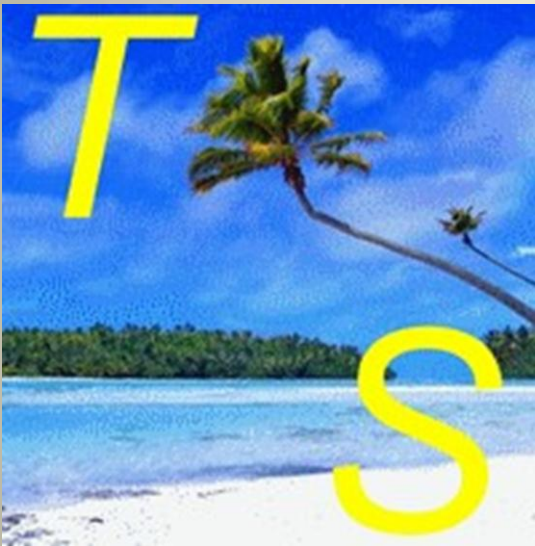


Tropical Supply,
Inc.



User Manual: Invoicing Application

by Kelly Frye
December 9, 2010

Table of Contents

Section 1: User Information

Introduction.....	3
Our Mission Statement.....	3
The Splash Screen	4
Login Form.....	5
Main Invoice Entry Form.....	6
Button Controls	9
Menu Options.....	10
Navigation Bar	11
Status Bar	12
Informational Website Link.....	12
Note Pad.....	13
NotePad Menu Options	14
DataBase Grid Viewer	15
Invoice Report Viewer.....	16
Report Viewer Menu and Nav Bar	17
Report Viewer Reports	17
FAQ.....	18
Hints and Tips	18
About.....	19

Section 2: Programmer Information

Divisions of Code Regions	20
Public Variable Declarations	21
Form Load.....	21
Main Entry Data Field Controls	22
Main Entry Button Controls	24
Main Entry Menu Options Controls	25
Navigation Bar Sub Routines.....	26
NotePad Controls.....	27
Splash Screen Code.....	30
Login Form Code	31
About Box Code	32
Data Base SQL Code.....	33

INTRODUCTION

Welcome to Tropical Supply's Invoicing Application. And, Thank You for being one of our Users!

This application is not just for a supplier. It is designed for any User that issues invoices – in other words, for every company or organization. It is a simple and easy to use invoice entering, tracking, and reporting application. In the following sections, you will be introduced to all of its features and learn how to use them.

Using this manual: For the Data Entry User, Section 1 of this manual provides step-by-step, graphical instructions on how to use this application. For the Software Developer, Section 2 of this manual provides all of the code for all of the functions of this application. The software code is grouped into logically ordered “regions,” as listed in the Table of Contents.

Our Mission Statement

It is our Mission to provide all Users of this Application with the best experience possible in entering, tracking, and reporting their invoices by providing an intuitively simple Invoicing Application – to transform tediously mundane data entry into joyfully simple invoicing.

We accomplish our mission by providing a simple “graphical user interface” (GUI), and automating data entry error checking, data recording, and invoice reporting.

THE SPLASH SCREEN

When you launch this application, it announces its start-up by displaying a “splash screen” in the center of your monitor. This lets you know that the program is loading. It looks like this:





LOGIN FORM

Once the program has loaded, the User is prompted to enter his or her login information. This consists of a user Name and a Password. If the User does not properly enter this information, or forgets his or her user name or password, the login utility will give the User three and only three login attempts. The program will immediately close, if the third login is incorrect. The login form appears as below:



Of course, we all forget or miss-place our login information from time to time. So, we have built-in a “backdoor.” This is a small area on the form that a User or

Administrator can click to gain entry into the application. This “backdoor” information should be known only to an Administrator for security reasons. The shaded area  does not appear on the form. It is shown here for illustration purposes.



Backdoor

MAIN INVOICE ENTRY FORM

Next we describe all of the data fields on the Main Invoice Entry Form, and the use and function(s) of each data field. The picture of the form below depicts all of the data fields:

Tropical Supply Invoicing

File Edit Tools Help | 1 of 7 | + X

Main Invoice Entry Page | Note Pad | DataBase Grid Viewer | Invoice Report Viewer

Main Invoice Entry

1 → Today's Date 12/5/2010

2 → Invoice Number 101

3 → Customer Name Bob Smith

4 → Item Description Safari Trip

5 → Part # T1001

6 → Quantity 1

7 → Unit Price \$100

Sales Tax \$7.00

SubTotal Invoice \$100.00 Total Invoice \$107.00

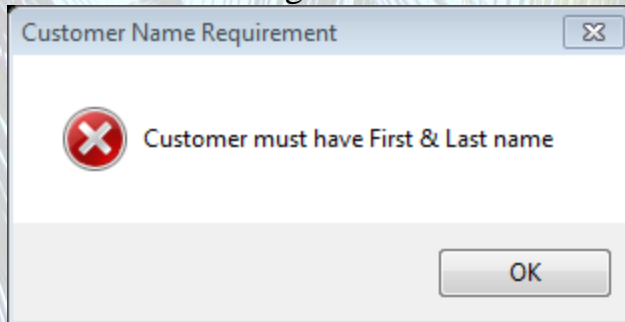
Save Next EXIT

Link for Helpful Info [General Accounting Practices](#)

In Progress Invoices Entered

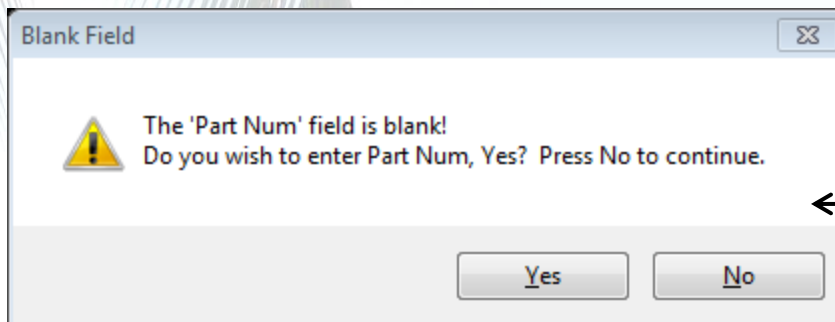
1. "Today's Date" Field: This field will always display the current date and cannot be modified by the User.
2. "Invoice Number" Field: A User entered number. This field allows any alpha-numeric entry and does not check its value, so that the entry can be customized or omitted, as necessary.

3. “Customer Name” Field: User Entered. Upon tabbing or clicking off of this field, the program will check the entry for first and last name. This will also work with a company name, because it checks for the presence of at least one blank space. If a company’s name is a single name, its correct legal name should be entered, such as: “ACompany, Inc.” An example of this field’s error message box is shown here:



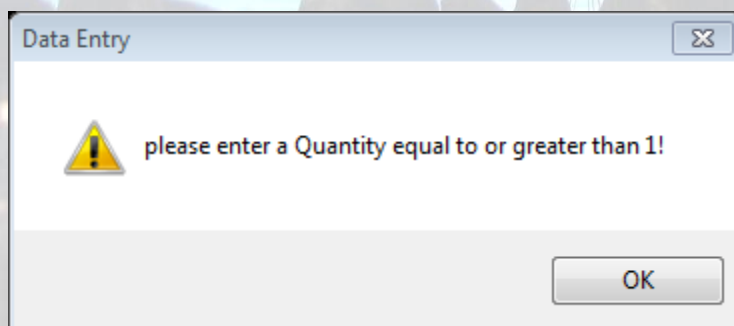
Also, A sound will occur when this error is encountered. Simply click “OK” and re-enter the customer name.

4. “Item Description” Field: User Entered. It is an optional field. It is recommended that an item description be entered, but it is not required.
5. “Part Number” Field: User Entered. For inventory tracking purposes, a part number is requested. The error check message box appears as shown here. The User does have the option of continuing without

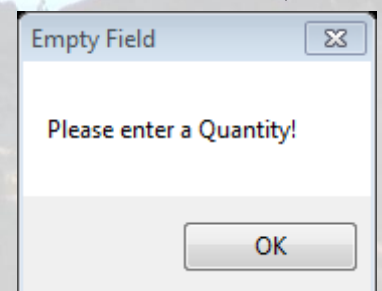


a part number. But, this is not recommended for inventory control reasons. A part number must be four alpha-numeric characters long. The error check routine will automatically discard any character after the fourth and capitalize all characters.

6. “Quantity” Field: User Entered. As you might image, one can’t have an invoice amount without a quantity and unit price. So, entering a quantity is a requirement. If a quantity isn’t entered, this message box appears.

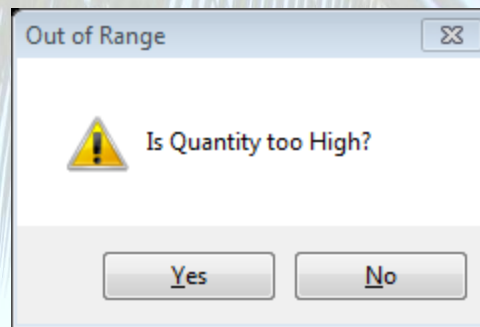


This field also checks for valid numeric entry. This message appears, if a quantity value of less than 1 is entered.



If a quantity amount of 1000 or greater is entered, the warning message box shown here is displayed:

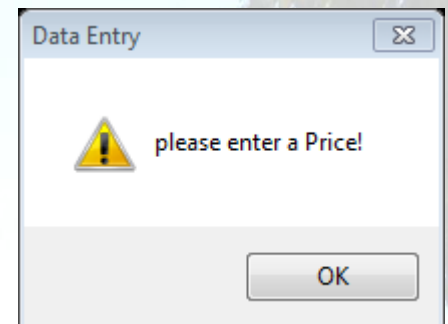
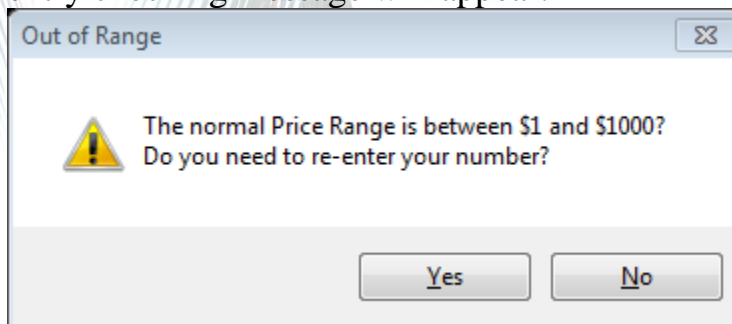
mistaken entry.
functions of this data
noticed, mistaken
of the invoice record.



It is easy to make a
These error checking
field helps to prevent un-
entries to become a part

7. “Unit Price” Field: User Entered. A price must be entered. The program will not let you leave this field without a valid price entered. This field performs similar entry validation (error checking) as the “Quantity” field. If the field is left blank, this message box will appear:

If the price entered is \$1000 or more, another entry checking message will appear:



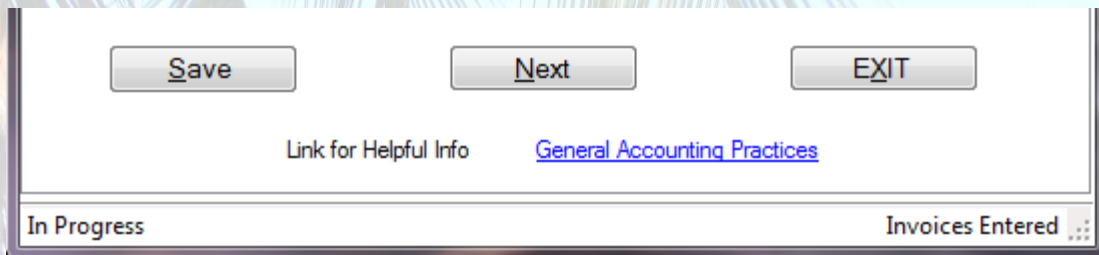
This will help the User catch accidental entries, or validate an entry that is more than the customary price.

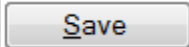
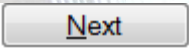
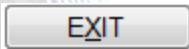
8. “Sub Total Invoice,” “Sales Tax,” and “Total Invoice” Fields: These fields are all automatically calculated once the last User entered field (Unit Price) is completed.

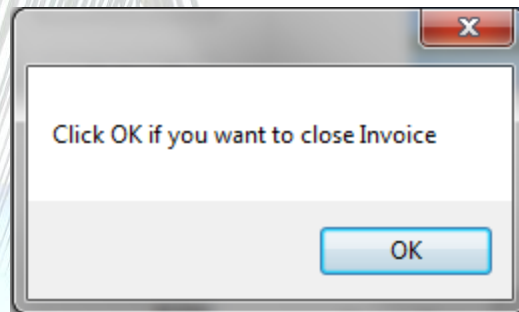
A screenshot of a software interface for creating an invoice. It features several input and display fields: "Unit Price" with a value of "\$100", "SubTotal Invoice" with a value of "\$100.00", "Sales Tax" with a value of "\$7.00", and "Total Invoice" with a value of "\$107.00". Below these fields are three buttons: "Save", "Next", and "EXIT". A yellow box with the number "8" is placed over the "Next" button. At the bottom of the window, there is a status bar with "In Progress" on the left and "Invoices Entered" followed by a small icon on the right. A link for "General Accounting Practices" is also visible.

Button Controls

On the Main Invoice Entry form, there are three button controls: “Save,” “Next,” and “Exit.” The keyboard shortcuts are as follow, respectively: Alt-S, Alt-N, and Alt-X. When the User hovers the cursor over each button, a tooltip note will appear. The tooltip note informs the User what the button will perform when clicked. The buttons are shown in the picture below:



- | | |
|---|---|
|  | Button will save data entered. |
|  | Button does not save data, but moves all fields to the next blank record for entry of a new invoice. |
|  | Button closes the program. To guard against accidental exit, and lose of data, a message box prompts to User to confirm the exit. |



See also “Navigation Bar” and “Status Bar” for more information.

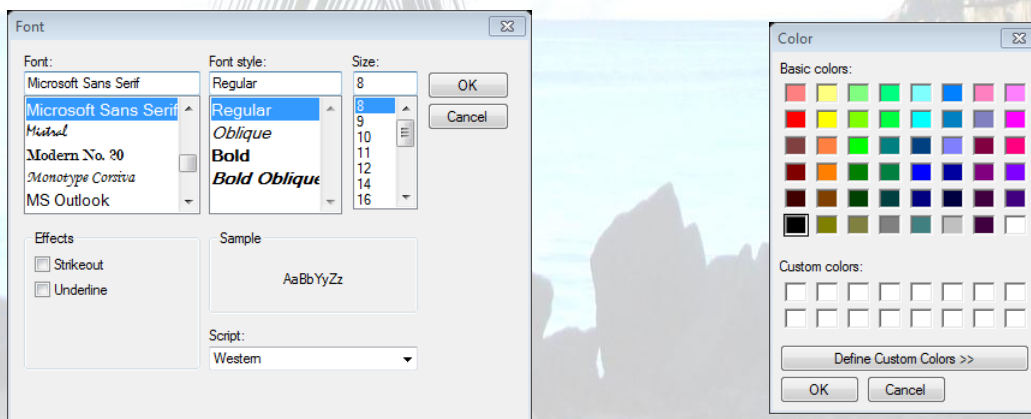
Menu Options

The Main Invoice Entry form has the menu options as shown here:



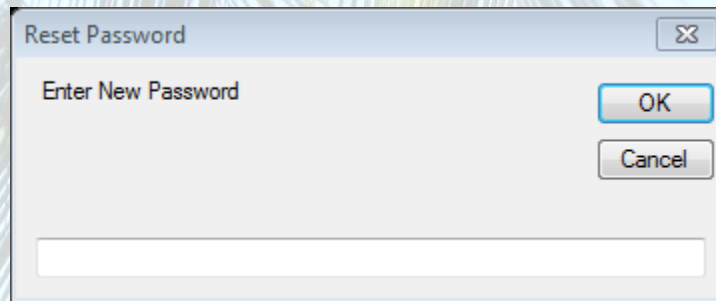
All menu items have keyboard shortcut keys, which follow the format of simultaneously pressing the Alt key and the letter, as underlined, in each menu item selection. Since these menu selections perform self-evident functions, we draw your attention to only a few unique menu items, as follow:

Edit > Fonts, and Edit > Color will change the font and color of all form labels. Here are depictions of the Font and Color dialogue boxes:



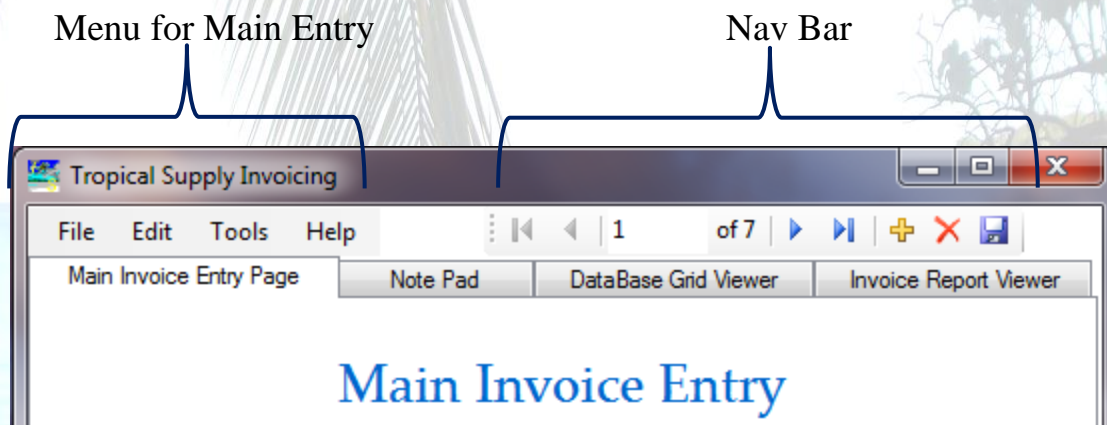
Additionally, the Font or Color dialogue boxes can be accessed by right-clicking on any blank space on the form.




Also, Tools > Reset Password is important, because this is the only utility that allows an Administrator to reset the User Password. This tool should be used if a User has forgotten his or her password, and the backdoor is used to gain access to the program. Simply enter the new password and press OK. Password Reset message box is depicted below:



Navigation Bar

The Navigation Bar performs some features that the aforementioned button controls perform. However, the ease and power of the Navigation Bar is to facilitate moving through all invoice records, like flipping a page, adding a new invoice record to enter, and saving all changes with the save icon. The Navigation's Bar (Nav Bar) exclusive use of icons makes it instinctively easy to use:



In the Nav Bar, press the left or right arrows to page thru records. The text box will indicate which record of total records you are currently on. The  icon will add a new blank record for new invoice entry. The  icon deletes the record shown. The  icon saves the record currently displayed.

Status Bar

The status bar is at the bottom of the Main Entry form:



A screenshot of the status bar at the bottom of the Main Entry form. It consists of a horizontal bar with a light beige background and a dark border. On the left side, the text "In Progress" is displayed. On the right side, the text "Invoices Entered" is displayed, followed by a small icon of three dots.

In the left corner, “In Progress” will become highlighted in yellow when an invoice is being entered. Each time an invoice is entered, “In Progress” turns black & white and “Invoices Entered” shows the number of invoices entered and saved, so far.



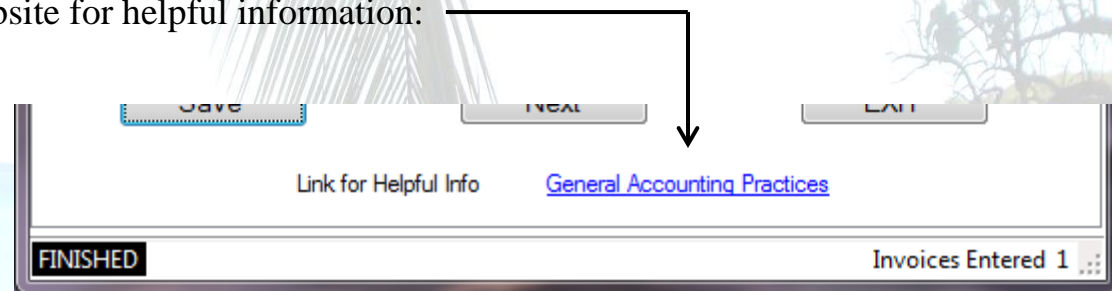
A screenshot of the status bar at the bottom of the Main Entry form. The text "In Progress" on the left is highlighted in yellow. The text "Invoices Entered" is on the right, followed by a small icon of three dots.



A screenshot of the status bar at the bottom of the Main Entry form. The text "FINISHED" on the left is in a black box. The text "Invoices Entered 1" is on the right, followed by a small icon of three dots.

Informational Website Link

Please notice that there is a link to the National General Accounting Practices website for helpful information:

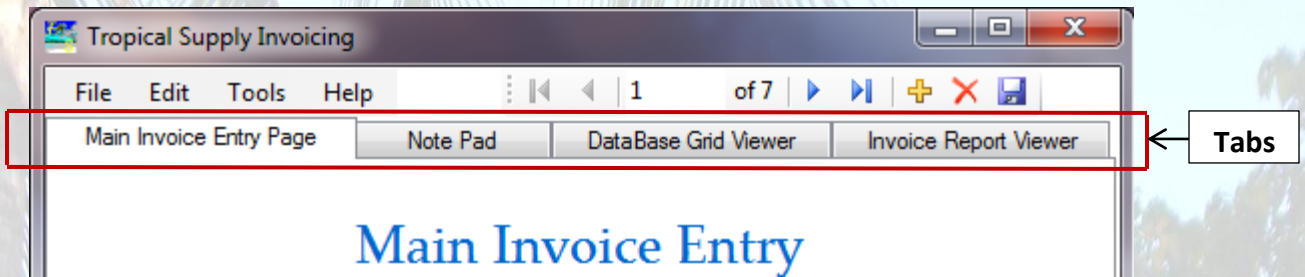


A screenshot of the Main Entry form. At the top, there are three buttons: "Save", "Next", and "Exit". Below these buttons, there is a text label "Link for Helpful Info" and a blue underlined hyperlink "General Accounting Practices". An arrow points from the text "Please notice that there is a link to the National General Accounting Practices website for helpful information:" to the hyperlink. At the bottom of the form, there is a status bar with "FINISHED" on the left and "Invoices Entered 1" on the right, followed by a small icon of three dots.

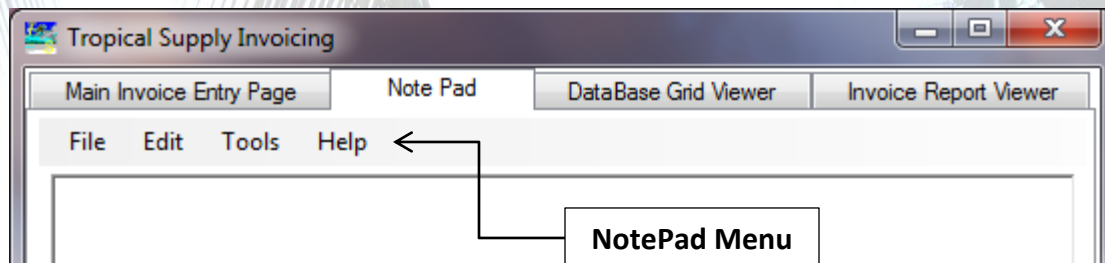
NOTE PAD

This application contains its own NotePad. NotePad is a richtext, text editor with many of the basic features of a word processor.

Please note that there are 'Tabs' located below the menu options and navigation bar, as shown below:



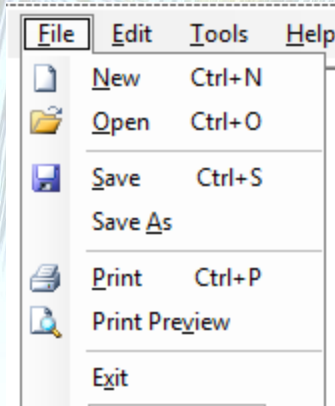
To access the NotePad, simply click on the 'NotePad' tab. When the NotePad tab is clicked, the tabs move up and hide the Menu and Nav Bar that is used with the 'Main Entry Invoice', as shown below:



NotePad has its own menu, as shown above. When a User clicks off of NotePad by clicking on any other tab, the top Menu and Nav Bar reappears. And, the NotePad menu disappears. Since the Menu and Nav Bar does not control any NotePad functions, it is important to hide the Menu and Nav Bar, while using the NotePad, so as to prevent any unwanted and inadvertent modifications to Main Entry Invoice data.

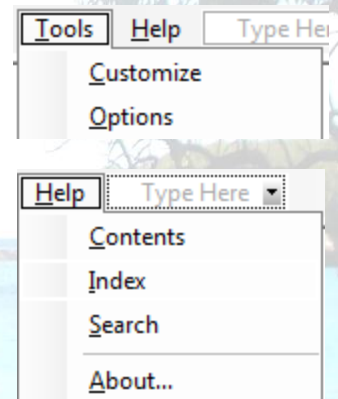
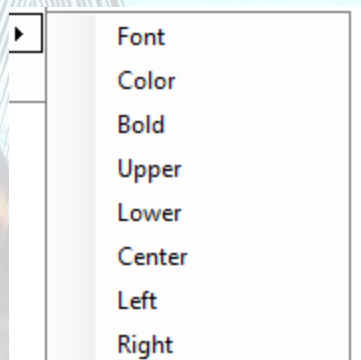
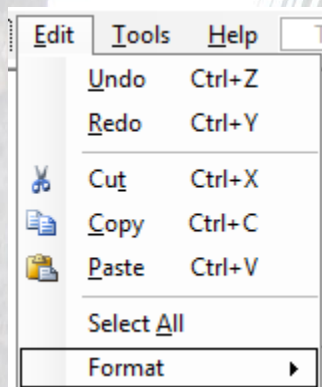
NotePad Menu Options

NotePad contains the same menu options as most word processors. Please note the keyboard shortcuts available under File >



The Edit >
Edit > Format >
Tools >
Help >

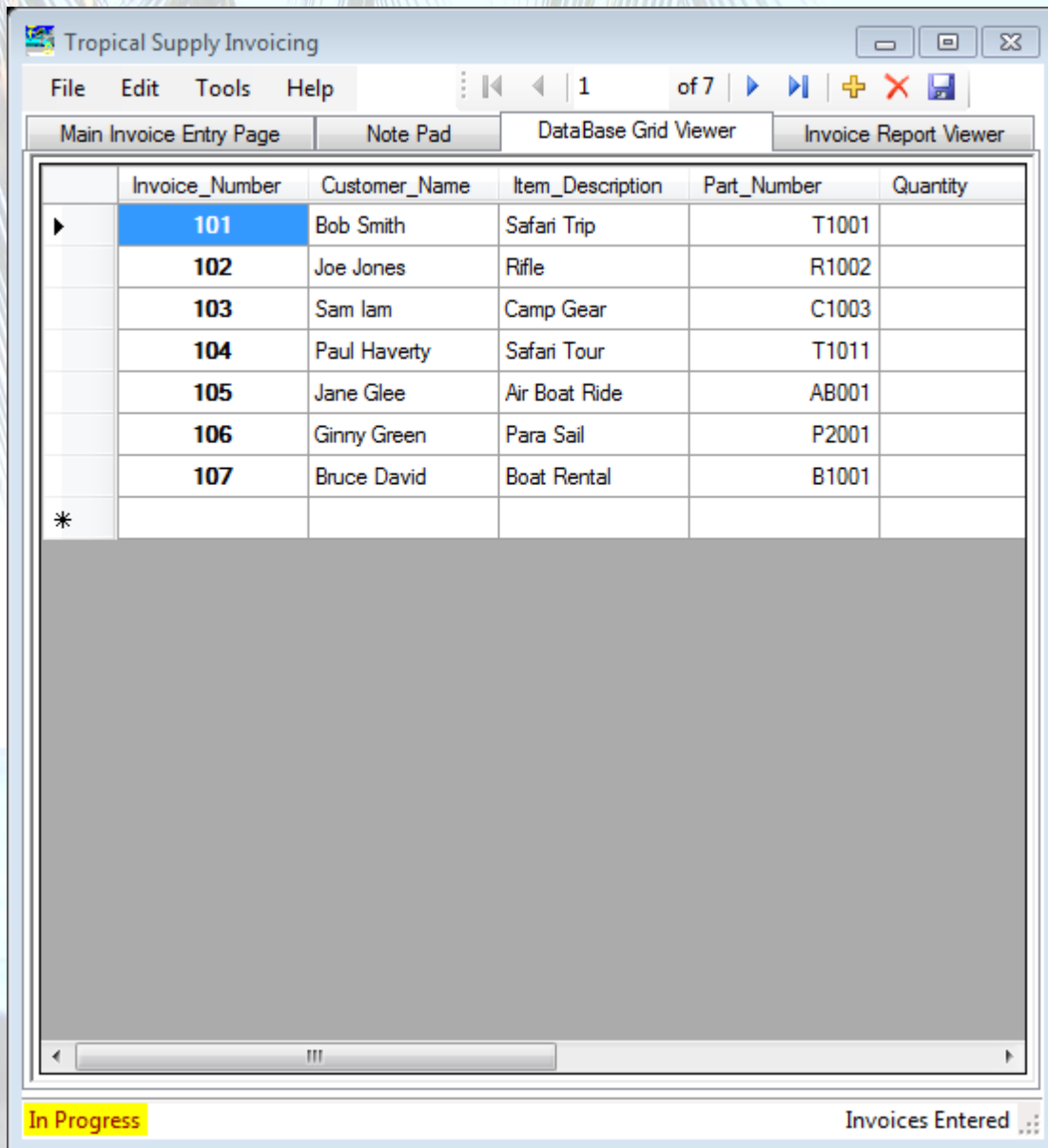
Menu options are as shown below:



DATABASE GRID VIEWER

This Invoice Application is linked to an Access database. Access is a database software program by Microsoft.

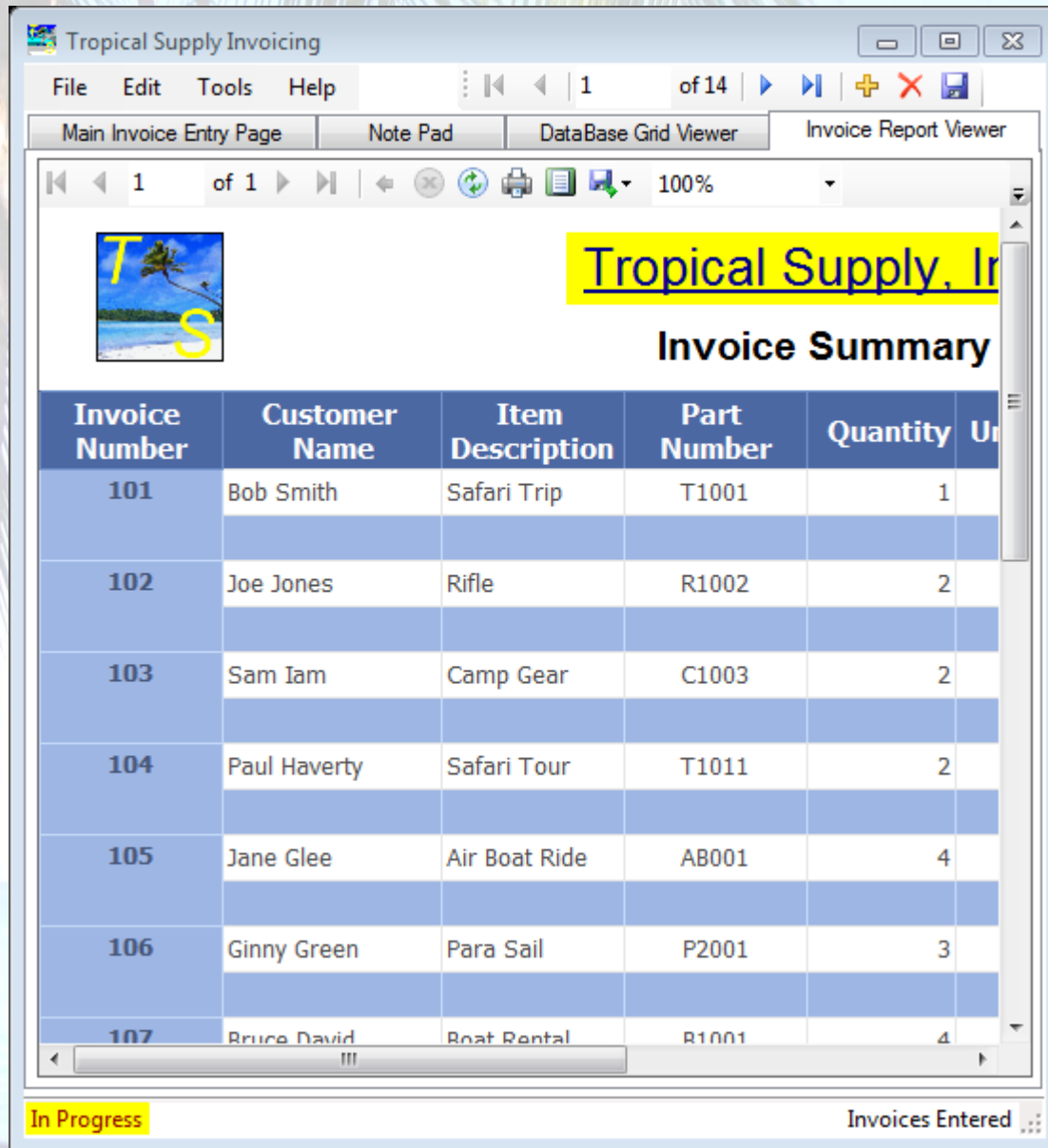
The User may view the invoice records stored in the database simply by clicking the 'DataBase Grid Viewer' tab. Below is a depiction of the DataBase Grid Viewer:



Use the scroll at the bottom of the viewer to see all data fields.

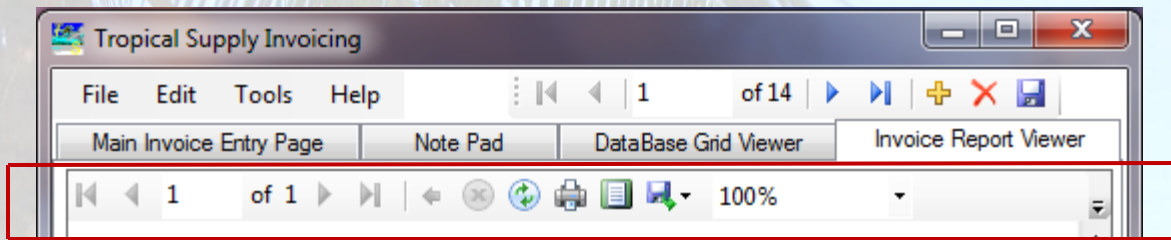
INVOICE REPORT VIEWER

In addition to viewing the data saved in the database, a User may automatically generate attractive reports and print them. The Report Viewer is depicted below:

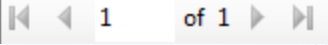







Use the scroll bar at the bottom and right side to view the report, or change the view size from 100% to 'Whole Page' to view the entire report.

As you can see, the report viewer has built in menu options and navigation bar:



Report Viewer Menu and Nav Bar

Use navigational arrows  to page thru a multi-page report. Use the refresh  icon to add newly entered invoices to the report. Print  report. Show print  layout. Save  or save as. Change viewing size to a percentage  100% or to whole page view.

Report Viewer Reports

A typical invoice report is shown below. This example is formatted in landscape view. However, reports may be formatted as desired.



Tropical Supply, Inc.

Invoice Summary



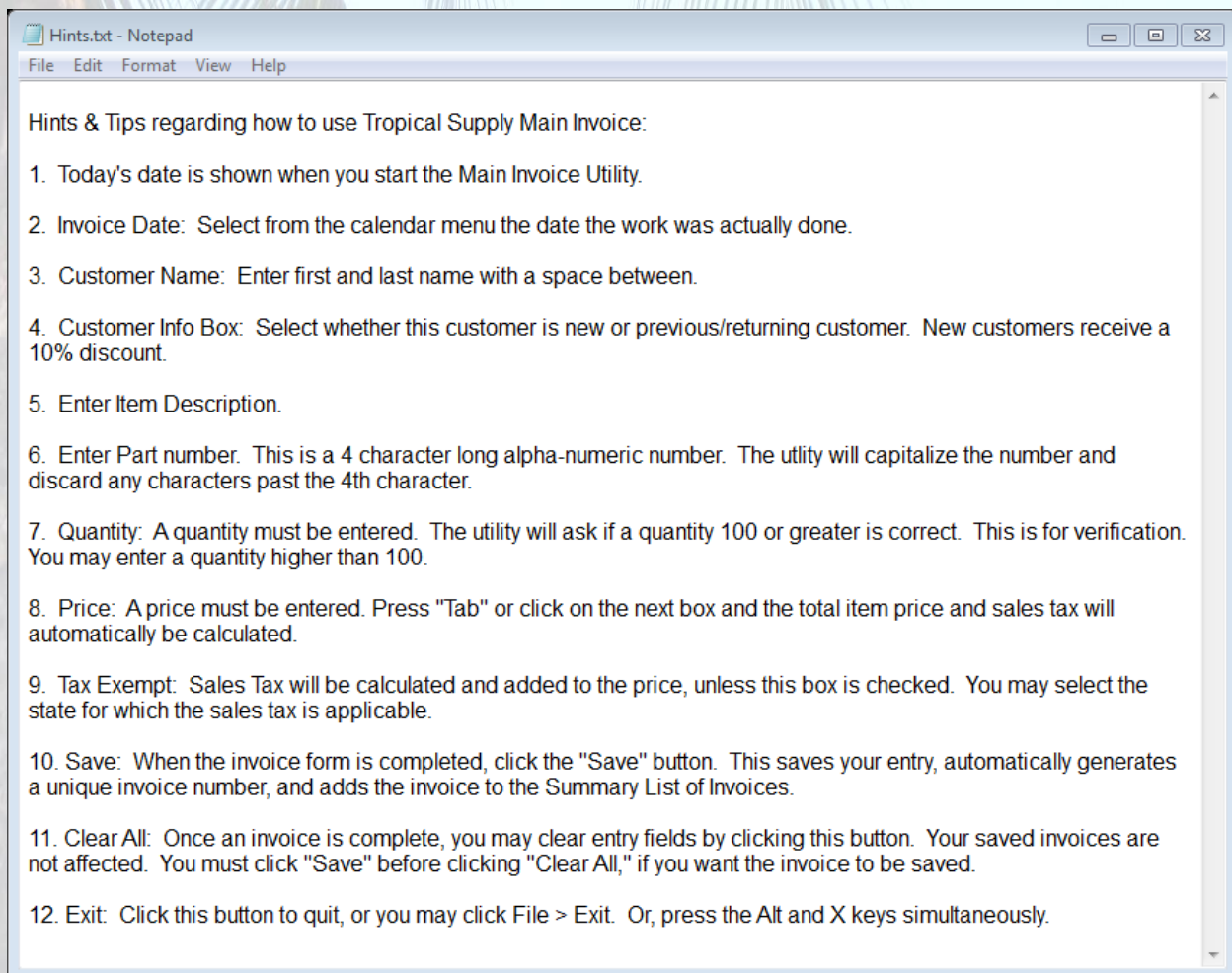
Invoice Number	Customer Name	Item Description	Part Number	Quantity	Unit Price	Sub Total Invoice	Tax	Total
101	Bob Smith	Safari Trip	T1001	1	\$ 100.00	\$ 100.00	\$ 7.00	\$ 107.00
102	Joe Jones	Rifle	R1002	2	\$ 200.00	\$ 400.00	\$ 28.00	\$ 428.00
103	Jane Smith	Swim Suit	SW101	2	\$ 60.00	\$ 120.00	\$ 8.40	\$ 128.40
104	Bruce Allan	Para Sail	P2002	3	\$ 165.00	\$ 495.00	\$ 34.65	\$ 529.65
105	Nancy Drew	Body Bag	BB010	4	\$ 110.00	\$ 440.00	\$ 30.80	\$ 470.80
106	Tim Street	Ammunition	A3011	2	\$ 50.00	\$ 100.00	\$ 7.00	\$ 107.00
107	Jack Wong	Scub Lesson	SC410	2	\$ 250.00	\$ 500.00	\$ 35.00	\$ 535.00
108	Herb Alpert	Safari Trip	T1001	4	\$ 100.00	\$ 400.00	\$ 28.00	\$ 428.00
Grand Total								\$ 2,733.85

FAQ

You may access 'Hints and Tips' and 'About' under the Help menu option in the top Menu and Nav Bar, as shown on page 9, under Menu Options.

Hints and Tips

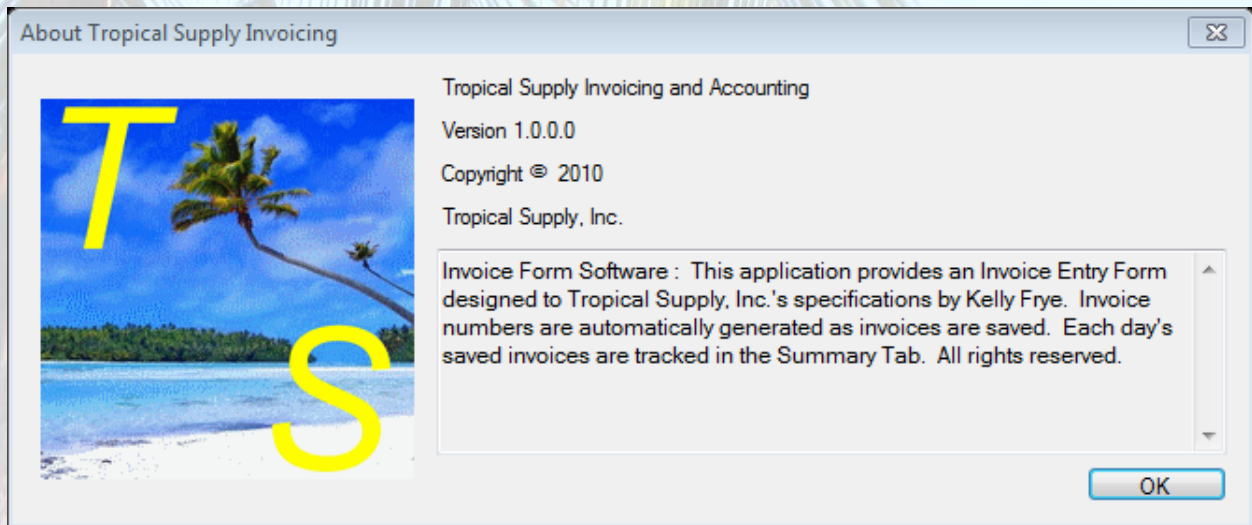
Under the top Menu and Nav Bar, Help > Hints and Tips, the following window will appear:



As shown on page 9, 'Hints and Tips' is accessible by pressing F1. Included is information similar to that narrated in previous pages, herein.

About

As shown on page 9, the 'About' window is accessed from the top Menu and Nav Bar: Help > About. The 'About' window provides basic information regarding the release version of this software:



Program Developer's Code Reference

DIVISIONS OF CODE REGIONS

```
- Public Class TropicalMainInvKF
+ Main Invoice Variables
+ Form Load
+ Main Entry Data Field Controls
+ Main Entry Button Controls
+ Main Entry Menu Options Controls
+ Navigation Bar Sub Routines
+ NotePad Controls
End Class
```

PUBLIC VARIABLE DECLARATIONS

```
#Region "Main Invoice Variables"
```

```
Public invFormLabels As New Collection()  
Public ctrlFormLabels As Control  
Public Const arrSIZE As Integer = 50  
Public arrInvNum(arrSIZE) As String  
Public arrLName(arrSIZE) As String  
Public arrFName(arrSIZE) As String  
Public arrInvAmt(arrSIZE) As Decimal  
Public GrandTotalSales As Decimal  
Public Grand_TotalSales As Decimal  
Public counter As Integer = 0  
Public TotalInvoices As Integer = 0  
Public invDate As String
```

```
#End Region
```

FORM LOAD

```
#Region "Form Load"
```

```
Private Sub TropicalMainInvKF_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Me.Shown  
    lblInvDateOutput.Text = Today.ToString("d")  
    txtInvNumbr.Focus()
```

```
With invFormLabels ' Build a collection of all Labels on the Form
```

```
    .Add(lblCustName)  
    .Add(lblGrdTotInvs)  
    .Add(lblInvDate)  
    .Add(lblItmDescrpt)  
    .Add(lblInvDate)  
    .Add(lblMainInvTitle)  
    .Add(lblPartNum)  
    .Add(lblQty)  
    .Add(lblSysDate)  
    .Add(lblTax)  
    .Add(lblTotPrice)  
    .Add(lblUntPrice)
```

```
End With
```

```
End Sub  
#End Region
```

MAIN ENTRY DATA FIELD CONTROLS

```
#Region "Main Entry Data Field Controls"
```

```
Private Sub txtInvNumbr_Leave(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtInvNumbr.Leave
    lblStatusInProgress.Text = "In Progress"
    lblStatusInProgress.BackColor = Color.Yellow
    lblStatusInProgress.ForeColor = Color.DarkRed
End Sub
```

```
Private Sub txtCustName_Leave(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtCustName.Leave
    If txtCustName.Text.Contains(" ") = False Then
        MessageBox.Show("Customer must have First & Last name", "Customer Name
Requirement", MessageBoxButtons.OK, MessageBoxIcon.Error)
        txtCustName.Focus()
    End If
End Sub
```

```
Private Sub txtPartNumbr_Leave(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPartNumbr.Leave
    Dim Response As Integer

    If txtPartNumbr.Text = "" Then
        Response = MessageBox.Show("The 'Part Num' field is blank!" &
ControlChars.CrLf & "Do you wish to enter Part Num, Yes? Press No to continue.", "Blank
Field", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button1)
        If Response = Windows.Forms.DialogResult.Yes Then txtPartNumbr.Focus()
        txtPartNumbr.SelectAll()

    ElseIf txtPartNumbr.Text.Length() < 4 Then
        MessageBox.Show("Part number must be at least 4 characters." &
ControlChars.CrLf & "Please re-enter Part Number. Thank You!", "Entry Error",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        txtPartNumbr.Focus()
        txtPartNumbr.SelectAll()
    Else
        txtPartNumbr.Text = txtPartNumbr.Text.ToUpper().Trim().Substring(0, 4)
    End If
End Sub
```

```
Private Sub txtQtyInput_Leave(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtQtyInput.Leave
    Dim Response As Integer

    If txtQtyInput.Text = "" Then
        MessageBox.Show("Please enter a Quantity!", "Empty Field",
MessageBoxButtons.OK)
        txtQtyInput.Focus()
        txtQtyInput.SelectAll()
    End If

    Dim intNum As Double
```

```

If Double.TryParse(txtQtyInput.Text, intNum) Then
    If Cdbl(txtQtyInput.Text) < 1 Then
        MessageBox.Show("please enter a Quantity equal to or greater than 1!",
            "Data Entry", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        txtQtyInput.Focus()
        txtQtyInput.SelectAll()
    End If
    If Cdbl(txtQtyInput.Text) > 100 Then ' Out of Range test
        Beep()
        Response = MessageBox.Show("Is Quantity too High?", "Out of Range",
            MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button1)
        If Response = Windows.Forms.DialogResult.Yes Then
            txtQtyInput.Focus()
        End If
        txtQtyInput.SelectAll()
    End If
    txtQtyInput.Text = FormatNumber(txtQtyInput.Text, 0)
    txtQtyInput.SelectAll()
End If
End Sub

Private Sub txtUnitPrice_Leave(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtUnitPrice.Leave
    Dim Response As Integer

    If txtUnitPrice.Text = "" Then
        MessageBox.Show("please enter a Price!", "Data Entry", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation)
        txtUnitPrice.Focus()
        txtUnitPrice.SelectAll()
    End If

    Dim intNum As Double
    If Double.TryParse(txtUnitPrice.Text, intNum) Then

        If Cdbl(txtUnitPrice.Text) < 1 Or Cdbl(txtUnitPrice.Text) > 1000 Then ' Out
of Range test
            Beep()
            Response = MessageBox.Show("The normal Price Range is between $1 and
$1000?" & ControlChars.CrLf & "Do you need to re-enter your number?", "Out of Range",
            MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button1)
            If Response = Windows.Forms.DialogResult.Yes Then
                txtUnitPrice.Focus()
            Else
                txtUnitPrice.Text = FormatCurrency(txtUnitPrice.Text)
            End If
            txtUnitPrice.SelectAll()
        Else
            txtUnitPrice.Text = FormatCurrency(txtUnitPrice.Text)
            txtUnitPrice.SelectAll()
        End If
        txtUnitPrice.Text = FormatCurrency(txtUnitPrice.Text)
        txtUnitPrice.SelectAll()
    End If
End Sub

#End Region

```

MAIN ENTRY BUTTON CONTROLS

```
#Region "Main Entry Button Controls"
```

```
    Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles btnNext.Click
```

```
        BindingNavigatorAddNewItem.PerformClick()
```

```
    End Sub
```

```
    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles btnSave.Click
```

```
        TropicalInvoicesBindingNavigatorSaveItem.PerformClick()
```

```
    End Sub
```

```
    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles btnExit.Click
```

```
        MessageBox.Show("Click OK if you want to close Invoice")
```

```
        Me.Close()
```

```
    End Sub
```

```
#End Region
```

MAIN ENTRY MENU OPTIONS CONTROLS

```
#Region "Main Entry Menu Options Controls"
```

```
Private Sub mnuEditFonts_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mnuEditFonts.Click  
    If fdDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then  
        For Each Me.ctrlFormLabels In invFormLabels  
            ctrlFormLabels.Font = fdDialog1.Font  
        Next  
    End If  
End Sub
```

```
Private Sub mnuEditColor_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mnuEditColor.Click  
    ColorDialog1.ShowDialog()  
    For Each Me.ctrlFormLabels In invFormLabels  
        ctrlFormLabels.ForeColor = ColorDialog1.Color  
    Next  
End Sub
```

```
Private Sub mnuFontsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mnuFontsToolStripMenuItem.Click  
    mnuEditFonts.PerformClick()  
End Sub
```

```
Private Sub mnuColorToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mnuColorToolStripMenuItem.Click  
    mnuEditColor.PerformClick()  
End Sub
```

```
Private Sub mnuToolsResetPassword_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mnuToolsResetPassword.Click  
    Dim FileName As System.IO.StreamWriter  
    Dim InputLine As String  
  
    InputLine = InputBox("Enter New Password", "Reset Password")  
    If InputLine = "" Then  
        Exit Sub  
    ElseIf InputLine <> "" Then  
        FileName = File.CreateText("PassWords.txt")  
        FileName.WriteLine(InputLine)  
        FileName.Close()  
    End If  
End Sub
```

```
Private Sub mnuHelpHintsAndTips_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles mnuHelpHintsAndTips.Click  
    Process.Start("NotePad", "Hints.txt")  
End Sub
```

```
Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As  
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked  
    Process.Start("http://www.fasab.gov/accepted.html")
```

```

End Sub

Private Sub mnuFileExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFileExit.Click
    Me.Close()
End Sub

Private Sub mnuHelpAbout_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuHelpAbout.Click
    AboutBoxTropical.ShowDialog()
End Sub

#End Region

```

NAVIGATION BAR SUB ROUTINES

```

#Region "Navigation Bar Sub Routines"
Private Sub TropicalInvoicesBindingNavigatorSaveItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TropicalInvoicesBindingNavigatorSaveItem.Click
    counter += 1 ' This is a Counter, which adds one for each pass (invoice saved)
    TotalInvoices = counter

    lblStatusInProgress.Text = "FINISHED"
    lblStatusInProgress.BackColor = Color.Black
    lblStatusInProgress.ForeColor = Color.White
    lblStatusInvoicesEntered.Text = "Invoices Entered " & TotalInvoices

    Me.Validate()
    Me.TropicalInvoicesBindingSource.EndEdit()
    Me.TableAdapterManager.UpdateAll(Me.TropicalInvoicesDataSet)

End Sub

Private Sub TropicalMainInvKF_Load_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    'TODO: This line of code loads data into the
'TropicalInvoicesDataSet.TropicalInvoices' table. You can move, or remove it, as needed.
    Me.TropicalInvoicesTableAdapter.Fill(Me.TropicalInvoicesDataSet.TropicalInvoices)
    txtUnitPrice.Text = FormatCurrency(txtUnitPrice.Text, 0)
    lblCalcTotal.Text = FormatCurrency(lblCalcTotal.Text)
    lblCalcTax.Text = FormatCurrency(lblCalcTax.Text)
    lblGrdTotOutput.Text = FormatCurrency(lblGrdTotOutput.Text)
    Me.ReportViewer1.RefreshReport()

End Sub

Private Sub BindingNavigatorAddNewItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BindingNavigatorAddNewItem.Click
    txtInvNumbr.Focus()
    lblStatusInProgress.Text = "In Progress"
    lblStatusInProgress.BackColor = Color.Yellow
    lblStatusInProgress.ForeColor = Color.DarkRed
End Sub

```

```

        Private Sub BindingNavigatorMoveNextItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BindingNavigatorMoveNextItem.Click
            txtUnitPrice.Text = FormatCurrency(txtUnitPrice.Text, 0)
            lblCalcTotal.Text = FormatCurrency(lblCalcTotal.Text)
            lblCalcTax.Text = FormatCurrency(lblCalcTax.Text)
            lblGrdTotOutput.Text = FormatCurrency(lblGrdTotOutput.Text)
        End Sub
#End Region

```

NOTEPAD CONTROLS

```

#Region "NotePad Controls"

```

```

        Private Sub TabControl1_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TabControl1.SelectedIndexChanged
            If TabControl1.SelectedTab Is tabNotePad Then
                TabControl1.Top = 0
            Else
                TabControl1.Top = 25
            End If
        End Sub

```

```

        Private Sub mnuNotePadFileSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadFileSave.Click
            Dim outputFileName As System.IO.StreamWriter
            SaveFileDialog1.Filter() = "Rich Text File (*.rtf)|*.rtf|Text File
(*.txt)|*.txt|All files (*.*)|*.*"
            Try
                SaveFileDialog1.ShowDialog()
                outputFileName = File.CreateText(SaveFileDialog1.FileName)
                outputFileName.Write(RichTextBox1.Rtf)
                outputFileName.Close()
            Catch
                MessageBox.Show("Sorry, the file cannot be created.")
            End Try
        End Sub

```

```

        Private Sub mnuNotePadFileOpen_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadFileOpen.Click
            Dim fileName = String.Empty ' Define and Clear Contents of text box
            Dim inputFile As StreamReader
            Dim openResult As Integer
            OpenFileDialog1.Filter = "Rich Text files (*.rtf)|*.rtf|Text files
(*.txt)|*.txt|All files (*.*)|*.*"
            openResult = OpenFileDialog1.ShowDialog()
            If openResult = DialogResult.Cancel Then
                Exit Sub
            Else
                fileName = OpenFileDialog1.FileName
                inputFile = File.OpenText(fileName)
                RichTextBox1.Rtf = inputFile.ReadToEnd
            End If

```

```

End Sub

Private Sub mnuNotePadEditFormatColor_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditFormatColor.Click
    ColorDialog1.ShowDialog()
    RichTextBox1.SelectionColor = ColorDialog1.Color
End Sub

Private Sub mnuNotePadEditFormatFont_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditFormatFont.Click
    fdDialog1.ShowDialog()
    RichTextBox1.SelectionFont = fdDialog1.Font
End Sub

Private Sub mnuNotePadEditFormatBold_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditFormatBold.Click
    RichTextBox1.SelectionFont = New Font(RichTextBox1.SelectionFont, FontStyle.Bold)
End Sub

Private Sub mnuNotePadEditFormatUpper_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditFormatUpper.Click
    RichTextBox1.SelectedText = RichTextBox1.SelectedText.ToUpper
End Sub

Private Sub mnuNotePadEditFormatLower_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditFormatLower.Click
    RichTextBox1.SelectedText = RichTextBox1.SelectedText.ToLower
End Sub

Private Sub mnuNotePadEditFormatCenter_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles mnuNotePadEditFormatCenter.Click
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Center
End Sub

Private Sub mnuNotePadEditFormatLeft_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditFormatLeft.Click
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Left
End Sub

Private Sub mnuNotePadEditFormatRight_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditFormatRight.Click
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Right
End Sub

Private Sub mnuNotePadEditCopy_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditCopy.Click
    RichTextBox1.Copy()
End Sub

Private Sub mnuNotePadEditUndo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditUndo.Click
    RichTextBox1.Undo()
End Sub

Private Sub mnuNotePadEditRedo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditRedo.Click
    RichTextBox1.Redo()

```

```

End Sub

Private Sub mnuNotePadEditCut_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditCut.Click
    RichTextBox1.Cut()
End Sub

Private Sub mnuNotePadEditPaste_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditPaste.Click
    RichTextBox1.Paste()
End Sub

Private Sub mnuNotePadEditSelectAll_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadEditSelectAll.Click
    RichTextBox1.SelectAll()
End Sub

Private Sub mnuNotePadFilePrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNotePadFilePrint.Click
    Dim printResult As Integer
    printResult = PrintDialog1.ShowDialog()
    If printResult = DialogResult.Cancel Then
        Exit Sub
    Else
        PrintDocument1.PrinterSettings = PrintDialog1.PrinterSettings
        PrintDocument1.Print()
    End If
End Sub

Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    e.Graphics.DrawString(RichTextBox1.Text, New Font("Times New Roman", 12,
FontStyle.Regular), Brushes.Black, 10, 10)
End Sub

#End Region

```

SPLASH SCREEN CODE

```
Public NotInheritable Class SplashScreenTropical

    'TODO: This form can easily be set as the splash screen for the application by going
    to the "Application" tab
    ' of the Project Designer ("Properties" under the "Project" menu).

    Private Sub SplashScreenTropical_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        'Set up the dialog text at runtime according to the application's assembly
        information.

        'TODO: Customize the application's assembly information in the "Application" pane
        of the project
        ' properties dialog (under the "Project" menu).

        'Application title
        If My.Application.Info.Title <> "" Then
            ApplicationTitle.Text = My.Application.Info.Title
        Else
            'If the application title is missing, use the application name, without the
            extension
            ApplicationTitle.Text =
System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.AssemblyName)
        End If

        'Format the version information using the text set into the Version control at
        design time as the
        ' formatting string. This allows for effective localization if desired.
        ' Build and revision information could be included by using the following code
        and changing the
        ' Version control's designtime text to "Version {0}.{1:00}.{2}.{3}" or something
        similar. See
        ' String.Format() in Help for more information.
        '
        ' Version.Text = System.String.Format(Version.Text,
My.Application.Info.Version.Major, My.Application.Info.Version.Minor,
My.Application.Info.Version.Build, My.Application.Info.Version.Revision)

        Version.Text = System.String.Format(Version.Text,
My.Application.Info.Version.Major, My.Application.Info.Version.Minor)

        'Copyright info
        Copyright.Text = My.Application.Info.Copyright
    End Sub

End Class
```

LOGIN FORM CODE

```
Imports System.IO
Public Class LoginFormTropicalInv

    ' TODO: Insert code to perform custom authentication using the provided username and
    password
    ' (See http://go.microsoft.com/fwlink/?LinkId=35339).
    ' The custom principal can then be attached to the current thread's principal as
    follows:
    '     My.User.CurrentPrincipal = CustomPrincipal
    ' where CustomPrincipal is the IPrincipal implementation used to perform
    authentication.
    ' Subsequently, My.User will return identity information encapsulated in the
    CustomPrincipal object
    ' such as the username, display name, etc.

    Dim pwcounter As Integer = 0
    Private Sub OK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles OK.Click
        ' Create an instance of StreamReader to read from a file:
        Dim inputFile As StreamReader = File.OpenText("PassWords.txt")
        ' PassWords.txt is stored in the Debug folder, inside the Bin folder
        Dim LinePW As String ' This for each line that is read
        LinePW = inputFile.ReadLine()

        inputFile.Close() ' Close all files when done.
        If txtBxPassWord.Text = LinePW And txtBxUserName.Text = "user" Then
            TropicalMainInvKF.Show()
            Me.Close()
        Else
            pwcounter += 1
            MessageBox.Show("Wrong User Name or Password", "Try Again",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
            Me.Show()
            If pwcounter = 3 Then
                Me.Close()
            End If
        End If
    End Sub

    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Cancel.Click
        Me.Close()
    End Sub

    'Back Door
    Private Sub picBxBackDoor_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picBxBackDoor.Click
        TropicalMainInvKF.Show()
        Me.Close()
    End Sub
End Class
```

ABOUT BOX CODE

```
Public NotInheritable Class AboutBoxTropical

    Private Sub AboutBoxTropical_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        ' Set the title of the form.
        Dim ApplicationTitle As String
        If My.Application.Info.Title <> "" Then
            ApplicationTitle = My.Application.Info.Title
        Else
            ApplicationTitle =
System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.AssemblyName)
        End If
        Me.Text = String.Format("About {0}", ApplicationTitle)
        ' Initialize all of the text displayed on the About Box.
        ' TODO: Customize the application's assembly information in the "Application"
pane of the project
        ' properties dialog (under the "Project" menu).
        Me.LabelProductName.Text = My.Application.Info.ProductName
        Me.LabelVersion.Text = String.Format("Version {0}",
My.Application.Info.Version.ToString)
        Me.LabelCopyright.Text = My.Application.Info.Copyright
        Me.LabelCompanyName.Text = My.Application.Info.CompanyName
        Me.TextBoxAboutDescription.Text = My.Application.Info.Description
    End Sub

    Private Sub OKButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OKButton.Click
        Me.Close()
    End Sub

End Class
```

DATA BASE SQL CODE

TableAdapter Configuration Wizard

Enter a SQL Statement

The TableAdapter uses the data returned by this statement to fill its DataTable.

Type your SQL statement or use the Query Builder to construct it. What data should be loaded into the table?

What data should be loaded into the table?

```
SELECT Invoice_Number, Customer_Name, Item_Description, Part_Number, Quantity, Unit_Price,
Quantity*Unit_Price AS SubTotalInvoice, Quantity*Unit_Price*0.07 AS Tax, Quantity*Unit_Price
+ Quantity*Unit_Price*0.07 AS Total
FROM TropicalInvoices
```

Advanced Options... Query Builder...

< Previous Next > Finish Cancel