Module10: Stand-Alone Applications

You have written your Dyalog APL application and all you want to do now is dish it out. *"HOW DO YOU DO THAT?"*

§ 10.1 Building GUI Applications

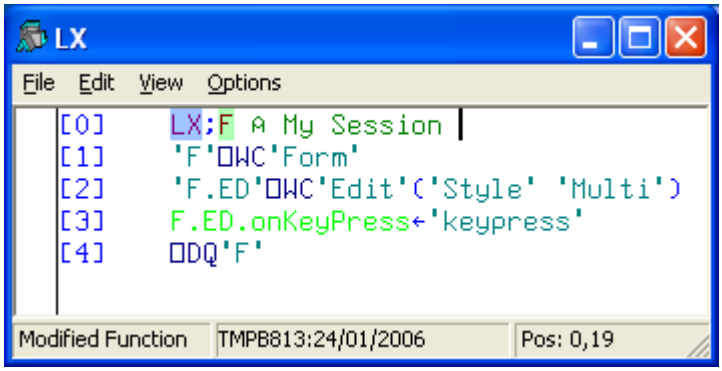
§§ 10.1.1 The bare Minimum

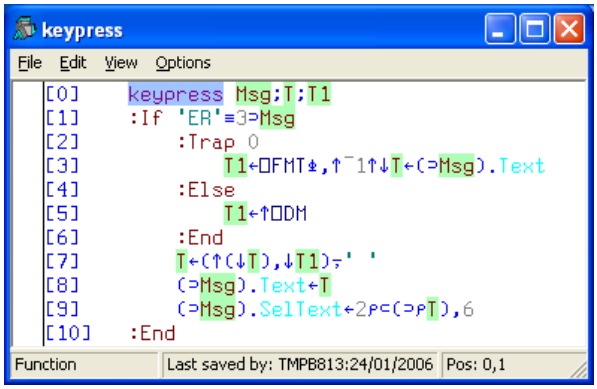
Your application code may be in a workspace and the interpreter may be in the runtime executable, Dyalogrt.exe. For this classic scenario see the document "Run-Time Applications in Dyalog APL/W Versions 7 and 8" which is downloadable by Dyalog Users open mail group members from the Dyalog Users section of Yahoo. This document also applies, essentially unchanged, to version 9.

From version 10 onwards, the smallest possible packaged Dyalog APL application requires just two files, an executable (EXE) file containing your code and the Dyalog APL dynamic link library, Dyalog10rt.DLL, to which it is bound. Together, these two files can constitute a complete application whose only reference to APL is the name of the Dyalog DLL. The application may use its own registry section and an application icon may be incorporated into the executable. This is explained in §10.2.

The program should trap all errors, but just in case the program should exit before `⎕OFF`, the configuration parameters RunTimeTitle and RunTimeError should be given appropriate entries in your registry file. This registry file will be described in §§10.2.2.

In this section we shall prepare an application for export. The runtime application agreement with Dyalog Limited clearly precludes the possibility of writing a Session-replacement cover for the Dyalog APL language kernel such as `⎕LX ↦ F ↪keypress` that executes (`⍎`) raw APL code below.

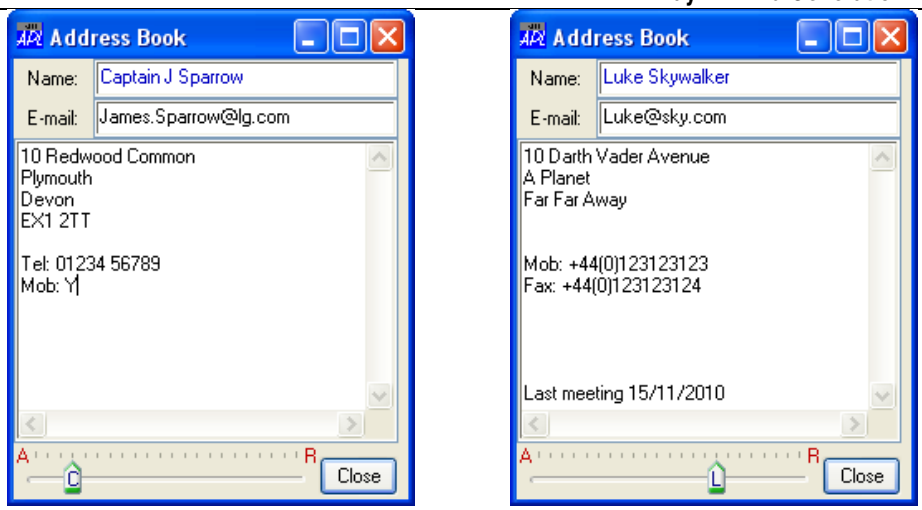




So we need an ***¡IDEA!*** for an application of our own. To assist in the quest we provide a small starter application that you might adopt and adapt to your liking.

10.1.1.1 Load workspace AddrBk.DWS and try using the address book. Add a new entry by dragging the `TrackBar` to the far right and entering a name. Drag the `TrackBar` back to the appropriate letter to view your new entry. Edit the entry at will.

Two of your friends and associates might be:



§§ 10.1.2 Completing the Address Book Application

You might have noticed that your entries are lost as soon as you exit the workspace ☹.

10.1.2.1 In order to make this little application useful, replace the functions `∇addrbk.GetMyStore∇` and `∇addrbk.PutMyStore∇` with APL component file read and write/create storage functions.

§§ 10.1.3 Enhancing your Address Book

There are many ways that you might wish to personalise and improve the Address Book application:

- Use a different extension (eg .AB) rather than .DCF for your component file name.
- Include `↪FileBox` in [File][Open] to select a different address book.
- Add `⎕TRAP↪ErrorLog` to deal with errors. (Start `ErrorLog[1]` with `⎕TRAP←0 'S'`.)
- And/or email errors home (see §§7.3.1) with information such as..
  `⎕AN ⎕AI ⎕DM ⎕SI ⎕XSI ⎕TS ⎕FNAMES ⎕WA ⎕TID` .. , but not `⎕WSID`!
- Create a Help file and add a [Help][Help] menu item.
- Add [Help][About] to identify yourself and your product.
- Change the system `Icon` defined in `∇addrbk.Icons32∇` to your own design.
  Tip: consider using supplied workspace BMED.DWS.
- Make the left and right arrow keys move the `TrackBar Thumb` left and right. Etc…

§ 10.2: Making runtime Executables

You can build free stand-alone applications with Dyalog APL up to version 10. (In version 11 the run-time interpreter is not free.) An important practical difference between version 10 and previous versions is that you can completely avoid difficulties involved in specifying the command line required to initialise a version 9 and prior run-time system.

§§ 10.2.1 Files to Include

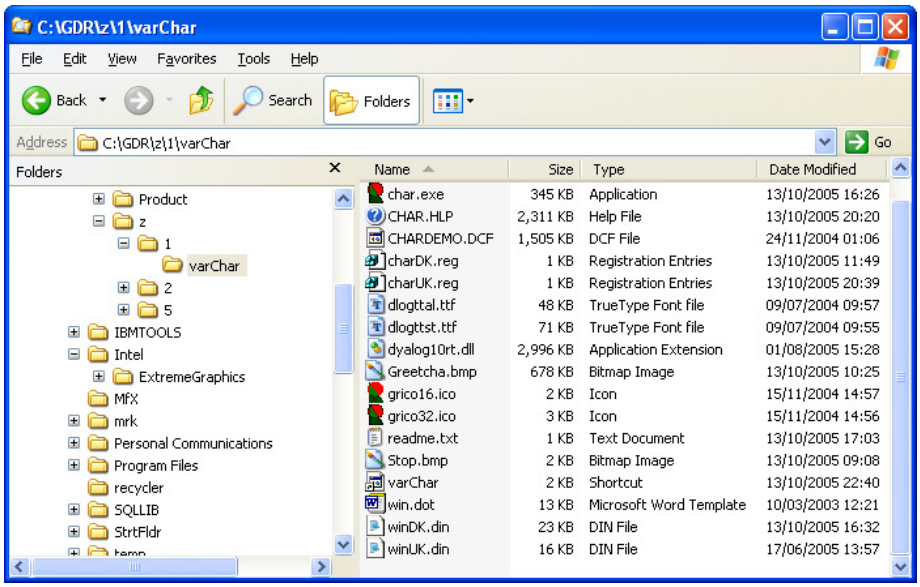
The only files that you have to include with a minimal runtime application in versions 10 and 11 are the application executable and the bound Dyalog runtime DLL. There is no need to include input (.DIN), output (.DOT) or APL font files, unless the application is explicitly going to use APL characters.

It is, however, often mandatory to include a registry (.REG) file, described in the next section.

Various other files are often necessary:

- APL component files or native files for data storage,
- the application help file,
- application-specific bitmaps and icons.

An example version 10.0 run-time application called **varChar** is included in this APL3&4 course. The files involved in this application are shown below. **Char.exe** with **Dyalog10rt.dll** is the core of the system. There are registry files and corresponding keyboard files for Danish and UK users. There is a demo DCF file and a help file. There are odd font, bitmap and icon files.



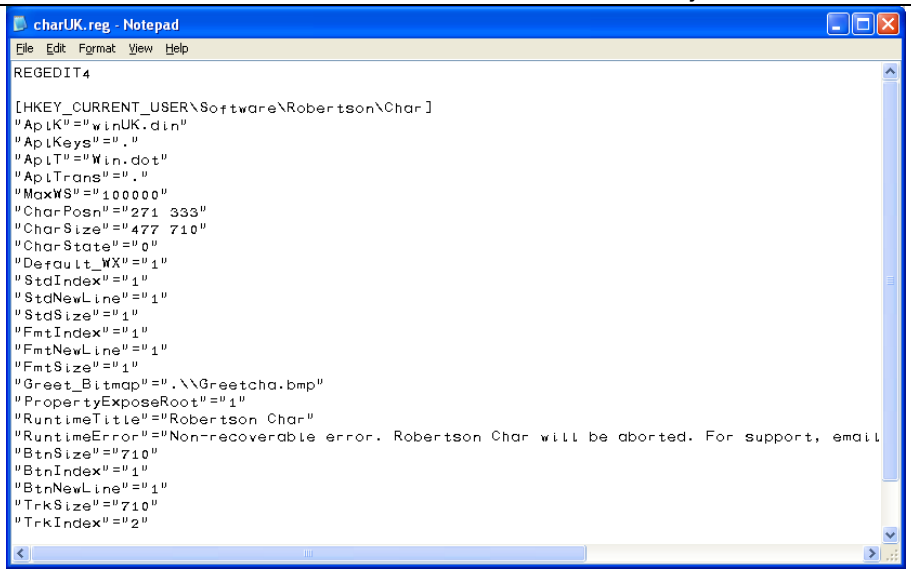
Tip: If your keyboard is other than Danish or UK then you should copy and rename your keyboard (.DIN) file to winUK.din and overwrite the one in **varChar**.

§§ 10.2.2 Inifiles and the Windows Registry

The Inifile parameter refers to a registry subkey in the HKEY_CURRENT_USER section of the Windows Registry. The default subkey in version 10.0 is \Software\Dyadic\Dyalog APL/W 10.0 but you can choose your own names to put under the \Software\ section.

REG files, which are simple text files and can therefore be edited in Notepad, should begin with `REGEDIT4` (*or* `Windows Registry Editor Version 5.00`) to identify the file as containing registry information. The next line identifies the section into which the following information should be added. If this section does not exist then it will be created. The name of the section typically contains the company name and the application name, *eg*
`[HKEY_CURRENT_USER\Software\MyCompany\MyApplication]`

10.2.2.1 Use REGEDIT.EXE [File][Export] to export a small section of your registry. View the resulting file in Notepad. Note that names and values of string entries (of type REG_SZ) are surrounded by double-quotes. Note also that backslash characters (\) have to be doubled up (\\) inside quotes.



To install information from a REG file into the registry, you simply need to double-click on the file in Windows Explorer. Alternatively you can run command

```
regedit c:\myapp\myapp.reg
```

in the command prompt. Or you can run the command

```
regedit /s c:\myapp\myapp.reg
```

to silently install the entry (without the popup message boxes). (The /d switch can delete a key.)

This can be done in APL under program control using `⎕CMD` (or WinExec), first making a call to `#.GetEnvironment'Inifile'` to check whether some particular entry already exists. This can make the whole process of establishing a registry entry completely transparent to the user.

One of the most significant entries in the registry file used to be the MAXWS parameter. This determines the maximum size of the workspace and often had to be set large for big applications. However, the DLL now dynamically allocates more memory as required and so this parameter is less important. (Also note that `⎕WSID` is empty in a .EXE file.)

One simple but nevertheless valuable parameter is greet_bitmap. If this parameter is set to the name of a bitmap file (*eg* "greet_bitmap"="c:\\myapp\\mybmp.bmp") then that bitmap will be displayed while the workspace is loading, and until the statement `#.GreetBitmap 0` is executed in your `⎕LX` after everything has been initialized and your application is ready for user activity. Sometimes this is too soon!

10.2.2.2 Make a 200 by 400 pixel bitmap (using the `FileWrite` method of a `Bitmap` object) and make a 32 by 32 pixel icon (using the `FileWrite` method of an `Icon` object.)

10.2.2.3 Create a personalised registry file with the greet_bitmap parameter set to your product bitmap file.

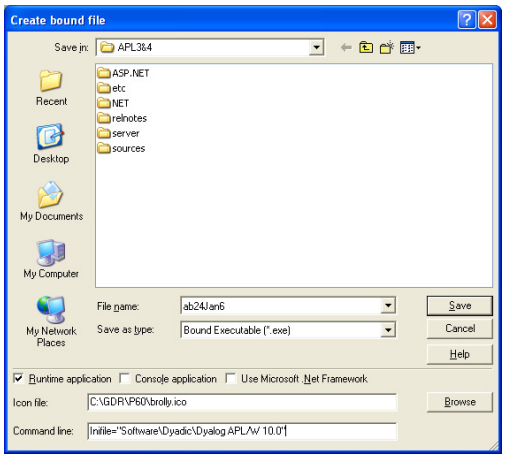
10.2.2.4 Add `⎕CMD'RegEdit /s ...'` to the Address Book `⎕LX`, to be run if and only if

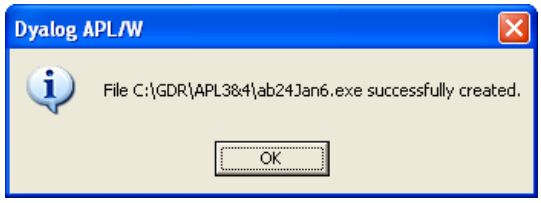
```
⍎'C:\..\MyBitmap.bmp'≠##.GetEnvironment'greet_bitmap'
```

10.2.2.5 Add the statement `#.GreetBitmap 0` to the Address Book `⎕LX` at a suitable point.

§§ 10.2.3 The [File][Export] MenuItem

10.2.3.1 XLoad your Address Book workspace in Dyalog version 10. Run [File][Export…]. Uncheck [Use Microsoft .Net Framework]. Enter your icon file name under *Icon file* and Inifile under *Command line*.





You now have an executable program bound to a dynamic link library. If the DLL is placed in the \Windows\System32\ directory then it should always be visible to the executable, otherwise it should be copied and moved around with the EXE file (that was your workspace).

10.2.3.2 Test your product and sell it to your friends.

§ 10.3 Aspects of Pocket APL

§§ 10.3.1 Pocket Platforms

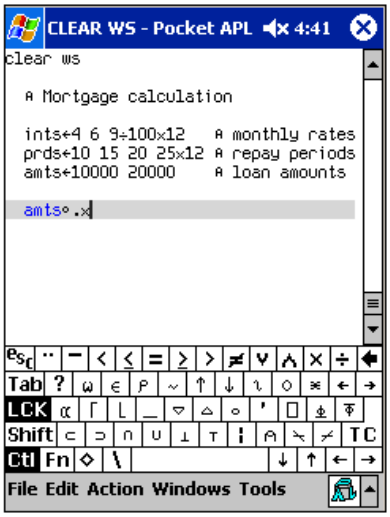
Dyalog Pocket APL runs under Microsoft Pocket PC 2002 and 2003 operating systems which run on a number of Personal Digital Assistants (PDAs) including Toshiba e740 and HP Jordana 560 Series.

←Toshiba e740

←HP Jornada 560 Series

For a list of suitable devices, see http://www.dyalog.com/ [Products][Pocket Dyalog]. The Dyalog web site and *Object Reference* are the primary sources of documentation for Pocket APL.

Often it is most practical to do most of the programming of a new pocket PC application on a big PC and make the executable on the PDA in for the final stages of development.



You can buy an add-on keyboard for your PDA to facilitate direct programming although a special APL keyboard input panel is provided to enable entry of APL expressions and user-defined functions. The APL keyboard (shown here with **Ctrl** pressed down) can be displayed or hidden. The presence of the keyboard is controlled by a tab on the screen or by the new *ShowSIP* method of a *Form* and Root (*#*). eg

```
#.ShowSIP 1
```

ShowSIP Bool	⍝ Displays or hides the Input Panel

Three new properties of a *Form* have been introduced for Pocket APL to facilitate some of the special characteristics of the device. They have no effect on other platforms.

SIPMode	⍝ Boolean determining automatic Input Panel display

If ⊢*SIPMode* then the Input Panel is displayed when a character input GUI object receives the focus.

SIPResize	⍝ Whether mode change generates a Configure event

If ⊢*SIPResize* then a *Configure* event is generated when the state of the Input Panel changes. The default behaviour of this event is to resize the *Form* to fit the available space.

OKButton	⍝ Whether an OK button appears in the title bar

If ⊢*OKButton* then an OK button like this - 

- replaces the X button at the top right of the *Form*. Clicking this button has the effect of pressing the *Button* (on the *Form*) which has ⊢*Default*. If no *Button* has ⊢*Default* then a *Close* event is generated. The *OKButton* property may only be set at ⎕*WC* time.

Pocket APL 10.0.4 is almost identical to Dyalog APL 10.0.4 for the PC, but without the DDE interface and *Form* docking facilities, and without the *Animation*, *BrowseBox*, *ComboEx*, *MDIClient*, *Metafile* and *SysTrayItem* objects. (Note that in all versions of Dyalog, the runtime executable EXE is almost identical to the development version EXE.)

A number of supporting workspaces are freely available from www.dyalog.com [Download Zone][Pocket Dyalog], including an APL tutorial, sample functions and useful applications.

§§ 10.3.2 Creating the executable Program

The Address Book GUI was built using the WDESIGN workspace. You might like to modify the Address Book GUI using WDESIGN. To do this you should follow the instructions in the *User Guide* where a brief description of the distributed WDESIGN workspace is given.

10.3.2.1 Remove the comment symbols on lines [1] and [3] of `▽addrbk.RUN▽` and save the workspace.

10.3.2.2 XLoad the workspace in **Pocket APL** on your **iPAQ** and select [File][Make Executable] to create an executable file. Tick the [Runtime Executable] checkbox and untick the [Evaluation Executable] checkbox to create a basic product.

§§ 10.3.3 Building a distributable Application

A pocket APL runtime application requires the Pocket APL Runtime Engine, Dyalog10.DLL, which should be installed in the PDA Windows directory. This Pocket APL Runtime Engine must be purchased by the user of your application from www.handango.com for around **$5**.

Instructions as to how you can package and distribute your application yourself through www.handango.com is give in www.dyalog.com [Products][Pocket Dyalog].

Apart from the APL runtime DLL all that you need to distribute is the system executable (plus any other files required by your) application, such as APL component files.

> Note if you are an individual holding personal information only for domestic reasons (eg an address book or Christmas card list) then you are not required to comply with the UK Data Protection Act 1998. Phew! ☺.

10.3.3.1 Consider joining the pocketapl@dyalog.com mailbox group. That is the end of Day 1. Please come back for more tomorrow. ☻