



## Reference Guide

**Ref No : RG-00106**

Date : 11/05/2005

Product : SQL Accounting

Module : DIY Script

Description : POS - Scripting for POS Pole Display Unit -- Guidelines on how to use - Model CD7220

### Some common command for Pole Display

```
=====
Clear Screen: 0C
Display on First line: 1B 51 41 nx20 0D
Display on Second line: 1B 51 42 nx20 0D
Scroll uppper line: 1B 51 44 nx20 0D
```

To add respective events listed on below, please follow the instruction written below:

- ```
=====
```
- 1)Go to **tools**.Under **tools**, go to **SQL Control Center**.
  - 2)When SQL Control Center window is opened,expand the treelist on the left hand side and go to the branch where the script should belongs to. For example, here we assume in Cash Sales.(So we go to **Business Objects>Sales>Sales Cash Sales**)
  - 3)Right Click on the sales cash sales and choose to **add a new event**.
  - 4)Give the event a name and then select from the drop down list the event type listed as below.
  - 5)Copy the respective codes to the event and save it.
  - 6)Repeat step 4 and 5 in order to add more events. You are now ready to use it.

### OnBeforeNew Event:

```
=====
function GetGreeting: string;
var
    hourNow: word;
begin
    hourNow := HourOf(Time);
    if hourNow <12 then
        Result := 'Good Morning!';
    else if hourNow<17 then
        Result := 'Good Afternoon!';
    else
        Result := 'Good Evening!';
end;

var
    c: TCommDevice;
    str: string;
begin
    C := TCommDevice.Create(Self);
    try
        try
            c.OpenPortWithParam('COM1:9600,n,8,1');
            c.PrintFromString(#$0C);
            str := CurrentUser;
            c.PrintFromString(#$1b#$51#$44'' + GetGreeting + ' You are service by ' + str + '.
'#13);
            c.Close;
        finally
            C.Free;
        end;
    except
        end;
end.
```

Basically, this event is invoked when a user started a new invoice. The Pole will display the greeting depending on the hours and also current user login to SQL Accounting.

### OnGridColumnValueChanged Event

=====

```
var
  C: TCommDevice;
  D: TDataSet;
  FDesc, FQty, FUnitPrice, S: String;
  A, T, Value: Currency;
begin
  D := TDataSource(Self.FindComponent('dsDocDetail')).DataSet;
  C := TCommDevice.Create(Self);
  try
    try
      C.OpenPortWithParam('COM1:9600,n,8,1');
      C.PrintFromString('#$0C');
      // FDesc      := D.FindField('Description2').AsString;
      FDesc      := D.FindField('Description').AsString;
      FQty       := FormatCurr(TNumericField(D.FindField('Qty')).DisplayFormat,
D.FindField('Qty').AsCurrency)+ 'X';

      Value := D.FindField('UnitPrice').AsCurrency;
      T := Value * D.FindField('Qty').AsCurrency;
      try
        if Trim(D.FindField('Disc').AsString) <> '' then begin
          if Pos('%', D.FindField('Disc').AsString) = 0 then
            Value := (T - StrToFloat(D.FindField('Disc').AsString)) / D.FindField
('Qty').AsCurrency
          else begin
            S := Copy(D.FindField('Disc').AsString, 1, Pos('%', D.FindField
('Disc').AsString) - 1);
            A := T * (StrToFloat(S) / 100);
            Value := (T - A) / D.FindField('Qty').AsCurrency;
          end;
        end;
      finally
        FUnitPrice := FormatCurr('0.00', Value);
      end;

      C.PrintFromString('#$1B#$51#$41' + FDesc + '#13');
      C.PrintFromString('#$1B#$51#$42' + FQty + ' ' + FUnitPrice + '#13');
      C.Close;
    finally
      C.Free;
    end;
  except
  end;
end.
```

As for this event is invoked when a user started adding or deleting a new item in the invoice. The Pole will display the item description, qty, unit price on the pole display.

### OnBeforeSave Event:

=====

```
var
  A, T, P: String;
  D: TDataSet;
  C: TCommDevice;
begin
  if TPanel(Self.FindComponent('DetailControl')).Visible then begin
    C := TCommDevice.Create(Self);
    D := TDataSource(Self.FindComponent('dsDocMaster')).DataSet;
    try
      T := 'RM ' + FormatCurr('0.00',D.FindField('DocAmt').AsCurrency);

      { Please open the line 14, 15 and 21, if the database version is 59 and above ...
    }
    // P := FormatCurr('0.00', D.FindField('P_PaidAmount').AsCurrency);
    // A := FormatCurr('0.00', D.FindField('P_PaidAmount').AsCurrency - D.FindFiel
```

```

('DocAmt').AsCurrency);

C.OpenPortWithParam('COM1:9600,n,8,1');
C.PrintFromString(#$0C);
C.PrintFromString(#$1b#$51#$41'Total: ' + T + #13);

//      C.PrintFromString(#$1b#$51#$42'Pay: ' + P + ' ' + 'C:' + A + #13);
C.Close;
finally
  C.Free;
end;
end;
end.

```

This event will generates a total amount for the items recorded on the invoice and displaying it on the Pole display.

#### OnBeforeCancel Event:

```

=====
var
  c: TCommDevice;
begin
  C := TCommDevice.Create(Self);
  try
    try
      c.OpenPortWithParam('COM1:9600,n,8,1');
      c.PrintFromString(#$0C);
      c.Close;
    finally
      C.Free;
    end;
  except
  end;
end.

```

This event will clear the screen whenever an invoice is being cancelled. Whatever appears on the Pole display previously will be clear instantly.

#### OnClose Event:

```

=====
var
  c: TCommDevice;
  i: integer;
begin
  C := TCommDevice.Create(Self);
  try
    try
      c.OpenPortWithParam('COM1:9600,n,8,1');
      c.PrintFromString(#$0C);
      c.PrintFromString(#$1b#$51#$44'Thank You! Have a nice day!'#13);
      c.Close;
    finally
      C.Free;
    end;
  except
  end;
end.

```

This event will prompt customer with greeting when a transaction is completed successfully and the invoice is being closed.

#### OnOpen Event:

```

=====
var FEvent: TNotifyEvent;

procedure OnBeforeSave(Sender: TObject);
var C: TCommDevice;
    D: TDataSet;
    T: String;
begin
  if TPanel(Self.FindComponent('DetailControl')).Visible then begin
    C := TCommDevice.Create(Self);

```

```

D := TDataSource(Self.FindComponent('dsDocMaster')).DataSet;
try
  try
    T := 'RM ' + FormatCurr('0.00',D.FindField('DocAmt').AsCurrency);
    C.OpenPortWithParam('COM1:9600,n,8,1');
    C.PrintFromString(#$0C);
    C.PrintFromString(#$1b#$51#$41'Total: ' + T + #13);
    C.Close;
    // MessageDlg(Format('Total Amount: %s', [T]), mtInformation, [mbOK], 0);
  finally
    C.Free;
  end;
except
end;
end;

if FEvent <> nil then FEvent(Sender);
end;

var C: TWinControl;
    B: TcxButton;
begin
  C := TWinControl(Self.FindComponent('frDataSetButton1'));
  B := TcxButton(C.FindComponent('BtnSave'));
  if Assigned(B) then begin
    FEvent := B.OnClick;
    B.OnClick := @OnBeforeSave;
  end;
end.

```

This event will display Total Amount on the POS when the user click on Save button.