$k ![logo] varChar A system for characterizing Dyalog APL variables and component files.

In second generation APLs, since the introduction of enclosed arrays, it has become more difficult to grasp the details of complicated nested structures. Initially there were three different types of depth one arrays, viz. numeric, character and mixed. APL2 introduced a very useful function called DISPLAY which goes a long way towards revealing precise details of arbitrarily nested arrays of these types. The varChar system is based on a (recursive) DISPLAY function such as the Dyalog version below. (Note: not all characters always display well; in particular look out for cup, domino and the line-drawing characters (eg see second line below)).

```
DISPLAY←{⎕IO ⎕ML←0                    ⍝ Boxed display of array.

    α←0 ◇ chars←α⊃'..''''|-' '_____'  ⍝ α: 0-clunky, 1-smooth.

    tl tr bl br vt hz←chars              ⍝ Top left, top right, ...

    box←{                               ⍝ Box with type and axes.
        vrt hrz←(¯1+⍴⍵)⍴¨vt hz          ⍝ Vert. and horiz. lines.
        top←(hz,'⊖→')[¯1↑α],hrz         ⍝ Upper border with axis.
        bot←(⊃α),hrz                    ⍝ Lower border with type.
        rgt←tr,vt,vrt,br                ⍝ Right side with corners.
        lax←(vt,'⌽↓')[¯1↓1↓α],¨⊂vrt     ⍝ Left side(s) with axes,
        lft←⍉tl,(↑lax),bl               ⍝ ... and corners.
        lft,(top⍪ω⍪bot),rgt             ⍝ Fully boxed array.
    }

    deco←{α←type open ⍵ ◇ α,axes ⍵}     ⍝ Type and axes vector.
    axes←{(-2⌈⍴⍴⍵)↑1+×⍴⍵}               ⍝ Array axis types.
    open←{(1⌈⍴⍵)⍴⍵}                     ⍝ Expose null axes.
    trim←{(~1 1⍷∧⌿⍵=' ')/⍵}            ⍝ Remove extra blank cols.
    type←{{(1=⍴⍵)⊃'+'⍵}∪,char¨⍵}        ⍝ Simple array type decorator.
    char←{θ≡⍴⍵:hz ◇ (⊃⍵∈'¯',⎕D)⊃'#~'}∘⍕ ⍝ Simple scalar type decorator.

    {                                   ⍝ Recursively box arrays:
        0=≡⍵:' '⍪(⎕FMT ⍵)⍪(⊃⍵ ⍵∈⎕AV)⊃' -' ⍝ Simple scalar.
        1 θ≡(≡⍵)(⍴⍵):'∇' 0 0 box ⎕FMT ⍵  ⍝ Object rep: ⎕OR.
        1=≡⍵:(deco ⍵)box ⎕FMT open ⍵     ⍝ Simple array.
        ('∈'deco ⍵)box trim ⎕FMT ∇¨open ⍵ ⍝ Nested array.
    }⍵
}
```

As Dyalog APL has developed beyond the second generation paradigm, Dyalog APL arrays have become richer and more varied in their content. There are now eleven different possible basic types of these super-mixed arrays

```
    11=+/2 3 4!4
```

---

constructed from combinations of numeric, character, reference and null scalar elements. This system elucidates these, and other, sometimes very complex and quite large, structures.

```
      ≡2 3 4 'a' 'b' □null #
1
```

Another version of this system is implemented as a new button in the Dyalog APL session. It can pick up current objects from the APL session for examination by using the *CurObj* property of □*SE*. That version (*CHAR*.*DWS*) is of real value to an APL programmer who needs to know the exact structure of some array with which (s)he is dealing.

This free version is distributed in the from of a stand-alone system based on CHAR.EXE . This version is of value to those wishing to understand details of arrays created by an arbitrary line of APL code or to those wishing to view components of Dyalog APL (*.DCF) files.

Installation of varChar version 2.1

This help file relates to **varChar version 2.1**, distributed free to APL programmers and other interested parties.

varChar Version 2.1 comes in a directory called varChar, the contents of which should be copied into a directory of the user's choice - say `C:\varChar\` or `C:\ProgramFiles\varChar\`. The directory then contains these files:

- **char.exe** This is the main system file. It may placed in a shortcut with target `C:\ProgramFiles\varChar\Char.exe` and start-in directory `C:\ProgramFiles\varChar`
- **winUK.din** Key input file, which should be used in conjunction with the fixed width Dyalog Std TT font (`dlogttst.ttf`), to relate keystrokes to APL symbols (the standard key output file (`win.din`) is also required).
- **winDK.din** Alternative key input file for Danish users.
- **char.hlp** This varChar help file.
- **charUK.reg** Creates the initial registry entry. Double-click on this file in Windows Explorer to import the contents into your registry. View this file first if you want to know what `regedit.exe` will do with it. APL characters will not be available until the registry entry has been installed.
- **charDK.reg** Creates the initial registry entry for Danish users.
- **charDemo.dcf** Demonstration file which contains a random collection of more or less complicated examples of Dyalog APL arrays.
- **Dyalog10rt.dll** This is the Dyalog APL Version 10.0 run-time engine.
- **greetcha.bmp** Greeting bitmap which is displayed when varChar is started. Also used in the [Help][Greet] gravity animation.
- **GRIco16.ico** Small logo displayed on the main form caption bar.
- **GRIco32.ico** Logo displayed on the shortcut and in the Alt+Tab menu.
- **Stop.bmp** A stop sign that appears if an error is encountered. A file called `trapLog.Dcf` is created. It can be deleted at any time without affecting system performance. However, system performance might improve in future releases if you send your trapLog file with irritating or serious errors (or indeed any type of comment you might wish to convey) to Graeme Robertson Ltd, 15 Little Basing, Old Basing, Basingstoke RG24 8AX, England, tel +44(0)1256-364071, email GraemeDR@nildram.co.uk All requests for changes and improvements will be gratefully received.

Installation of this system involves four or five simple steps.

1. Copy the contents of the varChar directory into a directory of your choice, say C:\varChar

2. Doubleclick on the file charUK.reg or charDK.reg in Windows Explorer to install the registry entry HKEY_CURRENT_USER\Software\Robertson\Char .

3. If you do not have the fixed-width APL font called "Dyalog Std TT" installed on your computer or have difficulty displaying APL characters in the help file, then install font file Dlogttst.ttf which is included in the package, or is downloadable from http://www.dyalog.com/ .

4. Doubleclick on the file `Char.exe`, or,

5. Create a shortcut on the desktop with Target: `C:\varChar\Char.exe` and Start In: `C:\varChar` and doubleclick on that.

- [Contents Page](#) Return to front page of help file.
- [Main Window](#) Initial appearance of the main window.
- [Menus and ToolBars](#) Explanation of system features.
- [Demonstration File](#) Examples brought together in a demonstration component file.
- [Known Problems](#) Top Twenty Known Problems

---

install

$ Installation of Char

k Installation;Setup;Char.din;Char.reg;GreetCha;.exe;.dll;.ico;Char.hlp;Dyalog Std TT;.ttf;.bmp;
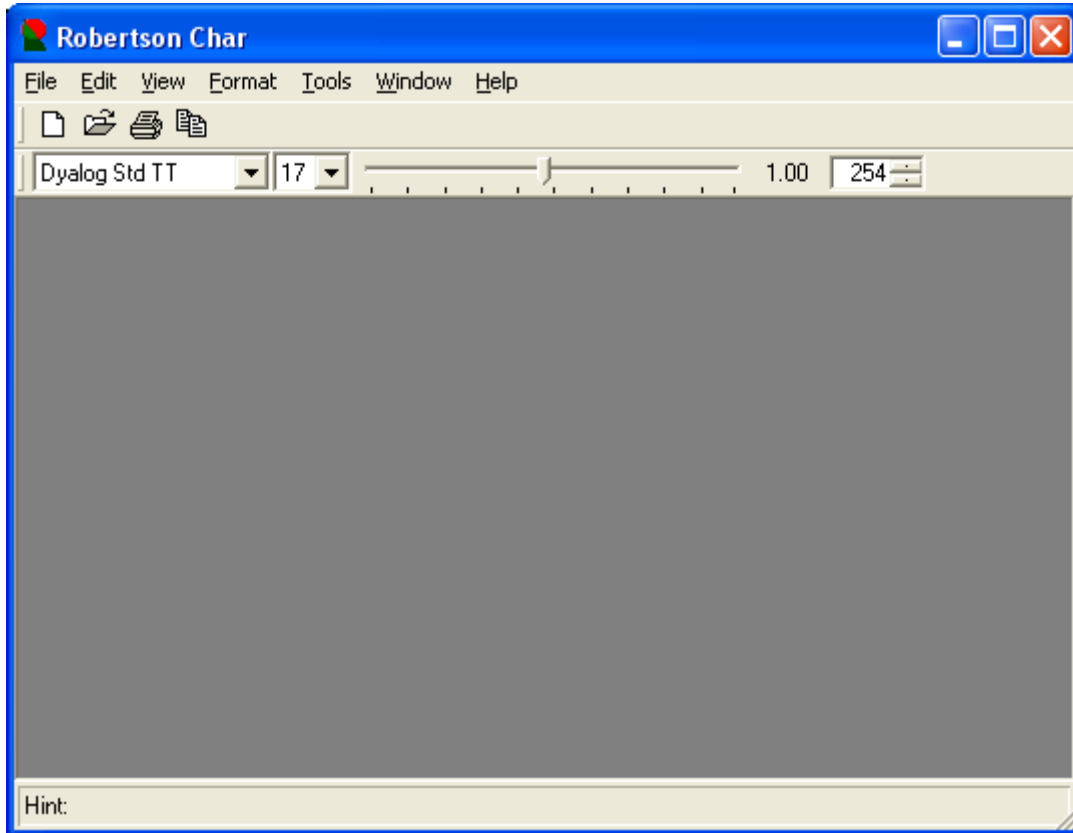
+ top:02

- <u>EULA</u>   End User Licence Agreement

#$kMain Window

When varChar is first started, an instance of your default printer is created. If there is no printer available then varChar will not be able to proceed. After the printer object has been created and the GreetBitmap has been displayed, the system will appear something like this:

Subsequent initiation of the system will reflect the size and position of the Form as it was last used. The current positions of the two ToolBars are also stored in the registry.

The Format ToolBar contains a FontName Combo, a FontSize Combo, the Zoom TrackBar and the Opacity Spinner. When you select a new font or font size, the next and subsequent arrays to be viewed make use of this new font – only *fixed width* fonts are displayed for use. The Zoom TrackBar controls the magnification of the active array. The Opacity Spinner (255-0) controls the translucence of the main varChar window, thus allowing windows behind to be made partially visible. (Beware of zero opacity!)

- Contents Page   Return to front page of this help file.
- Installation   How to install this product.
- Menus and ToolBars   Explanation of system features.
- Demonstration File   Examples brought together in a demo file.

#$kMenus and ToolBars

[File]   The **FILE** Menu

  [Current]   creates a representation of session CurObj (only available in the **Session Button** version (**Char.dws**) of varChar.)

  [New] [†]   creates a representation of any legal **APL array expression**, or **workspace variable**

  [Open] [†]   opens component(s) from a **.DCF file**
  [Close]   closes active array (or use **Esc** key)
  [Setup]   displays printer setup dialogue box
  [Print] [†]   **prints** pending pages
  [Properties]   describes the active array
  [Exit]   exits Robertson varChar

[Edit]   The **EDIT** Menu

  [Copy] [†]   copies current **metafile** onto the clipboard
  [Find]   finds a string in the active array representation (*unavailable in Char.exe*)

[View]   The **VIEW** Menu

  [Screens]   switches to the screen representation of the active array
  [Paper]   switches to the paper representation of the active array
  [Cross]   toggles the background cross on/off
  [Zoom] [†]   specifies the Zoom TrackBar thumb value in Char.dws version (use Zoom TrackBar directly in Char.exe version.)

[Format]   The **FORMAT** Menu

  [Font] [†]   switches to another fixed width true type font in Char.dws version (use Format ToolBar controls in Char.exe version.)

  [Elements]   allows the user to select a different colour scheme (only in the Session Button version (Char.dws)).

  [Scale to Fit]   scales the active array representation to fit the current window (only in the Session Button version (Char.dws)).

  [Centre]   positions the centre of the picture at the centre of the window (only in the Session Button version (Char.dws)).

[Tools]   The **TOOLS** Menu

  [Locate]   places a specified part of the array within the given window (only in the Session Button version (Char.dws)). (Related to Locator on bottom right square static.)

  [Pan]   moves the array to a specified position (only switchable in the Session Button version (Char.dws)).

  [Diagnostic Log]   displays all the errors which have been logged while you have been using varChar. Sending this file to Graeme Robertson (GraemeDR@nildram.co.uk) is a good way to report errors.

  [Log an error]   forces an artificial error so that you can write a note in the diagnostic log about a problem or suggestion

[Window]   The **WINDOW** Menu

  [Cascade]   cascades the open windows
  [Tile Vertically/Horizontally]   vertically/horizontally tiles the open windows
  [Minimize All Windows]   minimizes all the open windows
  [Restore All Windows]   restores all the minimized open windows
  [Close All Windows]   closes all the open windows

[Help]   The **HELP** Menu

[#] menu

[$] Menus and ToolBars

[k] Menus;ToolBars;Menu Items;Buttons;File Menu;View Menu;Edit Menu;Format Menu;Tools Menu;Window Menu;Help Menu

[†] top:04

[varChar Help]   opens this help file
[Greeting]   displays an animated system greeting
[About Robertson varChar]   program information, version and copyright


† Also can be accessed from the ToolBars.

- Main Window   Initial appearance of the main window.
- Menus and ToolBars   Explanation of system features.
- Demonstration File   Examples brought together in a demonstration component file.
- Known Problems   Top Twenty Known Problems

If you have the Dyalog Std TT font installed on your computer, and winUK.din is correctly assigned to APLKeys in the Registry (via charUK.reg) then you can type any APL character into the Combo box. In this way you can write arbitrary one-line APL expressions into the edit field.
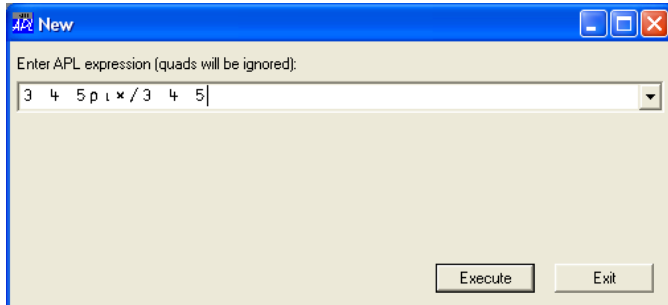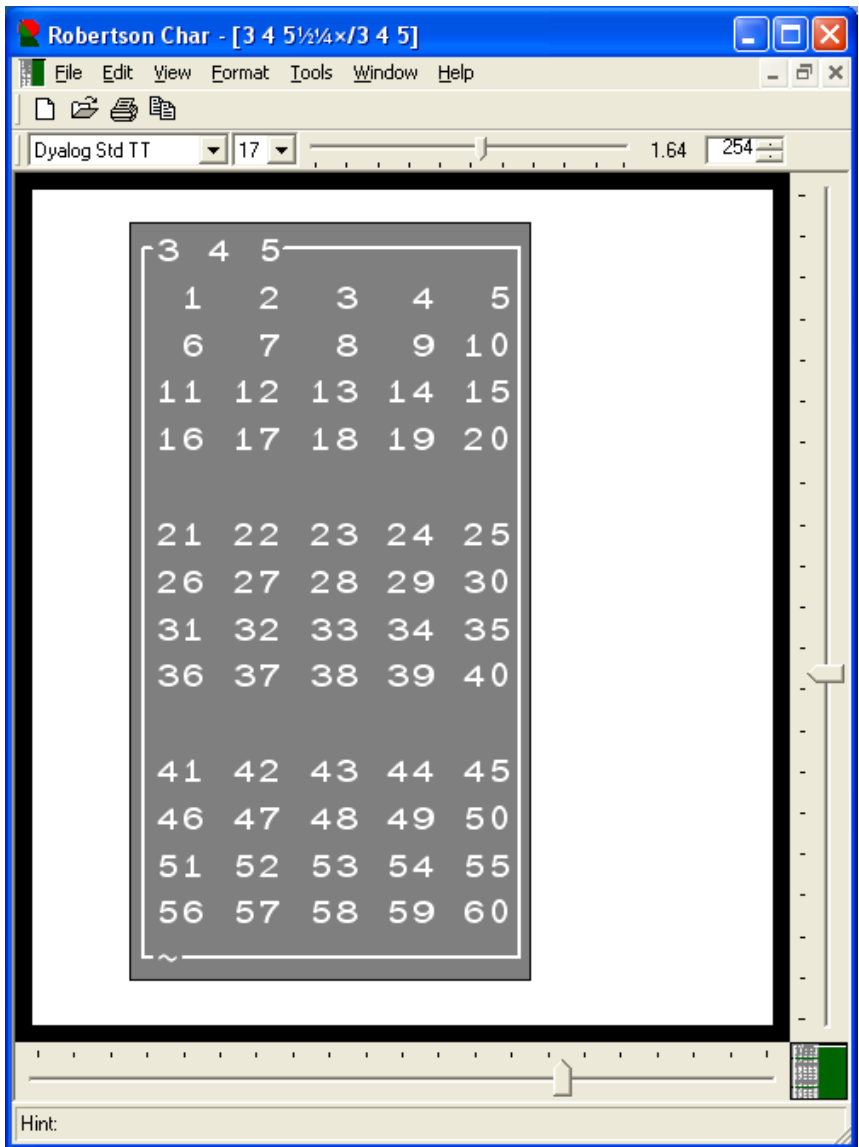

If they return array results then these results will be displayed on an MDI SubForm as shown below.

Robertson Char - [3 4 5½¼×/3 4 5]

File Edit View Format Tools Window Help

Dyalog Std TT   17   1.64   254

```
┌3  4  5─────────────┐
    1    2    3    4    5
    6    7    8    9   10
   11   12   13   14   15
   16   17   18   19   20

   21   22   23   24   25
   26   27   28   29   30
   31   32   33   34   35
   36   37   38   39   40

   41   42   43   44   45
   46   47   48   49   50
   51   52   53   54   55
   56   57   58   59   60
└~───────────────────┘
```

Hint:

Features to note are:- the shape (3 4 5) in the top left corner, the type (~) in the bottom left corner and the colour (white on grey). The significance of type and colour are described in [Format][Elements].

If an APL error is encountered as a result of an ungrammatical expression typed into the Combo box then the error will be noted on the [File][New] window.

New

Enter APL expression (quads will be ignored):

3+'a'

DOMAIN ERROR

3+'a'

Execute   Exit
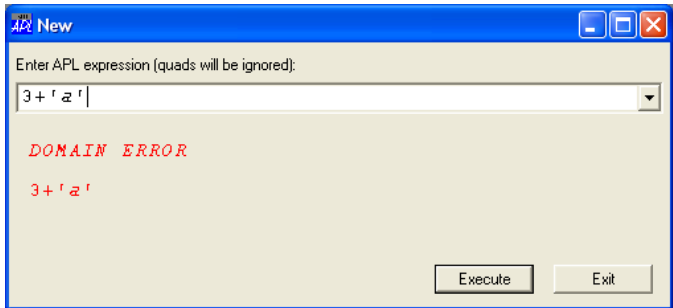
- <u>Menus and ToolBars</u>   Explanation of system features.

#$k+File Open   represent file component(s)

This option displays a standard FileBox and allows you to select any .DCF file. Sometimes Dyalog APL component files are given an extension other than .DCF in which case it can be hard to identify them as such. The "Files of type" Combo may be used to select "All files" and any file may be selected. If the file is not an APL component file its content is presented as a square character array.
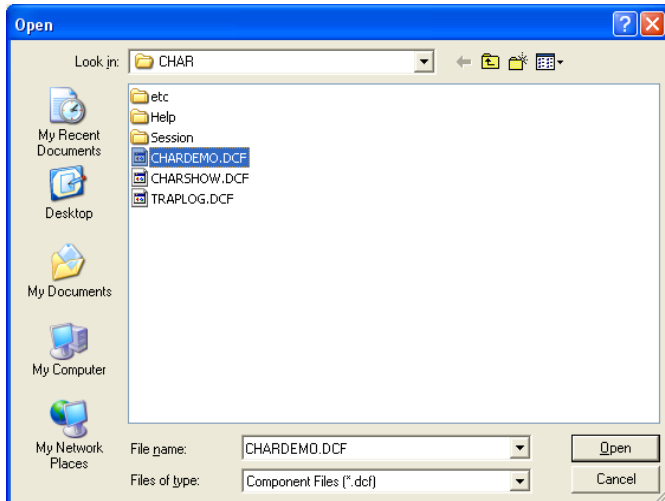
If the selected file is a component file with only one component then that component is displayed immediately. If the file contains a number of components then a dialogue box is presented and you can select one or many components to display. Each component will be displayed in a separate MDI SubForm.
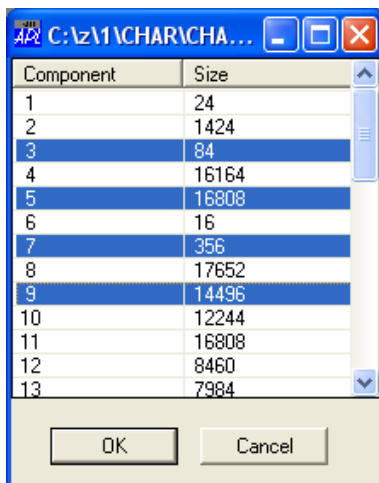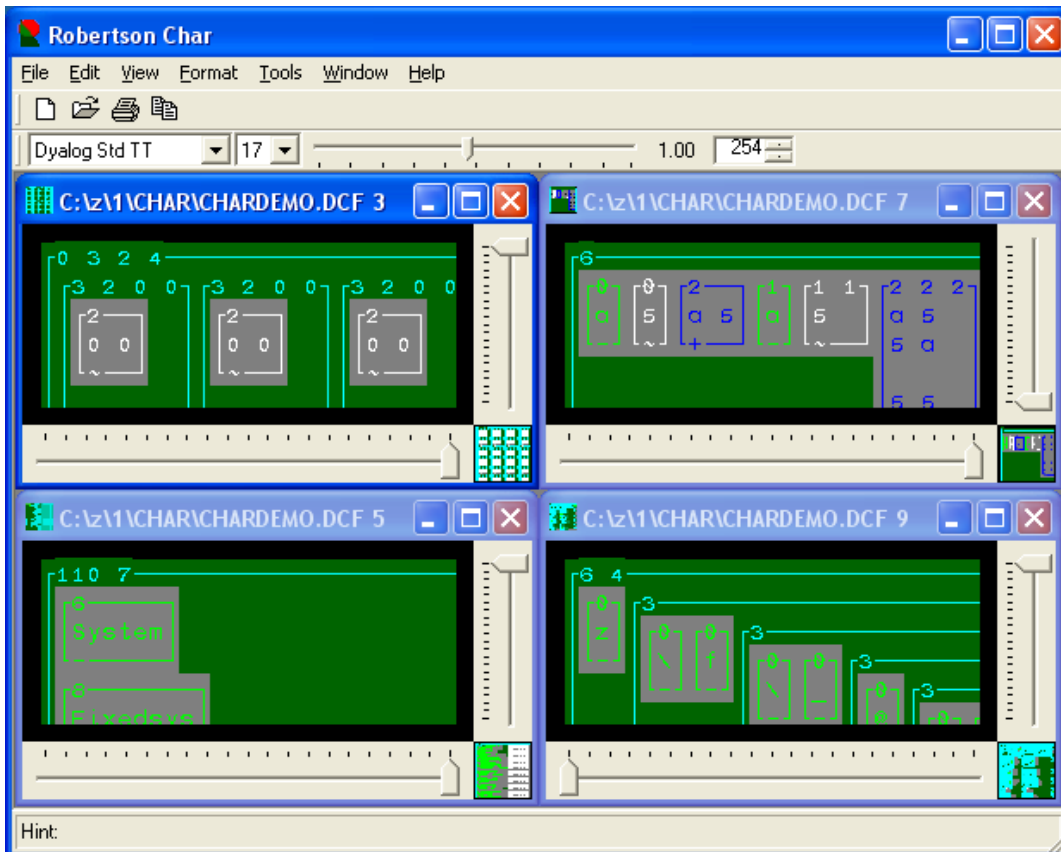

These SubForms may be tiled and arranged in the usual ways for an MDIClient using options on the [Windows] menu.

---

open

$ File Open MenuItem

k FileBox;.dcf;*.*;APL component file;thin array;too non-square;component;Metafile

+ item:06

Sometimes an array is too narrow to display effectively. In this case it may be best to reshape the array into a more presentable shape. Sometimes the array is simply too large for the underlying Metafile to hold. In either case a window pops up asking you if you would like to try again with a smaller font. Even after this adjustment it might still be too non-square to display and the same popup reappears and an even smaller font may be used. Often the results produced in these very small fonts are not handled well by the Metafile and the results are only useful in certain cases. Note also that subsequent use of varChar retains the small font (see problem 12 below). Restart Char to restore normality.
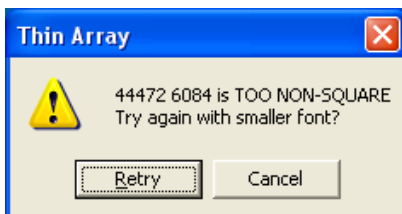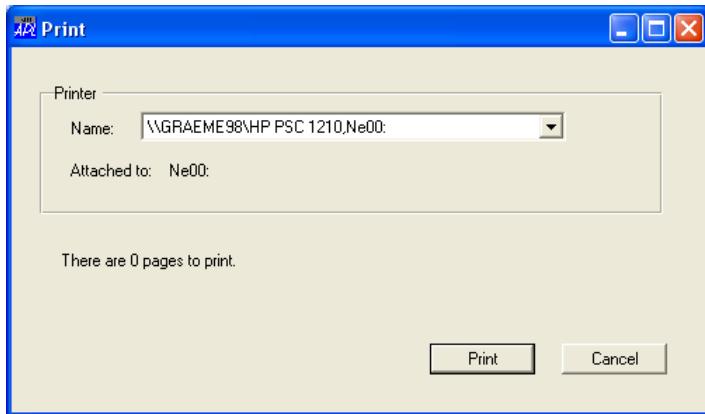

- <u>Menus and ToolBars</u>   Explanation of system features.

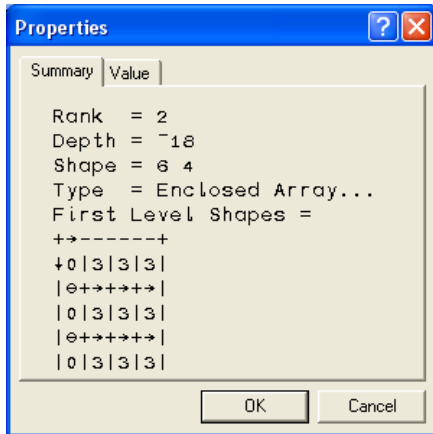# #$k+File Print   print selected parts of array

When pages have been selected in [View][Paper] mode they may be printed.

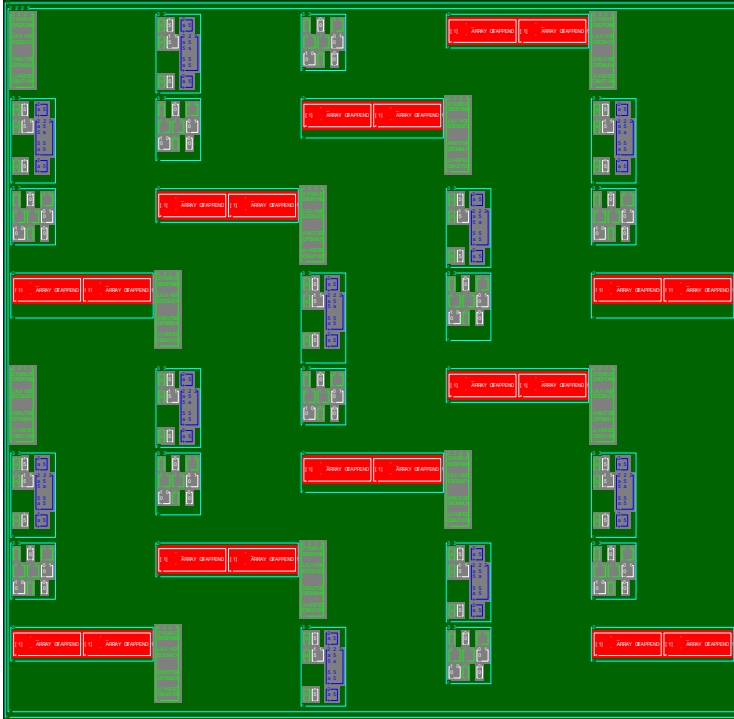- • Menus and ToolBars   Explanation of system features.

---

print
$ File Print MenuItem
k View Paper MenuItem;Pages to print;Printer name
+ item:07

The basic properties of the active SubForm array may be examined in some detail.

```
Properties                    [?][X]
 Summary | Value |
   Rank  = 2
   Depth = ¯18
   Shape = 6 4
   Type  = Enclosed Array...
   First Level Shapes =
   +→------+
   ↓0|3|3|3|
   |⊖→→→→→|
   |0|3|3|3|
   |⊖→→→→→|
   |0|3|3|3|
          [ OK ]  [ Cancel ]
```

- Menus and ToolBars   Explanation of system features.

---

props
$ File Properties MenuItem
k properties of array;rank;depth;shape;type
+ item:08

#$k+Edit Copy   copy representation to clipboard

The underlying Metafile may be placed on the Clipboard ready for pasting into another application, like Microsoft Word.

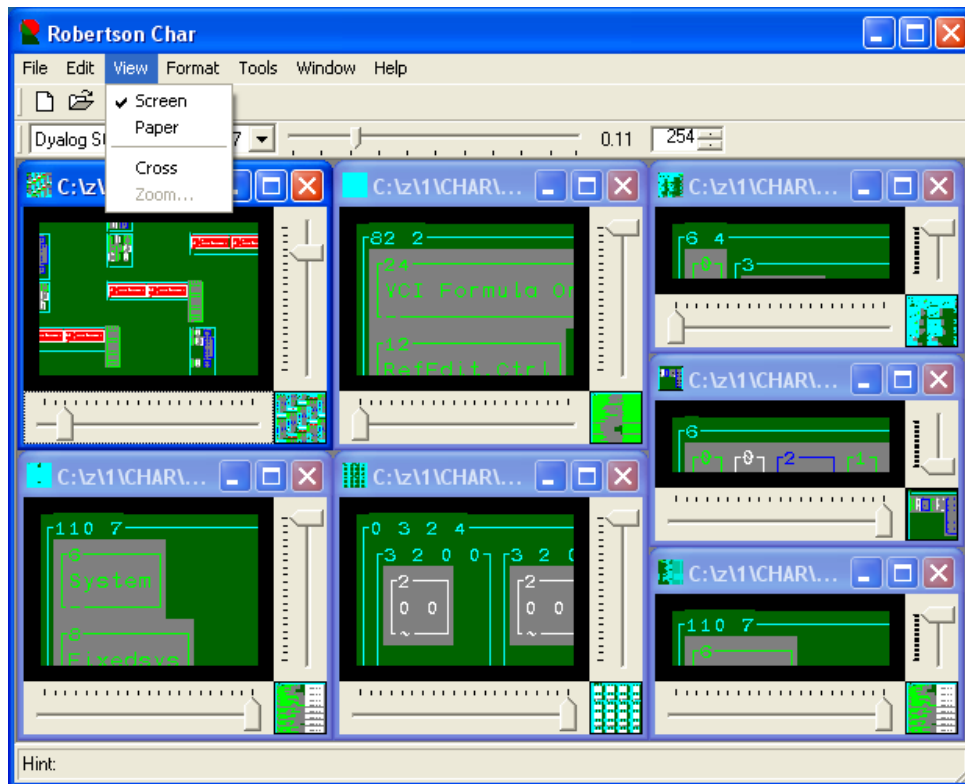- **Menus and ToolBars** Explanation of system features.

#_{$k+}View Screen   screen representation

The default viewing mode is [View][Screens]. In this mode the colour scheme used is that for a screen in [Format][Elements].

Clicking on any of the SubForm windows will activate it. When a window is active, holding down the left mouse button and dragging the mouse will cause the Metafile to be dragged around ("panned") within the window. The Metafile may also be moved using the scrolling TrackBars below and to the right of each window.

Alternatively, a Metafile may be magnified or shrunk to a point by means of the Zoom TrackBar control. Dragging the thumb of the TrackBar to the left will shrink the entire array so that the entire object may be viewed in the window. Dragging to the right will enlarge the array. The magnification factor, or zoom power, is given to the right of the TrackBar.

Another, rather experimental way of moving arrays is by means of a velocity Timer object in each window. Activate a window by clicking on it. Position your sprite on the Metafile, hold down the right mouse button and drag the sprite out of the Metafile. The Metafile should begin to move in the opposite direction under the control of a Timer. The motion will stop if the sprite is moved back into the Metafile or the Metafile hits an edge. In this way all the windows can simultaneously display moving arrays. This can be useful for methodical examination of extended arrays (but see problem 7).

- Menus and ToolBars   Explanation of system features.

[#] screen

[$] View Screen MenuItem

^k view mode;colour scheme;pan;drag;scroll;magnify;shrink;zoom;velocity timer;sprite;moving arrays;Metafile

⁺ item:10

If the view mode is set to paper, then arrays or parts of arrays may be selected and printed. The dashed grid sitting on top of the array (now showing in appropriate paper colours as set by [Format][Elements]) indicates page boundaries. A page is selected or deselected by a single mouse click on the corresponding page. Pages may be of portrait or landscape orientation as determined by [File][Setup].
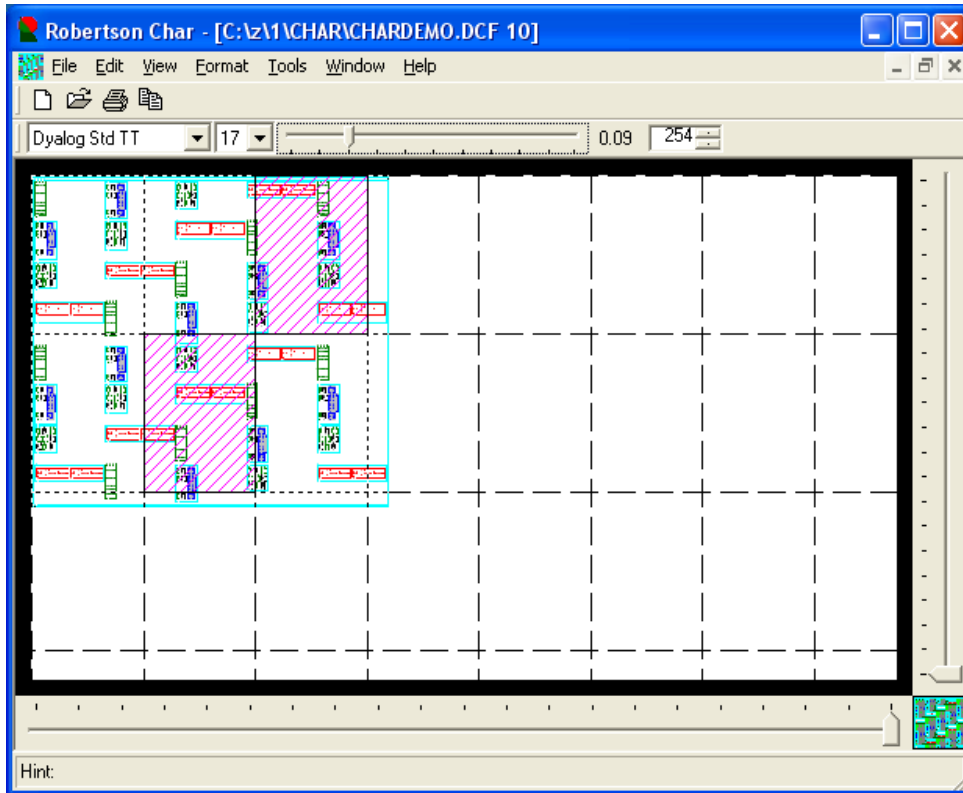
There are two entangled ways of changing the page contents. One is using the Zoom TrackBar to vary the size of the underlying Metafile. The other is by dragging the page boundaries to enlarge or shrink the amount contained on a page. Some experimentation is needed to find the best fit in any particular situation.

- <u>Menus and ToolBars</u>   Explanation of system features.

---

paper
$ View Paper MenuItem
k view mode;select page;print page;grid lines;paper;page boundaries;orientation;Zoom
TrackBar;Metafile;best fit
+ item:11

In the case of a large array it is sometimes difficult to focus in on the part of interest. In this case, Zooming can be used in conjunction with the centralized cross to quickly find the point of interest. First shrink the Metafile to less than a quarter size of the window and move to one size. Display the cross such that the central point is visible. Move the sprite such that the hotspot points precisely at the point of interest. Click the left mouse button and drag the Metafile over the cross so that the hotspot is precisely on the cross intersection. Drop the Metafile by releasing the mouse button. Now use the Zoom TrackBar to expand the point of interest to a suitable size for examination.
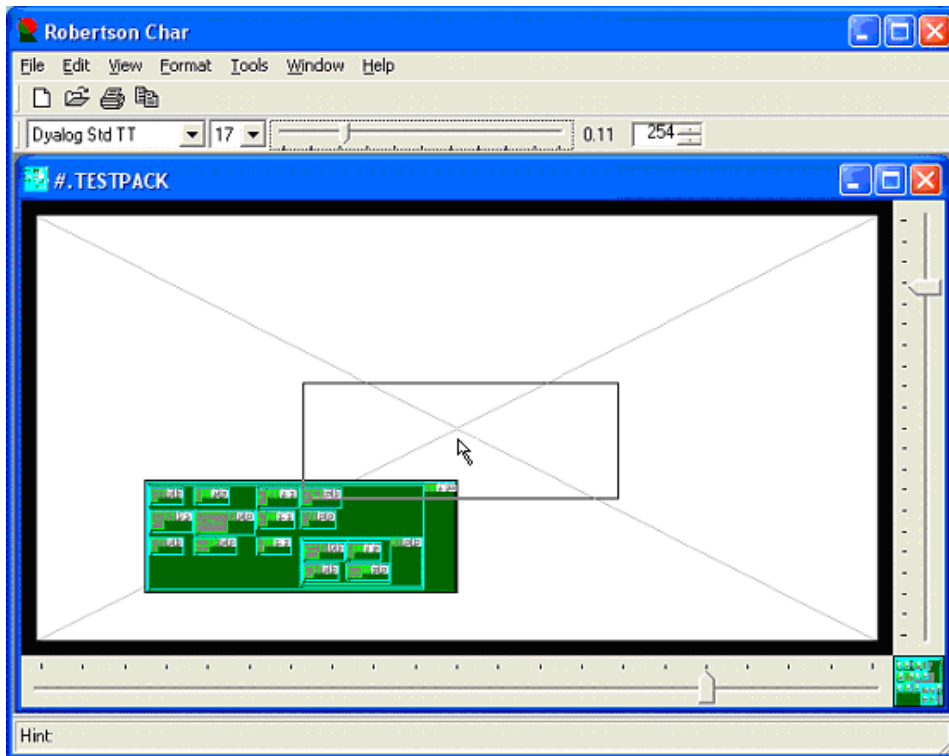
Another way of locating a given area of the picture, but which is not functional in this version of Char (.EXE), is by means of the square Static at the bottom right of the SubForm. Right-clicking and dragging in this Static opens a rectangular Locator which identifies a given area to be viewed.

- Menus and ToolBars   Explanation of system features.

---

cross
$ View Cross MenuItem
k large array;focus;Zoom TrackBar;cross;centralize;point of interest;quarter size;sprite;hotspot;drag;drop;mouse button;Static;Locator;area
+ item:12

#$k+**Format Font**	open font dialogue

This option is not functional in this version of Robertson varChar. Instead the font face name and font size should be set in the controls on the Format ToolBar. The font and size set in the ToolBar controls will affect the next and all subsequent arrays to be opened. The subset of fonts available is the set of fixed-width fonts which are available on both the screen and the current printer.
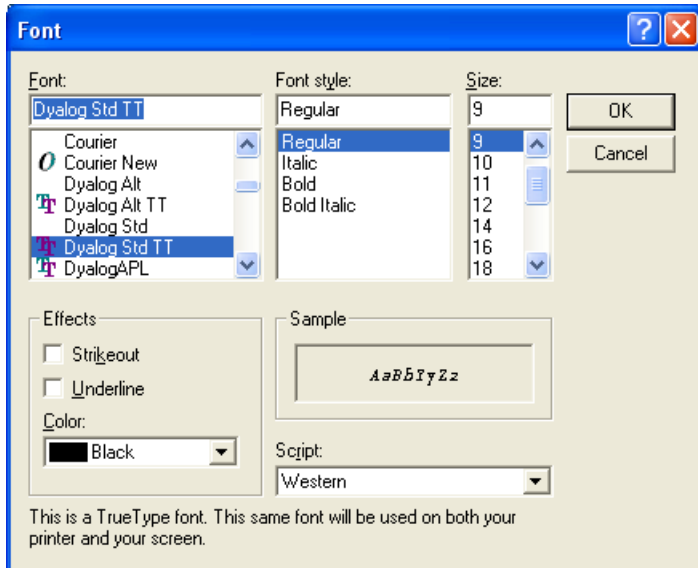
- Menus and ToolBars   Explanation of system features.

---

font
$ Format Font MenuItem
k font;face name;font size;Format ToolBar;fixed-width;printer fonts;screen fonts
+ item:13

Robertson varChar identifies seven or eight elements which determine the colour scheme and the symbols used in the bottom left corner of each array and subarray.

Character and numeric arrays will be most familiar. Mixed arrays are simple (depth 1) arrays composed exclusively of character scalars and numeric scalars. (Super-mixed arrays are simple arrays (depth 1) composed exclusively of character scalars , numeric scalars, $\Box NULL$'s and scalar refs.)

References are named or unnamed instances of namespaces. They will be coloured yellow on dark green on the screen. $\Box NULL$'s are scalars, coloured white on black and having a jot symbol in the bottom left corner. $\Box OR$'s are immiscible with other types of scalars and are coloured white on red.

Enclosures - arrays which have been enclosed - are marked with a cyan on dark green border – one line round the perimeter for each time it is enclosed.
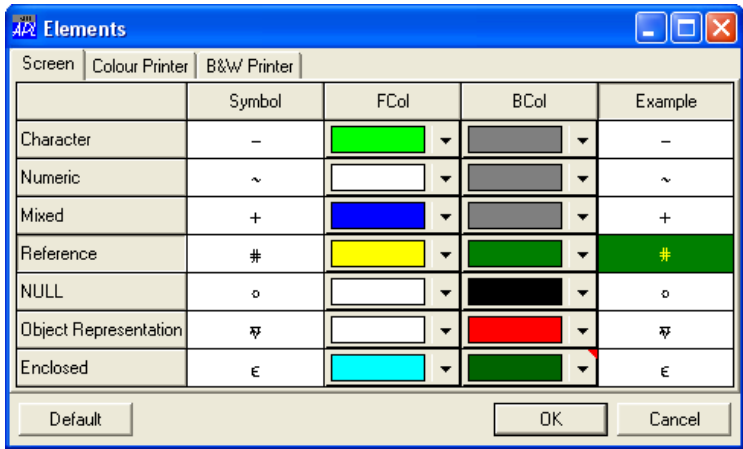
Not all printers have colours. A different default scheme is chosen for black & white printers.
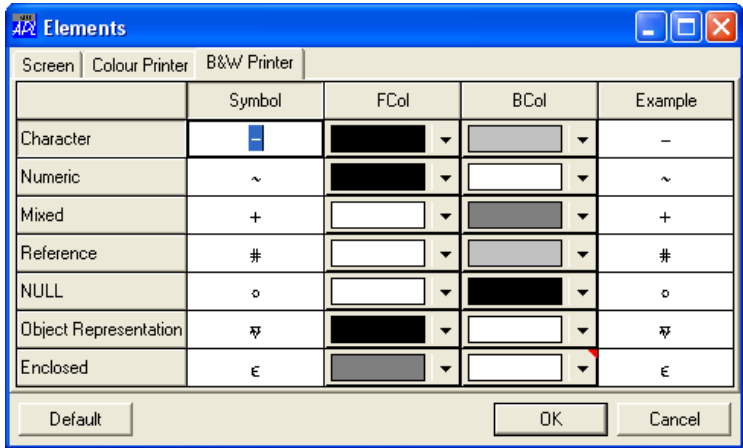

Note the comment indicator on the BCol of the enclosed element. This comment explains that the colour chosen as the background colour for the enclosed element is also the background colour used for the array as a whole.

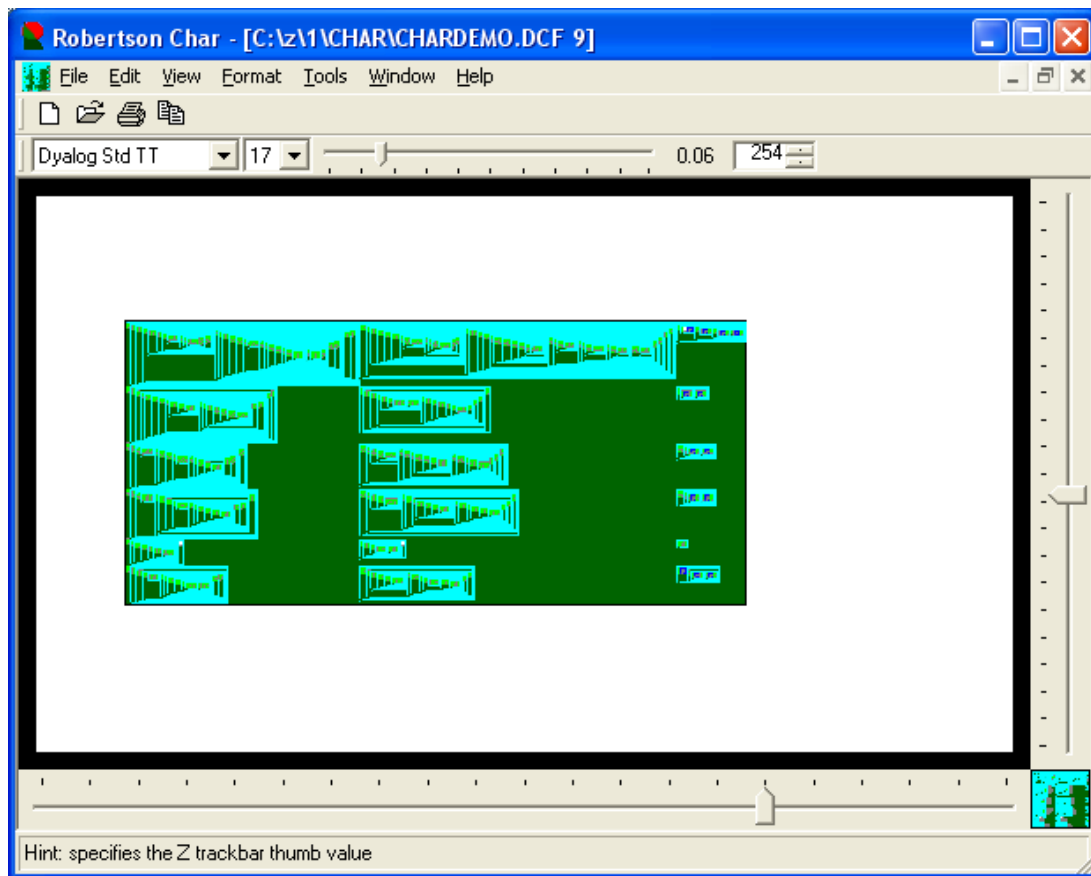- <u>Menus and ToolBars</u>   Explanation of system features.

---

elements

$ Format Elements MenuItem

k elements;colour scheme;symbols;mixed arrays;scalars;super-mixed arrays;depth;simple arrays;null item;object representation;colour;enclosed arrays;black & white printers;background colour

+ item:14

#$k+**Tools Pan**	toggle pan facility

The panning facility is permanently "on" in this Char .EXE version. Left click on the array and drag in order to move it around. Release the left mouse button to fix the new position.
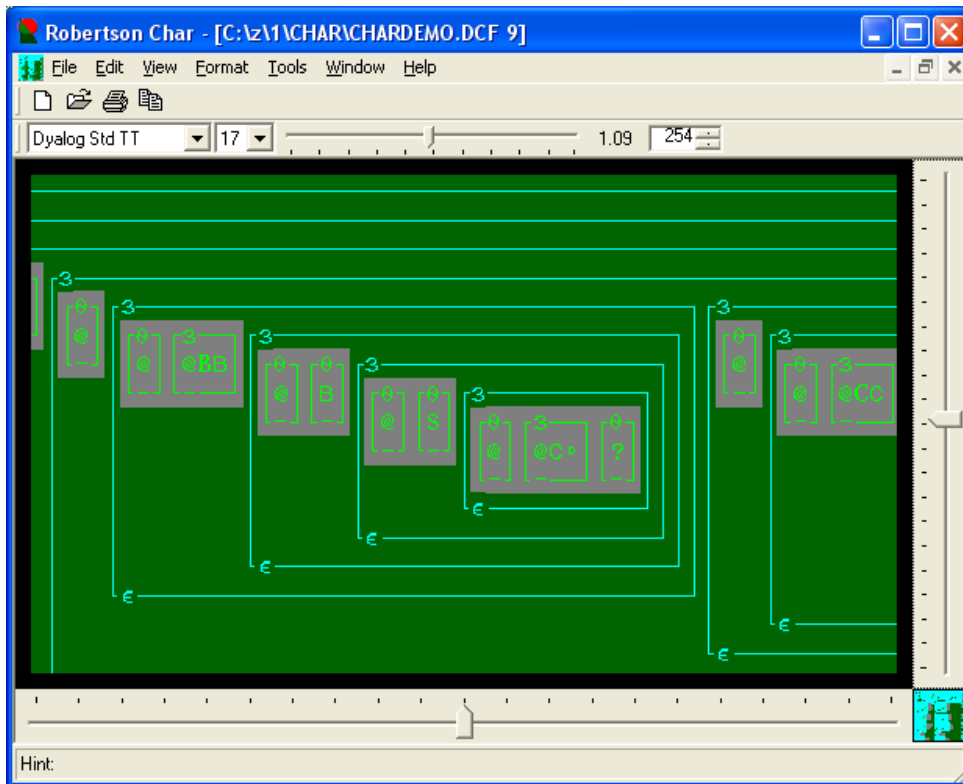
Use the Zoom TrackBar to adjust the overall size before or after panning around.

---

pan
$ Tools Pan MenuItem
k pan;Zoom TrackBar;velocity timer
+ item:15

Use the constant velocity Timer, as described in [View][Screens], as an alternative to panning.

- Menus and ToolBars   Explanation of system features.

Since it is so easy to produce lots of windows from a multi-component file, the Window menu is even more useful than usual. You can Cascade and Tile Vertically or Horizontally as usual. You can also Minimize All Windows, Restore All Windows or Close All Windows in just two keystrokes or two mouse clicks.
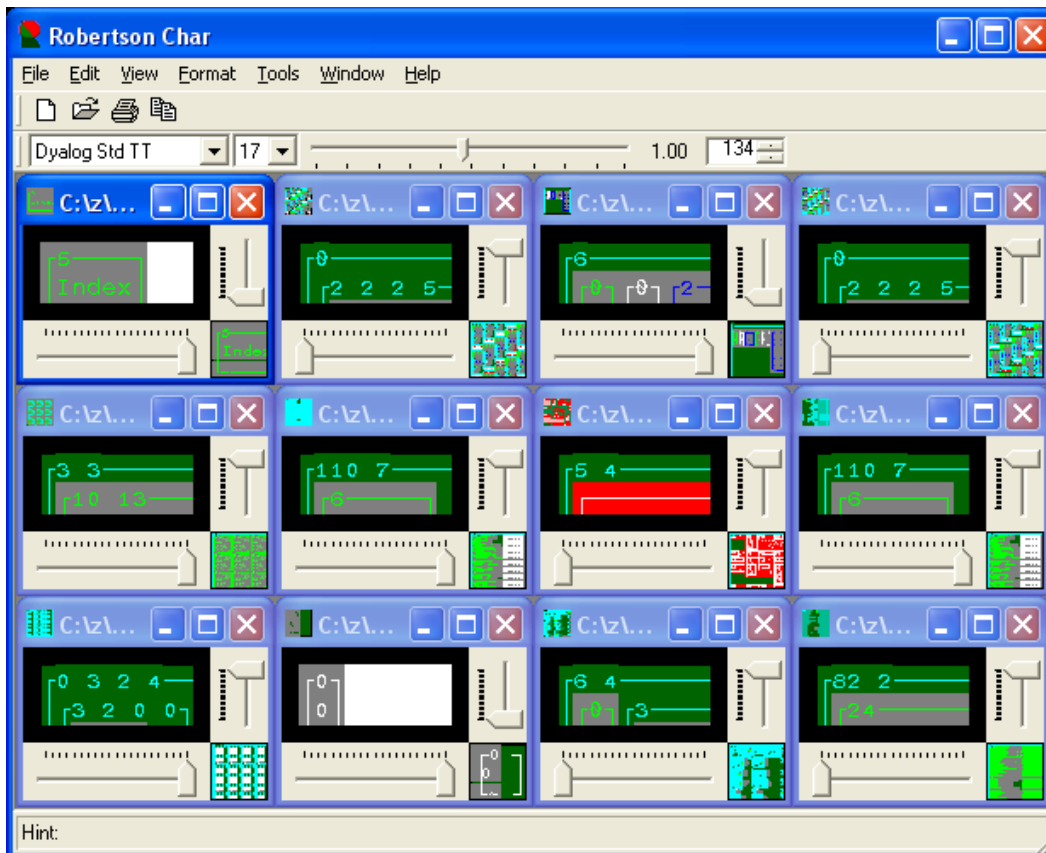
The experimental Opacity control on the Format ToolBar (not available under Windows 9x) allows you to see through Robertson varChar to the screens behind. This can be useful as varChar with lots of open windows can hog valuable screen space – although it can, of course, be minimized with a single click.

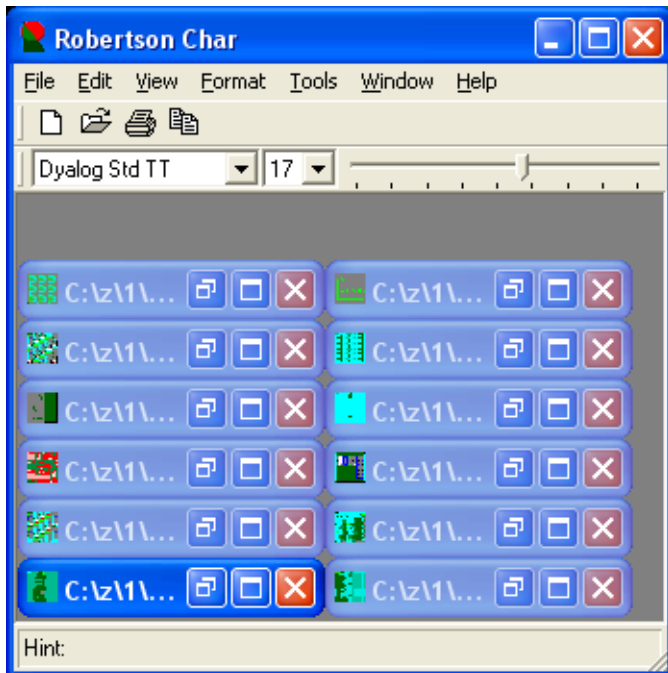- <u>Menus and ToolBars</u> Explanation of system features.

#$k+Window Minimize All Windows   minimize open windows

The result below can be achieved in a single click. And can be reversed in a single click too with [Restore All Windows].


Note the small *square impression* of the contained Metafile which is placed as an Icon on each minimized SubForm.

- <u>Menus and ToolBars</u>   Explanation of system features.

---

min

$ Window Minimize All Windows MenuItem

k minimize all windows;restore all windows;Icon;SubForm

+ item:17

#$k+Help Greeting	display system greeting

When varChar starts, the varChar greeting bitmap below is positioned on the screen while the system initializes.
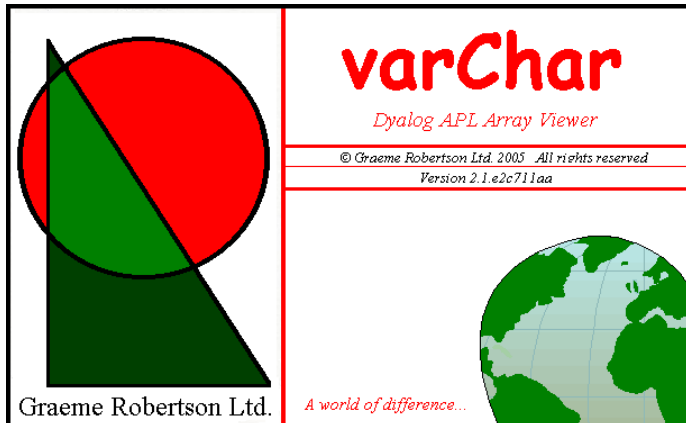
When this bitmap is displayed via [Help][Greeting] a 2D working APL model of Newtonian gravitation is superimposed demonstrating a semi-stable binary star with ten incoming potential planets. The model demonstrates the fact that planets in binary stellar systems, as opposed to single star systems such as our own, are far less likely to be in anything like a stable orbit when close to the centre of mass.

- Menus and ToolBars   Explanation of system features.

---

greet
$ Help Greeting
k GreetCha;GreetBitmap;Newtonian gravity;Dyalog APL array viewer;version;copyright
+ item:18

#$ᵏDemonstration File

Accompanying this system is a relatively small file called `charDemo.DCF`. It contains a number of more or less interesting examples of Dyalog APL arrays. In the first component is a character vector giving a quick explanation of the contents of each component. All 36 components can be opened and tiled in one fell swoop.
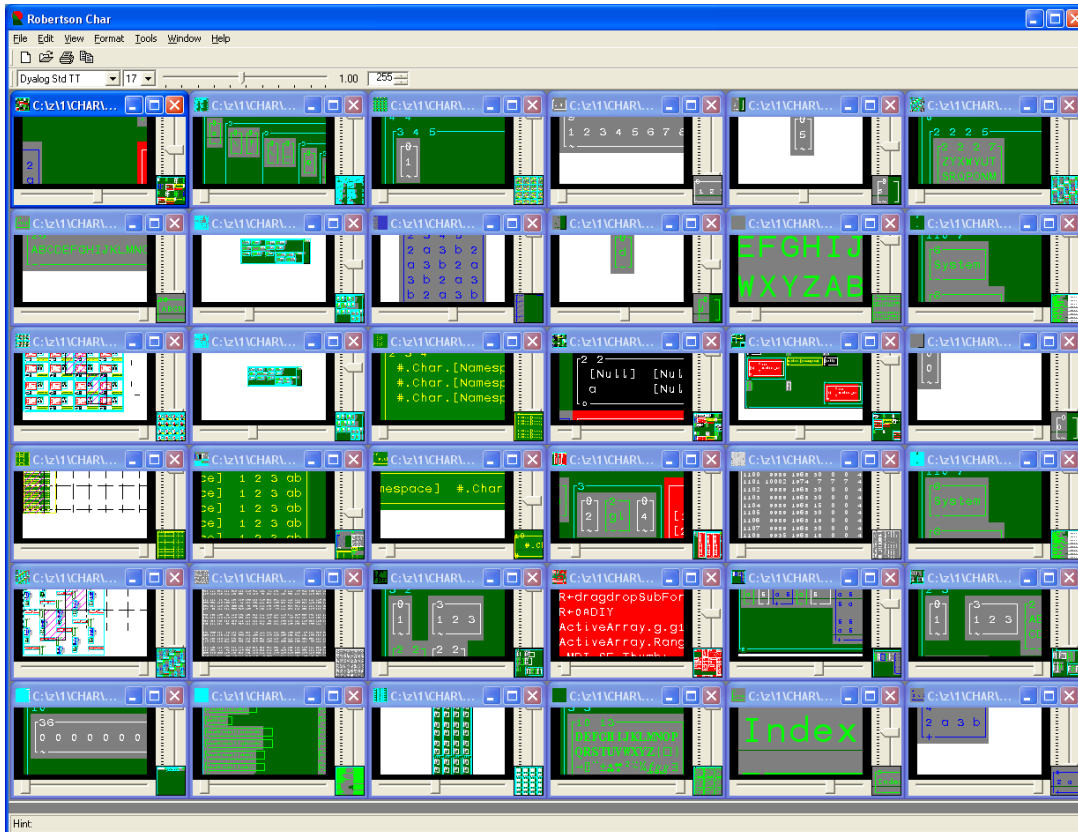
| Component	Description
1	This character vector INDEX (with embedded newline characters).
2	A rank 2, depth 2 character array.
3	A rank 4, depth 3 numeric array.
4	A rank 3, depth -4 array of characters, numbers, mixed arrays and $\square OR$ arrays.
5	A rank2, depth -2 array of font information. Try entering ⍏`#.FontList` in [File][New]. Note that ⍏ is entered using Ctrl+Y according to table winUK.din.
6	Zilde, ie `0⍴0`
7	An enclosed vector of character, numeric and mixed arrays.
8	A depth 2 matrix of $\square OR$'s of fns.
9	A depth -18 matrix of highly nested arrays of character scalars and vectors.
10	A depth -4 enclosed array of arrays of various types.
11	One million numbers:.to test, type `?20 50 1000⍴99` into [File] [New].
12	The result of ⍏`#.OLEControls`
13	A numeric matrix.
14	A depth -3 vector of numeric, character, $\square OR$s and ref arrays.
15	A rank 2, depth -3 enclosed array.
16	Component 15 reshaped.
17	Nested GUI property values.
18	A depth -6 nesting of $\square CR$s and other character and numeric elements.
19	A rank 2, depth -3 matrix of $\square NULL$, character, numeric, ref and enclosed $\square OR$ elements.
20	A matrix containing $\square NULL$ and character mixtures, character and numeric mixtures and unenclosed $\square OR$s.
21	Simple nesting of all types except $\square NULL$
22	An array of arrays of refs.
23	A depth -5 array of all types except mixed.
24	The multiply enclosed super-mixture `⊂⊂⊂⊂⊂⊂1'a'(⎕NS'')⎕NULL`
25	A rank 3 character array of `2 3 44⍴⎕A`
26	A rank 0, depth 0 scalar `'d'`. To test, type `'d'` into [File] [New].
27	$\square A$
28	`2 3 4 5⍴2'a'3'b'` To test type this into [File] [New].
29	`2'a'3'b'` To test type this into [File] [New].
30	A rank 4 array of random numbers between 1 and 1000? eg `?2 3 4 55 ⍴1000`
31	A numeric scalar 5. To test, type `5` into [File] [New].
32	`⍳9` To test type `⍳9` into [File] [New]. Note that `⍳999` is too non-square to display well, but, for example, `500 500⍴⍳999` is not, and `999 999⍴⍳999` is nearly OK.
33	A vector of unnamed namespaces.
34	A 3D array of unnamed namespaces.
35	A rank 3, depth 1 super-mixed array of reference, numeric and character scalars.
36	A nested structure of various types of arrays.

Some of the SubForms in the picture above have been magnified or shrunk using the Zoom TrackBar. Some have been switched to paper mode using [View][Paper].

Workspace Examples

There are also some simple examples saved in the Char workspace, in namespace $\#.eg$, which can be accessed by the dropdown Combo in [File][New]. This Combo holds all the expressions which have been entered by you during your current session with Robertson varChar. It also holds all the variables in all the namespaces in the currently loaded workspace in the session button version. In this stand-alone version, the current workspace is always the same and therefore the $\#.eg$ namespace variables are always available for examination. There is also a variable called $TESTPACK$ in the Root.
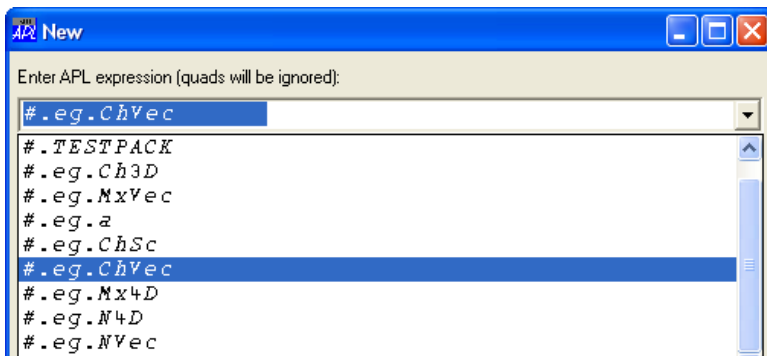
• Contents Page	Return to front page of help file.
• Installation	How to install the product.
• Main Window	Initial appearance of the main window.
• Menus and ToolBars	Explanation of system features.
• Known Problems	Top Twenty Known Problems
• EULA	End User Licence Agreement

#$k Some Known Problems with Version 2.1.

1	[Format] [Elements] is not operational in this version.
2	[Format][Scale to Fit]/[Centre] is inactive - use Zoom and Cross
3	Dragging page borders in [View] [Paper] sometimes drops the borders in the wrong place. It may behave better if you drag the finer dotted line on the picture itself.
4	Green horizontal lines appear in some Metafiles at some magnification scales. And test and background colours scale at slightly different rates when you zoom in/out Metafile due to a problem in the Metafile object.
5	Printing width (click to select page) is slightly different from WYSIWYG screen width.
6	[View][Zoom] is inactive - use the Zoom Trackbar.
7	Auto-move of Metafile (by right clicking and dragging off Metafile) does not get direction exactly correct and does not always stop at the edge.
8	[Tools][Locate]/[Pan] are inactive, as is (left click) Locator on bottom right Static.
9	There should be an egg-timer Cursor when opening new arrays.
10	[File] [New] can't cope with function expressions like $+ \circ +$
11	If you cancel a thin array which is "too non-square" a blue empty SubForm is left behind.
12	If you successfully create a thin array which was "too non-square", the new Font sticks and remains inappropriately small thereafter for the reset of the varChar session.
13	Pages selected for printing before being resized with the Zoom TrackBar are not deselected automatically.
14	The interpreter can crash when creating an instance of the first item in PrintList if there is no printer attached.
15	Can lose top and bottom of contents of MDI SubForms in a large MDI tiling.
16	Symbols for Unique and Matrix Inverse don't show up.
17	[File] [Open] can't cope with scalar GUI objects nor with scalar object representations of scalar GUI objects.

- Contents Page   Return to front page of help file.
- Installation   How to install the product.
- Main Window   Initial appearance of the main window.
- Menus and ToolBars   Explanation of system features.
- Demonstration File   Examples brought together in a demonstration component file.

---

prob

$ Known Problems

k problems;elements;page boundaries;drawing problems;print width;

+ top:20

#$ᵏEnd User Licence Agreement

PLEASE READ THE TERMS OF THIS SOFTWARE LICENCE AGREEMENT

1. *General.* The varChar software ("the software") is licensed not sold to you by Graeme Robertson Limited ("the licensor"). This licence is given free of charge. Further information regarding the licensing of dyalog10rt.dll may be sought from Dyalog Ltd, the suppliers of dyalog10rt.dll.

2. *Permitted Licence Uses and Restrictions.* This non-exclusive and transferable licence allows you to install and use the software on a computer. You may make the software available over a network where it might be used by multiple computers at the same time. You may copy the software in machine-readable form. You may distribute the software. You may decompile, reverse engineer, disassemble, or otherwise reduce the software to a human perceivable APL form. You may modify, adapt, translate, distribute, or create derivative works based upon the APL software.

3. *Transfer.* The software and licence is freely transferable. Char.exe is free of charge. There might be restrictions in the transferability of dyalog10rt.dll which can be clarified by the author of this software, Dyalog Ltd.

4. *Disclaimer of Warranties.* You expressly acknowledge and agree that use of the software is at your sole risk and that the entire risk as to satisfactory quality, performance, accuracy and effort is with you. The software is provided "as is", with all faults and without warranty of any kind. The licensor hereby disclaims all warranties and conditions with respect to the software, either expressed, implied or statutory, including, but not limited to, the implied warranties and/or conditions of merchantability, of satisfactory quality, of fitness for a particular purpose, of accuracy, of uninterrupted or error-free performance, or that defects in the software will be corrected.

5. *Limitation of Liability.* To the extent not prohibited by law, in no event shall the licensor be liable for personal injury, or any incidental, special, indirect or consequential damages whatsoever, including, without limitation, damages for loss of profits, loss of data, business interruption or any other commercial damages or losses, arising out of or related to your use or inability to use the software, however caused, regardless of the theory of liability (contract, tort or otherwise). In no event shall the licensor's total liability to you for all damages exceed the amount of one pound sterling (£1.00).

6. *Documentation.* You may make and use an unlimited number of copies of any documentation such as this help file.

7. *Trial or Evaluation or Educational Use.* A licence is granted for all educational uses and there is no licence fee to pay.

8. *Governing Law and Jurisdiction.* This licence will be governed by and construed in accordance with the laws of England and Wales. The licensor and you submit unconditionally and irrevocably to the non-exclusive jurisdiction of the courts of England and Wales.