



▶ **The Development Environment**

▶ **Editor**

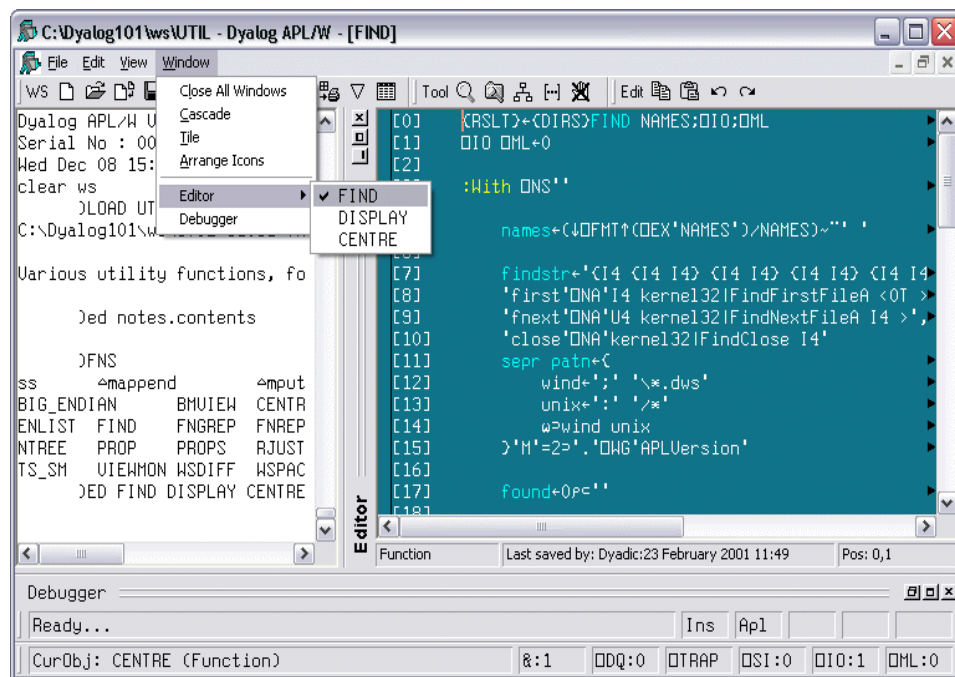
### The Editor

The Editor is an MDI window that may contain any number of sub-windows for objects (functions and variables) being edited or viewed. The Editor itself may appear as a completely separate window, or be docked in the Session window. It is also possible to drag individual Edit windows out of the MDI Editor window onto the desktop as independent windows.

The Editor provides syntax colouring (see below) to allow you to quickly identify syntax elements in your code.

The picture below illustrates what happens when you open one or more Edit windows. Notice that the Editor itself automatically springs up at and adopts the size it was before it was minimized.

Also, the Session menubar is replaced by the Editor's own menubar, and then restored when the Session window gets back the focus.



You may arrange the individual Edit windows using standard MDI facilities. For example, the following picture shows them tiled horizontally (*Window/Shift+T*).

**Dyalog Ltd**  
 South Barn  
 Minchens Court  
 Minchens Lane  
 Bramley  
 Hampshire  
 RG26 5BH  
 United Kingdom

**Phone:**  
 + 44 (0) 1256 830 030

**Fax:**  
 + 44 (0) 1256 830 031

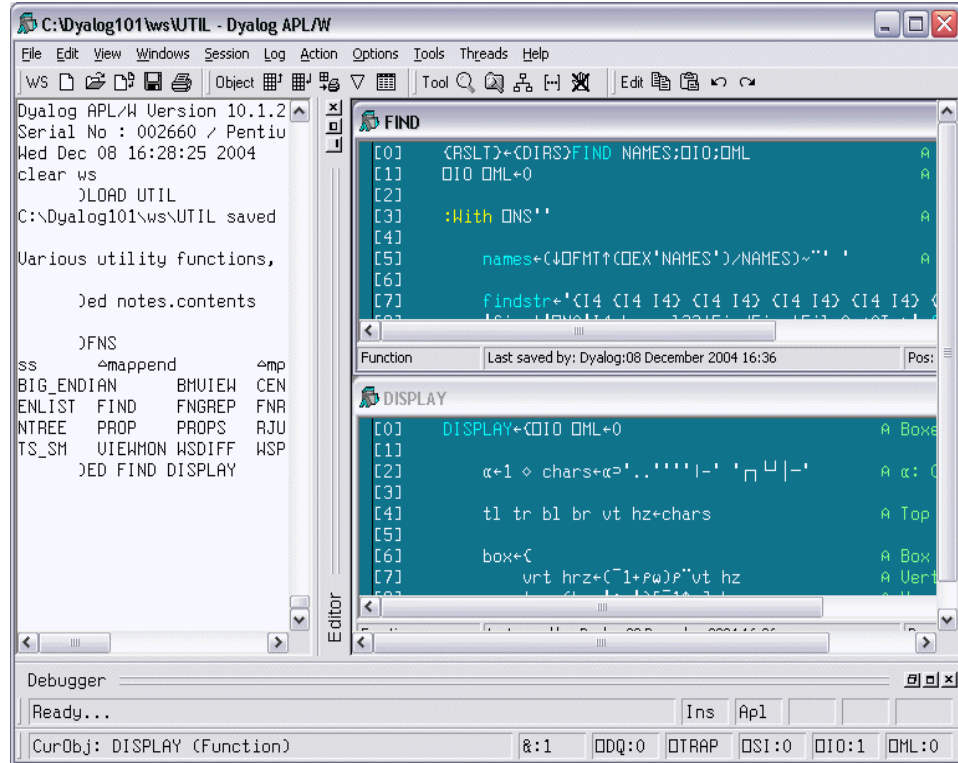
**e-mail:**  
 sales@dyalog.com



▶ **The Development Environment**

▶ **Editor**

### The Editor



When docked, the Editor has a horizontal or vertical title bar which contains its caption (displayed in a bold font if the Editor has the focus), and 3 buttons that are used to minimize, maximize, and close the window.

For example, if you want to concentrate on editing, you can quickly maximise the Editor as shown below.

If, while editing, you need to use the Session window, you can minimize the Editor to give you maximum Session space, or restore the Editor to have both visible.

**Dyalog Ltd**  
 South Barn  
 Minchens Court  
 Minchens Lane  
 Bramley  
 Hampshire  
 RG26 5BH  
 United Kingdom

**Phone:**  
 + 44 (0) 1256 830 030

**Fax:**  
 + 44 (0) 1256 830 031

**e-mail:**  
 sales@dyalog.com



▶ The Development Environment

▶ Editor

### The Editor

```

C:\Dyalog101\ws\UTIL - Dyalog APL/WS - [FIND]
File Edit View Windows Session Log Action Options Tools Threads Help
WS [Icons] Object [Grids] [Tools] [Edit] [Undo] [Redo]
[0] <RSLT>+<DIRS>FIND NAMES;DIO;DHL      A NAMES in workspaces in DIRS.
[1]   DIO DHL+0                          A <DIRS> defaults to HSPATH.
[2]
[3]   :With DNS''                          A Temp space avoids DCY clash.
[4]
[5]     names+⊖⊖DFMT↑(COEX'NAMES')/NAMES)~''  A Copy names to avoid DCY clash
[6]
[7]   findstr+⊖'C14 {I4 I4} {I4 I4} {I4 I4} {I4 I4} T[260] T[141]'
[8]   'first'DNA'I4 kernel32IFindFirstFileA <OT >',findstr
[9]   'fnext'DNA'U4 kernel32IFindNextFileA I4 >',findstr
[10]  'close'DNA'kernel32IFindClose I4'
[11]  sepr patn+⊖                          A HSPATH separator and file pat
[12]    wind+':': '\*.dws'                  A windows style.
[13]    unix+':': '/*'                      A unix style
[14]    ω=wind unix                         A select appropriate style.
[15]    ⋄'M'=2>'.'DWG'APLVersion'          A Windows / Unix.
[16]
[17]    found+0P←''                          A Accumulator for found list.
[18]
[19]    :Trap 1000                           A Interrupt: result so far.
[20]
[21]    :For dir :In (COEX'DIRS')/C          A For each directory in list.
[22]      2=DNQ ω:(⊖DFMT↑ω)~''              A use supplied directory li
[23]      sepr⊖DHL+3                          A partition HSPATH,
[24]      (((1+ω=α)=ω)~''α)~⊖←''          A at ';'.
[25]      ⋄2 DNQ',.'GetEnvironment' 'HSPATH' A directory list from HSPATH
  
```

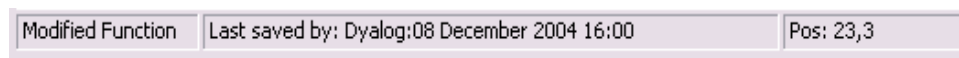
Function Last saved by: Dyalog:08 December 2004 16:00 Pos: 5,12

Debugger Ready... Ins Apl

CurObj: DISPLAY (Function) R:1 ODQ:0 DTRAP OSI:0 DIO:1 DHL:0

### Status Bar

Each Edit window has a status bar containing 3 fields



1. The left field contains a description of the object such as *Function*, *Character Array* or *Nested Array*. The type of object will be preceded by the word *New* or *Modified* if appropriate.
2. The middle field shows the name of the user who last modified the object and the timestamp of modification.
3. The right field displays the current cursor position in the object.

**Dyalog Ltd**  
 South Barn  
 Minchens Court  
 Minchens Lane  
 Bramley  
 Hampshire  
 RG26 5BH  
 United Kingdom

**Phone:**  
 + 44 (0) 1256 830 030

**Fax:**  
 + 44 (0) 1256 830 031

**e-mail:**  
 sales@dyalog.com



▶ **The Development Environment**

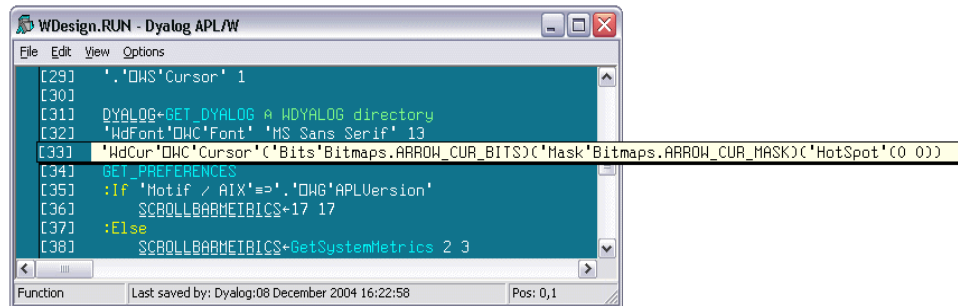
▶ **Editor**

## The Editor

### Popup window for long lines

If a line extends beyond the right edge of an Edit window, a small black triangle is displayed where the line is truncated

If you press and hold down the left mouse button over the black triangle, the entire line will be displayed in a pop-up window as shown below. Releasing the mouse button will remove the window.



The pop-up window for long lines is accessible from Trace windows too.

## Syntax Colouring

Syntax colouring allows you to visually identify the following Syntax Elements in a function edit window by assigning different colours to them:

- Global references (functions and variables)
- Local references (functions and variables)
- Primitive functions
- System functions
- Localised System Functions
- Comments
- Character constants
- Numeric constants
- Labels
- Control Structures
- Unmatched parentheses, quotes, braces and other errors

The *Schemes* combo box allows you to select one of a number of pre-defined Syntax Colouring schemes. To customise your colour scheme, choose a Syntax Element (by clicking on the text in the dummy function, or by selecting from the Syntax Element combo box) and then choose a colour for it. In seconds, you can design the colour scheme of your choice.

**Dyalog Ltd**  
 South Barn  
 Minchens Court  
 Minchens Lane  
 Bramley  
 Hampshire  
 RG26 5BH  
 United Kingdom

**Phone:**  
 + 44 (0) 1256 830 030

**Fax:**  
 + 44 (0) 1256 830 031

**e-mail:**  
 sales@dyalog.com

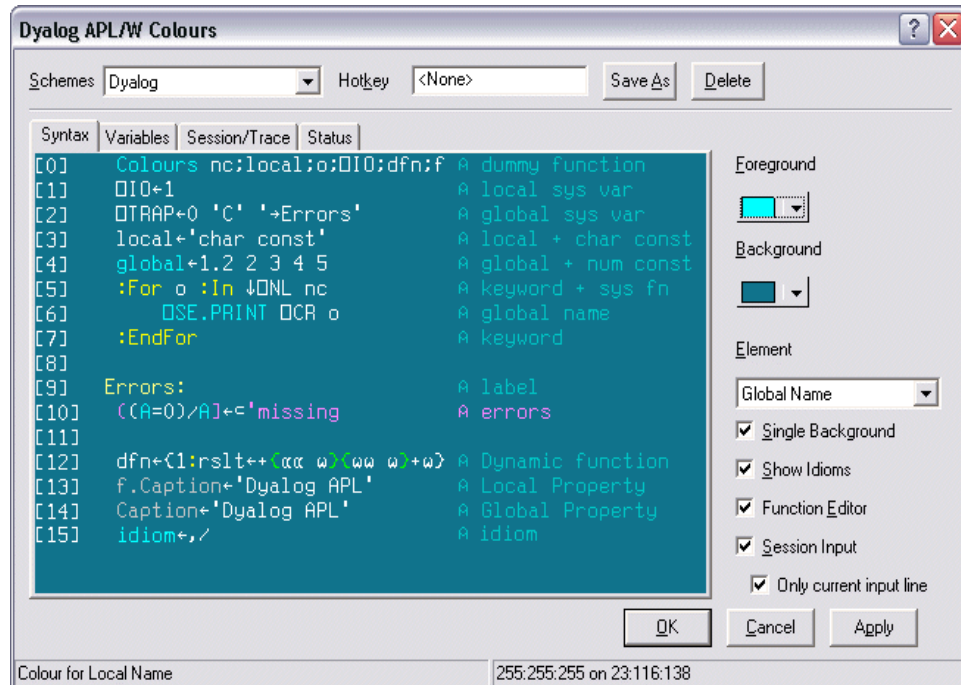


▶ **The Development Environment**

▶ **Editor**

### The Editor

It is also very easy to quickly switch from one colour scheme to another while editing a function, so it makes sense to have special schemes designed to highlight specific Syntax Elements, in addition to your regular ones. For example, there is a supplied 2-colour scheme that highlights global references.



The colour selection dialog box shown above also allows you select colours for:

- Edit, Trace and Session windows
- Status window

The colour selection dialog box is selected by the [[ChooseColor](#)] system action which may be attached to a suitable tool button and/or menu item on the Session window. The default session files have this attached to the Tools menu and to the pop-up menu.

**Dyalog Ltd**  
 South Barn  
 Minchens Court  
 Minchens Lane  
 Bramley  
 Hampshire  
 RG26 5BH  
 United Kingdom

**Phone:**  
 + 44 (0) 1256 830 030

**Fax:**  
 + 44 (0) 1256 830 031

**e-mail:**  
 sales@dyalog.com



▶ The Development Environment

▶ Editor

## The Editor

### Syntax Colour Hotkeys

You may also associate *hot keys* with a syntax colour scheme. When you depress a hot key over a function in an Edit window, the function is displayed using the scheme associated with the hot key. Releasing the hot key causes it to be displayed in the normal scheme.

This feature is intended to allow you to quickly check for certain syntax elements.

For example, you may define a special scheme that only highlights global names and associate a hot key with it. Pressing the hot key will temporarily highlight the globals for you.

The picture below shows the effect of toggling between two different syntax colour schemes, *Default* and *Show Globals*, in a function edit window.

Screen shot missing

#### Dyalog Ltd

South Barn  
Minchens Court  
Minchens Lane  
Bramley  
Hampshire  
RG26 5BH  
United Kingdom

**Phone:**

+ 44 (0) 1256 830 030

**Fax:**

+ 44 (0) 1256 830 031

**e-mail:**

sales@dyalog.com