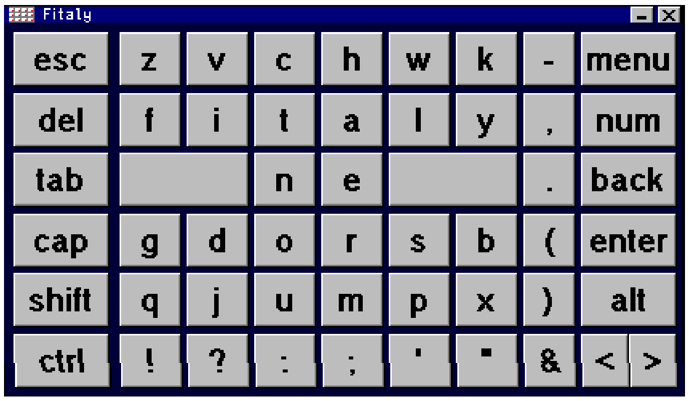
Textware SOLUTIONS

83 Cambridge Street
Burlington MA 01803-4181
617 272 3200
617 272 1432 fax

The API for the Fitaly™ Keyboard

An Application Programming Interface for the Fitaly Keyboard



Textware Solutions

The API for the Fitaly Keyboard

How to command the Fitaly Keyboard from your Application

Introduction

The Application Programming Interface (API) for the Fitaly Keyboard allows you to command the position and appearance of the Fitaly keyboard from your application. The API is supplied as a DLL (dynamic link library), which can be called by your application. For example, you can call functions and procedures of the DLL to do the following:

- Start the Fitaly Keyboard, minimize it, and restore it.
- Position the Fitaly Keyboard according to the needs of you application.
- Park the iconized Fitaly Keyboard at a given position.
- Command the display of numbers and function keys.

For Pen Computer applications, this allows the Fitaly on-screen keyboard to be integrated with your application. Your application can decide when to show or hide the Fitaly and where it should appear.

Contents:

The following subjects are covered in the remaining sections of this document:

Description of the API Functions and Procedures

The following describes the function and procedures provided by the Application Programming Interface.

For each entry, we give the Pascal syntax as well as the Visual Basic syntax. The use of the API is however not limited to these languages and, for example, calls from C are also possible.

Some function returning an answer use the constant values Yes and No. (This has been preferred to a formulation using booleans in order to avoid the difficulties associated with different representations of the values true and false for different programming languages.)

Functions and procedure using keyboard sizes are formulated in terms of integer constants for the five size supported.

Constant	Value
Yes	1
No	0
Tiny_Size	1
Small_Size	2
Medium_Size	3
Large_Size	4
Finger_Size	5

The API function and procedures are subdivided into:

- Functions and Procedures for Starting and Configuring
- Functions and Procedures for Positioning

Functions and Procedures for Starting and Configuring

Fitaly_Running

Pascal Syntax:

```
function Fitaly_Running: Integer;
```

Visual Basic Syntax:

```
Declare Function Fitaly_Running Lib "Fit16.dll" _
      () As Integer
```

Returns Yes if the Fitaly keyboard is currently running, No otherwise.

Fitaly_Minimized

Pascal Syntax:

```
function Fitaly_Minimized: Integer;
```

Visual Basic Syntax:

```
Declare Function Fitaly_Minimized Lib "Fit16.dll" _
      () As Integer
```

Returns Yes if the Fitaly keyboard is not currently running or is mimimized, No otherwise.

Start_Fitaly

Pascal Syntax:
```
procedure Start_Fitaly;
```

Visual Basic Syntax:
```
Declare Sub Start_Fitaly Lib "Fit16.dll" ()
```

This procedure starts the Fitaly keyboard if it is not currently running. If the Fitaly keyboard is currently running and minimized, the effect is to restore it. Otherwise it has no effect.

Terminate_Fitaly

Pascal Syntax:
```
procedure Terminate_Fitaly;
```

Visual Basic Syntax:
```
Declare Sub Terminate_Fitaly Lib "Fit16.dll" ()
```

This procedure ends the execution of the Fitaly keyboard if it is currently running.
Otherwise it has no effect.

Set_Size

Pascal Syntax:
```
procedure Set_Size (Size: Integer);
```

Visual Basic Syntax:
```
Declare Sub Set_Size Lib "Fit16.dll" _
     (ByVal Size As Integer)
```

This procedure changes the size of the Fitaly keyboard if it is currently running.
Otherwise it has no effect.

The Size parameter is used to define the size of the keyboard. The allowed values are defined by the constants listed in the table below. Calling Set_Size with any other value has no effect.

Constant	Value
Tiny_Size	1
Small_Size	2
Medium_Size	3
Large_Size	4
Finger_Size	5

Get_Fitaly_Size

Pascal Syntax:
```
function Get_Fitaly_Size: Integer;
```

Visual Basic Syntax:
```
Declare Function Get_Fitaly_Size Lib "Fit16.dll" _
     () As Integer
```

Returns the current size of the Fitaly keyboard, where the size number is defined as for the procedure Start_Fitaly. Returns a default size equal to 0 if the Fitaly keyboard is not currently running.

Get_Fitaly_Width

Pascal Syntax:
```
function Get_Fitaly_Width: Integer;
```

Visual Basic Syntax:
```
Declare Function Get_Fitaly_Width Lib "Fit16.dll" _
     () As Integer
```

Returns the current width of the Fitaly keyboard expressed in pixels. Returns a default width equal to 0 if the Fitaly keyboard is not currently running.

Get_Fitaly_Height

Pascal Syntax:
```
function Get_Fitaly_Height: Integer;
```

Visual Basic Syntax:
```
Declare Function Get_Fitaly_Height Lib "Fit16.dll" _
     () As Integer
```

Returns the current height of the Fitaly keyboard expressed in pixels. Returns a default height equal to 0 if the Fitaly keyboard is not currently running.

Minimize_Fitaly

Pascal Syntax:
```
procedure Minimize_Fitaly;
```

Visual Basic Syntax:
```
Declare Sub Minimize_Fitaly Lib "Fit16.dll" ()
```

This procedure minimizes the Fitaly keyboard. This procedure has no effect if the Fitaly keyboard is not currently running.

Restore_Fitaly

Pascal Syntax:
```
procedure Restore_Fitaly;
```

Visual Basic Syntax:
```
Declare Sub Restore_Fitaly Lib "Fit16.dll" ()
```

This procedure restores the Fitaly keyboard. This procedure has no effect if the Fitaly keyboard is not currently running.

Has_Phone_Layout

Pascal Syntax:
```
function Has_Phone_Layout: Integer;
```

Visual Basic Syntax:
```
Declare Function Has_Phone_Layout Lib "Fit16.dll" _
     () As Integer
```

Returns Yes if the Fitaly keyboard is currently running and has numbers in the phone layout, No otherwise.

Has_Numbers

Pascal Syntax:
```
function Has_Numbers: Integer;
```

Visual Basic Syntax:
```
Declare Function Has_Numbers Lib "Fit16.dll" _
     () As Integer
```

Returns Yes if the Fitaly keyboard is currently running and has numbers, No otherwise.

Has_Function_Keys

Pascal Syntax:
```
function Has_Function_Keys: Integer;
```

Visual Basic Syntax:
```
Declare Function Has_Function_Keys Lib "Fit16.dll" _
     () As Integer
```

Returns Yes if the Fitaly keyboard is currently running and has function keys, No otherwise.

Set_Phone_Layout

Pascal Syntax:
```
procedure Set_Phone_Layout;
```

Visual Basic Syntax:
```
Declare Sub Set_Phone_Layout Lib "Fit16.dll" ()
```

This procedure establishes the phone layout for numbers. This procedure has no effect if the Fitaly keyboard is not currently running.

Set_Calculator_Layout

Pascal Syntax:
```
procedure Set_Calculator_Layout;
```

Visual Basic Syntax:
```
Declare Sub Set_Calculator_Layout Lib "Fit16.dll" ()
```

This procedure establishes the calculator layout for numbers. This procedure has no effect if the Fitaly keyboard is not currently running.

Show_Numbers

Pascal Syntax:
```
procedure Show_Numbers;
```

Visual Basic Syntax:
```
Declare Sub Show_Numbers Lib "Fit16.dll" ()
```

This procedure shows numbers on the Fitaly keyboard. This procedure has no effect if the Fitaly keyboard is not currently running.

Hide_Numbers

Pascal Syntax:
```
procedure Hide_Numbers;
```

Visual Basic Syntax: `Declare Sub Hide_Numbers Lib "Fit16.dll" ()`

This procedure hides numbers on the Fitaly keyboard. This procedure has no effect if the Fitaly keyboard is not currently running.

Show_Function_Keys

Pascal Syntax: `procedure Show_Function_Keys;`

Visual Basic Syntax: `Declare Sub Show_Function_Keys Lib "Fit16.dll" ()`

This procedure shows function keys on the Fitaly keyboard. This procedure has no effect if the Fitaly keyboard is not currently running.

Hide_Function_Keys

Pascal Syntax: `procedure Hide_Function_Keys;`

Visual Basic Syntax: `Declare Sub Hide_Function_Keys Lib "Fit16.dll" ()`

This procedure hides function keys on the Fitaly keyboard. This procedure has no effect if the Fitaly keyboard is not currently running.

Get_Fitaly_Handle

Pascal Syntax: `function Get_Fitaly_Handle: Integer;`

Visual Basic Syntax:

```
Declare Function Get_Fitaly_Handle Lib "Fit16.dll" _
     () As Integer
```

Returns the window handle of the Fitaly keyboard if it is currently running.
Otherwise returns the default value 0.

Positioning

The following functions and procedures allow:

- definition of the units used (Pixels or Twips)
- setting the Fitaly keyboard at a specified position
- getting the current position, width and height.

Use_Pixel   Use_Twips

Pascal Syntax:

```
procedure Use_Pixels;
procedure Use_Twips;
```

Visual Basic Syntax:

```
Declare Sub Use_Pixels "Fit16.dll"  ()
Declare Sub Use_Twips  "Fit16.dll"  ()
```

These procedures establish whether coordinates are expressed in Pixels or in Twips.
By default, coordinates are expressed in Pixels.

Calibrate

Pascal Syntax:

```
procedure Calibrate (XTwips, YTwips: Longint);
```

Visual Basic Syntax:

```
Declare Sub Calibrate "Fit16.dll"  _
     (ByVal XTwips As Long, ByVal XTwips As Long)
```

This procedure establishes the current values for twips per pixels. This procedure is useful mostly for Visual Basic applications. The call from Visual Basic should be as follow:

```
Calibrate (Screen.TwipsPerPixelX, Screen.TwipsPerPixelY)
```

This procedure must be called before the first positioning operation that uses values expressed in Twips. Prior to the first call, default values equal to 20 Twips per Pixel are used.

Set_Fitaly_above

Pascal Syntax:

```
procedure Set_Fitaly_above (X, Y: Longint);
```

Visual Basic Syntax:

```
Declare Sub Set_Fitaly_above "Fit16.dll"  _
     (ByVal X As Long, ByVal Y As Long)
```

This procedure positions the Fitaly keyboard to be above the point (X,Y): this means that (X,Y) become the coordinates of the lower left corner of the Fitaly Keyboard. . The coordinates are expressed in the current units established by the last call to Use_Pixels or Use_Twips.

This procedure has no effect if the new position would cause the upper left corner of the Fitaly Keyboard to be out of the screen.

This procedure has no effect if the Fitaly keyboard is not currently running.

Set_Fitaly_below

Pascal Syntax:

```
procedure Set_Fitaly_below (X, Y: Longint);
```

Visual Basic Syntax:

```
Declare Sub Set_Fitaly_below "Fit16.dll"  _
     (ByVal X As Long, ByVal Y As Long)
```

This procedure positions the Fitaly keyboard to be below the point (X,Y): this means that (X,Y) become the coordinates of the upper left corner of the Fitaly Keyboard. The coordinates are expressed in the current units established by the last call to Use_Pixels or Use_Twips.

This procedure has no effect if the new position would cause the lower left corner of the Fitaly Keyboard to be out of the screen.

This procedure has no effect if the Fitaly keyboard is not currently running.

Fitaly_Possible_above

Pascal Syntax:

```
function Fitaly_Possible_above (Y: Longint): Integer;
```

Visual Basic Syntax:

```
Declare Function Fitaly_Possible_above Lib "Fit16.dll" _
      (ByVal Y As Long) As Integer
```

This function return Yes if the Fitaly keyboard can be placed above the Y coordinate, No otherwise. The Y coordinate is expressed in the current units established by the last call to Use_Pixels or Use_Twips.

This function returns No if the Fitaly keyboard is not currently running.

Fitaly_Possible_below

Pascal Syntax:

```
function Fitaly_Possible_below (Y: Longint): Integer;
```

Visual Basic Syntax:

```
Declare Function Fitaly_Possible_below Lib "Fit16.dll" _
      (ByVal Y As Long) As Integer
```

This function return Yes if the Fitaly keyboard can be placed below the Y coordinate, No otherwise. The Y coordinate is expressed in the current units established by the last call to Use_Pixels or Use_Twips.

This function returns No if the Fitaly keyboard is not currently running.

Set_Fitaly_Parking

Pascal Syntax:

```
procedure Set_Fitaly_Parking (X, Y: Longint);
```

Visual Basic Syntax:

```
Declare Sub Set_Fitaly_Parking "Fit16.dll"  _
     (ByVal X As Long, ByVal Y As Long)
```

This procedure defines the position for the Fitaly keyboard icon to be the point (X,Y). The coordinates are expressed in pixels.

This procedure has no effect if the position specified is not within the screen.

Values of the parking positions are stored in the Fitaly.ini file after conversion (if needed) in Pixels.

Fitaly_X   Fitaly_Y   Fitaly_Width   Fitaly_Height

Pascal Syntax:

```
function Fitaly_X      : Longint;
function Fitaly_Y      : Longint;
function Fitaly_Width  : Longint;
function Fitaly_Height : Longint;
```

Visual Basic Syntax:

```
Declare Function Fitaly_X Lib "Fit16.dll" () As Long
Declare Function Fitaly_Y Lib "Fit16.dll" () As Long
Declare Function Fitaly_Width  Lib "Fit16.dll" () As Long
Declare Function Fitaly_Height Lib "Fit16.dll" () As Long
```

Each of these function returns the corresponding value or coordinate for the current position of the Fitaly Keyboard. The values returned are expressed in the current units established by the last call to Use_Pixels or Use_Twips.

If the Fitaly keyboard is iconized, the functions Fitaly_X and Fitaly_Y return the position of the icon.
If the Fitaly keyboard is not currently running, the functions Fitaly_X and Fitaly_Y return the value 0.

If the Fitaly keyboard is iconized or not currently running, the functions Fitaly_Width and Fitaly_Height return the width and height of the icon.

Pascal Listing of the API

```
unit FITALY_API;
interface
uses WinTypes;

function     Fitaly_Running      : Integer;
function     Fitaly_Minimized    : Integer;

procedure    Start_Fitaly;
procedure    Terminate_Fitaly;
procedure    Set_Size            (Size: Integer);

function     Get_Fitaly_Size     : Integer;
function     Get_Fitaly_Width    : Integer;
function     Get_Fitaly_Height   : Integer;

procedure    Minimize_Fitaly;
procedure    Restore_Fitaly;

function     Has_Phone_Layout    : Integer;
function     Has_Numbers         : Integer;
function     Has_Function_Keys   : Integer;

procedure    Set_Phone_Layout;
procedure    Set_Calculator_Layout;

procedure    Show_Numbers;
procedure    Hide_Numbers;
procedure    Show_Function_Keys;
procedure    Hide_Function_Keys;

{ System function:  }

function   Get_Fitaly_Handle: HWnd;

{ Positioning: }

procedure    Use_Pixels;
procedure    Use_Twips;

procedure    Calibrate              (XTwips, YTwips: Longint);

procedure    Set_Fitaly_above       (X, Y: Longint);
procedure    Set_Fitaly_below       (X, Y: Longint);

function     Fitaly_Possible_above  (Y: Longint): Integer;
function     Fitaly_Possible_below  (Y: Longint): Integer;

procedure    Set_Fitaly_Parking     (X, Y: Longint);

function     Fitaly_X      : Longint;
function     Fitaly_Y      : Longint;
function     Fitaly_Width  : Longint;
function     Fitaly_Height : Longint;
```

```
implementation
uses WinProcs;
function      Fitaly_Running;               external 'FIT16.DLL';

procedure     Start_Fitaly;                 external 'FIT16.DLL';
procedure     Set_Size;                     external 'FIT16.DLL';

function      Get_Fitaly_Size;              external 'FIT16.DLL';
function      Get_Fitaly_Width;          external 'FIT16.DLL';
function      Get_Fitaly_Height;            external 'FIT16.DLL';

procedure     Minimize_Fitaly;              external 'FIT16.DLL';
procedure     Restore_Fitaly;               external 'FIT16.DLL';

function      Has_Phone_Layout;             external 'FIT16.DLL';
function      Has_Numbers;                  external 'FIT16.DLL';
function      Has_Function_Keys;            external 'FIT16.DLL';

procedure     Set_Phone_Layout;             external 'FIT16.DLL';
procedure     Set_Calculator_Layout;        external 'FIT16.DLL';
procedure     Show_Numbers;                 external 'FIT16.DLL';
procedure     Hide_Numbers;                 external 'FIT16.DLL';
procedure     Show_Function_Keys;           external 'FIT16.DLL';
procedure     Hide_Function_Keys;           external 'FIT16.DLL';

{ System function: }

function      Get_Fitaly_Handle;            external 'FIT16.DLL';

{ Positioning: }

procedure     Use_Pixels                    external 'FIT16.DLL';
procedure     Use_Twips;                    external 'FIT16.DLL';

procedure     Calibrate;                    external 'FIT16.DLL';

procedure     Set_Fitaly_above              external 'FIT16.DLL';
procedure     Set_Fitaly_below              external 'FIT16.DLL';

function      Fitaly_Possible_above         external 'FIT16.DLL';
function      Fitaly_Possible_below         external 'FIT16.DLL';

procedure     Set_Fitaly_Parking         external 'FIT16.DLL';

function      Fitaly_X                      external 'FIT16.DLL';
function      Fitaly_Y                   external 'FIT16.DLL';
function      Fitaly_Width                  external 'FIT16.DLL';
function      Fitaly_Height                 external 'FIT16.DLL';

end.
```

Visual Basic Listing of the API

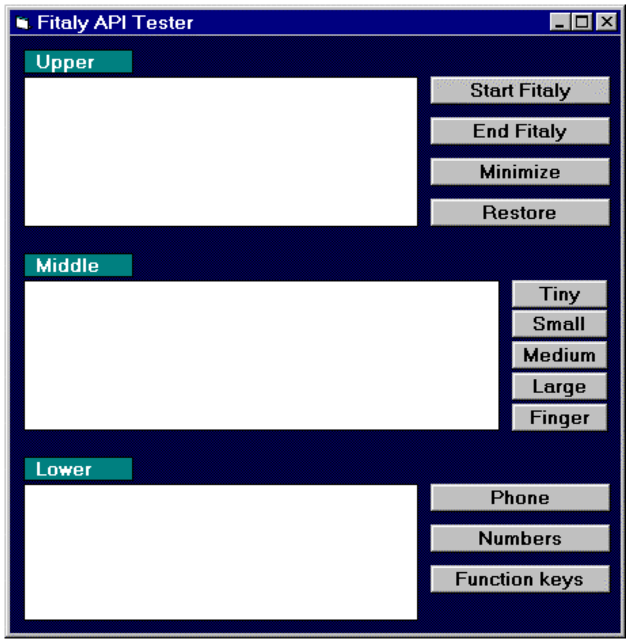
The following declarations should appear in the declaration section of the main form. The constants specify values returned by the functions and used by some of the subprograms.

```
'  Constant declarations:

Const Yes           = 1
Const No            = 0

Const Tiny_Size     = 1
Const Small_Size    = 2
Const Medium_Size = 3
Const Large_Size    = 4
Const Finger_Size = 5


'  Starting and Configuring

Declare Function Fitaly_Running Lib "Fit16.dll" () As Integer
Declare Function Fitaly_Minimized Lib "Fit16.dll" () As Integer

Declare Sub Start_Fitaly Lib "Fit16.dll" ()
Declare Sub Terminate_Fitaly Lib "Fit16.dll" ()
Declare Sub Set_Size Lib "Fit16.dll" (ByVal Size As Integer)

Declare Function Get_Fitaly_Size Lib "Fit16.dll" () As Integer
Declare Function Get_Fitaly_Width Lib "Fit16.dll" () As Integer
Declare Function Get_Fitaly_Height Lib "Fit16.dll" () As Integer

Declare Sub Minimize_Fitaly Lib "Fit16.dll" ()
Declare Sub Restore_Fitaly Lib "Fit16.dll" ()

Declare Function Has_Phone_Layout Lib "Fit16.dll" () As Integer
Declare Function Has_Numbers Lib "Fit16.dll" () As Integer
Declare Function Has_Function_Keys Lib "Fit16.dll" () As Integer

Declare Sub Set_Phone_Layout Lib "Fit16.dll" ()
Declare Sub Set_Calculator_Layout Lib "Fit16.dll" ()

Declare Sub Show_Numbers Lib "Fit16.dll" ()
Declare Sub Hide_Numbers Lib "Fit16.dll" ()
Declare Sub Show_Function_Keys Lib "Fit16.dll" ()
Declare Sub Hide_Function_Keys Lib "Fit16.dll" ()

'  For system calls:

Declare Function Get_Fitaly_Handle Lib "Fit16.dll" () As Integer
```

```
'  Positioning

Declare Sub Use_Pixels Lib "Fit16.dll" ()
Declare Sub Use_Twips Lib "Fit16.dll" ()

Declare Sub Calibrate Lib "Fit16.dll" (ByVal XTwips As Long, ByVal YTwips As Long)

Declare Sub Set_Fitaly_Above Lib "Fit16.dll" (ByVal X As Long, ByVal Y As Long)
Declare Sub Set_Fitaly_Below Lib "Fit16.dll" (ByVal X As Long, ByVal Y As Long)

Declare Function Fitaly_Possible_above Lib "Fit16.dll" (ByVal Y As Long) As Integer
Declare Function Fitaly_Possible_below Lib "Fit16.dll" (ByVal Y As Long) As Integer

Declare Sub Set_Fitaly_Parking Lib "Fit16.dll" (ByVal X As Long, ByVal Y As Long)

Declare Function Fitaly_X Lib "Fit16.dll" () As Long
Declare Function Fitaly_Y Lib "Fit16.dll" () As Long
Declare Function Fitaly_Width Lib "Fit16.dll" () As Long
Declare Function Fitaly_Height Lib "Fit16.dll" () As Long
```

Tester – A Simple Example of Use of the API

The following example illustrates the use of the Fitaly API to build a small Pen-based application that interacts with The Fitaly keyboard.

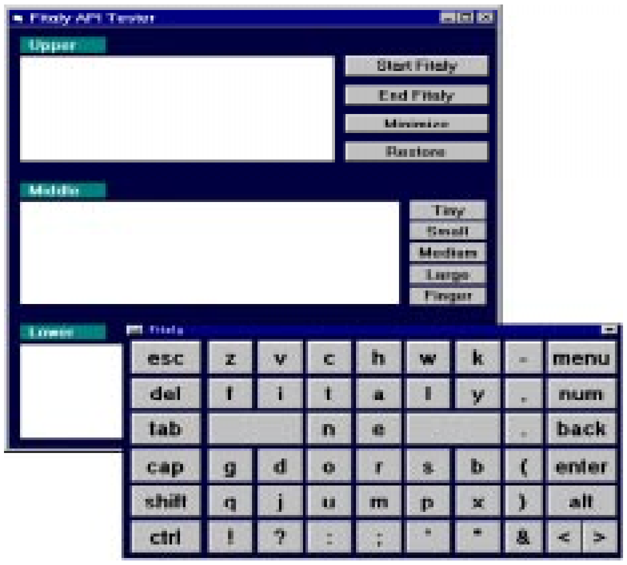


The buttons can be used to accomplish most of the functions offered by the API, such as starting and terminating the Fitaly Keyboard, minimizing or restoring it.

The buttons Tiny through Finger command the resizing of the keyboard very much in the same way as what can be accomplished with the Menu dialog.

Similarly, the buttons Phone, Numbers, and Function keys have been programmed as toggles between two possible values. As the toggles are exercized, the caption of the button is changed. Here too, the API achieves by program what can be otherwise done with buttons of the Menu dialog.

This mini-application also has three text boxes called Upper, Middle, and Lower.



A click in the Upper box will position the keyboard as shown in this example. Similarly, a click in the Lower box will position the keyboard at the top of the screen to allow entry into the Lower box.

In addition, the Middle box shows how to program text entry so that the keyboard is minimized when the Enter key is tapped.

Finally, if the keyboard is minimized, a click in any of the text boxes restores it with a position allowing text entry in that box.

Tester in Visual Basic

The example given below is programmed as a Visual Basic application.

It includes the decarations for the API functions (see previous section) and then the following subprograms associated with corresponding events.

Loading the form

```
Sub Form_Load()
 Dim XTwips As Long
   Dim YTwips As Long

   XTwips = Screen.TwipsPerPixelX
   YTwips = Screen.TwipsPerPixelY

   Call Calibrate(XTwips, YTwips)
   Use_Twips
End Sub
```

Upon loading, call Calibrate with appropriate values

Start, End, Minimize, and Restore Buttons

```
Sub StartButton_Click()
   Start_Fitaly

   If Has_Phone_Layout() = Yes Then
      PhoneButton.Caption = "Calculator"
   Else
      PhoneButton.Caption = "Phone"
   End If

   If Has_Numbers() = Yes Then
      NumberButton.Caption = "Hide Numbers"
   Else
      NumberButton.Caption = "Show Numbers"
   End If

   If Has_Function_Keys() = Yes Then
      FKeyButton.Caption = "Hide F-Keys"
   Else
      FKeyButton.Caption = "Show F-Keys"
   End If
End Sub
```

Start the Fitaly keyboard and establish the captions for the three toggles. The correct values are obtained by calling the Has_... functions.

```
Sub EndButton_Click()
   Terminate_Fitaly
End Sub

Sub MinimizeButton_Click()
   Minimize_Fitaly
End Sub

Sub RestoreButton_Click()
   Restore_Fitaly
End Sub
```

Direct calls to the corresponding API procedures

Sizing Buttons

```
Sub TinyButton_Click()
   Set_Size (Tiny_Size)
End Sub

Sub SmallButton_Click()
   Set_Size (Small_Size)
End Sub

Sub MediumButton_Click()
   Set_Size (Medium_Size)
End Sub

Sub LargeButton_Click()
   Set_Size (Large_Size)
End Sub

Sub FingerButton_Click()
   Set_Size (Finger_Size)
End Sub
```

Direct calls to the Set_Size procedure with the appropriate size constant

Phone, Numbers, and Function Keys Toggles

```
Sub PhoneButton_Click()
   If Has_Phone_Layout() = Yes Then
      Set_Calculator_Layout
      PhoneButton.Caption = "Phone"
   Else
      Set_Phone_Layout
      PhoneButton.Caption = "Calculator"
   End If
End Sub

Sub NumberButton_Click()
   If Has_Numbers() = Yes Then
      Hide_Numbers
      NumberButton.Caption = "Show Numbers"
   Else
      Show_Numbers
      NumberButton.Caption = "Hide Numbers"
   End If
End Sub

Sub FKeyButton_Click()
  If Has_Function_Keys() = Yes Then
      Hide_Function_Keys
      FKeyButton.Caption = "Show F-Keys"
   Else
      Show_Function_Keys
      FKeyButton.Caption = "Hide F-Keys"
   End If
End Sub
```

For each of the three toggles a call to Has_... finds the current state. The proper procedure is then called. Finally, the caption is changed to reflect the toggling.

Lower and Upper Text Boxes

```
Sub LowBox_GotFocus()
   Dim X As Long
   Dim Y As Long

   X = UpperLabel.Left + Tester.Left + UpperLabel.Width
   Y = 0

   If Fitaly_Possible_below(Y) Then
      Call Set_Fitaly_Below(X, Y)
      LowBox.SetFocus
   End If
End Sub
```

Position Fitaly at the top of the screen so as to allow typing into the Lower box.

```
Sub UpperBox_GotFocus()
   Dim X As Long
   Dim Y As Long

   X = LowLabel.Left + Tester.Left + LowLabel.Width
   Y = LowLabel.Top + Tester.Top + LowLabel.Height

   If Fitaly_Possible_below(Y) Then
      Call Set_Fitaly_Below(X, Y)
      UpperBox.SetFocus
   End If
 End Sub
```

Position Fitaly just below the Lower Label so as to allow typing into the Upper box.

Middle Text Box

```
Sub MidBox_GotFocus()
   Dim X As Long
   Dim Y As Long

   X = LowLabel.Left + Tester.Left + LowLabel.Width
   Y = LowLabel.Top + Tester.Top + LowLabel.Height

   If Fitaly_Possible_below(Y) Then
      Call Set_Fitaly_Below(X, Y)
      MidBox.SetFocus
   End If
End Sub
```

Position Fitaly just below the Lower Label so as to allow typing into the Middle box.

```
Sub MidBox_KeyDown(KeyCode As Integer, Shift As Integer)
   If KeyCode = 13 Then
      ' process the information and then:

      Minimize_Fitaly
   End If
End Sub
```

Minimize Fitaly when the Enter key is pressed (after processing the information - not shown here)

INI File Parameters

The INI file for the Fitaly Keyboard is as follows

```
[Fitaly]
Fitaly16 File=C:\Fitaly\Fitaly16.exe
Numbers=Y
Phone=Y
Function keys=N
Size=Huge
Hardware Keyboard=N
Closing Allowed=N
Switch Allowed=Y
Parking X=3000
Parking Y=3000
```

KeyWord	Values	Comment
Numbers	Y N	Yes if numbers are shown. No if numbers are hidden
Phone	Y N	Yes if the phone layout is used No if the calculator layout is used
Function keys	Y N	Yes if function keys are shown. No if function keys are hidden.
Size	Tiny Small Medium Large Huge	The initial size for the keyboard
Hardware Keyboard	Y N	Yes if there is a hardware keyboard. No otherwise
Closing Allowed	Y N	No if Closing of the keyboard is not allowed. In such a case the system menu and the option dialog will allow minimizing the keyboard but not closing it. This option is recommended if the unit has no hardware keyboard attached. Yes if closing is allowed.
Switch Allowed	Y N	Yes if the system menu includes a swich item. No otherwise
Parking X and Y		The parking positions in Pixels for the Fitaly icon. A value of 3000 (greater than the screen size) indicates no specified parking position.