

```

//*****
/**    MACRO: REVERSE.WCM
/**    PURPOSE: Applies colors and fill style to selected text or table cells.
//*****
Application (A1; "WordPerfect"; Default; "UK")

DefinedColors:=16
DefinedPatterns:=12
WaitMessage:="Initialising"
Call(WaitMessage@)
Call(Declarations@)

                                     // Set defaults here...
TxtColor:=Clr[1; 1]                 // white
FillColor:=Clr[2; 1]                // black
PatFill:=Pat[2; 1]                  // 100% Fill
CntrTxt:=0
LockCell:=0
HeaderRow:=0
Paragraph:=0
TextBox:=1
EndOfDoc:=False
CodesOn:=False

If(?RevealCodesActive)
    RevealCodes(Off!)
    CodesOn:=True
EndIf

If(?TableInTable=0)
    InTable:=False
Else
    InTable:=True
    InParagraph:=False
    InTextBox:=False
EndIf

If(?BlockActive=0)
    If(InTable)
        SelectCell()
    Else
        SelectWord()
    EndIf
    If(?BlockActive=0) // Double check to see if something was selected
        MessageBox("No Items Selected"; "You must select the text or table cells you
would like to modify before running this macro."; IconStop!)
        Go(End@)

```

```

        EndIf
    EndIf

    StatusPrompt:="Initialising dialog..."
    Call(StatusPrompt@)

    If(InTable)
        DlgHeight:=125
    Else
        DlgHeight:=137
    EndIf
    DialogDefine("Dlg1"; 50; 50; 198; DlgHeight; 1+2+16; "Reverse Text Options")
    DialogAddGroupBox("Dlg1"; "Gb1"; 4; 4; 90; 52; "&Text Colour")
    DialogAddComboBox("Dlg1"; "Cmb1"; 14; 16; 70; 50; 16; TxtColor; 20)

    DialogAddGroupBox("Dlg1"; "Gb2"; 98; 4; 90; 52; "&Fill Style/Colour")
    DialogAddComboBox("Dlg1"; "Cmb2"; 108; 16; 70; 50; 16; PatFill; 20)
    For(Count; 1; Count<=DefinedPatterns; Count+1)
        DialogAddListItem("Dlg1"; "Cmb2"; Pat[Count;1])
    EndFor

    DialogAddComboBox("Dlg1"; "Cmb3"; 108; 37; 70; 50; 16; FillColor; 20)

    For(Count; 1; Count<=DefinedColors; Count+1)
        DialogAddListItem("Dlg1"; "Cmb1"; Clr[Count;1])
        DialogAddListItem("Dlg1"; "Cmb3"; Clr[Count;1])
    EndFor

    If(InTable)
        DialogAddGroupBox("Dlg1"; "Gb3"; 4; 60; 184; 26; "Table &Options")
        DialogAddCheckBox("Dlg1"; "Cb1"; 14; 70; 50; 11; "C&entre Text"; CntrTxt)
        DialogAddCheckBox("Dlg1"; "Cb2"; 72; 70; 50; 11; "&Lock Cell"; LockCell)
        DialogAddCheckBox("Dlg1"; "Cb3"; 130; 71; 50; 11; "&Header Row"; HeaderRow)
    Else
        DialogAddGroupBox("Dlg1"; "Gb3"; 4; 60; 184; 38; "Text &Options")
        DialogAddRadioButton("Dlg1"; "Cb1"; 14; 70; 127; 11;"Place selected text in Text
        &Box";TextBox)
        DialogAddRadioButton("Dlg1"; "Cb2"; 14; 82; 127; 11;"Apply attributes to whole
        &Paragraph";Paragraph)
    EndIf
    StatusPrompt:="Off"
    Call(StatusPrompt@)
    Display(On!)
    Call(HideWaitMessage@)
    DialogDisplay("Dlg1"; "Cmb1")
    Button:=MacroDialogResult

```

```

DialogDestroy("Dlg1")
Display(Off!)
If(Not InTable)
    If(Paragraph=1)
        InParagraph:=True
        InTextBox:=False
    Else
        InParagraph:=False
        InTextBox:=True
    EndIf
EndIf
If(Button=2)
    Go(End@)
Else
    If(Button=1)
        WaitMessage:="Applying attributes"
        Call(KillWaitMessage@)
        Call(WaitMessage@)
        // FIRST DO THE SETUP
        If(InTextBox)
            StatusPrompt:="Creating in line text box..."
            Call(StatusPrompt@)
            EditCut()
            BookmarkCreate("Heading Macro Place Holder")
            BoxCreate(TextBox!)
            TextBoxNumber:=?BoxNumber
            BoxAttachTo(Character!)
            BoxVerticalAlignment(Baseline!)
            BoxContentEdit()
            EditPaste()
            PosDocTop()
        Else
            If(InParagraph)
                SelectParagraph()
                PosSelectBottom()
                If(?RightCode=0 And ?RightChar="")
                    EndOfDoc:=True
                    SelectOff()
                    HardReturn()
                    SelectParagraphPrevious()
                EndIf
            EndIf EndIf
            // NOW SET THE COLOR FOR THE TEXT
            If(Not InTextBox)
                StatusPrompt:="Setting text colour..."
                Call(StatusPrompt@)
            EndIf
        EndIf
    EndIf
EndIf

```

```

EndIf
For(SelClr; 1; Clr[SelClr;1]<>TxtColor; SelClr+1)
EndFor
TextColor(;Clr[SelClr;2]; Clr[SelClr;3]; Clr[SelClr;4])
If(InTextBox)
    SubstructureExit ()
    BoxWidth(AutoWidth!)
    BoxBorder(0)
EndIf
//      NOW SET THE FILL STYLE
If(Not InTextBox)
    StatusPrompt:="Setting fill style..."
    Call(StatusPrompt@)
EndIf
For(SelPat; 1; Pat[SelPat;1]<>PatFill; SelPat+1)
EndFor
If(InTable)
    TableCellFillStyle(Pat[SelPat;2])
Else
    If(InParagraph)
        TextBorderCreate(SpacingOnly!; Pat[SelPat;2]; ParagraphBorder!)
    Else
        If(InTextBox)
            BorderSetSpacing(Yes!)
            BorderOutsideSpacing(0;0;0;0)
            BoxFill(Pat[SelPat;2])
        EndIf EndIf EndIf
        //      NOW SET THE FILL COLORS
        If(Not InTextBox)
            StatusPrompt:="Setting fill colours..."
            Call(StatusPrompt@)
        EndIf
        For(SelfClr; 1; Clr[SelfClr;1]<>FillColor; SelfClr+1)
        EndFor
        If(Pat[SelPat;2]<10)
            FillPercent:=(Pat[SelPat;2]+1)*10
            If(InTable)
                TableCellFillColors(; Clr[SelfClr;2]; Clr[SelfClr;3]; Clr[SelfClr;4];
                FillPercent;; 255; 255; 255; 100)
            Else
                If(InParagraph Or InTextBox)
                    FillColors(; Clr[SelfClr;2]; Clr[SelfClr;3]; Clr[SelfClr;4]; FillPercent;;
                    255; 255; 255; 100)
                EndIf EndIf
            EndIf EndIf EndIf
        EndIf
    EndIf
EndIf

```

```

//      NOW SET THE OTHER OPTIONS
If(InTable)
    If(CntrTxt=1)
        TableCellJustification(Center!)
    EndIf
    If(LockCell=1)
        TableCellLock(Yes!)
    EndIf
    If(HeaderRow=1)
        TableHeader(Yes!)
    EndIf
EndIf
//      NOW DO WHATEVER CLEANUP IS NECESSARY
If(InParagraph)
    TextBorderEnd(Save!)
    If(EndOfDoc)
        PosDocBottom()
        PosLineUp()
        PosLineEnd()
        DeleteCharNext()
    EndIf
    PosWordNext()
Else
If(InTextBox)
    BoxUpdateDisplay ()
    BoxEnd(Save!)
    BoxEdit(TextBoxNumber)
    EditCut()
    BookmarkFind("Heading Macro Place Holder")
    BookmarkDelete("Heading Macro Place Holder")
    EditPaste()
    CloseGraphicsControlBar()
EndIf EndIf
If(CodesOn)
    RevealCodes(On!)
EndIf
SelectMode(Off!)
EndIf

Label(End@)
StatusPrompt:="Off"
Call(StatusPrompt@)
Call(KillWaitMessage@)
Quit

```

```

//*****

```

```

/**  ROUTINE: StatusPrompt@
/**  INPUT VARIABLES: StatusPrompt
/**  OUTPUT VARIABLES: None
/**  DESCRIPTION: Displays a macro status prompt if not in a substructure.
/**  *****
Label(StatusPrompt@)
If(Not ?Substructure)
    If(StatusPrompt="Off")
        MacroStatusPrompt(Off!)
    Else
        MacroStatusPrompt(On!; StatusPrompt)
    EndIf
EndIf
Return
/**  *****

/**  *****
/**  ROUTINE: WaitMessage@
/**  INPUT VARIABLES: WaitMessage
/**  OUTPUT VARIABLES: WaitMessageResult
/**  DESCRIPTION: Presents a Please Wait message to user. Call KillWaitMessage to turn
it off.
/**  *****
Label(WaitMessage@)
WaitTitle:="WAIT PROMPT"
WaitMessageWidth:=61
DialogDefine(WaitTitle; 50; 50; WaitMessageWidth+22; 43; 16+256; WaitTitle)
DialogAddText(WaitTitle; "T1"; 8; 8; 46; 10; 1; "Please Wait...")
DialogAddText(WaitTitle; "T2"; 8; 19; WaitMessageWidth; 10; 1; WaitMessage)
DialogDisplay(WaitTitle; 1; WaitDlgCallBack@)
Return

Label(WaitDlgCallBack@)
Return
/**  *****

/**  *****
/**  ROUTINE NAME: HideWaitMessage@
/**  INPUT VARIABLES: WaitDlgId
/**  OUTPUT VARIABLES: None
/**  DESCRIPTION: Hides a previously defined wait message dialog.
/**  *****
Label(HideWaitMessage@)
DialogUndisplay(WaitTitle; 1)
Return
/**  *****

```

```

//*****
//*   ROUTINE: KillWaitMessage@
//*   INPUT VARIABLES: none
//*   OUTPUT VARIABLES: none
//*   DESCRIPTION: Turns off wait message.
//*****
Label(KillWaitMessage@)
If(Exists(WaitTitle)=1)
    DialogDestroy(WaitTitle)
    Discard(WaitTitle)
EndIf
Return
//*****

```

```

//*****
//*   ROUTINE: Declarations@
//*   INPUT VARIABLES: None
//*   OUTPUT VARIABLES: Clr[]; Pat[]
//*   DESCRIPTION: Initializes arrays for color and pattern.
//*****

```

```

Label(Declarations@)
Declare(Clr[DefinedColors;4])
Clr[1;1]="White"
Clr[1;2]:=255
Clr[1;3]:=255
Clr[1;4]:=255

```

```

Clr[2;1]="Black"
Clr[2;2]:=0
Clr[2;3]:=0
Clr[2;4]:=0

```

```

Clr[3;1]="Dark Blue"
Clr[3;2]:=0
Clr[3;3]:=0
Clr[3;4]:=127

```

```

Clr[4;1]="Dark Green"
Clr[4;2]:=0
Clr[4;3]:=127
Clr[4;4]:=0

```

```

Clr[5;1]="Dark Cyan"
Clr[5;2]:=0
Clr[5;3]:=127

```

Clr[5;4]:=127

Clr[6;1]:="Dark Red"

Clr[6;2]:=127

Clr[6;3]:=0

Clr[6;4]:=0

Clr[7;1]:="Dark Magenta"

Clr[7;2]:=127

Clr[7;3]:=0

Clr[7;4]:=127

Clr[8;1]:="Dark Grey"

Clr[8;2]:=127

Clr[8;3]:=127

Clr[8;4]:=127

Clr[9;1]:="Brown"

Clr[9;2]:=127

Clr[9;3]:=127

Clr[9;4]:=0

Clr[10;1]:="Light Grey"

Clr[10;2]:=192

Clr[10;3]:=192

Clr[10;4]:=192

Clr[11;1]:="Light Blue"

Clr[11;2]:=0

Clr[11;3]:=0

Clr[11;4]:=255

Clr[12;1]:="Light Green"

Clr[12;2]:=0

Clr[12;3]:=255

Clr[12;4]:=0

Clr[13;1]:="Light Cyan"

Clr[13;2]:=0

Clr[13;3]:=255

Clr[13;4]:=255

Clr[14;1]:="Light Red"

Clr[14;2]:=255

Clr[14;3]:=0

Clr[14;4]:=0

Clr[15;1]:="Light Magenta"

Clr[15;2]:=255

Clr[15;3]:=0

Clr[15;4]:=255

Clr[16;1]:="Yellow"

Clr[16;2]:=255

Clr[16;3]:=255

Clr[16;4]:=0

Declare(Pat[DefinedPatterns;2])

Pat[1;1]:="<None>"

Pat[1;2]:=127

Pat[2;1]:="100% Fill"

Pat[2;2]:=9

Pat[3;1]:="90% Fill"

Pat[3;2]:=8

Pat[4;1]:="80% Fill"

Pat[4;2]:=7

Pat[5;1]:="70% Fill"

Pat[5;2]:=6

Pat[6;1]:="60% Fill"

Pat[6;2]:=5

Pat[7;1]:="50% Fill"

Pat[7;2]:=4

Pat[8;1]:="40% Fill"

Pat[8;2]:=3

Pat[9;1]:="30% Fill"

Pat[9;2]:=2

Pat[10;1]:="20% Fill"

Pat[10;2]:=1

Pat[11;1]:="10% Fill"

Pat[11;2]:=0

