CENTERTEXT
                       - Center body text in requester window.
>or CENTER

                HIGHLIGHTTEXT
                       - Highlight body text in requester window.
>or HIGHLIGHT

                BACKFILL
                       - Turn on backfill.

                WIDTH
                 <num> - Width of requester window. (ignored)

                POSITION
                 <pos> - Open requester window at <pos>. Values are:
>or POS                  CENTERSCR, TOPLEFTSCR, CENTERWIN, TOPLEFTWIN, POINTER.

                TOPOFFSET
                 <num> - Offset position relative to above.
>or TOP

                LEFTOFFSET
                 <num> - Offset position relative to above.
>or LEFT

                PUBSCREEN
                <screen>- Public screen name.

                TO
                  <file>- Save output number to this ENV filename.



DESCRIPTION

 rtGetString opens a requester that takes in a text string.
 The string retrieved is sent to an ENV file with the TO keyword.
```

## 1.52 rtgl

```
                Usage : rtGetLong

 TITLE/A,BODY/A,BUTTONTEXT,DEFAULT/K,MIN/N/K,MAX/N/K,INVISIBLE/S,
 ALLOWEMPTY/S,CENTERTEXT=CENTER/S,HIGHLIGHTTEXT=HIGHLIGHT/S,
 BACKFILL/S,WIDTH/N/K,POSITION=POS/K,TOPOFFSET=TOP/N/K,
 LEFTOFFSET=LEFT/N/K,PUBSCREEN/K,TO/K

Arguments:


                TITLE
                 text> - Title text for requester window.

                BODY
                 text> - Body text for requester window.

                BUTTONTEXT
                 text> - Button text. (default "Ok|Cancel")

                DEFAULT
                 <num> - Initial value.

                MIN
                 <num> - Minimum allowed value.

                MAX
                 <num> - Maximum allowed value.

                INVISIBLE
                       - Do not echo keypresses (passwords etc)

                ALLOWEMPTY
                       - Empty string returns TRUE.

                CENTERTEXT
                       - Center body text in requester window.
>or CENTER

                HIGHLIGHTTEXT
                       - Highlight body text in requester window.
>or HIGHLIGHT

                BACKFILL
                       - Turn on backfill.

                WIDTH
                 <num> - Width of requester window. (ignored)

                POSITION
                 <pos> - Open requester window at <pos>. Values are:
>or POS                  CENTERSCR, TOPLEFTSCR, CENTERWIN, TOPLEFTWIN, POINTER.

                TOPOFFSET
                 <num> - Offset position relative to above.
>or TOP

                LEFTOFFSET
                 <num> - Offset position relative to above.
>or LEFT

                PUBSCREEN
                 <screen>- Public screen name.
                TO
                   <file>- Save output number to this ENV filename.



DESCRIPTION

     Use rtGetLong to get a number from the user. Only numbers in the
     range of -2,000,000 to +2,000,000 will be accepted.

     Only whole numbers are accepted. You may specify an allowable
     range with the MIN and MAX arguments. These arguments, when
     called will be shown in the window. If the user tries to enter
     a number greater than the MAX or smaller than the MIN the screen
     will flash, warning of an improper entry. You may use either MIN
     or MAX or both.

STORING AND USING THE OUTPUT

     To get the number back into your script as a variable, you must
     first specify where the number is to be stored, and it's name.
     Use the TO argument to send the number to an ENV file.
     Example:

          TO ENV:age

     Assuming the number is to be used as an age, this argument will
     store the number in the RAM:ENV directory with the filename age.
     The number will now be available to your script as the variable
     named $age.

INTERACTIVE EXAMPLE OF RTGETLONG

     Here is an example of using rtGetLong:

 rtGetLong "Information" "What is your age?" "OK" MIN 15 MAX 70 TO ENV:age
 rtEZRequest "Sensus Data" "Are you are $age years old?" "_Yes|_No"
QUIT

     We will take in a number between 15 and 70, store the number in
     ENV:age, and then use the number in the next requester as $age.

 Now click  to see it work.
```

## 1.53 inst

```
 Installation is very simple. Just click the Install button
 and answer two questions. This will show you a bit of what DOSReqTools
 can do for you. It will also create a file in Env-Archive so the
 rtMan program can find out where you put this document.

 You could also execute Installer.script from the shell.
```

## 1.54 over

```
DOSReqTools by BOTH Software

 Until now, only programmers who have mastered C or other high level
 programming languages could use the power of Nico Francois' ReqTools.
 His requesters have become the defacto standard in the Amiga community.
 Now his requesters may be used by anyone who writes AmigaDOS scripts.

 DOSReqTools allows you to write complete installer scripts, including
 on-line Help. Every facility of Commodore's Installer can be emulated.

 Other uses for DOSReqTools include automating your backup scripts and
 system boot-up scripts.

 You can now include decision making abilities within an AmigaGuide
 database. This database is an example. Your programs could actually
 be installed from within the the Guide Doc. Throughout this Manual,
 you will see interactive examples of this facility. The syntax for
 calling your script from a label is:

       *{"label" SYSTEM "EXECUTE demo.overview"}

                            label

 The requester will pop to the front screen, right on top of the
 AmigaGuide (or MultiView) screen. Can you see some possibilities here?

          About using the demos from this manual:

      As long as the demos and this AmigaGuide database
      are in the same  directory, the demos will pop up
      on the screen.

 This manual is designed to help you learn how to use the various
 requesters, not how to use AmigaDOS. It is assumed you have some
 knowledge of DOS. There are plenty of interactive examples to help
 you learn the syntax of the requesters.

 Each type of requester may be called with a variety of "arguments",
 like TITLE or BODY. These arguments are presented in each section
 of this manual that describes a requester. The sections begin with
 a complete list of the arguments that the requester can use. These
 are labels that will direct you to an explaination of that argument.

 The Command Template for that requester is presented in each section.

 We hope you find DOSReqTools useful. If Nico Francois updates
 ReqTools, we will port the changes into DOSReqTools.

 You will need a passing knowledge of templates to use the requesters.
 There are more that 40 demo scripts included. You can learn quickly
 by studying them.
```

## 1.55 requ

You must have the ReqTools.library installed.
ReqTools.library is not included here.
WorkBench 2.x and up is required.

## 1.56 rule

DOSReqTools gives you access to the entire ReqTools Library.

ReqTools is a library of seven requesters. Each requester pops up a unique window to gather information from the user. When you call a requester from your script you must follow the rules that ReqTools expects. The most important rule is to pass the proper arguments, spelled correctly, and in the proper order.

You can get this information by reading this manual or from the shell. We suggest you read the individual sections that outline the usage of each requester.

The Command Template of each requester is available from the shell by typing the requester's name followed by a question mark. This will show you the command template for that requester. Example:

```
    rtEZrequest ?
```

You can get a verbose listing of the template and the kind of information required for each argument by entering the name of the requester and HELP. Example:

```
    rtEZRequest help
```

The command template for each requester is in the section that describes that requester. Templates are useful to figure out what each requester needs. When you first look at a command template, it may appear to be alphabet soup. Arguments, keywords and qualifiers are presented in a very cryptic manner. This is an AmigaDOS requirement.

What you need to know:

Qualifiers are letters preceeded by a / (slash).

Required Arguments are terms that are followed by a /A.
They must appear in the same order that they are presented in the command template.

Optional Arguments are terms followed by a /N, /K, /S, /F.
Optional Arguments without qualifiers are considered POSITIONAL and REQUIRED. There are three arguments that meet this requirement: BODY, FILE, DRAWER. Positional arguments must appear in the order they appear in the template.

When in doubt, use the Keyword

```
 Examples using DOSReqTools requesters:

 TITLE/A     - This argument is required. All requesters need TITLE.
               Required arguments are presented first in the command
               template and you must call them in the order that
               they are presented. Since they are POSITIONAL, you do
               not have to enter the argument’s name, just the string
               that represents the argument.


 DEFAULT/N/K - This argument is optional. If you use it, you must
               enter the name (DEFAULT) followed by a number. The
               /N tells you the argument will only accept a number.
               If /N is not specified, the argument requires a string.
               The /K means the argument is a Keyword. It is optional
               and if used, the name must be used.


 HIGHLIGHT/S - This argument is optional, and needs no extra
               information following it. Just entering the name of
               the argument turns on that function. The /S means
               this is a Switch.


 CENTERTEXT=CENTER/S - If an argument has two names separated by an
                       equal sign, that argument may be called by
                       either name. These are usually used to shorten
                       a long argument name.


 For a more complete explaination of Arguments and Keywords, see the
 AmigaDOS Reference Guide, Fourth Edition, by Sheldon Leemon.
 This reference is published by Compute Books.

COMMAND TEMPLATE RECAP

 To see the command template for a requester, enter the requester’s
 name in the shell, followed by help and press enter. Example:

     rtGetString help

 The arguments with /A qualifiers are POSITIONAL. They must be used,
 and they must be in order. Using the actual name of the argument is
 optional. Example:

 rtEZRequest TITLE "System Advice" BODY "ReqTools Lib not present"

     is the same as

 rtEZRequest "System Advice" "ReqTools Lib not present"


GETTING INFORMATION BACK TO YOUR SCRIPT

 The requesters gather data from the user. That information, either
```

numbers or strings, must be available to your script as a variable.

Numbers and strings are stored as an ENVirionment variable by using the TO argument. This is exactly the same as redirecting the output to a file in RAM:ENV. You cannot redirect the output of DOSReqTools, so you MUST use the TO argument to store data. You define the filename and you decide how it is to be used in your script. Example:

```
rtGetString "Data" "What is your Phone Number?" TO ENV:phone
```

The phone number entered by the user is now available to you as $phone.

In addition to the data gathered, you may need to know which button the user pressed. You define the number of buttons and the text of each button. DOSReqTools returns the number of the button pressed in the system variable $RC (return code). Example:

```
    SET Result $RC
```

The value of the button pressed is now available as $Result.
For more information see
BUTTONTEXT
.

## ABOUT THIS MANUAL

We feel that the likeliest users of DOSReqTools will be those people who do not write in C, or Pascal, or any other high level language. For that reason, we have tried to include simple to understand programming examples throughout each section. Many of the concepts and details are presented twice.

The unique nature of DOSReqTools allows us to present an actual script, and then let you click a button to see the script in action. None of the demo scripts will do anything to your system, so feel free to click any button that is presented.

Important subheadings are presented in HIGHLIGHT (white for most).
Scripts are presented in FILL (blue for most).
This applies only to users of AmigaGuide v37 (OS3.0).

We have tried to use the same calling syntax to ReqTools that a C programmer would use. The obvious difference would be the TO argument. This argument had to be added so that data could be made available to the calling script.

## rtMan -"MAN PAGES" for DOSReqTools

This is a big complicated package to memorize. If you are like me, you might use DOSReqTools occasionally to jazz up a script. It’s hard to remember the calling syntax for every requester, so a small scale Man page script is included to help you.

rtMan is the script and it requires a requester name. If you forget this, just type in rtman from the shell and a help text will prompt you. Example:

```
   ->rtman rtezrequest

 This command  will call this AmigaGuide database and display the document
 (node) that describes that requester. Don’t change the call to  Multiview.
 It can’t display a database. rtMan will not function unless you use the
 Installer.script. See
                Install DOSReqTools
                 We understand that rtFontRequest and rtScreenModeRequest are  ←
                    pretty
 useless when called from a script.

 If you can think of an actual use for these requesters, contact us
 and perhaps we can rewrite them to make they useful to you.
```

## 1.57 lice

```
               These programs use the reqtools.library
               which is copright (c) by Nico Francois.
               =======================================
```

## 1.58 cred

```
Our thanks go to:

 Nico Francois for his pioneering efforts in standardizing the look of
 our beloved Amiga.  His ReqTools Library of requesters has become the
 defacto standard for the programming community. We hope our efforts
 will widen the use of his requesters to a group of programmers that
 have not had access to them before.

 And a big Thank You goes to Rainer Scharnow because he gave me the
 idea. Ranier wrote MultiRequestChoice and I used it in my scripts.
```

## 1.59 xx

```
     This argument is not described in this manual.
     Please refer to the ReqTools docs for more information.
```

## 1.60 both

If you find any bugs or have any comments, please contact the Authors. Email messages will be answered immediately. Snail Mail letters will take longer since we never learned to read.

C & UNIX Code<br>
--------------------------<br>
Will Bow<br>
1531 Corinth Unit 1<br>
West Los Angeles, CA 90025<br>
(310) 478-4913<br>
InterNet:wbow@crl.com

Program Design, GUIs, Docs<br>
--------------------------<br>
Colin Thompson<br>
9606 Carroll Cyn. Rd. H9<br>
San Diego, CA 92126<br>
(619) 695-2181<br>
InterNet: colin@cts.com

------------------------------------------------------------------

COMALITES UNITE !!!!

FIELD TESTERS NEEDED FOR
A COMAL PORT TO THE AMIGA

25 November 1994

We are writing a replacement for the Amiga Shell, based on COMAL.20. The COMAL Shell will be DOS compliant with COMAL structures. It is based on the COMAL 85 standard, with several extensions added.

If you were a COMAL programmer on the C64, and would like to see COMAL ported to the Amiga, please contact us. Len Lindsay has closed the COMAL Users Group, so he won't be able to help us much on this project. We have the BNF notation for the language and have just finished writing the parsor. We need your input on how the language should be implimented. As we see it, COMAL will be used as a replacement for the Shell. We will include all of COMAL.20 except turtle graphics, PEEK and POKE. The effect of this will be to give you access to all of the looping structures, string handling, file handling, and other wonderful features COMAL has, but DOS doesn't.

Our implementation will incorporate a script driven version of MUI. This will give COMAL an infinate variety of requesters and GUIs. It will also give COMAL access to intuition. Try THAT from DOS!

COMAL on the Amiga would be interpretive, including the LOAD and RUN commands. We have looked at EVERY shell ever written for the Amiga and, frankly they are mostly UNIX creatures, not intended for amateur programmers. We hope to put a structured, high level language back in the hands of those who haven't learned C. COMAL was invented by Borg Christiansen in 1976. He did it because Pascal was simply too difficult to teach to most people.

> A COMAL compiler is also planned. This would give registered developers the ability to write commercial programs in COMAL and compile them for distribution.

We believe the resurrection of COMAL would be a boon to the Amiga Community. If an easy-to-learn language was available, perhaps we would see a new generation of programmers showing off their skills. Maybe Amiga User Groups would flourish again.  Maybe pigs will fly. Who knows. No one will ever know unless we try.

If you don’t remember COMAL, but are skilled in high level languages, we want to hear from you. We’ve never written a language before.

Will Bow  
Colin Thompson

## 1.61 dist

DOSReqTools is copyright 1994 by BOTH Software. It is GIFTWARE. That means if you like it and use it, you may send us a Gift. A birthday card or email would be nice. We would really like to hear about how you’ve used the requesters.

If you feel a requester could be made more useful, please contact BOTH Software.

DOSReqTools is free to use and copy as you see fit. The Authors retain all rights to the programs. The Authors are not responsible for any damage the programs may cause. If you distribute the programs, you MUST distribute the entire package. The exception to this requirement would be if you wrote a script that used some of the requesters or extensions, you could distribute just those programs. We ask that you DO NOT RENAME the DOS extension’s filenames. This could cause confusion.

DOSReqTools may be included in any archive collection.

Tested on a 3000, 2000, 1000, 600, and a 1200 without problems.

## 1.62 dosx

EXTRA PROGRAMS INCLUDED IN THIS PACKAGE

Since DOSReqTools is intended for use by script writers, we chose to release some DOS Extensions with this package.

These programs are intended to be used in a DOS Script:

```
    GetString
```

Usage is simple. It will take in a string from the user, from the shell, and send the response to an ENVironment Variable.

```
    GetString >ENV:name "What is your name:"
```

The string "What is your name?" is a prompt that will appear on the command line, prompting the user to type in his name. The response is available as $name.

Do not use this command in ‘backticks‘.

### InString

This will compare two strings to see if one string appears in the other string. If the test is true, InString returns WARN and the starting and ending position. If the test fails, InString returns a zero in $RC and a starting/ending position of 0:0

InString is presumed to be case insensitive. If you want the test to be case sensitive, include the flag CASE.

Usage:

```
SET $filenamme "picture.iff"
InString ".iff" $filenamme
IF WARN
  ECHO $filenamme is an IFF file"
ELSE
  ECHO $filenamme is not an IFF file"
ENDIF
```

If .iff is found within $filenamme, InString will return a 5, or WARN. InString is not positional. If the substring is found anywhere in the string, the result will be TRUE (WARN)

To get the actual placement of the substring you are testing for, redirect the output to a variable.

```
InString >env:place "BC" "aBCdef" CASE
IF WARN
  ECHO $place
ELSE
  ECHO "Not Found"
ENDIF
```

Since the test is true, $place will hold "2:3".

InString may be used in backticks like this:

```
IF ‘InString ".iff" $filename‘
```

If the condition is TRUE, WARN is returned
If the condition is FALSE, a zero is returned

### SubString

This command will allow you to pull a substring from a string. The substring is defined by a starting position and an ending position.

Start_pos should be 1 or greater.

Usage:

```
     SET temp `SubString <string> <start_pos>:<end_pos>`
     SubString >variable <start_pos>:<end_pos>
```

An example:

```
     SET test "1234567890"
     SET result `SubString $test 3:6`
     ECHO $result
     3456
```

Using the example above, let’s look at how the command handles errors and out of range problems:

```
     0:5  returns 12345
     1:5  returns 12345
     5:15 returns 567890
     6:6  returns 6
     6:7  returns 78
    11:8  returns nothing or the previous value of $result
     6:5  returns nothing or the previous value of $result
     2:   returns 234567890
      :3  returns 123
      5   returns 567890
```

The returned value will not have a trailing carraige return. If you desire a carraige return, use ECHO like this:

```
     ECHO  `SubString <string> <start_pos>:<end_pos>`
```

### EqualsNull

This will test a string (or ENV variable) to see if contains data or is a null string.

Usage:

```
     EqualsNull <string>         (test a string)
     EqualsNull <a_file          (input from a file)
     EqualsNull $a_variable      (test a variable)
```

Examples:

```
     EqualsNull ""
     IF WARN
       ECHO "The string is empty"
     ENDIF


     EqualsNull <a_file_somewhere
     IF WARN
       ECHO "The string is empty"
```

```
    ENDIF


    EqualsNull $any_variable
    IF WARN
      ECHO "The string is empty"
    ENDIF
```

If the string is empty (null) a 5 is returned in $RC.
If the string is not empty, a 0 is returned in $RC.

```
    ChangeExt
```

This will change the filename extension of a filename that is stored in a variable. It won’t change a filename on a disk. If you need to do that, use NewExt, by Lee Kindness. It’s available on AmiNet in util/batch.

Example:

```
SET sourcename "picture.jpg"
ChangeExt >env:temp $sourcename ".gif"
SET sourcename $temp
ECHO $sourcename
picture.gif
```

If no extension is specified (with a null string), the existing extension will be removed.

```
SET source "DH5:graphics/picture.jpg"
ChangeExt >env:temp $source ""
SET source $temp
ECHO $source
DH5:graphics/picture
```

The following three Extensions work together to allow you to separate the filename from it’s path, and then reassemble them together. They were written so that a selection from rtFileRequest could be broken up, then reassembled with a new path or filename. This is useful for graphics conversion programs when you want to convert a file from one format to another, then send the new file to a different directory with a new name or extension.

Also see the description of rtFileRequest/BASENAME.

To help you understand how to use these Extensions, we will refer to this example:

```
rtFileRequest "Convert a JPEG to a JGIF" DRAWER DH3:Pix/JPEG TO ENV:source
(After selection, ENV:source = DH3:Pix/GIF/JTKirk.jpg)
```

```
   PathPart
```

This will assign "DH3:Pix/JPEG" to a variable:

```
   PathPart >env:sourcepath $source
   ECHO $sourcepath
   DH3:Pix/JPEG
```

```
   FilePart
```

```
   FilePart >env:sourcename $source
   ECHO $sourcename
   JTKirk.jpg
```

At this point we have split the path from the filename and have them in two variables: $sourcepath and $sourcename. If we continue with our example of a graphics convertor, we would call rtFileRequest again to get the target directory. Assume the target directory selected by the user is "DH3:Pix/GIF", and it was stored in ENV:targetdir. Now we will make up a new target filename.

```
   SETENV targetname $sourcename
   ChangeExt >env:tempname $targetname ".gif"
   SETENV targetname $tempname
   ECHO $targetname
   JTKirk.gif
```

```
   AddPart
```

Now we can make up the entire targetname by tacking on $targetname to $targetdir with AddPart.

```
   SETENV destination $targetdir
   Addpart >env:tempname  $destination $targetname
   SETENV destination $tempname
   ECHO $destination
   DH3:Pix/JPEG/JTKirk.gif
```

AddPart expects the first entry to be a path and the second entry to be a fileame. It merges the two together as $tempname. Syntax is checked and corrected if necessary.

Now we have all the variables necessary to pass to the conversion program. For this example, we will call DJPEGv5, which converts a JPEG file into a GIF. Here’s how it would look:

```
   djpeg -gr -gif -scale 1/2 $source $destination
```

This call would convert $source (DH3:Pix/GIF/JTKirk.jpg) and write it out to $destination (DH3:Pix/JPEG/JTKirk.gif)

A working version of the this example script is provided as demo.ext.
The explaination of these extensions may be hard to follow. The file demo.ext
is fully annotated and may offer a better description of syntax.

These extensions can be found in the directory DOSExtensions. The source
files for are included. If you Install DOSReqTools from the installer
section, these files will be copied along with the requesters.