

```

//*****
//      MACRO: PLEADING.WCM
//      PURPOSE: Creates/Modifies Pleading Paper
//*****
Application (A1; "WordPerfect"; Default; "US")
If((?MajorVersion=6) And (?MinorVersion=0) And (?InterimVersion=0))
    Display(On!)
    MessageBox(;"Old Version Detected";"This macro requires WordPerfect Version 6.0a or
greater to function properly.";OK!|IconStop!)
    Quit
EndIf
StartNum:=1                // Sets starting line number
StopNum:=28                // Sets ending line number
MaxNumber:=128            // Sets maximum number of pleading lines
Numbers:=1                 // Sets numbering On or Off (1=On 0=Off)
LinesLeft:=2              // Sets number of vertical lines on left
LinesRight:=1             // Sets number of vertical lines on right
NumToLineSpace:=0.094"    // Sets distance between numbers and vertical line
LineToTextSpace:=0.078"  // Sets spacing between vertical lines and text
VertLineInset:=0.188"    // Sets inset between lines and top/bottom of page
Call(WaitMsgInit@)
While(?Substructure)
    SubstructureExit()
EndWhile
MacroStatusPrompt(On!;"Pleading Paper...")

//*****
//      PROGRAM BEGIN
//*****
Call(GetJustification@)    // Get current justification setting
Call(EditStyle@)          // Open initial style for editing
Call(Dialog@)             // Create and display pleading dialog
Call(CloseStyle@)        // Close the initial style
Call(CalcLineHeight@)    // Calculate line height
Call(EditStyle@)          // Open initial style for editing
Call(LinesAndNumbers@)    // Insert lines and numbers in style
Call(CloseStyle@)        // Close the initial style
Call(InsertJustification@) // Insert justification at top of document
Call(WaitMsgHide@) // Hide wait message
Call(WaitMsgDestroy@)    // Destroy wait message
Quit                      // End of macro
//*****
//      PROGRAM END
//*****

//*****

```

```

//  ROUTINE NAME: GetJustification@
//  INPUT VARIABLES: None
//  OUTPUT VARIABLES: Justify
//  USES: Nothing
//  DESCRIPTION: Gets current justification setting.
//*****
Label(GetJustification@)
Switch(?Justification)
    CaseOf 0: Justify:="Left"
    CaseOf 1: Justify:="Full"
    CaseOf 2: Justify:="Center"
    CaseOf 3: Justify:="Right"
    CaseOf 4: Justify:="All"
    Default: Justify:="Full"
EndSwitch
Return
//*****

//*****
//  ROUTINE NAME: EditStyle@
//  INPUT VARIABLES: None
//  OUTPUT VARIABLES: None
//  USES: Nothing
//  DESCRIPTION: Opens initial style for editing.
//*****
Label(EditStyle@)
StyleEditBegin(DocStyle!;CurrentDoc!)
StyleCodes (WithoutOffCodes!; CurrentDoc!)
PosDocBottom()    // Move to bottom of style
Return
//*****

//*****
//  ROUTINE NAME: Dialog@
//  INPUT VARIABLES: MaxNumber
//  OUTPUT VARIABLES: StartNum, StopNum, LinesLeft, LinesRight
//  USES: Nothing
//  DESCRIPTION: Defines and displays main dialog. Also checks dialog input for errors.
//*****
Label(Dialog@)
DlgWidth:=186
DlgHeight:=162
ButtonWidth:=58
ButtonHPos:=DlgWidth-ButtonWidth-12
ButtonVPos:=DlgHeight-35
DialogDefine("Pleading";50;50;DlgWidth;DlgHeight;2+16;"Pleading Paper")

```

### // Push Buttons

```
DialogAddPushButton("Pleading";"OK";ButtonHPos;8;ButtonWidth;13;1;"OK")
DialogAddPushButton("Pleading";"Cancel";ButtonHPos;24;ButtonWidth;13;0;"Cancel")
DialogAddPushButton("Pleading";"Margins";ButtonHPos;45;ButtonWidth;13;0;"&Margins...")
DialogAddPushButton("Pleading";"Font";ButtonHPos;61;ButtonWidth;13;0;"&Font...")
DialogAddPushButton("Pleading";"Numbering";ButtonHPos;77;ButtonWidth;13;0;"Pg
&Numbers...")
DialogAddPushButton("Pleading";"Spacing";ButtonHPos;93;ButtonWidth;13;0;"Line
&Spacing...")
```

### // Justification

```
DialogAddText("Pleading";"Justification";116;114;50;10;1;"&Justification")
DialogAddPopUpButton("Pleading";"JustPop";116;125;ButtonWidth;13;Justify)
DialogAddListItem("Pleading";"JustPop";"Left")
DialogAddListItem("Pleading";"JustPop";"Right")
DialogAddListItem("Pleading";"JustPop";"Center")
DialogAddListItem("Pleading";"JustPop";"Full")
DialogAddListItem("Pleading";"JustPop";"All")
```

### // Line Numbering

```
DialogAddGroupBox("Pleading";"NumGrp";8;8;100;76;"Line Numbering")
DialogAddCheckBox("Pleading";"Numbers";18;26;85;11;"&Display Line Numbers";Numbers)
DialogAddText("Pleading";"Start";18;50;50;10;1;"Star&t Number:")
DialogAddCounter("Pleading";"StartCount";70;48;31;13;0;StartNum;1;MaxNumber-1;1)
DialogAddText("Pleading";"Stop";18;67;50;10;1;"Sto&p Number:")
DialogAddCounter("Pleading";"StopCount";70;65;31;13;0;StopNum;2;MaxNumber;1)
```

### // Vertical Lines

```
DialogAddGroupBox("Pleading";"LineGrp";8;91;100;47;"Vertical Lines")
DialogAddText("Pleading";"LinesLeft";18;105;50;10;1;"Lines on &Left:")
DialogAddCounter("Pleading";"LeftCount";70;103;31;13;0;LinesLeft;0;2;1)
DialogAddText("Pleading";"LinesRight";18;122;50;10;1;"Lines on &Right:")
DialogAddCounter("Pleading";"RightCount";70;120;31;13;0;LinesRight;0;2;1)
RegionShowWindow("Pleading.CancelBttn";Hide!)
```

```
Display(On!)
```

```
DialogDisplay("Pleading";1; Callback@)
```

```
NotDone:=True
```

```
While (NotDone)
```

```
EndWhile
```

```
DialogUndisplay("Pleading";"OK")
```

```
DialogDestroy("Pleading")
```

```
Display(Off!)
```

```
MacroStatusPrompt(On!;"Generating pleading...")
```

```
Return
```

```
Label(Callback@)
```

```
MacroStatusPrompt(On!;"Choose OK to Generate Pleading")
```

```
Switch(Callback[3])
```

```
CaseOf "Margins":
```

```

        MacroStatusPrompt(On!;"Set Margins for Pleading")
        FormatMarginsDlg()
    CaseOf "Font":
        MacroStatusPrompt(On!;"Set Font for Pleading")
        FontDlg()
    CaseOf "Numbering":
        MacroStatusPrompt(On!;"Set Page Numbering for Pleading")
        PageNumberingDlg()
    CaseOf "Spacing":
        MacroStatusPrompt(On!;"Set Line Spacing for Pleading")
        LineSpacingDlg()
    CaseOf "Cancel";2;"CancelBtn":
        DialogUndisplay("Pleading";"Cancel")
        DialogDestroy("Pleading")
        SubstructureExit()
        StyleEditEnd (Cancel!)
        Quit
    CaseOf "OK":
        RegionGetWindowText(StartNum;"Pleading.StartCount")
        RegionGetWindowText(StopNum;"Pleading.StopCount")
        If(StartNum>=StopNum)
            MessageBox(; "Line Numbers Overlap"; "Start Number must be less than Stop
            Number."; IconExclamation!)
        Else
            NotDone:=False
        EndIf
    EndSwitch
    Return
//*****

//*****
//    ROUTINE NAME: CloseStyle@
//    INPUT VARIABLES: None
//    OUTPUT VARIABLES: None
//    USES: Nothing
//    DESCRIPTION: Closes and saves initial style.
//*****
Label(CloseStyle@)
SubstructureExit ()
StyleEditEnd (Save!)
Return
//*****

//*****
//    ROUTINE NAME: CalcLineHeight@
//    INPUT VARIABLES: None

```

```

//      OUTPUT VARIABLES: LineHeight
//      USES: Nothing
//      DESCRIPTION: Calculates line height based on start and stop values from dialog.
//*****
Label(CalcLineHeight@)
CodesWereOff:=False
DefaultUnits(WPUnits!)
BookmarkCreate("_pleading.wcm")
PosPageBottom()
Advance(AdvanceDown!;27")
MaxLine:=?Line
If(Not ?RevealCodesActive)
    CodesWereOff:=True
    RevealCodes(On!)
EndIf
PosCharPrevious()
MinLine:=?Line
DeleteCharNext()
If(CodesWereOff)
    RevealCodes(Off!)
EndIf
PrintSpace:=MaxLine-MinLine
LineHeight:=Integer(PrintSpace/(2*(StopNum-StartNum)+1))
Return
//*****

//*****
//      ROUTINE NAME: LinesAndNumbers@
//      INPUT VARIABLES: LineHeight, LineToTextSpace, NumToLineSpace, LinesLeft,
LinesRight, VertLineInset, NumbersOn
//      OUTPUT VARIABLES: None
//      USES: WaitMsgDisplay@
//      DESCRIPTION: Inserts vertical lines and numbers in watermark B.
//*****
Label(LinesAndNumbers@)
LineHeight(LineHeight)
WatermarkB (Create!)
Call(WaitMsgDisplay@)
JustifyLeft()
TextShade (ShadingValue: 100)
LineHeight (LineHeight)
LineSpacing (2.0)
NumPosition:=?MarginLeft-LineToTextSpace-NumToLineSpace
TabSet (Origin: Absolute!; {Position: NumPosition; TabType: TabRight!})
//Create vertical lines
If(LinesLeft>0)

```

```

GraphicsLineCreate ()
GraphicsLineType(Vertical!)
Switch(LinesLeft)
    CaseOf 1: GraphicsLineStyle (Style: SingleLine!)
    CaseOf 2: GraphicsLineStyle (Style: DoubleLine!)
    Default: GraphicsLineStyle (Style: DoubleLine!)
EndSwitch
GraphicsLineHorizontalPosition(Left!)
GraphicsLineSpacing (TopSpace: LineToTextSpace)
GraphicsLineVerticalPosition (Position: Set!; Where: VertLineInset)
LineLength:=?PaperLength-(2*Integer(VertLineInset/1200))
GraphicsLineLength (Length: LineLength)
GraphicsLineEnd (State: Save!)
EndIf
If(LinesRight>0)
    GraphicsLineCreate ()
    GraphicsLineType(Vertical!)
    Switch(LinesRight)
        CaseOf 1: GraphicsLineStyle (Style: SingleLine!)
        CaseOf 2: GraphicsLineStyle (Style: DoubleLine!)
        Default: GraphicsLineStyle (Style: SingleLine!)
    EndSwitch
    GraphicsLineHorizontalPosition (Position: Right!)
    GraphicsLineSpacing (TopSpace: LineToTextSpace; BottomSpace: LineToTextSpace)
    GraphicsLineVerticalPosition (Position: Set!; Where: VertLineInset)
    LineLength:=?PaperLength-(2*Integer(VertLineInset/1200))
    GraphicsLineLength (Length: LineLength)
    GraphicsLineEnd (State: Save!)
EndIf
Advance(AdvanceFromTop!;MinLine)

// Insert line numbers
If(Numbers=1)
    For(Count;StartNum;Count<=StopNum-1;Count+1)
        MarginReleaseInsert()
        MarginReleaseInsert()
        Tab()
        Type(Count)
        HardReturn()
    EndFor
    MarginReleaseInsert()
    MarginReleaseInsert()
    Tab()
    Type(StopNum)
EndIf
SubstructureExit()

```

Return

```
*****
```

```
*****
```

```
// ROUTINE NAME: InsertJustification@  
// INPUT VARIABLES: Justify  
// OUTPUT VARIABLES: None  
// USES: Nothing  
// DESCRIPTION: Inserts justification at top of document.
```

```
*****
```

```
Label(InsertJustification@)
```

```
PosDocTop()
```

```
Switch(Justify)
```

```
  CaseOf "Left": JustifyLeft()  
  CaseOf "Right": JustifyRight()  
  CaseOf "Center": JustifyCenter()  
  CaseOf "Full": JustifyFull()  
  CaseOf "All": JustifyAll()  
  Default: JustifyFull()
```

```
EndSwitch
```

```
BookmarkFind("_pleading.wcm")
```

```
BookmarkDelete("_pleading.wcm")
```

```
Return
```

```
*****
```

```
*****
```

```
// ROUTINE NAME: WaitMsgInit@  
// INPUT VARIABLES: None  
// RETURN VARIABLES: None  
// USES: Nothing  
// DESCRIPTION: Initializes macro wait messages.
```

```
*****
```

```
Label(WaitMsgInit@)
```

```
// Include the following four lines for each wait message desired.
```

```
WaitDlgId="Wait1"
```

```
WaitTitle="WordPerfect for Windows"
```

```
WaitMessage="Generating pleading..."
```

```
DialogDefine(WaitDlgId;50;50;140;50;16;WaitTitle)
```

```
DialogAddText(WaitDlgId;"WaitText";8;12;124;10;1+4;WaitMessage)
```

```
Return
```

```
Label(WaitDlgCallBack@)
```

```
Return
```

```
*****
```

```
*****
```

```

//      ROUTINE NAME: WaitMsgDisplay@
//      INPUT VARIABLES: WaitDlgId
//      OUTPUT VARIABLES: None
//      USES: Nothing
//      DESCRIPTION: Displays a previously defined wait message dialog.
//*****
Label(WaitMsgDisplay@)
Display(On!)
DialogDisplay(WaitDlgId;0;WaitDlgCallBack@)
Display(Off!)
Return
//*****

//*****
//      ROUTINE NAME: WaitMsgHide@
//      INPUT VARIABLES: WaitDlgId
//      OUTPUT VARIABLES: None
//      USES: Nothing
//      DESCRIPTION: Hides a previously defined wait message dialog.
//*****
Label(WaitMsgHide@)
DialogUndisplay(WaitDlgId;1)
Return
//*****

//*****
//      ROUTINE NAME: WaitMsgDestroy@
//      INPUT VARIABLES: WaitDlgId
//      OUTPUT VARIABLES: None
//      USES: Nothing
//      DESCRIPTION: Destroys a previously defined wait message dialog. Hide the dialog
with WaitMsgHide before destroying it.
//*****
Label(WaitMsgDestroy@)
DialogDestroy(WaitDlgId)
Return
//*****

```