ARx_PrgUcr.ag

COLLABORATORS

	TITLE : ARx_PrgUcr.ag		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 8, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

ARx_PrgUcr.ag

1.1 Tutorial: A program to automate dissolving of archived files

```
   AN AMIGAGUIDE® TO ARexx                 Edition: 1.0
   by Robin Evans
```

Note: This is a subsidiary file to ARexxGuide.guide. We recommend using that file as the entry point to this and other parts of the full guide.

1.2 Tutorial: A program to automate dissolving of archived files

The program that follows is but one way to accomplish a task which is often one of the more tedious for those who use files from networks and bulletin boards: uncrunching files from archives created with a variety of archiving utilities.

This program provides a single, consistent CLI interface for the many different utilities.

View program

The tutorial demonstrates the use of internal functions , the ARG instruction, comparison operators , and of the several built-in functions that allow manipulation of character strings.

Pressing any of the highlighted lines calls up explanatory text for that line. The explanations include extensive links to nodes elsewhere in this guide. Pressing any of those link points with the shift key depressed will open a new window for the link. If a new window is not opened, pressing -Retrace- will return the the program explanation node.

The complete text for this program is included in the file in the distribution archive: ARx_UnCrunch.rexx.

Next: Program code | Prev: Tutorials | Contents: Tutorials

1.3 ARexxGuide | Tutorials | UNCRUNCH.REXX

```
/**** UnCrunch.rexx ****************  Unarchive any file ****************
**
**    A tutorial example to ARexxGuide
*/

 arg FileName UCdir .

 if FileName = '?' then
 do
    options prompt 'UCR FILENAME/A, DESTINATION/F: '
    pull FileName UCdir
 end

 if FileName = '' then
    do
      call Usage('You must supply the name of a file to uncrunch.')
      exit 19
    end
    else
      if ~exists(FileName) then
      do
         call Usage(Can't open' FileName 'for input - object not found)
         exit 20
      end
 if UCdir = '' then
    UCdir = 'RAM:'
 else
    if right(UCdir, 1) ~= ':' & right(UCdir, 1) ~= '/' then
       UCdir = UCdir || '/'

 ArcExt = upper(right(FileName, 3))

 address command
 select
    when ArcExt = 'LZH' | ArcExt = 'LHA' then
       /* Substitute 'lz' for 'lha'if you prefer to use it */
       'Lha -x x' FileName '#?' UcDir
    when ArcExt = 'ZOO' then
       call SetDest('ZOO x//')
    when ArcExt = 'ARC' then
       call SetDest('ARC x')
    when ArcExt = 'ZIP' then
       call SetDest('UNZIP')
    otherwise
       say FileName 'is not valid archive'
       say 'Filename must end in "LZH", "LHA", "ZOO", "ZIP", or "ARC"'
       exit 11
 end
 exit


   /* ZOO and ARC uncrunch all files to the current directory           **
   ** The following subroutine changes the current directory to that    **
   ** specified for uncrunching                                         */


 SetDest:
    parse arg AProg
    cdir = pragma('D')          /* Store the current directory */

       /* Change archive filename to a path/file that can be      */
       /* understood from a different directory                   */
    FDir = CDir
    select
       when pos(':', FileName) > 1 then
          FDir = ''   /* device is already specified. Don't change*/
       when left(FileName, 1) == '/' then
             /* Strip off leading '/' in filename */
       do while left(FileName, 1) == '/'      /* is 1st char '/'? */
          FileName = substr(FileName, 2)      /* strip it off     */
          PLn = length(FDir) - 1
          DivPos = max(lastpos('/', FDir,Pln),,
                            lastpos(':', FDir, Pln))
          FDir = left(FDir, DivPos)
       end
       when left(FileName,1) = ':' then
          parse var CDir FDir ':' .
       otherwise
          if verify(right(FDir, 1), '/:', 'M') ~= 0 then
             FDir = FDir'/'
    end
    FileName = FDir||FileName

       /* Perform the uncrunching                                 */
    'cd \
     ''AProg '\
    'cd "'CDir'"'

 return

Usage:
   if arg(1, 'E') then
      say arg(1)
    say 'Usage:'
    say '  ucr <filename> [destination]'
    say '      <filename> must end with valid archive extension.'
    say '      [destination] is the dir in which files will be uncrunched.'
    say ''
    exit 20
 end
```